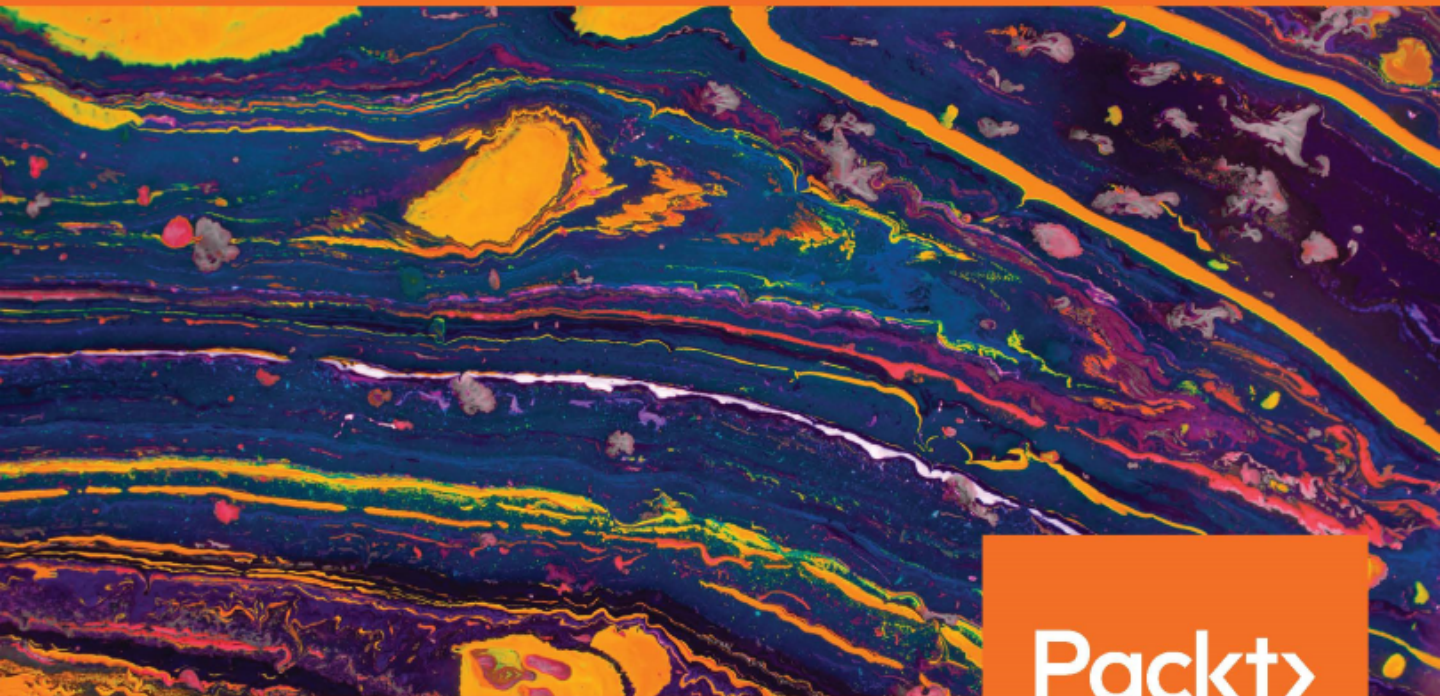


# Learn Kali Linux 2019

Perform powerful penetration testing using Kali Linux, Metasploit, Nessus, Nmap, and Wireshark



**Packt>**

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

Glen D. Singh

# Learn Kali Linux 2019

Perform powerful penetration testing using Kali Linux, Metasploit, Nessus, Nmap, and Wireshark

**Glen D. Singh**



**BIRMINGHAM - MUMBAI**

# Learn Kali Linux 2019

Copyright © 2019 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 author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been 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.

**Commissioning Editor:** Vijin Boricha  
**Acquisition Editor:** Heramb Bhavsar  
**Content Development Editor:** Alokita Amanna  
**Senior Editor:** Rahul Dsouza  
**Technical Editor:** Mohd Riyan Khan  
**Copy Editor:** Safis Editing  
**Project Coordinator:** Anish Daniel  
**Proofreader:** Safis Editing  
**Indexer:** Manju Arasan  
**Production Designer:** Jyoti Chauhan

First published: November 2019

Production reference: 1141119

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

ISBN 978-1-78961-180-9

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

*I would like to dedicate this book to those people in our society who have always worked hard in their field of expertise and who have not been recognized for their hard work, commitment, sacrifices, and ideas, but who, most importantly, believed in themselves when no one else did. This book is for you. Always have faith in yourself. With commitment, hard work, and focus, anything can be possible. Never give up because great things take time.*

*- Glen D. Singh*



Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

## Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

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.packt.com](http://www.packt.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [customercare@packtpub.com](mailto:customercare@packtpub.com) for more details.

At [www.packt.com](http://www.packt.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.

# Contributors

## About the author

**Glen D. Singh**, CEH, CHFI, 3xCCNA (cyber ops, security, and routing and switching) is a cyber security instructor, author, and consultant. He specializes in penetration testing, digital forensics, network security, and enterprise networking. He enjoys teaching and mentoring students, writing books, and participating in a range of outdoor activities. As an aspiring game-changer, Glen is passionate about developing cyber security awareness in his homeland, Trinidad and Tobago.

*I would like to thank Danish Shaikh, Swathy Mohan, Abhishek Jadhav, Amitendra Pathak, Alokita Amanna, Mohd Riyan Khan, and Rahul Dsouza, the wonderful team at Packt Publishing, who have provided amazing support and guidance throughout this journey. To the technical reviewers, Rishalin and Lystra, thank you for your outstanding contribution to making this an amazing book.*

## About the reviewers

**Lystra K. Maingot** is a trained ethical hacker and digital forensics investigator. He has conducted numerous tests and investigations, and has worked in penetration testing and digital forensics investigation training for several years. He is also trained in networking and earned his MSc in network security from Anglia Ruskin University in the UK. He intends to pursue his passion for cyber security in the hope of making our cyber environment a safer place.

**Rishalin Pillay** has over 12 years' cyber security experience, and has acquired a vast number of skills consulting for Fortune 500 companies while participating in projects involving tasks associated with network security design, implementation, and vulnerability analysis. He has reviewed several books, and authored the book *Learn Penetration Testing*. He holds many certifications that demonstrate his knowledge and expertise in the cyber security field from vendors such as (ISC)2, Cisco, Juniper, Checkpoint, Microsoft, and CompTIA. Rishalin currently works at a large software company as a senior cyber security engineer.

## Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit [authors.packtpub.com](https://authors.packtpub.com) and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

# Table of Contents

<b>Preface</b>	1
<b>Section 1: Kali Linux Basics</b>	
<b>Chapter 1: Introduction to Hacking</b>	7
<b>Who is a hacker?</b>	8
Types of hackers	8
Black hat hacker	9
White hat hacker	9
Gray hat hacker	10
Suicide hacker	10
State-sponsored hacker	11
Script kiddie	11
Cyber terrorist	11
<b>Exploring important terminology</b>	12
Threat	12
Asset	13
Vulnerability	13
Exploit	14
Risk	14
Zero-day	14
Hack value	15
<b>Penetration testing phases</b>	15
Pre-engagement	15
Information gathering	17
Threat modeling	18
Vulnerability analysis	18
Exploitation	18
Post-exploitation	19
Report writing	19
<b>Penetration testing methodologies</b>	20
OWASP	20
NIST	20
OSSTMM	21
SANS 25	21
<b>Penetration testing approaches</b>	21
White box	22
Black box	22
Gray box	22
<b>Types of penetration testing</b>	23
Web application penetration testing	23



Mobile application penetration testing	23
Social engineering penetration testing	24
Network penetration testing	24
Cloud penetration testing	24
Physical penetration testing	25
<b>Hacking phases</b>	25
Reconnaissance or information gathering	26
Scanning	26
Gaining access	27
Maintaining access	27
Covering tracks	28
<b>Summary</b>	28
<b>Questions</b>	28
<b>Further reading</b>	29
<b>Chapter 2: Setting Up Kali - Part 1</b>	30
<b>Technical requirements</b>	30
<b>Lab overview</b>	31
Virtualization	31
Hypervisors	34
Type 1 hypervisor	34
Type 2 hypervisor	36
Additional components	37
Virtual switches	37
Operating systems	38
<b>Building our lab</b>	39
Creating a virtual network	42
Setting up Kali Linux	47
Attaching the virtual network to a virtual machine	51
Installing Nessus	54
Setting up Android emulators	59
Installing Metasploitable 2	62
<b>Summary</b>	65
<b>Questions</b>	65
<b>Further reading</b>	65
<b>Chapter 3: Setting Up Kali - Part 2</b>	66
<b>Technical requirements</b>	66
<b>Installing Windows as a VM</b>	67
Creating a user account	74
Opting out of automatic updates	75
Setting a static IP address	76
Adding additional interfaces	80
<b>Installing Ubuntu 8.10</b>	83
Creating and using snapshots	90
<b>Troubleshooting Kali Linux</b>	91

Network adapter and USB incompatibility	92
VM memory problems	94
<b>Summary</b>	95
<b>Further reading</b>	95
<b>Chapter 4: Getting Comfortable with Kali Linux 2019</b>	96
<b>Technical requirements</b>	97
<b>Understanding Kali Linux</b>	97
<b>What's new in Kali Linux 2019?</b>	98
<b>Basics of Kali Linux</b>	99
The Terminal and Linux commands	99
Navigating in Kali Linux	101
Updating sources and installing programs	105
The find, locate, and which commands	109
The locate command	109
The which command	109
The find command	110
Managing Kali Linux services	112
<b>Summary</b>	114
<b>Questions</b>	115
<b>Further reading</b>	115
<b>Section 2: Reconnaissance</b>	
<hr/>	
<b>Chapter 5: Passive Information Gathering</b>	117
<b>Technical requirements</b>	118
<b>Reconnaissance and footprinting</b>	118
Reconnaissance	118
Footprinting	119
<b>Understanding passive information gathering</b>	121
<b>Understanding OSINT</b>	121
<b>Using the top OSINT tools</b>	124
Maltego	124
Recon-ng	134
theHarvester	144
Shodan	147
OSRFramework	149
<b>Identifying target technology and security controls</b>	152
Discovering technologies using Shodan	152
The power of Netcraft	154
Recognizing technologies with WhatWeb	156
<b>Finding data leaks in cloud resources</b>	158
<b>Understanding Google hacking and search operators</b>	160
<b>Leveraging whois and copying websites with HTTrack</b>	163
whois	164

HTTrack	165
<b>Finding subdomains using Sublist3r</b>	166
<b>Summary</b>	167
<b>Questions</b>	168
<b>Further reading</b>	168
<b>Chapter 6: Active Information Gathering</b>	169
<b>Technical requirements</b>	170
<b>Understanding active information gathering</b>	170
<b>DNS interrogation</b>	171
What is DNS and why do we need it on a network?	171
Performing DNS enumeration and zone transfer using dnsenum	175
Using the host utility to perform DNS analysis	178
Finding subdomains with dnsmap	179
DNS interrogation using Fierce	180
<b>Scanning</b>	181
<b>Nmap</b>	183
Performing a ping sweep with Nmap	185
Obtaining operating system and service versions using Nmap	186
Scanning host devices with ICMP disabled	188
Performing a stealth scan using Nmap	188
Scanning UDP ports using Nmap	191
Evading detection using Nmap	191
Evading firewalls with Nmap	192
Checking for a stateful firewall	192
<b>NSE scripts</b>	195
<b>Zenmap</b>	196
<b>Hping3</b>	199
<b>SMB, LDAP enumeration, and null sessions</b>	201
SMBmap and SMBclient	201
Enum4linux	203
LDAP enumeration	204
Null sessions	207
<b>User enumeration through noisy authentication controls</b>	207
<b>Web footprints and enumeration with EyeWitness</b>	209
<b>Metasploit auxiliary modules</b>	211
<b>Summary</b>	212
<b>Questions</b>	213
<b>Further reading</b>	213
<b>Section 3: Vulnerability Assessment and Penetration</b>	
<b>Testing with Kali Linux 2019</b>	
<hr/>	
<b>Chapter 7: Working with Vulnerability Scanners</b>	215
<b>Technical requirements</b>	216

<b>Nessus and its policies</b>	216
Nessus policies	217
<b>Scanning with Nessus</b>	219
<b>Exporting Nessus results</b>	223
<b>Analyzing Nessus results</b>	226
<b>Using web application scanners</b>	228
Nikto	229
WPScan	230
Burp Suite	235
Using Intruder for brute force	241
<b>Summary</b>	248
<b>Questions</b>	248
<b>Further reading</b>	248
<b>Chapter 8: Understanding Network Penetration Testing</b>	249
<b>Technical requirements</b>	249
<b>Introduction to network penetration testing</b>	250
Types of penetration test	251
<b>Understanding the MAC address</b>	252
How to spoof the MAC address	254
<b>Connecting a wireless adapter to Kali Linux</b>	255
<b>Managing and monitoring wireless modes</b>	259
Enabling monitor mode manually	260
Enabling monitor mode using airmon-ng	261
<b>Summary</b>	263
<b>Questions</b>	263
<b>Further reading</b>	264
<b>Chapter 9: Network Penetration Testing - Pre-Connection Attacks</b>	265
<b>Technical requirements</b>	266
<b>Getting started with packet sniffing using airodump-ng</b>	266
<b>Targeted packet sniffing using airodump-ng</b>	269
<b>Deauthenticating clients on a wireless network</b>	270
<b>Creating a rogue AP/evil twin</b>	273
<b>Performing a password spraying attack</b>	278
<b>Setting up watering hole attacks</b>	282
<b>Exploiting weak encryption to steal credentials</b>	283
<b>Summary</b>	288
<b>Questions</b>	289
<b>Further reading</b>	289
<b>Chapter 10: Network Penetration Testing - Gaining Access</b>	290
<b>Technical requirements</b>	291
<b>Gaining access</b>	291

<b>WEP cracking</b>	293
<b>WPA cracking</b>	295
<b>Securing your network from the aforementioned attacks</b>	298
SSID management	299
MAC filtering	300
Power levels for antennas	300
Strong passwords	301
Securing enterprise wireless networks	302
<b>Configuring wireless security settings to secure your network</b>	302
<b>Exploiting vulnerable perimeter systems with Metasploit</b>	306
EternalBlue exploitation	311
<b>Penetration testing Citrix and RDP-based remote access systems</b>	315
Citrix penetration testing	315
Breaking into RDP	316
Leveraging user credentials	319
<b>Plugging PWN boxes and other tools directly into a network</b>	321
<b>Bypassing NAC</b>	323
<b>Summary</b>	324
<b>Questions</b>	325
<b>Further reading</b>	325
<b>Chapter 11: Network Penetration Testing - Post-Connection Attacks</b>	326
<b>Technical requirements</b>	327
<b>Gathering information</b>	327
Scanning using Netdiscover	327
Scanning using AutoScan-Network	329
Scanning using Zenmap	331
<b>MITM attacks</b>	333
ARPspooF	334
MITMf	335
Use cases of MITMf	337
<b>Session hijacking</b>	339
<b>DHCP attacks</b>	349
<b>Exploiting LLMNR and NetBIOS-NS</b>	354
<b>WPAD protocol attacks</b>	358
<b>Wireshark</b>	362
Basic overview of Wireshark and how to use it in MITM attacks	362
Configuring a SPAN port	364
Configuring a monitor (sniffer) interface on Wireshark	365
Parsing Wireshark packet captures to find the goods	367
<b>Escalating privileges</b>	376
<b>Lateral movement tactics</b>	378
<b>PowerShell tradecraft</b>	381
Removing Windows Defender virus definitions	381

Disabling Windows Antimalware Scan Interface	383
<b>Launching a VLAN hopping attack</b>	384
<b>Summary</b>	386
<b>Questions</b>	386
<b>Further reading</b>	386
<b>Chapter 12: Network Penetration Testing - Detection and Security</b>	387
Technical requirements	387
Using Wireshark to understand ARP	388
Detecting ARP poisoning attacks	389
Detecting suspicious activity	391
MITM remediation techniques	393
Encryption	393
Dynamic ARP inspection	395
Sniffing remediation techniques	397
<b>Summary</b>	397
<b>Questions</b>	398
<b>Further reading</b>	398
<b>Chapter 13: Client-Side Attacks - Social Engineering</b>	399
Technical requirements	400
Basics of social engineering	400
Types of social engineering	401
Human-based social engineering	402
Eavesdropping	402
Shoulder surfing	402
Dumpster diving	403
Computer-based social engineering	403
Phishing	403
Spear phishing	404
Mobile-based social engineering	404
Social engineering through social networking	405
Phone-based social engineering (vishing)	405
<b>Defending against social engineering</b>	406
Protecting your perimeter security	406
Protecting the help desk and general staff	407
Additional countermeasures	407
Detecting phishing emails	407
<b>Recon for social engineering (doxing)</b>	409
<b>Planning for each type of social engineering attack</b>	410
<b>Social engineering tools</b>	411
Social-Engineer Toolkit	411
Ghost Phisher	414
<b>Summary</b>	417
<b>Questions</b>	417

