Mastering Python for Networking and Security

Second Edition

Leverage the scripts and libraries of Python version 3.7 and beyond to overcome networking and security issues

José Manuel Ortega

Packt

BIRMINGHAM—MUMBAI

Mastering Python for Networking and Security *Second Edition*

Copyright © 2020 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: Shrilekha Inani

Senior Editor: Rahul Dsouza

Content Development Editor: Carlton Borges, Sayali Pingale

Technical Editor: Sarvesh Jaywant

Copy Editor: Safis Editing

Project Coordinator: Neil Dmello

Proofreader: Safis Editing

Indexer: Manju Arasan

Production Designer: Alishon Mendonsa

First published: September 2018

Second edition: December 2020

Production reference: 1031220

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

ISBN 978-1-83921-716-6

`www.packt.com`

I would like to thank my friends and family for their help in both the professional and personal fields. I would especially like to thank Shrilekha Inani (Acquisition Editor at Packt Publishing), Carlton Borges, and Sayali Pingale (Content Development Editors at Packt Publishing) for supporting me during the course of completing this book.

– José Manuel Ortega

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 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 for more details.

At 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

José Manuel Ortega has been working as a Software Engineer and Security Researcher with focus on new technologies, open source, security and testing. His career target has been to specialize in Python and DevOps security projects with Docker. Currently he is working as a security tester engineer and his functions in the project are analysis and testing the security of applications both web and mobile environments.

He has collaborated with universities and with the official college of computer engineers presenting articles and holding some conferences. He has also been a speaker at various conferences both national and international and is very enthusiastic to learn about new technologies and loves to share his knowledge with the developers community.

About the reviewers

Christian Ghigliotty is a writer and security engineer. He specializes in detection and response, incident response, and network security. When he's not wrestling with computers, he enjoys reading, cycling, and baseball. You can find him on Twitter: `@ harveywells`.

To my wife Mary, for her love and encouragement. She also tolerates my occasional loud chewing. To my children, who make me laugh and help me see the world differently.

Greg Smith is an experienced security professional who has worked in a variety of roles across the full stack of engineering disciplines including offensive security, software development, security architecture, security operations, WAN/SATCOM, engineering management, and systems management.

This experience has been built up across a variety of roles within the UK government, most recently within the Ministry of Justice Digital Offensive Security team and is now building the Application Security function in fintech at GoCardless.

Greg is an active member of the infosec community and has spoken at NCSC CyberUK In Practice, BSidesLDN, and BSidesMCR, conferences in recent years.

Thank you to my wife and family for supporting me and allowing me the space to contribute to further the knowledge of others in the infosec community.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `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

Preface

Section 1: The Python Environment and System Programming Tools

1

Working with Python Scripting

Technical requirements 4

Introduction to Python scripting 4

Why choose Python? 4

Multi-platform capabilities and versions 5

Python 3 features 5

Exploring Python data structures 6

Lists 6

Tuples 10

Python dictionaries 10

Python functions, classes, and managing exceptions 13

Python functions 13

Python classes 16

Python inheritance 17

Managing exceptions 18

Python modules and packages 23

What is a module in Python? 23

Getting information from standard modules 24

Difference between a Python module and a Python package 25

Python Module Index 25

Managing parameters in Python 25

Managing dependencies and virtual environments 28

Managing dependencies in a Python project 28

Generating the requirements.txt file 28

Working with virtual environments 29

Configuring virtualenv 29

Development environments for Python scripting 31

Setting up a development environment 31

PyCharm 31

Debugging with PyCharm 32

Debugging with Python IDLE 34

Summary 35

Questions 35

Further reading 35

2
System Programming Packages

Technical requirements 38
Introducing system modules in Python 38
The system (sys) module 39
The operating system (os) module 40
The platform module 44
The subprocess module 45
Working with the filesystem in Python 48
Working with files and directories 49
Reading and writing files in Python 50
Opening a file with a context manager 52
Reading a ZIP file using Python 53
Managing threads in Python 54
Creating a simple thread 55
Working with the threading module 56
Multithreading and concurrency in Python 59
Multithreading in Python 59
Limitations of classic Python threads 60
Concurrency in Python with ThreadPoolExecutor 61
Executing ThreadPoolExecutor with a context manager 63
Working with socket.io 64
Implementing a server with socket.io 64
Implementing a client that connects to the server 66
Summary 66
Questions 67
Further reading 67

Section 2: Network Scripting and Extracting Information from the Tor Network with Python

3
Socket Programming

Technical requirements 72
Introducing sockets in Python 72
Network sockets in Python 72
The socket module 74
Basic client with the socket module 77
Implementing an HTTP server in Python 78

