Learn Penetration Testing with Python 3.x

An ethical hacker's blueprint for offensive security

2nd Edition

Yehia Elghaly

www.bpbonline.com

Second Revised and Updated Edition 2024

First Edition 2022

ISBN: 978-93-55519-436

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My wife ***Faten Hmem***

My father ***Mamdouh Elghaly***

and

My daughter ***Elina Elghaly***

About the Author

Yehia Elghaly brings over 11 years of experience in offensive cybersecurity and red teaming. He has successfully led more than 200 projects across various sectors, including government, banking, telecommunications, aviation, oil & gas, education, construction, energy, healthcare, marine, ports & terminal, and critical country infrastructure systems. These projects were executed across diverse regions, including Asia, Europe, Africa, the Gulf, and Latin America.

Throughout his career, Yehia has held several prestigious positions, including Senior Penetration Testing Consultant and Security Researcher at DTS Solution Dubai, Group Manager of Cyber Security Assurance at DP World Dubai, and is currently serving as a Senior Consultant for the Red Team at CPX Abu Dhabi.

Yehia holds a Bachelor's degree in Business Administration from The Open University Business School, UK, and a Master's degree in Information Security and Digital Forensics from the University of East London, UK. He is the author of "Lean Penetration Testing with Python 3.x" and has published articles in renowned international cybersecurity magazines such as Hakin9 and Pentest. Yehia also has experience in exploitation development as he discovered 18+ CVE's. His name has been mentioned in the Hall of Fame of many websites.

His research in offensive security has established him as a keynote speaker at numerous international cybersecurity conferences, including the Middle East Info Security Summit 2015, QuBit Conference 2016 and 2019, DefCamp 2016 and 2019, and Blackhat 2023. Yehia also holds multiple cybersecurity certifications, underscoring his expertise and commitment to the field.

About the Reviewer

Gjoko Krstic is a security engineer and vulnerability researcher, with a bachelor's degree in Computer Systems and Networks, and various certifications specializing in cyber security. He has over 17 years of experience in security architecture, exploit development, reverse engineering, red teaming and penetration testing for various corporate and government organizations. Gjoko is the founder of Zero Science Lab, a Macedonian information security research and development laboratory, discovering and responsibly disclosing a wide range of vulnerabilities in commercial products. He is also the author of security research papers related to WAFs and BMS including embedded systems and a speaker at various conferences.

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the completion of this book.

First and foremost, I extend my heartfelt appreciation to my wife, family and friends for their unwavering support and encouragement throughout this journey. Their love and encouragement have been a constant source of motivation.

I am immensely grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. Their support and assistance were invaluable in navigating the complexities of the publishing process.

I would also like to acknowledge the reviewers, technical experts, and editors who provided valuable feedback and contributed to the refinement of this manuscript. Their insights and suggestions have significantly enhanced the quality of the book.

Last but not least, I want to express our gratitude to the readers who have shown interest in our book. Your support and encouragement have been deeply appreciated.

Thank you to everyone who has played a part in making this book a reality.

Preface

Understanding and keeping up to date with cyber threats is essential for those who work or are interested in cybersecurity, especially the offensive cyber security. Using programming languages to create tools or scripts in different security assessments is also essential to discover different IT, IoT and OT systems and applications bugs. Using programming languages is mandatory for saving time and creating more efficient results during offensive cyber security assessment. Python3 is one of the top selected programming languages by the cybersecurity community that is being used by penetration testers, red teams, SOC analysis›s, and IT Administrators.

Comprising of fourteen insightful chapters, this book covers a wide range of topics starting from teaching the basics of Python3 and creating your first script, moving to creating advanced scripts and tools that used in many areas such as web applications, network, wireless networks, crypto and cracking, network monitoring, fuzzing and exploitation development, forensics, burp suite extensions and using ChatGPT to enhance your scripts.

The book not only focuses on the technical part but each chapter is also designed to commence with a theoretical foundation offering the readers a profound depth understanding systems, network, protocols, web applications. This offers the readers a deep understanding of the underlying principle of cybersecurity and the digital domain. The practical part in each chapter will start with creating basic scripts, giving the readers a solid practical foundation, and then moving forward to more advanced scripts.