<b>Further reading</b>	418
<b>Chapter 14: Performing Website Penetration Testing</b>	419
<b>Technical requirements</b>	420
<b>Information gathering</b>	420
Discovering technologies that are being used on a website	420
Discovering websites on the same server	423
Discovering sensitive files	426
robots.txt	429
Analyzing discovered files	430
<b>Cryptography</b>	432
<b>File upload and file inclusion vulnerabilities</b>	433
XSS	434
Stored XSS	435
Reflected XSS	435
CSRF	435
SQLi	436
Insecure deserialization	437
Common misconfigurations	438
Vulnerable components	438
IDOR	439
<b>Exploiting file upload vulnerabilities</b>	439
<b>Exploiting code execution vulnerabilities</b>	443
<b>Exploiting LFI vulnerabilities</b>	445
<b>Preventing vulnerabilities</b>	446
<b>Summary</b>	447
<b>Questions</b>	447
<b>Further reading</b>	447
<b>Chapter 15: Website Penetration Testing - Gaining Access</b>	448
<b>Technical requirements</b>	448
<b>Exploring the dangers of SQL injection</b>	449
Dangers from SQL injection vulnerabilities	449
Bypassing logins using SQL injection	449
<b>SQL injection vulnerabilities and exploitation</b>	453
Discovering SQL injections with POST	460
Detecting SQL injections and extracting data using SQLmap	464
Preventing SQL injection	465
<b>Cross-Site Scripting vulnerabilities</b>	465
Understanding XSS	466
Discovering reflected XSS	467
Discovering stored XSS	468
Exploiting XSS – hooking vulnerable page visitors to BeEF	470
<b>Discovering vulnerabilities automatically</b>	476
Burp Suite	476

Acunetix	479
OWASP ZAP	485
<b>Summary</b>	489
<b>Questions</b>	489
<b>Further reading</b>	489
<b>Chapter 16: Best Practices</b>	490
<b>Technical requirements</b>	490
<b>Guidelines for penetration testers</b>	490
Gaining written permission	491
Being ethical	491
Penetration testing contract	492
Rules of engagement	492
Additional tips and tricks	493
<b>Web application security blueprints and checklists</b>	493
OWASP	494
Penetration testing execution standard	495
Reporting	495
Penetration testing checklist	496
Information gathering	497
Network scanning	497
Enumeration	497
Gaining access	498
Covering tracks	498
<b>Summary</b>	499
<b>Questions</b>	499
<b>Further reading</b>	499
<b>Assessments</b>	501
<b>Other Books You May Enjoy</b>	506
<b>Index</b>	509

---



# Preface

*Learn Kali Linux 2019* is an excellent book filled with amazing content and exercises designed with a student-centric approach, making it easy to adapt to and follow through each chapter easily. *Learn Kali Linux 2019* starts by introducing the reader to ethical hacking concepts and threat actors, before gradually moving into penetration testing approaches and methodologies. Each chapter smoothly flows onto the next. With each step along the journey, the stages of penetration testing are outlined, with the help of in-depth theory and hands-on labs using one of the most popular penetration testing platforms, Kali Linux.

The reader will learn how to build their own penetration testing lab environment, perform both passive and active reconnaissance using OSINT on the target organizations, perform vulnerability scanning using multiple tools such as Nessus, and perform wireless penetration, network penetration testing, website and web application penetration testing, and client-side attacks.

Furthermore, readers will gain the skills required to perform privilege escalation and lateral movement using the Metasploit framework. *Learn Kali Linux 2019* takes you from beginner to expert in terms of learning and understanding penetration testing, while keeping the reader in mind.

This title can also be used as a training guide in penetration testing, ethical hacking, and cyber security-related courses.

## Who this book is for

This book is designed for students, network and security engineers, cyber security/information security professionals, enthusiasts, and those who simply have an interest in ethical hacking and penetration testing. This title can also be used in both independent (self-study) and classroom-based training in penetration testing and cyber security courses alike.

Whether you're new to the field of information technology or a seasoned IT professional, *Learn Kali Linux 2019* has something for everyone. A detailed knowledge of networking and IT security is preferred but not mandatory, as the book is written for anyone.

## What this book covers

Chapter 1, *Introduction to Hacking*, introduces various types of threat actors and penetration testing methodologies and approaches.

Chapter 2, *Setting Up Kali - Part 1*, introduces you to virtualization concepts, how to build your own penetration testing lab, how to install Kali Linux, and vulnerable target machines.

Chapter 3, *Setting Up Kali - Part 2*, focuses on installing and configuring Windows and Ubuntu operating systems and troubleshooting Kali Linux.

Chapter 4, *Getting Comfortable with Kali Linux 2019*, teaches you about Kali Linux, its features, and commands to enable you to perform various tasks.

Chapter 5, *Passive Information Gathering*, examines the passive ways to gather information pertaining to the target from **Open Source Intelligence (OSINT)**, which means we will gather information about the target from publicly available resources.

Chapter 6, *Active Information Gathering*, explains the active ways of gathering information using DNS interrogation, scanning, and enumeration techniques.

Chapter 7, *Working with Vulnerability Scanners*, explores various network and web vulnerability scanner tools, including Nessus, Nikto, WPScan, and Burp Suite.

Chapter 8, *Understanding Network Penetration Testing*, covers some basic concepts of wireless penetration testing.

Chapter 9, *Network Penetration Testing - Pre-Connection Attacks*, explores a wireless hacking tool, aircrack-ng, the basic concept of deauthentication attacks, and how to create fake access points.

Chapter 10, *Network Penetration Testing - Gaining Access*, covers the basics of gaining access, and how to crack WEP and WPA encryption using dictionary and brute force attacks.

Chapter 11, *Network Penetration Testing - Post-Connection Attacks*, explores information gathering, how to perform man-in-the-middle attacks, sniffing using Wireshark, elevating privileges, and lateral movement on a network.

Chapter 12, *Network Penetration Testing - Detection and Security*, explains how to detect an ARP poisoning attack and suspicious activities using Wireshark and packet analysis.

Chapter 13, *Client-Side Attacks - Social Engineering*, explains various types of social engineering attacks and how to defend against them, while also covering how to create a phishing Facebook page and mitigation techniques.

Chapter 14, *Performing Website Penetration Testing*, covers the basics of web application penetration testing. Readers will learn about common web-based vulnerabilities such as SQL Injection, **Cross-Site Scripting (XSS)**, and **Cross-Site Request Forgery (CSRF)**.

Chapter 15, *Website Penetration Testing - Gaining Access*, explains how to bypass logins using a SQL injection attack, while also providing you with an explanation of reflected and store XSS attacks and how to perform client-side attacks using BeEF.

Chapter 16, *Best Practices*, provides guidelines for penetration testers and the web application security blueprint to ensure that, after completing this book, the reader has a wealth of knowledge and is able to adapt to good practices in the industry.

## To get the most out of this book

To get the most out of this book, readers should have a basic understanding of networking, including various network and application protocols, network devices and appliances, and a basic understanding of routing and switching concepts. Some prior knowledge of IT security is not mandatory, but help you grasp the concepts and exercises presented during the course of this book.

The only hardware required is a personal computer, such as a laptop or desktop, with an operation system capable of running Oracle VM VirtualBox or VMware Workstation 15 Pro. As for specifications, the recommended setup is as follows:

- Processor: Intel i5, i7, or better
- HDD: 200 GB hard drive
- RAM: 4 GB of RAM (8 GB is preferable)
- An internet connection
- Alfa Network AWUS036NHA wireless adapter

## Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: [https://static.packt-cdn.com/downloads/9781789611809\\_ColorImages.pdf](https://static.packt-cdn.com/downloads/9781789611809_ColorImages.pdf).

## Conventions used

There are a number of text conventions used throughout this book.

**CodeInText:** Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Use the `ifconfig` command to verify the status of the adapter."

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

```
airodump-ng --bssid <bssid value> -c <channel number> wlan0mon
```

**Bold:** Indicates a new term, an important word, or words that you see on screen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "If you're using VMware, the **New Virtual Machine Wizard** will prompt you to continue your setup in either a **Typical (recommended)** or **Custom (advanced)** mode."



Warnings or important notes appear like this.



Tips and tricks appear like this.

## Get in touch

Feedback from our readers is always welcome.

**General feedback:** If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at [customercare@packtpub.com](mailto:customercare@packtpub.com).

**Errata:** Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit [www.packtpub.com/support/errata](http://www.packtpub.com/support/errata), selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy:** If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at [copyright@packt.com](mailto:copyright@packt.com) with a link to the material.

**If you are interested in becoming an author:** If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit [authors.packtpub.com](http://authors.packtpub.com).

## Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit [packt.com](http://packt.com).

# 1

## Section 1: Kali Linux Basics

This section covers the basics of hacking by discussing the concepts of penetration testing and its value in combating cyber threats. In addition, the reader will learn how to build their own penetration testing lab filled with various operating systems to practice and sharpen their skill set.

This section comprises the following chapters:

- Chapter 1, *Introduction to Hacking*
- Chapter 2, *Setting Up Kali - Part 1*
- Chapter 3, *Setting Up Kali - Part 2*
- Chapter 4, *Getting Comfortable with Kali Linux 2019*

# 1

# Introduction to Hacking

Cybersecurity is one of the most rapidly growing fields in information technology. Every day, numerous attacks are executed against various entities, from individuals to large enterprises and even governments. Due to these threats in the digital world, new professions are being created within organizations for people who can protect assets. This book aims to give you the knowledge and techniques that an aspiring penetration tester needs in order to enter the field of cybersecurity. A penetration tester is a professional who has the skills of a hacker; they are hired by an organization to perform simulations of real-world attacks on their network infrastructure with the objective of discovering security vulnerabilities before a real attack occurs. The penetration tester does this task with written legal permission from the target organization. To become a highly skilled hacker, it's vital to have a strong understanding of computers, networking, and programming, as well as how they work together. Most importantly, however, you need creativity. Creative thinking allows a person to think outside the box and go beyond the intended uses of technologies and find exciting new ways to implement them, doing things with them that were never intended by their developers. In some ways, hackers are artists.

Throughout this book, we will be using one of the most popular operating systems for penetration testing, Kali Linux. The Kali Linux operating system has hundreds of tools and utilities designed to assist you during a vulnerability assessment, penetration test, or even a digital forensics investigation in the field of cybersecurity. We will use Kali Linux to take you through various topics using a student-centric approach, filled with a lot of hands-on exercises starting from beginner level to intermediate to more advanced topics and techniques.

In this chapter, you will become acquainted with what hackers are and how they can be classified based on motivations and actions. You'll learn important terminology and look at methods and approaches that will help you throughout this book and set you on your path to becoming a penetration tester. You'll be introduced to the workflow of a hack as well.

In this chapter, we will look at the following topics:

- Who is a hacker?
- Key terminology
- Penetration testing phases
- Penetration testing methodologies
- Penetration testing approaches
- Types of penetration testing
- Hacking phases

## Who is a hacker?

**Hacker**, **hack**, and **hacking** are terms that have become ubiquitous in the 21st century. You've probably heard about life hacks, business hacks, and so on. While these may be, in some sense of the word, forms of hacking, the traditional form of hacking we'll discuss in this book is computer hacking. Computer hacking is the art of using computer-based technologies in ways they were never intended to be used to get them to do something unanticipated.

Hacking has taken on many different names and forms throughout the years. In the late 20th century, a common form of hacking was known as **phreaking**, which abused weaknesses in analog phone systems. Computer hacking has been around for more than half a century and, over the past few decades, has become a pop culture sensation in Hollywood movies and on television shows. It's all over the news, almost daily. You hear about things such as the Equifax, NHS, and Home Depot data breaches all the time. If you're reading this book, you have made your first step toward better understanding this fringe form of engineering.

Now that we have a better idea of what a hacker is, let's explore the various classifications of hackers.

## Types of hackers

Hacking has many varieties or flavors, and so there are many classifications for hackers. In this section, we'll explore the various types of hackers, including the activities, skill sets, and values associated with each.



The following are the different types of hackers:

- Black hat
- White hat
- Gray hat
- Suicide
- State-sponsored
- Script kiddie
- Cyber terrorist

At the end of this section, you will be able to compare and contrast each type of hacker.

## **Black hat hacker**

Black hat hackers typically have a strong understanding of systems, networks, and application programming, which they use for malicious and/or criminal purposes. This type of hacker typically has a deep understanding of evasion and indemnification tactics, which they use to avoid imprisonment as a result of their actions.

They understand the common tools and tactics used by highly skilled ethical hackers. Hackers caught performing criminal hacking are usually blacklisted from ethical hacking, thus losing the ability to get employment as an ethical hacker.

Now that you have a better understanding of black hat hackers, let's take a look at another type—one that follows ethical practices and helps others: the white hat hacker.

## **White hat hacker**

White hat hackers, like black hat hackers, possess a strong understanding of systems, networks, and application programming. However, unlike black hats, they use their knowledge and skills to test systems, applications, and networks for security vulnerabilities. This testing is conducted with the permission of the target and is used to find weaknesses in security before unethical hackers exploit them. The motivation to safeguard systems and entities, while staying within the confines of the law and ethics, leads to white hats being called ethical hackers.

Like black hats, they possess a solid knowledge of hacking tools, attack vectors, and tactics used in the exploitation and discovery of vulnerabilities. They also need to think like black hats when testing and, therefore, must use creativity to imagine themselves in the shoes of those they wish to combat. Ethical or white hat hacking is the most common form of hacking and the focus of this book.

Now that we understand the difference between a white hat hacker and a black hat hacker, let's move on to a type of hacker who looks for vulnerabilities while inhabiting an ambiguous or **gray** area between ethical and unethical hacking: the gray hat hacker.

## **Gray hat hacker**

Gray hat hackers are similar to white hats but often conduct vulnerability research on their own, and then disclose these vulnerabilities to force vendors to remediate the issue by issuing a software patch. Their skills typically have a heavier emphasis on vulnerability research tactics, such as fuzzing, debugging, and reverse engineering.

At times, being a gray hat can be difficult as the balance and definition of ethical and unethical actions keep changing. Despite the difficult place that they occupy in the community, they share valuable information about security flaws, and are therefore important members of the cybersecurity community.

The next type of hacker uses unethical means to break into systems but does not do so for personal profit like a black hat—this type of hacker is the suicide hacker.

## **Suicide hacker**

Suicide hackers are typically less-skilled hackers who are just about capable enough to gain access to systems but are not able to evade detection. These hackers have no concern for being caught or imprisoned—they are happy as long as they succeed in entering and disrupting a system. Their actions are motivated by revenge, political ideologies, and so on. This type of hacker doesn't care whether they are caught or arrested, so long as the job is done.

Next, we'll take a look at hackers that work on behalf of or within governments.

## **State-sponsored hacker**

The state-sponsored hacker is usually employed by a national government to spy and launch cyberattacks against another nation. These hackers have dominated conversations about hacking in society.

This type of hacker enjoys access to all the tools and resources provided by the state, as well as protection from prosecution in order to execute their duties effectively.

However, not everyone has access to the cybersecurity training or tools. Most people start with limited resources and skills, such as the type we'll encounter in the next section.

## **Script kiddie**

A script kiddie is a type of hacker that does not fully understand the technical background of hacking. They use scripts and tools created by other hackers to perform their dirty work. However, even though script kiddies lack the technical knowledge of a real hacker, their actions can still cause a lot of damage in the digital world.

Most hackers start off as a script kiddie. Then, by developing their knowledge and skill set, they are able to become more accomplished at hacking. This ultimately leads them to choose a life as one of the various other types of hackers mentioned in this section.

The last type of hacker has a different set of motives, for example, ideological or political motives that are extreme in nature: they are cyber terrorists.

## **Cyber terrorist**

Cyber terrorists are either individuals or groups with hostile intent to cause havoc for their targets, such as a nation. Their motivation is political in nature. Cyber terrorists carry out quite a wide variety of hacking based on what they do, from causing chaos by compromising cybersecurity to even compromising physical security by hacking into confidential databases.

Having completed this section, you are now able to differentiate between each type of hacker, and you know about their motives and skill sets. The skill sets of hackers can range from script kiddie to black hat level.

Next, let's move on to some important terminologies so that you can become better acquainted with the language of the cybersecurity community. You may have already encountered these terms, and you will continue to do so in this book and on your journey to becoming a penetration tester through discussions, books, training, and so on.

## Exploring important terminology

Every field has certain terms that become a major part of the language of that field. Information security and cybersecurity are no different. The following are the most common terms, and we'll explore them in detail in this section:

- Threat
- Asset
- Vulnerability
- Exploit
- Risk
- Zero-day
- Hack value

Let's delve into these terms in more detail.

### Threat

A threat in terms of cybersecurity is something or someone that intends to cause harm to another person or system. Furthermore, we can look at a threat as something that has the potential to cause malicious damage to a system, network, or person.

Whether you're on the offensive or defensive side in cybersecurity, you must always be able to identify threats. However, while we need to be aware of threats, we also need to know what has to be protected against threats. We call the entity in need of safeguarding an asset. Let's look at what constitutes an asset.

## Asset

Assets, in terms of cybersecurity, are systems within a network that can be interacted with and potentially expose the network or organization to weaknesses that could be exploited and give hackers a way to escalate their privileges from standard user access to administrator/root-level access or gain remote access to the network. It is important to mention that assets are not and should not be limited to technical systems. Other forms of assets include humans, physical security controls, and even data that resides within the networks we aim to protect.

Assets can be broken down into three categories:

- **Tangible:** These are physical things such as networking devices, computer systems, and appliances.
- **Intangible:** These are things that are not in a physical form, such as intellectual property, business plans, data, and records.
- **Employees:** These are the people who drive the business or organization. Humans are one of the most vulnerable entities in the field of cybersecurity.

One key step in vulnerability assessment and risk management is to identify all the assets within an organization. All organizations have assets that need to be kept safe; an organization's systems, networks, and assets always contain some sort of security weakness that can be taken advantage of by a hacker. Next, we'll dive into understanding what a vulnerability is.

## Vulnerability

A vulnerability is a weakness or defect that exists within technical, physical, or human systems that hackers can exploit in order to gain access to or control over systems within a network. Common vulnerabilities that exist within organizations include human error (the greatest of vulnerabilities on a global scale), web application injection vulnerabilities, and the oldest of vulnerabilities, the buffer overflow.

Now that we know what a vulnerability is, let's take a look at what is used by a hacker to take advantage of a security weakness in the next section.

## Exploit

Exploit attacks are the ways hackers take advantage of weaknesses or vulnerabilities within systems. For example, take a hammer, a piece of wood, and a nail. The vulnerability is the soft, permeable nature of wood, and the exploit is the act of hammering the nail into the wood.

As a cybersecurity professional, you must understand vulnerabilities and exploits to reduce the likelihood of being compromised. In the next section, we will describe risk.

## Risk

Risk is the potential impact that a vulnerability, threat, or asset presents to an organization calculated against all other vulnerabilities, threats, and assets. Evaluating risk helps to determine the likelihood of a specific issue causing a data breach that will cause harm to an organization's finances, reputation, or regulatory compliance.

Reducing risk is critical for many organizations. There are many certifications, standards, and frameworks that are designed to help companies understand, identify, and reduce risks. Later, in the *Penetration testing methodologies* section, we will cover such standards and frameworks. Next, we'll look at threats that companies do not know about because no one has identified them yet—zero-day attacks.

## Zero-day

A zero-day attack is an exploit that is unknown to the world, including the vendor, which means it is unpatched by the vendor. These attacks are commonly used in nation-state attacks, as well as by large criminal organizations. The discovery of a zero-day exploit can be very valuable for ethical hackers and can earn them a bug bounty. These bounties are fees paid by vendors to security researchers that discover previously unknown vulnerabilities in their applications.

Today, many organizations have established a bug bounty program, which allows interested persons who discover a vulnerability within a system of a vendor to report it. The person who reports the vulnerability, usually a zero-day flaw, is given a reward. However, there are hackers who intentionally attempt to exploit a system or network for some sort of personal gain; this is known as the hack value, which we will explore next.

## **Hack value**

The hack value is commonly referred to as the motivation or the reason for performing a hack on a system or network. It is the value of accomplishing the goal of breaking into a system.

You are now able to better describe the terminology used in penetration testing. In the next section, we will look at each phase of a penetration test.

## **Penetration testing phases**

While penetration testing is interesting, we cannot attack a target without a battle plan. Planning ensures that the penetration testing follows a sequential order of steps to achieve the desired outcome, which is identifying vulnerabilities. Each phase outlines and describes what is required before moving onto the next steps. This ensures that all details about the work and target are gathered efficiently and that the penetration tester has a clear understanding of the task ahead.

The following are the different phases in penetration testing:

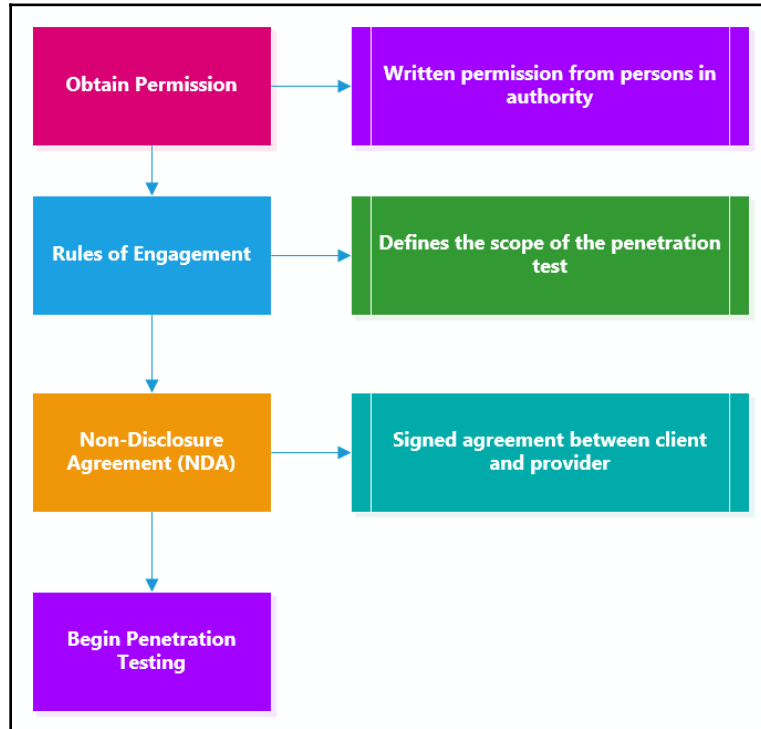
1. Pre-engagement
2. Information gathering
3. Threat modeling
4. Vulnerability analysis
5. Exploitation
6. Post-exploitation
7. Report writing

Each of these phases will be covered in more detail in the following sections.

## **Pre-engagement**

During the pre-engagement phase, key personnel are selected. These individuals are key to providing information, coordinating resources, and helping testers understand the scope, breadth, and rules of engagement in the assessment.

This phase also covers legal requirements, which typically include a **non-disclosure agreement (NDA)** and a **consulting services agreement (CSA)**. The following is a typical process overview of what is required prior to the actual penetration testing:



An NDA is a legal agreement that specifies that a penetration tester will not share or hold onto any sensitive or proprietary information that is encountered during the assessment. Companies usually sign these agreements with cybersecurity companies who will, in turn, sign it with employees working on the project. In some cases, companies sign these agreements directly with the penetration testers from the company carrying out the project.

The scope of a penetration test defines the systems that the testers can and cannot hack or test. To ensure that the penetration tester remains within the legal boundaries, he or she must acquire legal permission in writing from the client or company who is requesting the services. Additionally, the penetration tester must provide an NDA. The agreement between the ethical hacker and the client also defines sensitive systems as well as testing times and which systems require special testing windows. It's incredibly important for penetration testers to pay close attention to the scope of a penetration test and where they are testing in order to always stay within the testing constraints.



The following are some sample pre-engagement questions to help you define the scope of your penetration test:

- What is the size/class of your external network? (Network penetration testing.)
- What is the size/class of your internal network? (Network penetration testing.)
- What is the purpose and goal of the penetration test? (Applicable to any form of penetration testing.)
- How many pages does the web application have? (Web application penetration testing.)
- How many user inputs or forms does the web application have?



This is not an extensive list of pre-engagement questions, and all engagements should be given thorough thought to ensure that you ask all the important questions so you don't underscope or underprice the engagement.

Now that we've understood the legal limitation stages in penetration testing, let's move on to learn about the information-gathering phase and its importance.

## Information gathering

Most types of penetration tests involve an information-gathering phase, which is vital to ensuring that testers have access to key information that will assist them in conducting their assessment. This is not the case in a black box approach, which we will deal with later. Most information gathering is done for web-based application penetration testing, so the questions involved are generally geared toward web-based applications, such as those given here:

- What platform is the application written in?
- Does the application use any APIs?
- Is the application behind a **web application firewall (WAF)**?
- How does the application handle authentication?
- Does the application use active directory credentials to authenticate users?
- Do users access this application in any other way than through the web URL?
- Is the application internet-facing or internal?
- Does the application serve any sensitive information or system access?

Understanding the target is very important before any sort of attack as a penetration tester, as it helps in creating a profile of the potential target. Recovering user credentials/login accounts at this phase, for instance, will be vital to later phases of penetration testing as it will help us gain access to vulnerable systems and networks. Next, we will discuss the essentials of threat modeling.

## **Threat modeling**

Threat modeling is a process used to assist testers and defenders to better understand the threats that inspired the assessment or the threats that the application or network is most prone to. This data is then used to help penetration testers emulate, assess, and address the most common threats that the organization, network, or application faces.

Having understood the threats an organization faces, the next step is to perform a vulnerability assessment on the assets to further determine the risk rating and severity.

## **Vulnerability analysis**

Vulnerability analysis typically involves the assessors or testers running vulnerability or network/port scans to better understand which services the network or application is running and whether there are any vulnerabilities in any systems included in the scope of the assessment. This process often includes manual vulnerability testing/discovery, which is often the most accurate form of vulnerability analysis or vulnerability assessment.

There are many tools, both free and paid for, to assist us in quickly identifying vulnerabilities on a target system or network. After discovering the security weaknesses, the next phase is to attempt exploitation.

## **Exploitation**

Exploitation is the most commonly ignored or overlooked part of penetration testing, and the reality is that clients and executives don't care about vulnerabilities unless they understand why they matter to them. Exploitation is the ammunition or evidence that helps articulate why the vulnerability matters and illustrates the impact that the vulnerability could have on the organization. Furthermore, without exploitation, the assessment is not a penetration test and is nothing more than a vulnerability assessment, which most companies can conduct in-house better than a third-party consultant could.

To put it simply, during the information-gathering phase, a penetration tester will profile the target and identify any vulnerabilities. Next, using the information about the vulnerabilities, the penetration tester will do their research and create specific exploits that will take advantage of the vulnerabilities of the target—this is what exploitation is. We use exploits (malicious code) to leverage a vulnerability (weakness) in a system, which will allow us to execute arbitrary code and commands on the target.

Often after successfully exploiting a target system or network, we may think the task is done—but it isn't just yet. There are tasks and objectives to complete after breaking into the system. This is the post-exploitation phase in penetration testing.

## **Post-exploitation**

Exploitation is the process of gaining access to systems that may contain sensitive information. The process of post-exploitation is the continuation of this step, where the foothold gained is leveraged to access data or spread to other systems within the network. During post-exploitation, the primary goal is typically to demonstrate the impact that the vulnerability and access gained can pose to the organization. This impact assists in helping executive leadership better understand the vulnerabilities and the damage it could cause to the organization.

## **Report writing**

Report writing is exactly as it sounds and is one of the most important elements of any penetration test. Penetration testing may be the service, but report writing is the deliverable that the client sees and is the only tangible element given to the client at the end of the assessment. Reports should be given as much attention and care as the testing.

I will cover report writing in greater detail later in the book, but report writing involves much more than listing a few vulnerabilities discovered during the assessment. It is the medium in which you convey risk, business impact, summarize your findings, and include remediation steps. A good penetration tester needs to be a good report writer, or the issues they find will be lost and may never be understood by the client who hired them to conduct the assessment.

Having completed this section, you are now able to describe each phase of a penetration test. Furthermore, you have a better idea of the expectations of penetration testers in the industry. Next, we will dive into understanding various penetration testing methodologies, standards, and frameworks.

# Penetration testing methodologies

In the field of penetration testing, there are many official and standard methodologies that are used to perform a penetration test on a target system or network.

In the following sections, we will discuss the most popular standards and frameworks that are used in cybersecurity to ensure that organizations meet an acceptable baseline of operating in a secure environment.

## OWASP

**OWASP** stands for **Open Web Application Security Project**, and it provides methodologies as well as lists of the top 10 biggest security weaknesses present in web applications. This list is the de facto framework used by web application penetration testers and is what most corporations are looking for when hiring penetration testers to test their web applications. This is also the most common and prevalent form of penetration testing.

This is one of the most popular frameworks, and every penetration tester should have a clear understanding of it when it comes to web application testing. However, it's equally important to understand others, such as NIST.

## NIST

**NIST** stands for the **National Institute of Standards and Technology**. NIST is a division of the US government, and it publishes a number of special publications defining best practices as well as standards for organizations to employ in order to improve their security. It's important to understand NIST in order to map findings or discovered vulnerabilities to their appropriate rules in order to help organizations understand the compliance implications of the issues discovered during the assessment.

At times, a target organization may require security testing using a specific framework or standard. Being familiar with the OSSTMM can be useful for your engagements with the target organization as a penetration tester.

## OSSTMM

**OSSTMM** stands for the **Open Source Security Testing Methodology Manual**. This is a community-driven, frequently updated, and peer-reviewed set of security testing standards that every ethical hacker should be aware of and keep updated on. These standards tend to cover a wide array of testing subjects and are especially valuable to those entering the industry to help them better understand the process as well as testing best practices.

The knowledge found in OSSTMM will be a great asset as a penetration tester. In the next section, we will discuss the benefits of also understanding SANS 25.

## SANS 25

**SANS 25** is a list of the top 25 security domains as defined by the SANS Institute. When conducting assessments, it's good to be familiar with this list and understand how your findings pertain to the list. In addition, understanding the top 25 domains can assist in helping increase the breadth of your knowledge of security vulnerabilities. These issues typically extend far beyond what will be discovered through nothing but penetration testing, and understanding these issues may even help you identify additional vulnerabilities or risk trends during your assessments.

In my job opportunities, the employer usually wants to ensure that their penetration tester is familiar with and understands each of these penetration testing frameworks and standards. This information is useful when conducting a security test/audit on an organization of a particular industry.

Now that you have a better understanding of popular penetration testing methodologies, let's dive into the three penetration testing approaches.

## Penetration testing approaches

The following are different approaches to performing a penetration test on a target organization:

- White box
- Black box
- Gray box

Let's see what each of these entails.

## White box

A white box assessment is typical of web application testing but can extend to any form of penetration testing. The key difference between white, black, and gray box testing is the amount of information provided to the testers prior to the engagement. In a white box assessment, the tester will be provided with full information about the application and its technology, and will usually be given credentials with varying degrees of access to quickly and thoroughly identify vulnerabilities in the applications, systems, or networks.

Not all security testing is done using the white box approach; sometimes, only the target company's name is provided to the penetration tester. Next, we will cover the fundamentals of black box testing.

## Black box

Black box assessments are the most common form of network penetration assessment and are most typical among external network penetration tests and social engineering penetration tests. In a black box assessment, the testers are given very little or no information about the networks or systems they are testing. This particular form of testing is inefficient for most types of web application testing because of the need for credentials in order to test for authenticated vulnerabilities, such as lateral and vertical privilege escalation.

In situations where black box testing is not suitable, there's another approach that exists between white and black box; this is known as gray box.

## Gray box

Gray box assessments are a hybrid of white and black box testing, and are typically used to provide a realistic testing scenario while also giving penetration testers enough information to reduce the time needed to conduct reconnaissance and other black box testing activities. In addition to this, it's important in any assessment to ensure you are testing all in-scope systems. In a true black box, it's possible to miss systems and, as a result, leave them out of the assessment. The gray box is often the best form of network penetration testing as it provides the most value to clients.