This book is designed to cater to all undergraduate students, from diverse academic backgrounds, including computer science to information technology. Additionally, individuals who are new to offensive or defensive cyber security and those who are working as penetration testers, red team, IT administrators or SOC analysts, who can understand different attack tactics and techniques and learn how to defend against those attacks.

Through practical examples, comprehensive explanations, and a structured approach, this book aims to equip readers with a solid understanding to master Python3 to create advanced scripts and tools used in offensive security. Whether you are a novice or an experienced learner, we hope this book will serve as a valuable resource in your journey of exploring Python3 in cybersecurity.

Chapter 1: Starting with Penetration Testing and Basic Python – This chapter provides an introduction to penetration testing methodologies, red team different activities, mobile application and wireless penetration testing types. It gives an introduction of the most famous tools used in penetration testers nowadays. The second part of the chapter will teach the basics of Python3 for those who are unfamiliar with programming languages, reaching to creating your first program with Python3.

Chapter 2: Cracking with Python 3 – This chapter at the beginning will provide learning about different types of crypto world, and the difference between encoding, encryption, hashing, and obfuscation. Then it will move to teach the readers about different types of credential attacks and when to use them. In the practical part it teaches how to create scripts that can generate usernames and passwords list, decrypt MD5, SHA1 and crack zip-protected files using Python3.

Chapter 3: Service and Applications Brute Forcing with Python – This chapter provides a practical experience of to how to create scripts that can crack different network protocols such as FTP, SMTP, SSH, and web application broken authentication along with how to add custom functions that can brute force different network protocols using wordlists.

Chapter 4: Python Services Identifications: Ports and Banner – This chapter provides a deep learning of how ethernet networks operate, how communication takes place, and what the three-way handshake is. This should take us deeper inside IP, TCP, and UDP packet headers. Moving to teaching how wireless communication works by understanding the four-way handshake that wireless networks rely on. In the practical side it will teach how to create Python3 scripts, uncover different services and collect the banner information for each service. In addition to the way of using the socket library to conduct TCP and UDP scans, it will teach how to identify live hosts within the network and use the socket library to play with DNS.

Chapter 5: Python Network Modules and Nmap – This chapter focuses on creating scripts that can perform port scanners using Python3 and namp libraries, identify banner grabbing, and discover live hosts using the socket library. Perform advanced network reconnaissance using a Python module such as Scapy and Nmap library, and build a packet from scratch using Scapy.

Chapter 6: Network Monitoring with Python – This chapter will use Python3 but from the eye of the blue team's perspective. It will start with teaching network monitoring and its importance. The chapter gives an introduction of what is SOC and its role inside the organization. In the practical part it teaches how to create scripts that can network monitoring using the socket and scapy library. The chapter teaches how to create scripts

that can monitor HTTP and DNS and using scapy, along with how to create scripts that can analyze pcap file.

Chapter 7: Attacking Wireless with Python – This chapter will start with an introduction 802.11 packet headers, how are the wireless connections happening in the background, the wireless frequency, and the channels moving to wireless BSSID and SSID, and ESSID and the purpose of each of them. The chapter will provide an introduction of different types of wireless encryptions and their weaknesses from an offensive perspective, how to crack the WEP encryption key and use the dictionary method to attack the WPA/WPA2 keys. The chapter teaches how to use the evil twin method to attack WPA/WPA2 and understand how KRACK occurs. In the practical part it will teach how to create scripts that can scan close networks and grab information like SSID, BSSID, channel, and encryption, send death packets which allows to disconnect clients from the access point and grant the 4-way handshake.

Chapter 8: Analyzing Web Applications with Python – This chapter will start with an introduction of different HTTP methods and how to create Python3 scripts using the HTTP methods like GET, POST, PUT.etc. The chapter will discuss Python modules (Beautiful Soup and Requests) and how to create a script using them to parse URLs from different web applications. The chapter will teach how to create scripts to extract cookies and live sessions from web applications. teaches how to use Python to extract images and documents from web applications, and how to use Python modules to extract image metadata like date and time, camera module, and geolocation. The chapter will include a section on how to use Python to scan the web application for hidden web directories using dictionary files, along with how to use the Scrapy module to parse URLs from dynamic web application pages.