Each penetration test approach is different from the other, and it's vital that you know about all of them. Imagine a potential client calling us to request a black box test on their external network; as a penetration tester, we must be familiar with the terms and what is expected.

Now that we have covered the different approaches of testing, let's dive into the various types of penetration testing.

## Types of penetration testing

Vulnerability and port scanning cannot identify the issues that manual testing can, and this is the reason that an organization hires penetration testers to conduct these assessments. Delivering scans instead of manual testing is a form of fraud and is, in my opinion, highly unethical. If you can't cut it testing, then practice, practice, and practice some more. You will learn legal ways to up your tradecraft later in this book.

In the following sections, we will dive into various types of penetration tests.

## Web application penetration testing

**Web application penetration testing**, hereafter referred to as **WAPT**, is the most common form of penetration testing and likely to be the first penetration testing job most people reading this book will be involved in. WAPT is the act of conducting manual hacking or penetration testing against a web application to test for vulnerabilities that scanners won't find. Too often testers submit web application vulnerability scans instead of manually finding and verifying issues within web applications.

Now you have the essential understanding of WAPT, let's take a look at mobile application penetration testing in the next section.

## Mobile application penetration testing

Mobile application penetration testing is similar to web application penetration testing, but is specific to mobile applications that contain their own attack vectors and threats. This is a rising form of penetration testing with a great deal of opportunity for those who are looking to break into penetration testing and have an understanding of mobile application development.

As you may have noticed, the different types of penetration testing each have specific objectives. Next, we will look at a more human-oriented approach, social engineering.

## Social engineering penetration testing

Social engineering penetration testing, in my opinion, is the most adrenaline-filled type of testing. Social engineering is the art of manipulating basic human psychology to find human vulnerabilities and get people to do things they may not otherwise do. During this form of penetration testing, you may be asked to do activities such as sending phishing emails, make vishing phone calls, or talk your way into secure facilities to determine what an attacker targeting their personnel could achieve. I have personally obtained domain admin access over the phone, talked my way into bank vaults and casino money cages, and talked my way into a Fortune 500 data center.

There are many types of social engineering attacks, which will be covered later on in this book. Most commonly, you'll be tasked with performing security auditing on systems and networks. In the next section, we will discuss network penetration testing.

## Network penetration testing

Network penetration testing focuses on identifying security weaknesses in a targeted environment. The penetration test objectives are to identify the flaws in the target organization's systems, their networks (wired and wireless), and their networking devices such as switches and routers.

The following are some tasks that are performed using network penetration testing:

- Bypassing an **Intrusion Detection System (IDS)/Intrusion Prevent System (IPS)**
- Bypassing firewall appliances
- Password cracking
- Gaining access to end devices and servers
- Exploiting misconfigurations on switches and routers

Now that you have a better idea of the objectives in network penetration testing, let's take a look at the purpose of cloud penetration testing.

## Cloud penetration testing

Cloud penetration testing involves performing security assessments and penetration testing on risks to cloud platforms to discover any vulnerabilities that may expose confidential information to malicious users.



Before attempting to directly engage a cloud platform, ensure you have legal permission from the vendor. For example, if you are going to perform penetration testing on the Azure platform, you'll need legal permission from Microsoft.

In the next section, we will cover the essentials of physical penetration testing.

## Physical penetration testing

Physical penetration testing focuses on testing the physical security access control systems in place to protect an organization's data. Security controls exist within offices and data centers to prevent unauthorized persons from entering secure areas of a company.

Physical security controls include the following:

- **Security cameras and sensors:** Security cameras are used to monitor physical actions within an area.
- **Biometric authentication systems:** Biometrics are used to ensure that only authorized people are granted access to an area.
- **Doors and locks:** Locking systems are used to prevent unauthorized persons from entering a room or area.
- **Security guards:** Security guards are people who are assigned to protect something, someone, or an area.

Having completed this section, you are now able to describe the various types of penetration testing. Your journey ahead won't be complete without understanding the phases of hacking. The different phases of hacking will be covered in the next section.

## Hacking phases

During any penetration test training, you will encounter the five phases of hacking. These phases are as follows:

1. Reconnaissance
2. Scanning
3. Gaining access
4. Maintaining access
5. Covering tracks

In the following sections, we will describe each in detail.

## Reconnaissance or information gathering

The reconnaissance or information-gathering phase is where the attacker focuses on acquiring meaningful information about their target. This is the most important phase in hacking: the more details known about the target, the easier it is to compromise a weakness and exploit it.

The following are techniques used in the reconnaissance phase:

- Using search engines to gather information
- Using social networking platforms
- Performing Google hacking
- Performing DNS interrogation
- Social engineering

In this phase, the objective is to gather as much information as possible about the target. In the next section, we will discuss using a more directed approach, and engage the target to get more specific and detailed information.

## Scanning

The second phase of hacking is scanning. Scanning involves using a direct approach in engaging the target to obtain information that is not accessible via the reconnaissance phase. This phase involves profiling the target organization, its systems, and network infrastructure.

The following are techniques used in the scanning phase:

- Checking for any live systems
- Checking for firewalls and their rules
- Checking for open network ports
- Checking for running services
- Checking for security vulnerabilities
- Creating a network topology of the target network

This phase is very important as it helps us to create a profile of the target. The information found in this phase will help us to move onto performing exploitation on the target system or network.

## Gaining access

This phase can sometimes be the most challenging phase of them all. In this phase, the attacker uses the information obtained from the previous phases to exploit the target. Upon successful exploitation of vulnerabilities, the attacker can then remotely execute malicious code on the target and gain remote access to the compromised system.

The following can occur once access is gained:

- Password cracking
- Exploiting vulnerabilities
- Escalating privileges
- Hiding files
- Lateral movement

The gaining-access (exploitation) phase can at times be difficult as exploits may work on one system and not on another. Once an exploit is successful and system access is acquired, the next phase is to ensure that you have a persistent connection back to the target.

## Maintaining access

After exploiting a system, the attacker should usually ensure that they are able to gain access to the victim's system at any time as long as the system is online. This is done by creating backdoor access on the target and setting up a persistence reverse or bind connection between the attacker's machines and the victim's system.

The objectives of maintaining access are as follows:

- Lateral movement
- Exfiltration of data
- Creating backdoor and persistent connections

Maintaining access is important to ensure that you, the penetration tester, always have access to the target system or network. Once the technical aspect of the penetration test is completed, it's time to clean up on the network.

## Covering tracks

The last phase is to cover your tracks. This ensures that you do not leave any traces of your presence on a compromised system. As penetration testers, we would like to be as undetectable as possible on a target's network, not triggering any alerts while we remove any residual traces of the actions performed during the penetration test.

Covering tracks ensures that you don't leave any trace of your presence on the network, as a penetration test is designed to be stealthy and simulate real-world attacks on an organization.

## Summary

During the course of this chapter, we discussed the different types of hackers while outlining their primary characteristics. The various types of penetration tests and phases were covered, including an exploration of popular testing methodologies and approaches used in the cybersecurity industry.

You are now able to compare and contrast the different types of hackers. You have gained knowledge and understanding of various terms used within the cybersecurity industry, and you have got to grips with the importance of and different phases of penetration testing. You are able to distinguish between various types of penetration testing, such as network, web, and even cloud penetration testing.

In *Chapter 2, Setting Up Kali - Part 1*, and *Chapter 3, Setting Up Kali - Part 2*, we will be covering the steps involved in setting up your own virtual penetration testing lab for practicing and building your skill set. I hope this chapter has been helpful and informative for your studies and career.

## Questions

1. What type of hacker depends on instructions and tools created by others but does not understand the technical aspects of hacking?
2. What is the last phase of hacking?
3. Which penetration testing methodology is used on web applications?
4. What is the approach where the penetration tester has the least knowledge about the target?
5. What type of hacker is employed by a nation's government?

## Further reading

- **Penetration testing methodologies:** [https://www.owasp.org/index.php/Penetration\\_testing\\_methodologies](https://www.owasp.org/index.php/Penetration_testing_methodologies)
- **Penetration testing phases:** <https://www.imperva.com/learn/application-security/penetration-testing/>

# 2

## Setting Up Kali - Part 1

As a future ethical hacker and/or penetration tester, it is quite important when testing payloads or practicing hacking skills that you do not disrupt or cause any sort of harm or damage to other people's computers or network infrastructure, such as that of your organization. To elaborate further, we'll use a simple analogy. Imagine you work for a company called ACME (a fictional organization) and you're the network/system administrator. Your IT director has noticed you express an interest in cybersecurity and that you have significant potential in becoming a penetration tester or an ethical hacker. They, therefore, approve official training in penetration testing certification for you. Once the training has ended, access to the virtual labs through the **Authorized Training Centre (ATC)** is usually terminated, which poses a real challenge for you: how are you going to practice your hacking skills when the training course and lab access has ended? Another challenge is the fact that practicing hacking techniques on an organization's network is intrusive and illegal.

This brings us to the importance of building our own personal lab environment for practicing and improving our skill set. Furthermore, having our own penetration testing lab will allow us to try new attacks, tools, and techniques without worrying about being intrusive or creating a security breach in a company network. Most importantly, throughout this chapter, you will learn about the importance of building and designing a suitable penetration testing lab for practicing various hacking techniques on Windows and Linux operating systems.

In this chapter, we will cover the following topics:

- Lab overview
- Building our lab
- Setting up Kali Linux
- Installing Nessus
- Setting up Android emulators
- Installing Metasploitable 2

## Technical requirements

To follow along with the exercises in this chapter, please ensure that you have met the following hardware and software requirements:

- Oracle VM VirtualBox
- VMware Workstation Pro
- Kali Linux 2019.2
- Nessus vulnerability scanner
- Android operating system (x86 version 4.4-r4)
- Metasploitable 2

## Lab overview

In this section, we are going to discuss the methodology and components required for designing and setting up our own penetration testing lab. To build our lab, we are going to build a virtual lab infrastructure to ensure that we are able to save money, as opposed to having to buy physical computers and networking equipment.

In the following sections, we will begin our discussion on the importance of using virtualization in building our penetration testing lab environment, as virtualization plays an important role throughout this chapter and the remainder of the book. Afterward, we will dive into installing Kali Linux and creating a virtual network.

## Virtualization

In my experience as a student, instructor, and professional, when a person is embarking on their studies within the field of IT, that individual normally believes that a physical lab infrastructure is definitely required. To some extent, this is true, but there are many downsides associated with building a physical lab.

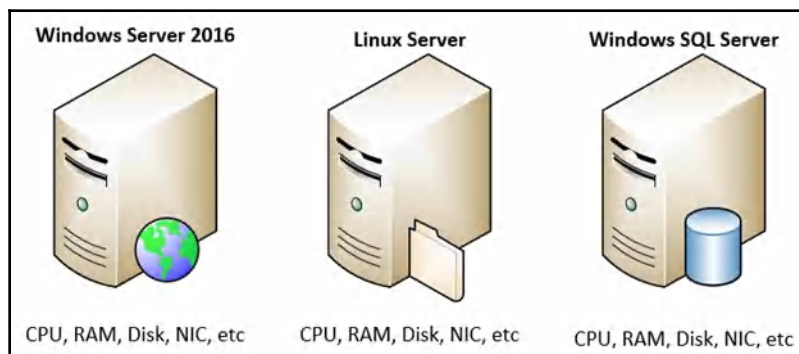
These downsides include, but are by no means limited to, the following:

- The physical space required to store the many servers and networking appliances that are needed.
- The power consumption per device will result in an overall high rate of financial expenditure.
- The cost of building/purchasing each physical device, whether it's a network appliance or a server.

These are just some of the primary concerns of a student or beginner. In most cases, a person has a single computer, be it a desktop or laptop machine. The conception of **virtualization**, emerging as a response to these downsides, opened a multitude of doors in IT and enabled many people and organizations to optimize and manage their hardware resources efficiently.

*What is virtualization and how is it helpful?* The concept of virtualization within the IT industry allows organizations to reduce the need for multiple items of physical equipment, such as servers and networking and security appliances. In the early days of IT, an operating system such as Microsoft Windows Server would need to be installed on a single physical device. Usually, a server-like device would consist of a high-end processor for the CPU, large amounts of RAM, and a lot of storage. However, there would be many times when the hardware resources (CPU and RAM) would be underutilized by the host operating system (Microsoft Windows Server). This wastage of resources is commonly known as **server sprawl**.

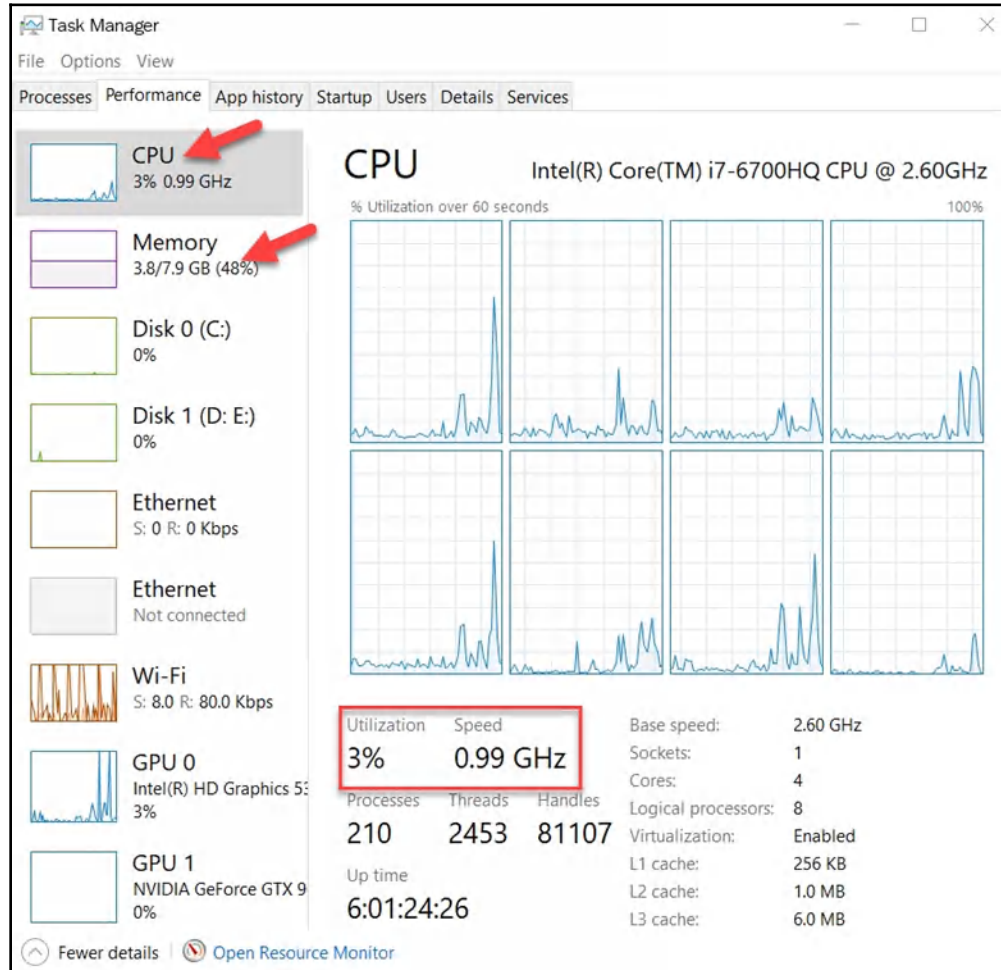
The following diagram shows three physical servers, each with their own host operating system and hardware resources available:



To quickly view the utilization of your resources on a Microsoft Windows operating system, simply open **Task Manager** and select the **Performance** tab. The following screenshot is a capture of my current device.



We can see that **CPU**, **Memory**, and other resources are currently underutilized; looking closely at the **CPU** and **Memory** graphs, we can see that they are not over 80%-90%, and less than 50% of their capacity is being used:



What if we were able to run multiple operating systems (such as Windows and Linux) on a single physical device? We could definitely utilize virtualization. This would enable us to better manage and efficiently maximize the resources available, using a component known as a **hypervisor**.

## Hypervisors

The hypervisor is the most important component in virtualization. It is responsible for creating an emulated environment that a guest operating system uses to function. Each type of operating system, irrespective of whether it is designed for a desktop, server, network, or mobile device, requires particular hardware components to ensure optimal and seamless functioning. This is where the hypervisor works its magic to make the impossible happen, allowing you to run multiple different operating systems on a single computer.