Chapter 9: Attacking Web Applications with Python – This chapter will focus on attacking web application. Starting with understanding of a variety of cybersecurity topics and methodologies. It will start with creating scripts that use Shodan, a powerful search engine used for security research, detailing how it operates and the methods for extracting critical data such as *domains*, *IP addresses*, and *banners.* The chapter delves into **cross-site trace** (**XST**) attacks, offering guidance on scripting to detect such vulnerabilities within web applications. The narrative then shifts to the identification of **web application firewalls** (**WAFs**), discussing the use of different tools and scripting techniques for detection purposes. The chapter also covers the intricacies of **cross-site scripting** (**XSS**), including its various forms and scripting strategies to validate false positives. Additionally, it explores the open redirect vulnerability, showcasing how to uncover such flaws through Python programming. A section on bypassing firewalls introduces methods for payload

encoding using diverse encoding techniques, enhancing the reader›s ability to navigate security measures. Lastly, the discussion on web logic vulnerabilities provides insights into their identification and exploitation, equipping readers with the necessary knowledge to address these security challenges effectively.

Chapter 10: Exploit Development with Python – This chapter will start with comprehensive understanding of various foundational concepts in computer architecture and cybersecurity. First, you will learn about Intel›s CPU architecture, specifically the x86, and delve into the roles of its general-purpose and special-purpose registers, along with segment and EFLAGS registers, understanding their unique purposes. Additionally, it will explore the structure of Windows memory and how programs or processes are managed within memory, offering insight into the complex workings of computer systems. Further, the chapter will guide through critical system components like the Kernel, Process **Environment Block** (**PEB**), **Thread Environment Block** (**TEB**), **Portable Executable** (**PE**) formats, and **dynamic-link libraries (DLLs**), demystifying how these elements interact within a computing environment. It will also teach about the stack and heap, two fundamental data structures in programming, and how manipulation of the stack can be leveraged in various computing contexts. Moreover, it will grasp the concepts of big- and little-endian data formats, which are essential for understanding data representation and manipulation in exploit development. The chapter will culminate in teaching the basics of writing buffer overflow exploits, an essential skill in exploit development. Through hands-on practice, it will teach the mechanics behind exploiting software vulnerabilities and the various protections in place to mitigate such attacks.

Chapter 11: Forensics with Python – This chapter will focus on forensics starting with teaching what is file analysis and how to extract different information from different file extensions. From network analysis, it will teach how to capture and dissect network traffic, perform packet analysis, and whip up network-based intrusion detection systems like a master. From memory analysis, it will dive into memory dumps to extract precious intel, detect nasty malware, and investigate memory-based attacks. Finally, it will teach how to perform disk analysis, such as recover deleted files, analyze disk images, and investigate file systems for evidence.

Chapter 12: Python with Burp Suite – This chapter will focus on Burp Suite, the most used proxy tools for testing web applications. It will start with an introduction of Burp Suite features and extensions. Teaches to create scripts that can be attached to Burp Suite, and can detect cross-domain resource sharing, sensitive data exposure, and default credentials or pages. The chapter will teach how to create Python3 scripts working as burp suite extensions that can detect open redirect and cross-site scripting.

Chapter 13: Fuzzing with Python – This chapter will focus on fuzzing, starting giving a comprehensive guide to the world of fuzzing. We will embark on a journey to explore the fundamentals of fuzzing, delve into various fuzzing techniques, and understand how Python can be leveraged to build effective fuzzing tools. The chapter will focus on the real world practical side and teaching how to create fuzzers targeting web applications and network protocols that will help to identify different bugs that will help testers to develop different exploits.

Chapter 14 ChatGPT with Python – This chapter aims to provide a comprehensive guide on using ChatGPT with Python in offensive cyber security. It will teach how to use ChatGPT to generate and complete Python codes, enhance the quality and efficiency of Python code for effective tasks, and use ChatGPT effectively in ML projects. It will also teach how ChatGPT can be integrated with Python libraries and frameworks. Finally, it gives an understanding of the challenges and best practices in ChatGPT-Python integration.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/a508f7

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Learn-Penetration-Testing-with-Python-3.x-2nd-Edition.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

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 **business@bpbonline.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 **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights 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.

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 BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
Starting with Penetration Testing and Basic Python

Introduction

Nowadays, cybersecurity is becoming a dire necessity due to the rise of cyber-attacks not only on an individual but also on corporate and government levels. Cyber-attacks have become more sophisticated and harder to detect. Ethical hackers may have different skills and may have followed different paths, but they all share one common skill: programming.