A hypervisor can be installed in one of two ways on a hardware device, which will be explored in more detail later in the chapter:

- It can be installed on top of a host operating system, such as Windows, Linux, or macOS.
- It can be installed directly on top of hardware in order to function as the native operating system.



A **host operating system** refers to the operating system that is installed directly on a device, such as a desktop or a laptop computer running Windows 10. A **guest operating system** is an operating system that is installed within a hypervisor (considered to be virtualized).

Listed here are the types of hypervisors available:

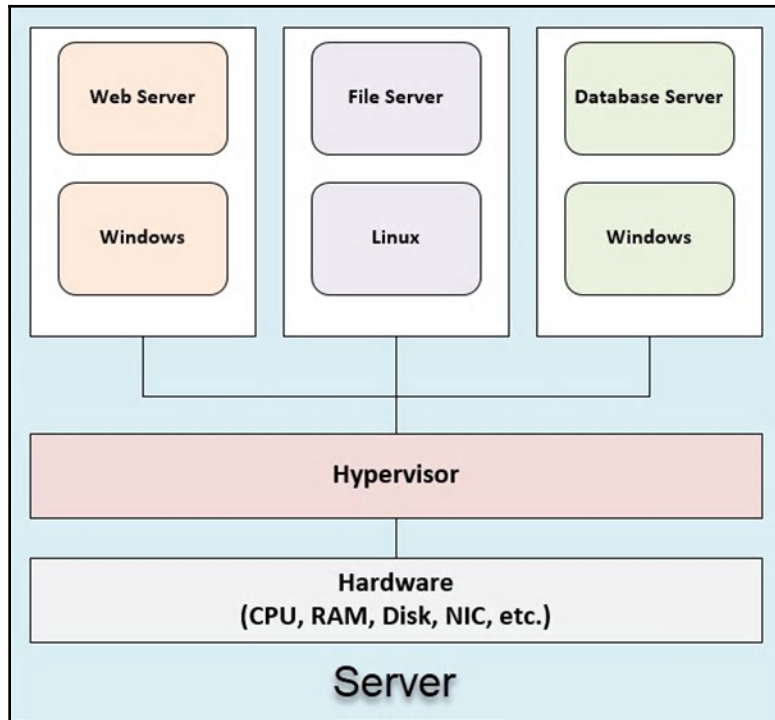
- Type 1
- Type 2

In the next two sections, we will look at the two types of hypervisors and understand their similarities and differences.

### Type 1 hypervisor

A type 1 hypervisor is sometimes referred to as a **bare-metal hypervisor** as it is typically deployed directly onto the hardware of the physical server. In this model, any operating system that is installed on the hypervisor has direct access to the hardware resources, such as the CPU, RAM, and **Network Interface Card (NIC)**. This model allows each guest operating system to interact directly with any hardware component available on the physical device; therefore, rendering the deployment model more efficient than the type 2 model.

The following diagram illustrates how each guest operating system (virtual machine) interacts with the physical hardware components of a single physical server chassis through the hypervisor. For example, virtual machines have direct access to the physical hardware through the hypervisor:



The following is a list of both free and commercial type 1 hypervisors:

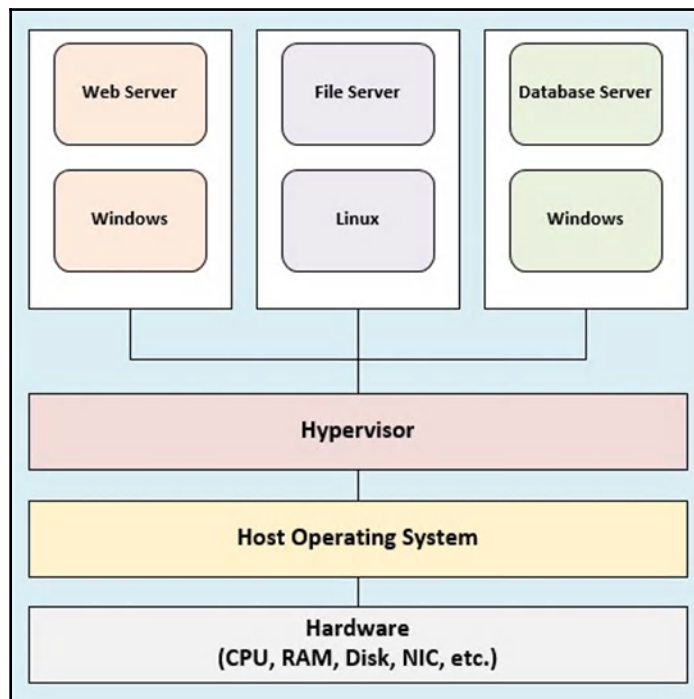
- VMware ESXi (free)
- VMware ESX (commercial)
- Microsoft Hyper-V Server (free)
- XCP-ng (free/commercial)

Now you have a better understanding of the type 1 hypervisor, let's learn about the type 2 hypervisor.

## Type 2 hypervisor

In a type 2 hypervisor deployment model, the hypervisor application is installed on top of a host operating system rather than on the hardware components directly. Examples of host operating systems include Microsoft Windows, the Apple macOS, and various flavors of Linux. The hypervisor does not have direct access to the hardware resources on the local system, as it would in the type 1 deployment model. Instead, the hypervisor in a type 2 deployment interfaces with the host operating system to access whatever resources are available. The host operating system usually requires a certain amount of resources, such as CPU and RAM utilization, in order to function optimally, and the remainder is then provided to the type 2 hypervisor for the guest virtual machines.

The following is a diagram illustrating how each component interfaces with the other on a single system, such as a desktop or laptop computer. Looking closely, each virtual machine has indirect access to the resources (CPU, memory, and so on). The operating system will have priority when it comes to hardware resources, and what is left is then made available to the running virtual machines:



The following is a brief list of type 2 hypervisors. Please note that some are free while others are commercial:

- Microsoft Virtual PC (free)
- Oracle VM VirtualBox (free)
- VMware Player (free)
- VMware Workstation Pro (commercial)
- VMware Fusion (commercial)
- Parallels Desktop for Mac (commercial)

You may be wondering which hypervisor is the better option—type 1 or type 2? Honestly, it really depends on your situation. Personally, I have a type 2 hypervisor installed on my laptop with a few virtual machines, which I use for training and other situations at remote locations. While at home, I have a type 1 hypervisor installed on an Intel NUC in my home lab, which has multiple virtual machines, each for a different purpose.

Now that you have a better idea of the concepts of hypervisors, let's learn about the features of a hypervisor, as that will help us to build a virtual network for creating our penetration testing lab.

## **Additional components**

In this section, we outline the additional components needed to complete our lab, including looking at what virtual switches are and the different types of operating systems we are going to use in the lab.

### **Virtual switches**

You may be wondering, since we are going to create a virtualized lab environment, how we are going to create a network to ensure that all the various virtual machines have connectivity with one another. Do we need some network cables, a network switch, or even other network appliances? Most importantly, we need to ensure that our virtual environment is isolated from the rest of our existing network and from the internet, as we do not want to be launching an inadvertent attack on a public server, as this would be illegal and entail legal complications.

Fortunately for us, each hypervisor contains a virtual switch, which provides us with a basic layer 2 switching functionality. Some hypervisors provide **virtual LAN (VLAN)** assignments on their virtual switches, while others do not. Since we are proceeding to build an isolated virtual lab, we'll need a single virtual switch to connect our attacker machine with the other vulnerable machines.

## Operating systems

As a future ethical hacker, penetration tester, or cybersecurity professional, it's recommended that you test various techniques to simulate real-world attacks on different types of operating systems. At times, when you are conducting a penetration test or performing a vulnerability assessment on an organization's network and servers, you will encounter many different operating systems. We will be using the following operating systems in our lab environment, and I'll provide a download link for each operating system:

- **Windows 10:** <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise>
- **Windows Server 2016:** <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2016>
- **Ubuntu Server:** <https://www.ubuntu.com/download/server>
- **Kali Linux:** <https://www.kali.org/downloads/>
- **Metasploitable:** <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>
- **OWASPBWA:** <https://sourceforge.net/projects/owaspbwa/>

Each operating system listed here has a unique purpose in our lab. In the remainder of this chapter, we will execute various types of attacks on each.

The Microsoft Evaluation Center (<https://www.microsoft.com/en-us/evalcenter/>) allows users to download and test drive any application and operating system available on their platform for a period of 180 days while providing full functional support for the application of your choice.



The **Open Web Application Security Project (OWASP)** (<https://www.owasp.org>) has created a virtual machine that allows cybersecurity professionals to execute various applications with known vulnerabilities; this is the **OWASP Broken Web Applications (OWASPBWA)** virtual machine. Metasploitable is a vulnerable Linux-based virtual machine created by Rapid7 (<https://www.rapid7.com>). Its objective is to help people learn about, and practice, penetration testing in a safe environment.

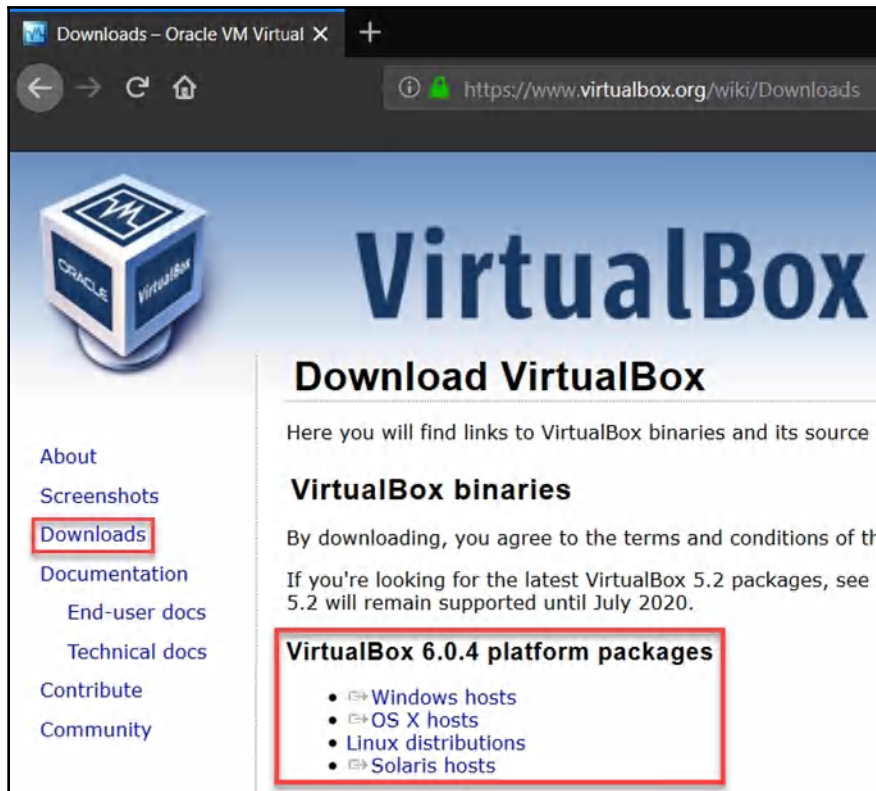
In this section, we covered the essentials of virtualization—including the core component, the hypervisor—and we are now ready to build virtual lab environments to support many operating systems and use cases. In the next section, we will be looking at putting all the pieces together and building our lab.

## Building our lab

Now it's time to assemble all the components and configure our own penetration testing lab. We'll need to decide what resources are currently available to us before choosing a type of hypervisor. If you currently have a single laptop or desktop computer, we'll be using a type 2 hypervisor, such as Oracle VM VirtualBox or VMware Workstation Pro. As mentioned previously, a type 2 hypervisor deployment will allow us to use our existing resources, such as a single laptop or desktop computer, to build our virtual lab environment, without being concerned about purchasing additional hardware components such as servers.

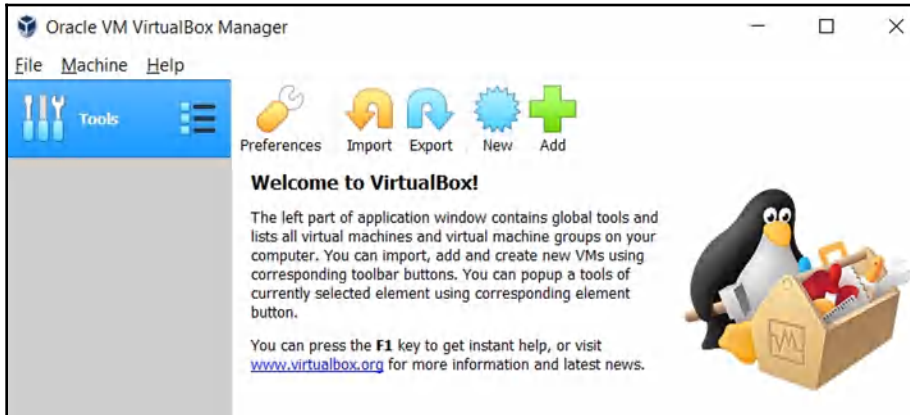
To begin installing our hypervisor, let's download and install Oracle VM VirtualBox:

1. Go to [www.virtualbox.org](https://www.virtualbox.org), then navigate to the **Downloads** section of the website, and choose your platform type based on your current operating system:

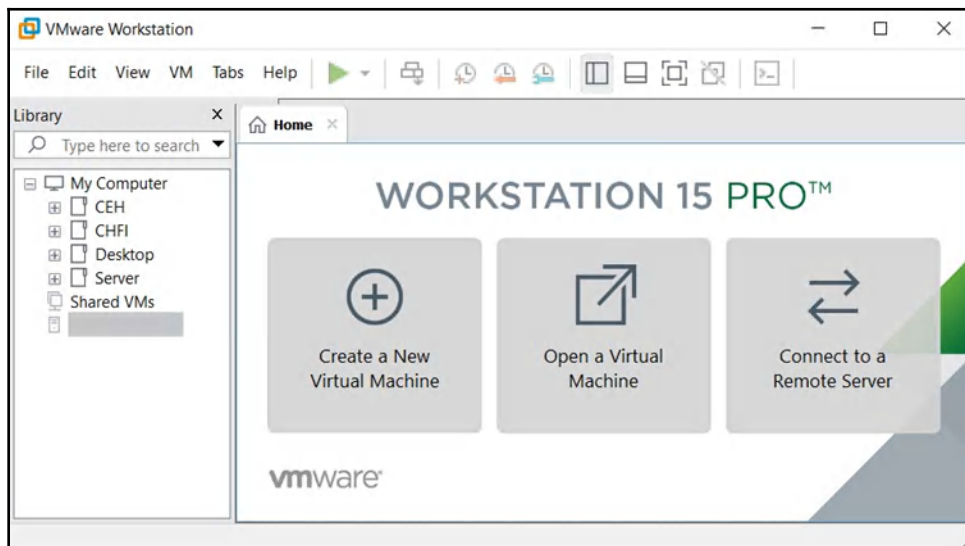




2. Once the application has been downloaded, it's time to install it. Be sure to use the default configurations presented during the installation wizard. Once completed, open VirtualBox to ensure that the installation was successful. You should be presented with something similar to the following screenshot:



3. Optionally, if you prefer using VMware Workstation for your lab, it is currently available at <https://www.vmware.com/products/workstation-pro.html>. Once downloaded, proceed to install the application using the default configuration during the installation process. Once completed, you should be presented with the user interface, as shown in the following screenshot:



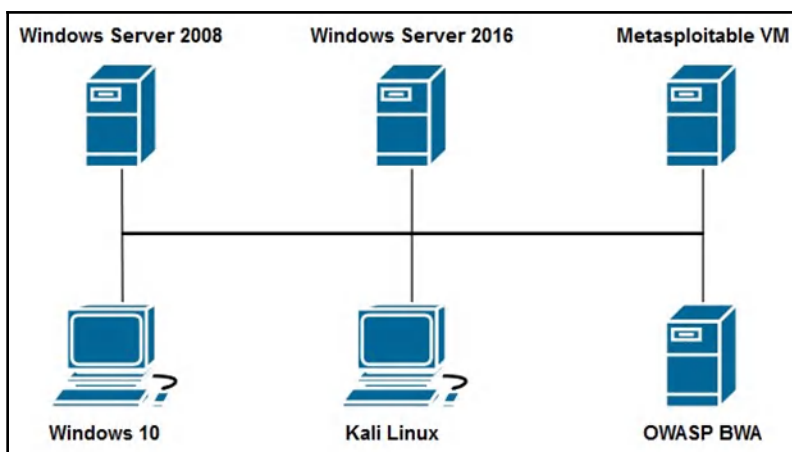


If you are using an older version of Oracle VM VirtualBox or VMware Workstation, you do not need to upgrade, as the previous editions already contain the features required to continue configuring our lab.