In the first part of this chapter, we will learn about the different penetration testing methodologies in use nowadays. We will also learn about wireless and mobile applications penetration testing and red teaming activities. We will familiarize ourselves with different tools used by **ethical hackers**.

If you are unfamiliar with the basics of Python programming, in the second part of this chapter, you will learn about the basics of Python programming and how to create your first program. If you already are a penetration tester, you can skip *Chapter 2, Cracking with Python*, which will shed light on how to use Python in cracking.

NOTE: The offensive tools and programming scripts you will learn in this book can be used only in your local environment. Using offensive tools in a live environment like companies requires written permission from the entity. Always remember that penetration testing without permission from clients is illegal. Note: Tools and programming scripts in this book will be tested in a local environment. You need to set up your environment using virtualization as a first step. You can choose between

VMWARE (https://www.vmware.com/products/) or Oracle Virtual Box (https:/ / www.virtualbox.org). In the second step, you must download and install an updated KALI Linux on a virtual machine from (https:// www.kali.org/downloads/). Also, you need to install Windows on a virtual machine, preferably a Windows 10. Ensure the two virtual machines are on the same IP range using (NAT or Bridged). If any other tools or software are required, they will be indicated when appropriate in the chapters.

Structure

In this chapter, we will cover the following topics:

- Introducing penetration testing
- Penetration testing types
- Mobile application penetration testing
- Penetration testing tools
- The basics of Python 3
- Python variables
- Statements in Python 3
- Operators in Python 3
- Functions in Python 3
- Classes, self, and detractors in Python 3

Objectives

When you complete the first part of this chapter, you will be able to understand different penetration testing methodologies and red teaming. You will also understand all the penetration testing phases. You will get practical knowledge of the most offensive tools used these days by ethical hackers.

When you complete the second part of this chapter, you will be writing your scripts using Python 3, in which you get the required programming skills to write advanced scripts in the coming chapters.

Introducing penetration testing

Penetration testing, pen testing, or ethical hacking all refer to the *process of testing* a computer system, network, web application, or wireless mobile application to find security holes or vulnerabilities that may be abused by malicious users or criminals to gain unauthorized access to a system. Malicious users are not only external users but can also be internal

ones or employees; many reported incidents were from inside organizations. Penetration testers use their skills to prove the existence of security holes in different systems so they can fix those security holes and, consequently, prevent access to attackers who may abuse them.

Penetration testers usually have the knowledge and expertise to test different environments to discover security holes. This expertise and knowledge should contain at least one programming language, which makes them not fully dependent on existing tools, and of course, the deep knowledge of systems, networks, and web protocols.

However, knowing how to use different hacking tools and their limits, and how they work in the background is necessary, as most systems that are being tested are in production, so any mistakes will affect the client's business. In the end, you can write everything.

Penetration testers should have mixed knowledge of how to use existing tools and the ability to write their scripts and programs. Depending only on existing tools to discover vulnerabilities is inefficient, as we have recently seen many private and governmental entities being attacked successfully despite penetration testing and red teaming assessments being carried out.

This is because many companies do not apply manual penetration testing that focuses on discovering vulnerabilities in application functionalities. Instead, they depend solely on vulnerability scanners, leading to inaccurate results and false positives.

The only systems not tested during operation are SCADA, which are systems used in countries' infrastructures. The rationale is that any mistake or wrong packet sent while testing will cause a disaster. Since SCADA systems control oil and gas operations, water and electricity, nuclear powers, vessels, and so on. Therefore, governments often clone the operating system to allow the penetration testers to try to find potential vulnerabilities. Besides, updating and patching the vulnerabilities and replacing software or hardware is not easy as they require suspending certain production functions.

There are two types of penetration testing:

- **Automated**: This is concerned with using different tools to discover existing vulnerabilities.
- **Manual**: This is concerned with viewing the application manually without using the tools or scanners and trying to abuse the application's functionality. Usually, manual testing is used in web applications and source code reviews.

Exploits writing

There is a common misunderstanding that penetration testers should be able to discover the **zero-day** vulnerability, which has a zero-day patch (that is, the vendor does not know about its existence). Usually, this type of vulnerability costs thousands of dollars in the black market. Penetration testers do not have to know how to discover zero days simply

because it takes a very long time to them on software or hardware. However, the pen tester must know basic reverse engineering and exploitation development. In certain companies and governments, exploit writers and reverse engineers are hired to create cyber weapons for the government.