One of the most important things about designing a proper penetration testing lab is ensuring that we have the optimal network design for interconnecting our virtual machines. In the next section, we will cover in detail how to create a virtual network using both Oracle VM VirtualBox and VMware Workstation Pro.

## Creating a virtual network

The following diagram shows the general network topology we are going to use in our virtual lab environment:



In the upcoming section, we will assign an appropriate IP address to each virtual machine within our lab. Each virtual machine is interconnected using a virtual switch within the hypervisor. Routers are not required, as this is just a simple lab design.



A Windows Server 2008 machine is optional and is not required.

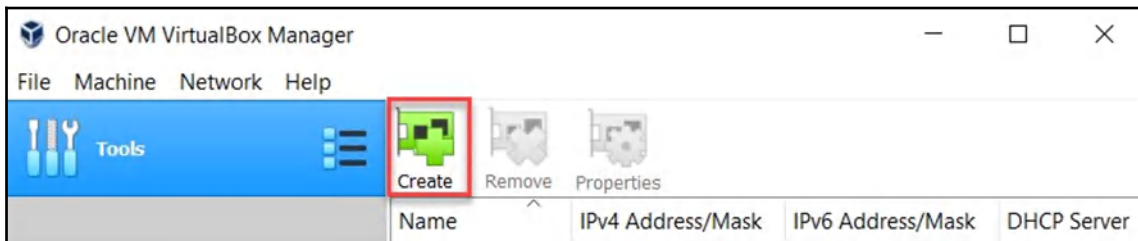
Let's see how to build a virtual network:

1. If you're using VirtualBox, click on the menu icon on the right-hand side of **Tools** | **Network**:



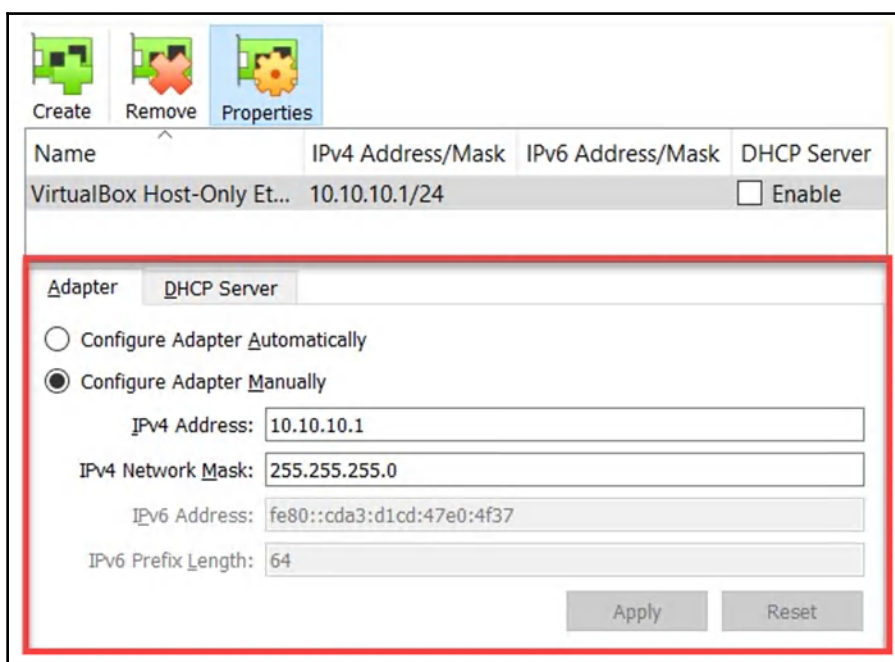
A new window will open, giving you the option to **Create**, **Remove**, or modify the properties of a virtual network adapter. In this exercise, we are going to create a new virtual adapter, which will be used to connect each of our virtual machines within the hypervisor. This accomplishes the effect of a virtual switch.

2. Click on **Create** to add a new virtual adapter:

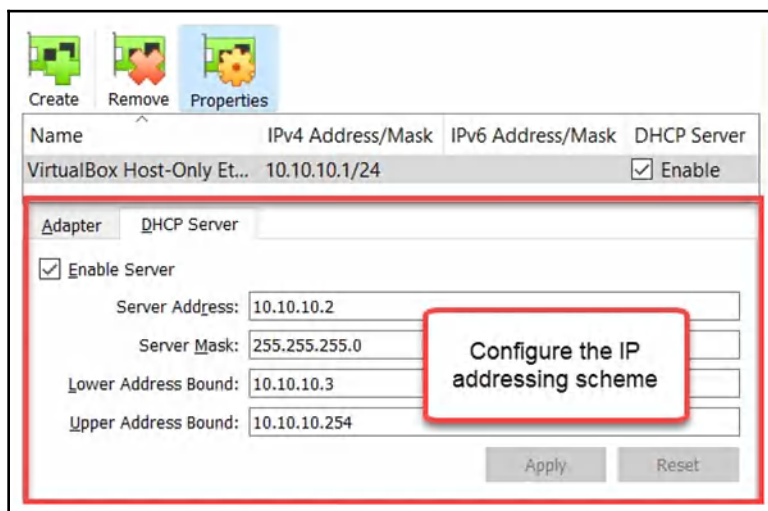


Your host operating system will take a few minutes to create the new virtual network adapter on your computer.

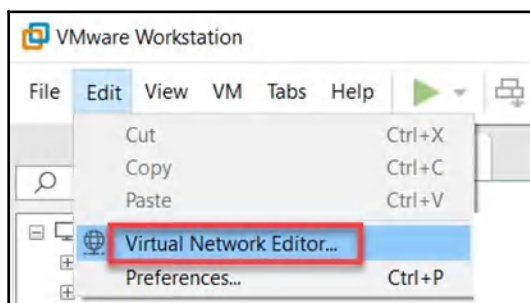
3. Once the virtual network adapter has been created, the network manager component within VirtualBox will automatically assign an IP address to the interface. However, we are going to configure the IP addressing scheme as per our preferences. To begin, simply select the virtual network adapter, and then click on **Properties** to modify the configurations.
4. Ensure that you choose the option to configure the adapter manually, using the IP address and subnet mask shown in the following screenshot. Click **Apply** to register the configurations on the network adapter:



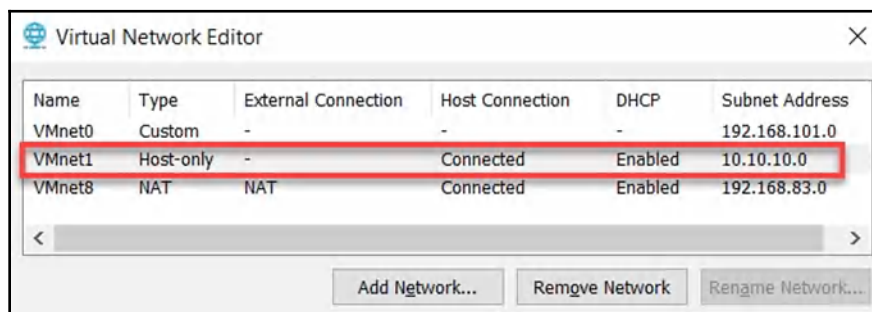
- Optionally, we can configure the **Dynamic Host Configuration Protocol (DHCP)** server on the virtual network adapter to provide a range of IP addresses to each virtual machine that is connected to this virtual network. If you would like to enable the DHCP service, please use the following configurations:



- For those of you who prefer VMware Workstation, we've got you covered. Configuring a virtual network within VMware Workstation is quite simple. Open the VMware Workstation application and select **Edit | Virtual Network Editor...**, as shown in the following screenshot:

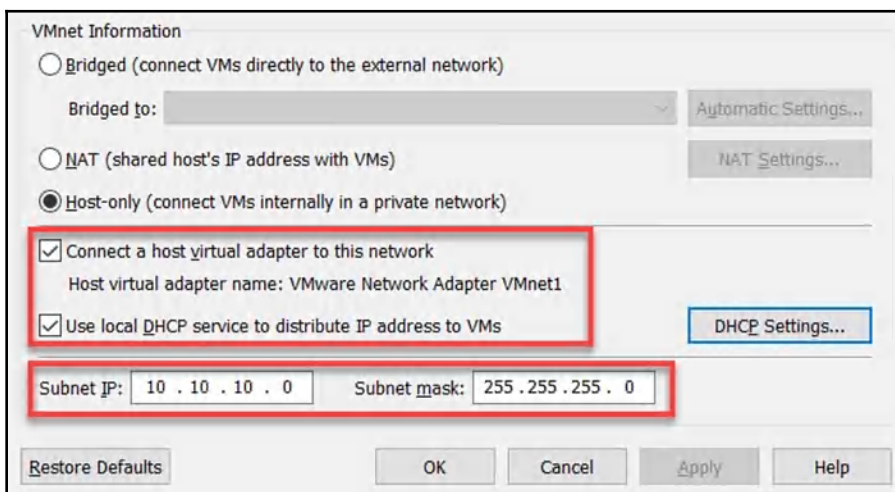


7. The **User Access Control (UAC)** on Windows will prompt you for administrator privileges. Upon providing the authorization, the **Virtual Network Editor** window will open. As you can see, there are three virtual network adapters present:



We are going to modify the **VMnet1** virtual adapter. The host-only adapter creates a virtual network for all connected virtual machines and the host computer. This type of configuration allows all virtual machines to communicate seamlessly while isolated and in the absence of an internet connection.

8. To modify the **VMnet1** adapter, select the adapter and adjust your configurations, as shown in the following screenshot:



These configurations replicate those executed previously within Oracle VM VirtualBox.

Now that we have the knowledge required to build our virtual network using both Oracle VM VirtualBox and VMware Workstation Pro, let's begin installing virtual machines and setting up Kali Linux in our lab.

## Setting up Kali Linux

Let's set up our first virtual machine, our attacker machine, Kali Linux. The Kali Linux operating system is a Debian-based Linux platform consisting of over 300 tools for both penetration testing and forensics. It's one of the most popular platforms used by penetration testers as it contains many features and considerable functionality, such as the following:

- Full-disk encryption
- Support for **Linux Unified Key Setup (LUKS)** encryption with emergency self-destruction (Nuke)
- Accessibility features
- Forensics mode
- Live USB with multiple persistence

To get started, Kali Linux can be found at the official website ([www.kali.org](http://www.kali.org)) and at the Offensive Security domain (<https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>). There are many methods when it comes to setting up Kali Linux, such as installing from an ISO file and importing a virtual preconfigured image into a hypervisor. For our setup procedure, we are going to use the latter approach. Importing the virtual appliance is seamless and takes very little time; it also avoids the chances of misconfiguration that come with installation using an ISO file.

In my personal experience, setting up Kali Linux using the preconfigured virtual image also works more efficiently in most situations. To get started, we can take the following steps:

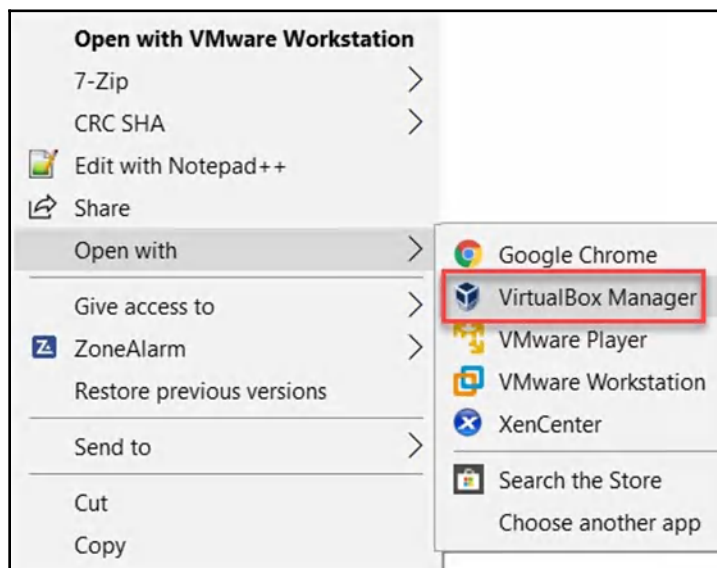
1. Navigate to <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/> and download either the 32-bit or 64-bit Kali Linux VMware image, based on your operating system architecture. Choose either the VMware or the VirtualBox image, based on the vendor of your hypervisor software:

Kali Linux VMware Images		Kali Linux VirtualBox Images		
Image Name	Torrent	Size	Version	SHA256Sum
<b>Kali Linux VMware 64-Bit 7z</b>	Torrent	2.4G	2019.2	4611f3797c53ed37c89443bd8bb94ac1fd860fb807865d8933783c0f6ef21007
<b>Kali Linux VMware 32-Bit 7z</b>	Torrent	2.5G	2019.2	c7f52865f5d0554ad1bc990684a0751eb46d1b8ab552d7c942d71e4fe20b7e67

- Whether you've downloaded the VirtualBox or VMware image, ensure that you unzip the contents. If you've downloaded the VirtualBox image, within the folder there will be a file with a similar naming convention, as shown in the following screenshot:

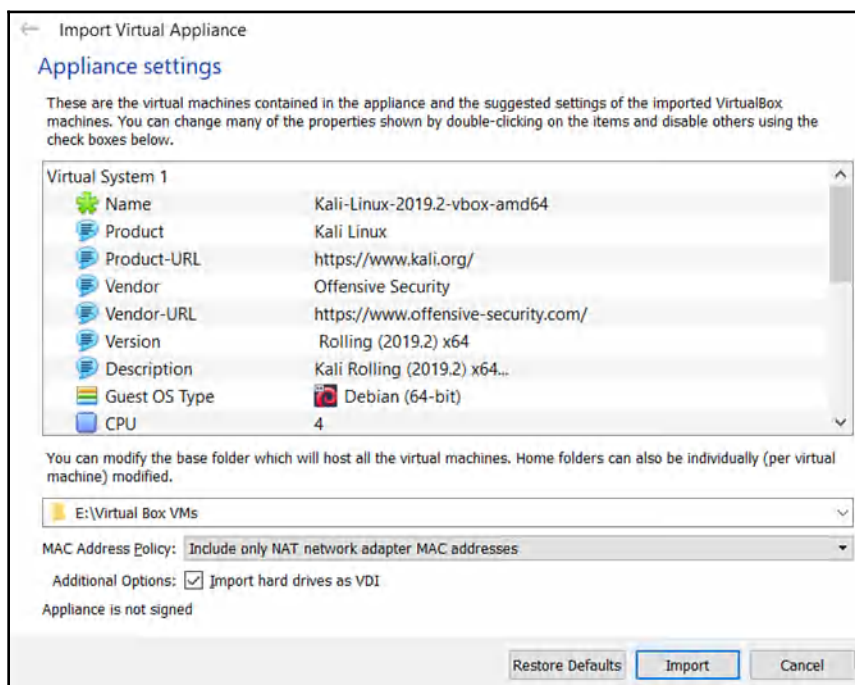


- You can right-click on **File** and choose **Open with | VirtualBox Manager**:

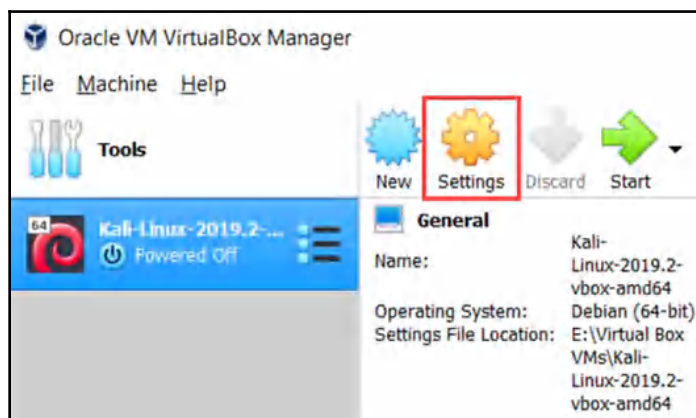











- Next, the **Import Virtual Appliance** wizard will appear. Simply click on **Import**. The importing process will take a few minutes to complete:



Once the importing process is complete, you'll see your new virtual machine available on the VirtualBox dashboard:



5. Import Kali Linux into VMware Workstation. Ensure that you've downloaded and unzipped the virtual image folder. The following are the contents of the extracted folder. Right-click on the highlighted file shown in the following screenshot, and choose **Open with | VMware Workstation**:

Name	Date modified	Type	Size
 Kali-Linux-2019.2-vmware-amd64	17-May-19 1:30 PM	VMware Virtual Machine nonvolatile RAM	9 KB
 Kali-Linux-2019.2-vmware-amd64	17-May-19 12:17 P...	VMDK File	2 KB
 Kali-Linux-2019.2-vmware-amd64	17-May-19 11:47 ...	VMware snapshot metadata	0 KB
 Kali-Linux-2019.2-vmware-amd64	17-May-19 1:31 PM	VMware virtual machine configuration	4 KB
 Kali-Linux-2019.2-vmware-amd64	17-May-19 11:47 ...	VMware Team Member	1 KB
 Kali-Linux-2019.2-vmware-amd64-s001	17-May-19 1:31 PM	VMDK File	2,602,176 ...
 Kali-Linux-2019.2-vmware-amd64-s002	17-May-19 1:31 PM	VMDK File	187,904 KB

6. VMware Workstation will then open, providing the **Import Virtual Machine** window. Click on **Import**:

Import Virtual Machine

Store the new Virtual Machine

Provide a name and local storage path for the new virtual machine.