A lot of penetration testers do not discover zero-day vulnerabilities in client applications or networks simply because this requires lots of time and persistence. Usually, zero-day vulnerabilities are only discovered in governmental entities. They must know, however, how the memory works during a program's execution. They should also be aware of the assembly languages to understand how to read programs from the debuggers and manipulate CPU registers which are small storage systems that store the programs' data during execution. We are going to talk about this in *Chapter 10*, *Exploit Development with Python*.

Origin of the term hacking

There is a huge misunderstanding about the term hacking. Hacking, in today's world, is related to criminal activities like *stealing money*, *fraud*, or *destroying a company's system*. This is particularly the case in the media. However, if we look at hacking from a cyber-security perspective, we realize that it is related to attacking systems to prove they are being vulnerable or exploitable, which is partly correct but not totally.

The term *hacking* came from *MIT* in 1955. It started with a group calling themselves *hackers* who edited and modified train models and elaborated miniature buildings. Hacking originally referred to understanding a system so deeply that you can modify it or add functions to it. It is not only related to computers; for example, a person who modifies the mechanics of a car is known as a **car hacker**. Returning to the cybersecurity domain, if we apply the original meaning of the word *hacking*, we can see that penetration testers should fully understand how the application or system works, and then they will be able to find not only existing vulnerabilities but also logical ones. This is where the importance of manual penetration testing comes in handy.

Vulnerability assessments

A **vulnerability assessment** (**VA**) is the process of identifying vulnerabilities and security holes in systems, networks, and applications. The difference between VAs and penetration testing is that in a VA, you only identify the existence of the vulnerability without taking any further steps. For example, let us say that we have an FTP server in a system that is vulnerable to remote code execution due to a version that has not been updated. In a VA, you would identify the vulnerability by the *FTP version number*, or you can identify vulnerabilities based on *known vulnerabilities* or *CVEs*. For example, the famous (CVE-2017-0144) is the *Eternal Blue* vulnerability that targets SMB protocol, but you do not exploit that vulnerability. In penetration testing, however, you would exploit that FTP server to gain access to the system. You might also carry out an escalation of privileges

and hash dumping or try to access other systems within the network, which is part of the post-exploitation process.

There is a misunderstanding that vulnerability assessments are carried out using only vulnerability scanners, which is inaccurate. Vulnerability scanners produce many false positives and false negatives. While they can provide a good starting point to indicate the most critical areas in a network, there are many blind spots that vulnerability scanners cannot see. There are also many different risk ratings in different vulnerability scanners. You may find that a vulnerability risk is rated as low in one scanner but medium in another, so we follow the international risk rating like security vulnerability database/information source (CVE) or **Common Vulnerability Scoring System Version 3.1 Calculator** (**CVSS**) **https://www.first.org/cvss/calculator/3.0**. It is more important to consider the business risk and how it might affect the company.

Red team assessments

Red team assessments and penetration testing are almost the same, but they differ in their approach and their final goal. We know that penetration testing is a process that identifies and exploit vulnerabilities in the targeted system, network, or application. In penetration testing, you discover and exploit all the vulnerabilities that can be found on the target network, application, or the provided assets from the client. Also, you show how the exploitable vulnerabilities affect the client business, and you show how far the target network or application, or system is weak and needs to be patched or requires mitigation.

In the penetration testing process, the client provides the pen tester with the targeted scope of IP addresses or applications to be tested. In other situations, the client asks the pen tester to search for the published services over the internet and then confirm with the client what they have found.

In a red team assessment, however, the tester acts like a real criminal who wants to access a specific company, either through network access or physical access. Red teaming does not involve discovering all vulnerabilities, but instead, one or a few that help them to reach their goal. More methods are used during red teaming assessments, including social engineering, physical break-ins, bypassing monitoring systems, or attacking the internet wirelessly. In red teaming, there is no specific scope because the aim is simply to test the company's incident response and the effectiveness of the detection of any malicious actions.

Social engineering

Social engineering is the art of manipulating people's behavior to gain information or unauthorized physical access. Social engineering is different from scamming. The emails you receive that offer you $1 million from the prince of Nigeria are scams, not examples of social engineering. Social engineering is a combination of psychology (including micro-expressions, colors, and their effect on the subconscious, body, and eye language) and