Name for the new virtual machine:

kali-linux-vm-amd64

Storage path for the new virtual machine:

E:\VMWare Machines\kali-linux-vm-amd64

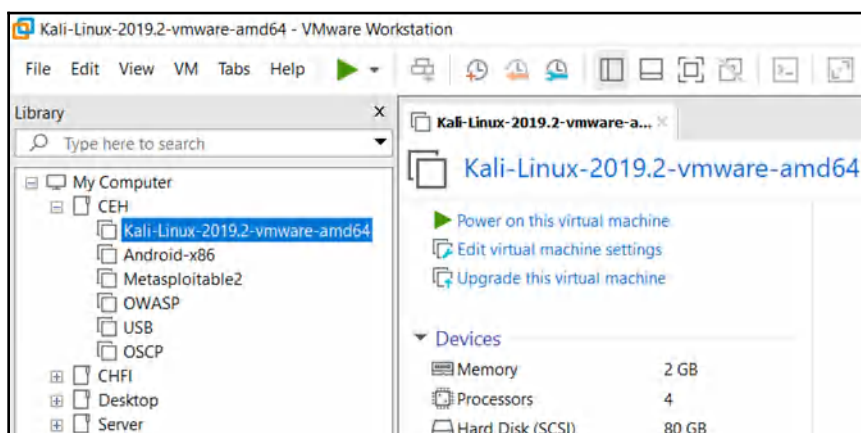
Browse...

Help

Import

Cancel

This process should take a few minutes to complete. Once complete, the new virtual machine will be available in your library on VMware Workstation:



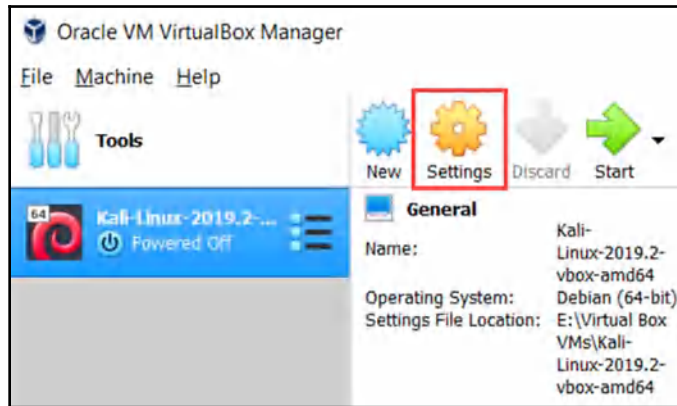
The benefit of importing a virtual image over manually installing an operating system using an ISO image is that all the configuration is done automatically. Configuration would include creating a virtual hard drive for storage and the assignment of resources, such as a processor, RAM, and NIC. Importing a virtual image eliminates the chances of any misconfigurations during the installation phase. Once the importing phase is complete, the user can subsequently make adjustments to the individual virtual machine, such as increasing or reducing the resources per virtual machine.

## Attaching the virtual network to a virtual machine

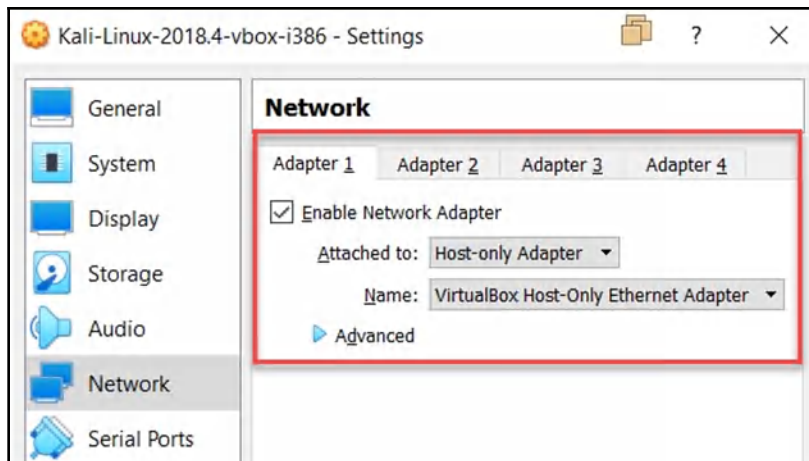
At this point, we have created our virtual network adapter and imported Kali Linux into our hypervisor. It's now time to attach our attacker machine, Kali Linux, to our virtual network (virtual switch).

Firstly, I'll guide you through the steps to configuring the hardware resources through Oracle VM VirtualBox:

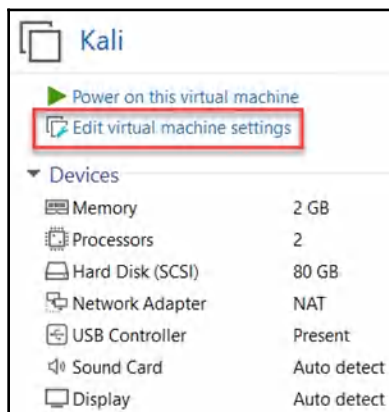
1. Select the Kali Linux virtual machine and click on **Settings**:



2. Once the settings window has opened, select the **Network** option. Here, you'll be able to enable/disable network adapters on the current virtual machine. Select the **Host-only Adapter** option, and the virtual network adapter will be automatically selected underneath:

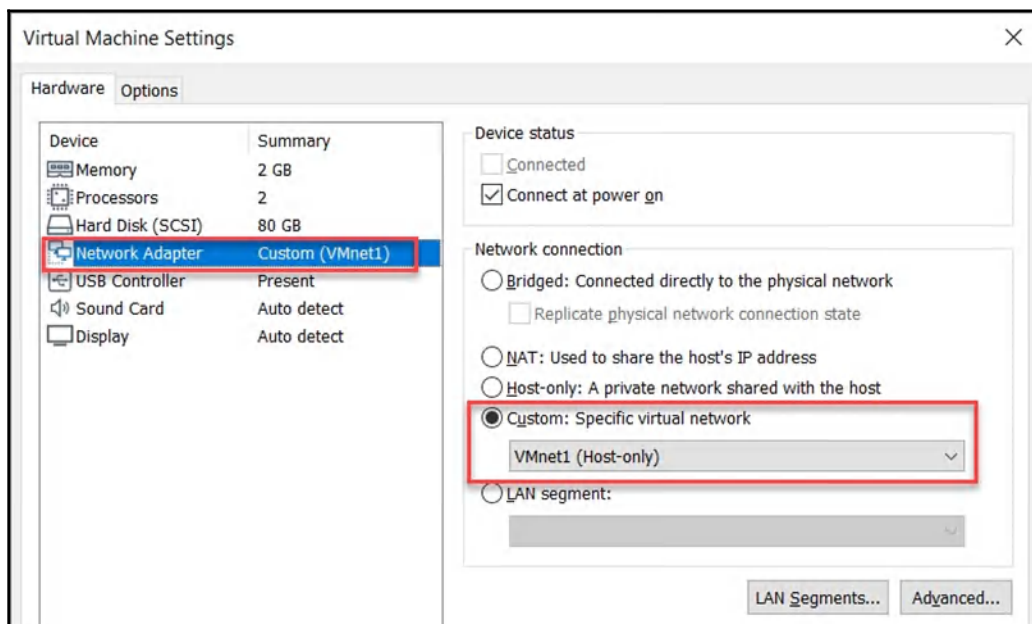


- Next, we are going to make the same adjustments on VMware Workstation. Firstly, click on **Edit virtual machine settings** on the Kali Linux virtual machine:



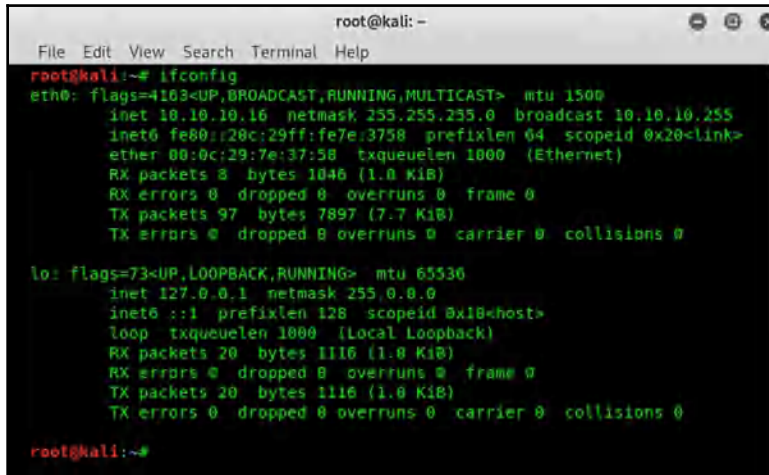
The **Virtual Machine Settings** window will open. Here, you can customize the settings on any hardware component within the hypervisor menu.

- Select **Network Adapter**, and then choose **Custom: Specific virtual network | VMnet1 (Host-only)**:



Remember that the VMnet1 adapter has our custom IP scheme.

5. We can power on our Kali Linux virtual machine to ensure that it is working properly. The default username/password for Kali Linux is `root/toor`.
6. Once you've successfully logged in, you'll have access to the desktop:



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.10.10.16 netmask 255.255.255.0 broadcast 10.10.10.255  
    inet6 fe80::20c:29ff:fe7e:3758 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:7e:37:58 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 1046 (1.0 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 97 bytes 7897 (7.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 20 bytes 1116 (1.0 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 20 bytes 1116 (1.0 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
root@kali:~#
```

Now we have a clear understanding of how to set up a virtual machine within Oracle VM VirtualBox and VMware Workstation, and how to configure a virtual network within each hypervisor application. Let's move on to setting up additional applications and other types of virtual machines within our lab.

## Installing Nessus

When you get into the field of penetration testing and vulnerability assessment, one tool you must be familiar with using is **Nessus**. Nessus is one of the most popular vulnerability assessment tools available on the market. The Nessus application is controlled using a web interface that allows its users to create customized scans. Additionally, Nessus contains prebuilt scanning templates for various types of industries, such as the **Payment Card Industry (PCI)** compliance scanner.

Tenable, the creator of Nessus, has indicated that Nessus is capable of detecting over 47,000 **common vulnerabilities and exposures (CVE)**. As a future ethical hacker/penetration tester, using Nessus during your security auditing phase will aid you significantly in discovering security vulnerabilities quickly.

Nessus is supported on many platforms, such as Windows and Kali Linux. The **Nessus Home** edition is free for personal use, and is capable of scanning up to 16 IP addresses per scan. To get the Nessus Home edition, simply go to <https://www.tenable.com/products/nessus-home> and complete the registration form in order to obtain an activation license. After registration, you'll be redirected to the download center, where you can choose a suitable version for your platform:

The screenshot shows the Nessus Home registration page. The header includes the Tenable logo and navigation links: Downloads, Blog, Contact, Login, and Global. Below the header, there are buttons for 'Free Trial' and 'Buy Now'. The main heading is 'Nessus Home'. The text below the heading states: 'Nessus® Home allows you to scan your personal home network (up to 16 IP addresses per scanner) with the same high-speed, in-depth assessments and agentless scanning convenience that Nessus subscribers enjoy.' It also includes a disclaimer: 'Please note that Nessus Home does not provide access to support, allow you to perform compliance checks or content audits, or allow you to use the Nessus virtual appliance. If you require support and these additional features, please purchase a Nessus subscription.' A note at the bottom left says: 'Nessus Home is available for personal use in a home environment only. It is not for use by any commercial organization.' The registration form on the right, highlighted with a red border, contains the following fields and options:

- Register for an Activation Code**
- First Name \*** (text input)
- Last Name \*** (text input)
- Email \*** (text input)
- ☐ Check to receive updates from Tenable
- Register** (button)

If you're installing Nessus on a Windows operating system, the procedure is quite simple. Download the Windows executable file, and run it.

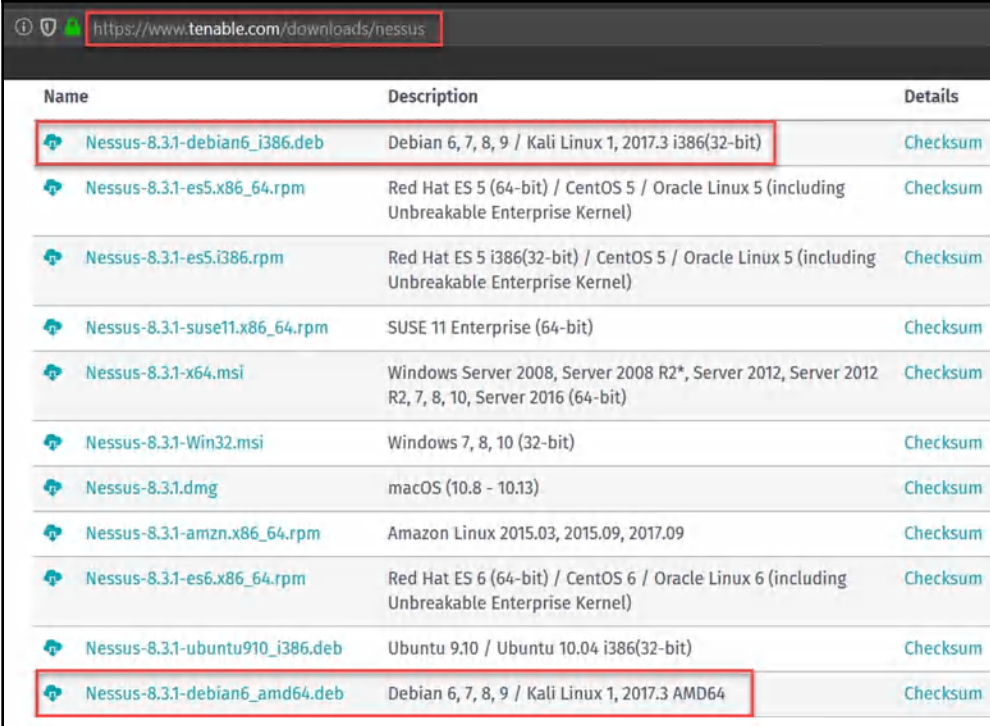
However, to install Nessus on Kali Linux, follow these steps:

1. Open Terminal and run the following commands to upgrade all currently installed applications on the platform:

```
apt-get update && apt-get upgrade
```

2. Obtain an activation code from Tenable by completing the registration form at <https://www.tenable.com/products/nessus/nessus-essentials>.
3. Navigate to the Nessus **Downloads** page at <https://www.tenable.com/downloads/nessus> and download either the 32-bit or the 64-bit version, based on your operating system architecture:





Name	Description	Details
Nessus-8.3.1-debian6_i386.deb	Debian 6, 7, 8, 9 / Kali Linux 1, 2017.3 i386(32-bit)	Checksum
Nessus-8.3.1-es5.x86_64.rpm	Red Hat ES 5 (64-bit) / CentOS 5 / Oracle Linux 5 (including Unbreakable Enterprise Kernel)	Checksum
Nessus-8.3.1-es5.i386.rpm	Red Hat ES 5 i386(32-bit) / CentOS 5 / Oracle Linux 5 (including Unbreakable Enterprise Kernel)	Checksum
Nessus-8.3.1-suse11.x86_64.rpm	SUSE 11 Enterprise (64-bit)	Checksum
Nessus-8.3.1-x64.msi	Windows Server 2008, Server 2008 R2*, Server 2012, Server 2012 R2, 7, 8, 10, Server 2016 (64-bit)	Checksum
Nessus-8.3.1-Win32.msi	Windows 7, 8, 10 (32-bit)	Checksum
Nessus-8.3.1.dmg	macOS (10.8 - 10.13)	Checksum
Nessus-8.3.1-amzn.x86_64.rpm	Amazon Linux 2015.03, 2015.09, 2017.09	Checksum
Nessus-8.3.1-es6.x86_64.rpm	Red Hat ES 6 (64-bit) / CentOS 6 / Oracle Linux 6 (including Unbreakable Enterprise Kernel)	Checksum
Nessus-8.3.1-ubuntu910_i386.deb	Ubuntu 9.10 / Ubuntu 10.04 i386(32-bit)	Checksum
Nessus-8.3.1-debian6_amd64.deb	Debian 6, 7, 8, 9 / Kali Linux 1, 2017.3 AMD64	Checksum

- Once Nessus has been downloaded on Kali Linux, open Terminal, change the directory to the Downloads folder, and begin installation using the following command:

```
dpkg -i Nessus-8.3.1-debian6_amd64.deb
```

The output of running the preceding command is as follows:

```
root@kali:~# cd Downloads
root@kali:~/Downloads# dpkg -i Nessus-8.3.1-debian6_amd64.deb
(Reading database ... 367983 files and directories currently installed.)
Preparing to unpack Nessus-8.3.1-debian6_amd64.deb ...
Shutting down Nessus : .
Unpacking nessus (8.3.1) over (8.3.1) ...
Setting up nessus (8.3.1) ...
Unpacking Nessus Scanner Core Components...

- You can start Nessus Scanner by typing /etc/init.d/nessusd start
- Then go to https://kali:8834/ to configure your scanner

Processing triggers for systemd (241-1) ...
root@kali:~/Downloads#
```



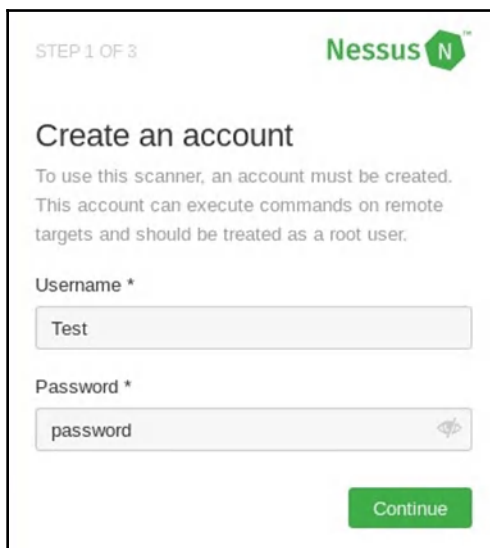
5. Once the installation is complete, use the following command to start the Nessus service on Kali Linux:

```
/etc/init.d/nessusd start
```

Optionally, if you would like the Nessus service to start automatically during the Kali Linux boot process, the following command can be used to enable this feature:

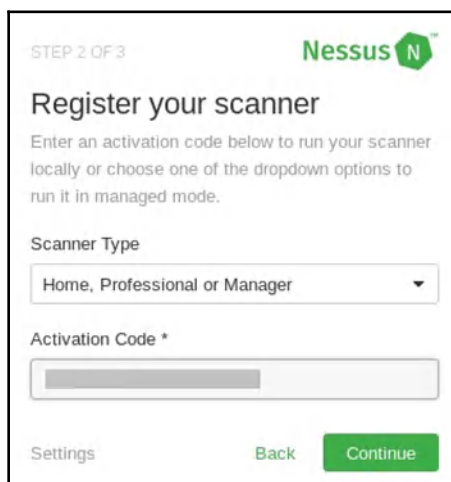
```
update-rc.d nessusd enable
```

6. Once the installation has been completed on Kali Linux, enter `https://localhost:8834/` into your web browser. At this point, you'll be prompted to create a user account:




The screenshot shows the Nessus web interface at the 'STEP 1 OF 3' stage. The title is 'Create an account'. Below the title, it states: 'To use this scanner, an account must be created. This account can execute commands on remote targets and should be treated as a root user.' There are two input fields: 'Username \*' with the value 'Test' and 'Password \*' with the value 'password'. A green 'Continue' button is at the bottom right.

7. Next, you'll be prompted to enter your Nessus license to activate the product. You'll need the activation code from *step 2* to complete this stage:



STEP 2 OF 3

Nessus 

## Register your scanner

Enter an activation code below to run your scanner locally or choose one of the dropdown options to run it in managed mode.

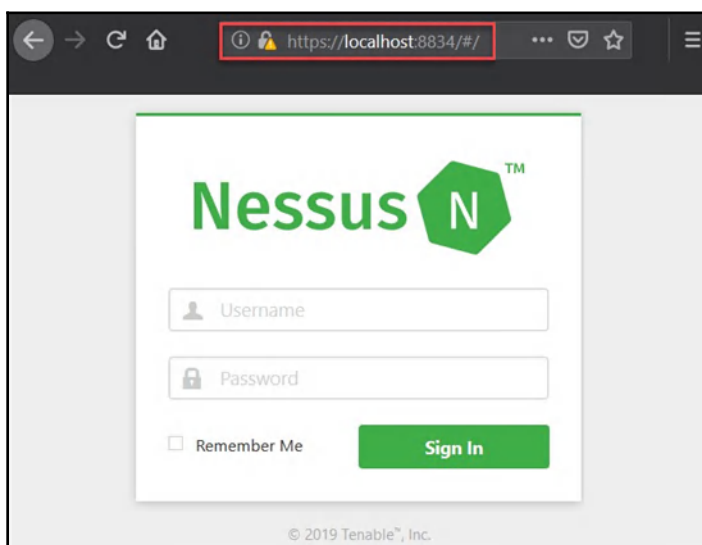
Scanner Type

Home, Professional or Manager


Activation Code \*

Settings Back Continue

8. After completing the activation phase, Nessus will attempt to connect to the internet to download additional resources. This process should take a few minutes to complete:



Browser address bar: <https://localhost:8834/#/>

Nessus 

Username

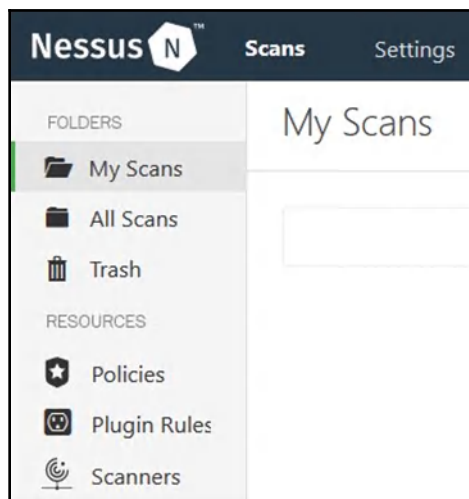
Password

☐ Remember Me

Sign In

© 2019 Tenable™, Inc.

9. Once you're logged in, your user dashboard will be available. Here, you can create new scans and templates and modify existing resources as per your preferences:



During the course of this book, we'll be exploring the capabilities of Nessus during our penetration testing phases.

Having completed this section, you are now able to install and set up the Nessus vulnerability scanner on Kali Linux. In the next section, you will learn how to install Android as a virtual machine within your lab environment.

## Setting up Android emulators

Being a penetration tester and/or ethical hacker, you'll encounter a lot of different types of targets and operating systems in the field. One type of operating system that has made its way into the field of cybersecurity is the mobile platform Android. In this section, we will discover how to set up the Android operating system version 4.4 as a virtual machine that will be part of your penetration testing lab environment.

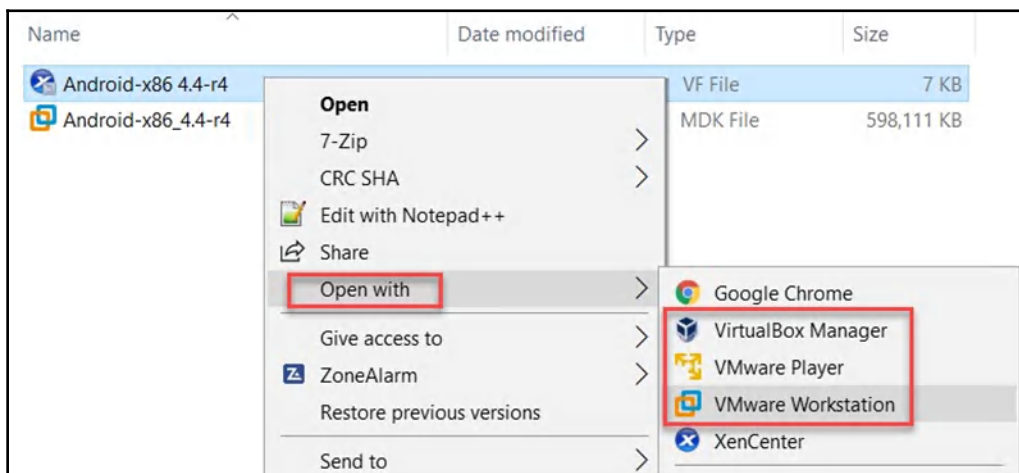
Note that [www.osboxes.org](http://www.osboxes.org) has a repository of virtual images of almost every type of operating system, including desktop, server, and even mobile operating systems. This website allows you to download a virtual image of your choice and simply load it seamlessly into a hypervisor, such as Oracle VM VirtualBox or VMware Workstation.

Let's learn how to create a virtual Android machine within your penetration testing lab:

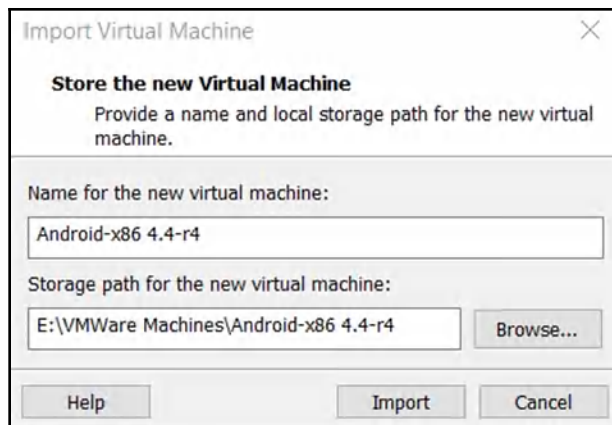
1. Firstly, go to <https://www.osboxes.org/android-x86/> to download the Android mobile operating system for your lab.
2. Search for the **Android-x86 4.4-r4** version and download either the **VirtualBox** or the **VMware** virtual image for your hypervisor:



3. Once the file has been downloaded onto your desktop computer, extract the zipped folder to view the contents.
4. Next, right-click on the `.ovf` file and choose the **Open with** option, then select the VMware or VirtualBox options, as shown in the following screenshot:



5. The import wizard will appear. Select **Import** to begin the process:



The importing process takes a few minutes to complete and the new Android virtual machine will appear in your hypervisor library.

6. I have chosen to use the following configuration on my Android virtual machine. However, you have the option to either increase or decrease the resources on your virtual machine as you see fit. Ensure that the virtual network adapter is assigned to **Custom (VMnet1)**, as shown in the following screenshot:








7. After booting your Android virtual machine, you'll be presented with an interface once it's fully loaded. The full functionality of Android 4.4 is available within your virtual machine.

Once the Android virtual machine is powered on, it acts as a real, physical Android device on your lab network. This simulates an environment that not only has typical operating systems, such as Windows and Linux, but also mobile platforms, such as Android. Now that you have a virtual Android machine within your lab, let's take a look at setting up a vulnerable Linux-based virtual machine in the next section.

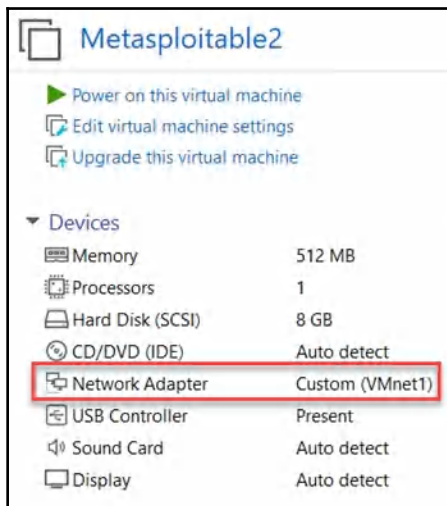
## Installing Metasploitable 2

As mentioned previously, the Metasploitable virtual machine was created by the team at Rapid7 ([www.rapid7.com](http://www.rapid7.com)) for the purpose of cybersecurity awareness and training. In this section, I'll walk you through the steps involved in setting up a Metasploitable virtual machine in your lab:

1. Firstly, you need to download the virtual image file from <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>. Once downloaded to your computer, extract the ZIP folder to view the contents.
2. Next, right-click on the following highlighted file, and either choose the option to import or open with a hypervisor of your choice:

Name	Date modified	Type	Size
 Metasploitable	11-Jan-19 4:26 PM	VMware Virtual Machine nonvolatile RAM	9 KB
 Metasploitable	01-Mar-19 6:13 PM	VMDK File	1,901,184 ...
 Metasploitable	11-Jan-19 4:26 PM	VMware snapshot metadata	1 KB
 Metasploitable	11-Jan-19 4:26 PM	VMware virtual machine configuration	3 KB
 Metasploitable	11-Jan-19 4:26 PM	VMware Team Member	1 KB

3. Once the importing process has finished, the new virtual machine will appear in your library in the hypervisor (VirtualBox or VMware). Ensure that the network adapter is set to **Custom (VMnet1)**, just like the virtual network for our lab is:



4. To test the virtual machine, power it on, and let it boot. Once the boot process is complete, you'll see that the login credentials (username/password) are part of the system banner, `msfadmin/msfadmin`:

```
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]
```

\_\_\_\_\_

/ \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \ / \_ \_ \

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|

|\_|

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login:

5. Log in to the virtual machine using the credentials, and use the `ifconfig` command to verify that it has a valid IP address:

```
No mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

nsfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:64:71:7f
          inet addr:10.10.10.129  Bcast:10.10.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe64:717f/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3339 (3.2 KB)  TX bytes:5066 (4.9 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:99 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21713 (21.2 KB)  TX bytes:21713 (21.2 KB)

nsfadmin@metasploitable:~$ _
```



For each virtual machine, ensure that you have taken a record of the IP addresses. The IP addresses I'll be using during the remaining chapters of the book may be a bit different to yours, but the operating systems and virtual machine configurations will be the same.

You now have a vulnerable Linux-based operating system in your lab. It's always recommended to have a mixture of various target operating systems in a lab when practicing penetration testing techniques and honing your skills. This method allows you to learn how to perform attacks on a variety of different targets, which is important since a corporate network usually has a blend of many different devices and operating systems. You don't want to be in a penetration test engagement where the target organization has mostly Linux devices but you're skills are geared toward only Windows-based systems; this would be a bad sign for you as a penetration tester! Therefore, emulating a corporate network in a lab as closely as possible will help you to improve your skills.



## Summary

In this chapter, we opened with a discussion of the importance of having our own isolated lab environment for practicing offensive security training. We delved into the concepts of virtualization and looked at how it's going to help us now and in the future. Later in the chapter, we covered configuring a virtual network on both Oracle VM VirtualBox and VMware Workstation, as these networks will be used to interconnect all of our virtual machines (our attacker and victim machines). We then walked through deploying Kali Linux and Android to our penetration testing lab.

Now that we have a fundamental understanding of designing and building our lab environment, let's continue deploying both Windows and Linux-based operating systems in our next chapter.

## Questions

1. Which type of hypervisor is installed on top of a host operating system?
2. What are some of the benefits of virtualization?
3. What are some examples of free hypervisors?
4. How is an offline package/application installed in Kali Linux?
5. What is an operating system within a hypervisor usually called?

## Further reading

The following links are recommended for additional reading:

- **Kali Linux documentation:** <https://docs.kali.org/>
- **Nessus user guide:** <https://docs.tenable.com/Nessus.htm>
- **Virtualization:** <https://www.networkworld.com/article/3234795/what-is-virtualization-definition-virtual-machine-hypervisor.html>