Testing the HTTP server 79

Implementing a reverse shell with sockets 80

Resolving IPS domains, addresses, and managing exceptions 82

Gathering information with sockets 82

Using the reverse lookup command 85

Managing socket exceptions 86

Port scanning with sockets 88

Implementing a basic port scanner 88

Advanced port scanner 92

Implementing a simple TCP client and TCP server 94

Implementing a server and client with sockets 94

Implementing the TCP server 96

Implementing the TCP client 97

Implementing a simple UDP client and UDP server 98

Implementing the UDP server 99

Implementing the UDP client 100

Summary 101

Questions 102

Further reading 102

4

HTTP Programming

Technical requirements 104

Introducing the HTTP protocol 104

Reviewing the status codes 104

Building an HTTP client with http.client 105

Building an HTTP client with urllib.request 106

Get response and request headers 109

Extracting emails from a URL with urllib.request 111

Downloading files with urllib.request 112

Handling exceptions with urllib.request 113

Building an HTTP client with requests 114

Getting images and links from a URL with requests 117

Making GET requests with the REST API 119

Making POST requests with the REST API 121

Managing a proxy with requests 124

Managing exceptions with requests 125

Building an HTTP client with httpx 126

Authentication mechanisms with Python 129

HTTP basic authentication with a requests module 129

HTTP digest authentication with the requests module 130

Summary 133

Questions 134

Further reading 134

5

Connecting to the Tor Network and Discovering Hidden Services

Technical requirements 136

Understanding the Tor Project and hidden services 136

Exploring the Tor network 137

What are hidden services? 141

Tools for anonymity in the Tor network 142

Connecting to the Tor network 142

Node types in the Tor network 144

Installing the Tor service 144

ExoneraTor and Nyx 148

Discovering hidden services with OSINT tools 150

Search engines 150

Inspecting onion address with onioff 151

OnionScan as a research tool for the deep web 152

Docker onion-nmap 153

Modules and packages in Python for connecting to the Tor network 154

Connecting to the Tor network from Python 155

Extracting information from the Tor network with the stem module 160

Tools that allow us to search hidden services and automate the crawling process in the Tor network 168

Scraping information from the Tor network with Python tools 168

Summary 171

Questions 171

Section 3: Server Scripting and Port Scanning with Python

6

Gathering Information from Servers

Technical requirements 176

Extracting information from servers with Shodan 176

Accessing Shodan services 176

The Shodan RESTful API 177

Shodan search with Python 179

Using Shodan filters and the BinaryEdge search engine 184

Shodan filters 184

BinaryEdge search engine 185

Using the socket module to obtain server information 188

Extracting server banners with Python 188

Getting information on DNS servers with DNSPython 191
DNS protocol 192
DNS servers 192
The DNSPython module 193
Getting vulnerable addresses in servers with fuzzing 198
The fuzzing process 198
Understanding and using the FuzzDB project 198
Summary 203
Questions 204
Further reading 204

7

Interacting with FTP, SFTP, and SSH Servers

Technical requirements 206
Connecting with FTP servers 206
Using the Python ftplib module 207
Using ftplib to brute-force FTP user credentials 214
Building an anonymous FTP scanner with Python 216
Connecting with SSH servers with paramiko and pysftp 218
Executing an SSH server on Debian Linux 219
Introducing the paramiko module 220
Establishing an SSH connection with paramiko 221
Running commands with paramiko 224
Using paramiko to brute-force SSH user credentials 227
Establishing an SSH connection with pysftp 229
Implementing SSH clients and servers with the asyncSSH and asyncio modules 230
Checking the security in SSH servers with the ssh-audit tool 233
Installing and executing ssh-audit 233
Rebex SSH Check 235
Summary 236
Questions 236
Further reading 237

8

Working with Nmap Scanner

Technical requirements 240
Introducing port scanning with Nmap 240
Scan modes with python-nmap 248
Implementing synchronous scanning 249
Implementing asynchronous scanning 255
Working with Nmap through the os and subprocess modules 260

Discovering services and vulnerabilities with Nmap scripts 261
Executing Nmap scripts to discover services 262
Executing Nmap scripts to discover vulnerabilities 265
Summary 270
Questions 270
Further reading 271

Section 4: Server Vulnerabilities and Security in Python Modules

9

Interacting with Vulnerability Scanners

Technical requirements 276
Understanding vulnerabilities and exploits 276
What is an exploit? 277
Vulnerability formats 277
Introducing the Nessus vulnerability scanner 279
Installing and executing the Nessus vulnerability scanner 280
Nessus vulnerabilities reports 283
Accessing the Nessus API with Python 285
Interacting with the Nessus server 286
Introducing the OpenVAS vulnerability scanner 293
Installing the OpenVAS vulnerability scanner 293
Understanding the web interface 295
Scanning a machine using OpenVAS 297
Accessing OpenVAS with Python 302
Summary 306
Questions 306
Further reading 307

10

Identifying Server Vulnerabilities in Web Applications

Technical requirements 310
Understanding vulnerabilities in web applications with OWASP 310
Testing XSS 313
Analyzing and discovering vulnerabilities in CMS web applications 317
Using CMSMap 318
Other CMS scanners 320

Discovering SQL vulnerabilities with Python tools 321
Introduction to SQL injection 321
Identifying pages vulnerable to SQL injection 322
Introducing SQLmap 324
Using SQLmap to test a website for a SQL injection vulnerability 326
Scanning for SQL injection vulnerabilities with the Nmap port scanner 331
Testing Heartbleed and SSL/TLS vulnerabilities 332
Vulnerabilities in the Secure Sockets Layer (SSL) protocol 332
Finding vulnerable servers in the Censys search engine 333
Analyzing and exploiting the Heartbleed vulnerability (OpenSSL CVE-2014-0160) 335
Scanning for the Heartbleed vulnerability with the Nmap port scanner 338
Scanning TLS/SSL configurations with SSLyze 340
Summary 342
Questions 342
Further reading 343

11

Security and Vulnerabilities in Python Modules

Technical requirements 346
Exploring security in Python modules 346
Python functions with security issues 346
Input/output validation 347
Eval function security 348
Controlling user input in dynamic code evaluation 351
Pickle module security 351
Security in a subprocess module 355
Using the shlex module 358
Insecure temporary files 359
Static code analysis for detecting vulnerabilities 360
Introducing static code analysis 360
Introducing Pylint and Dlint 361
The Bandit static code analyzer 361
Bandit test plugins 364
Detecting Python modules with backdoors and malicious code 367
Insecure packages in PyPi 367
Backdoor detection in Python modules 367
Denial-of-service vulnerability in urllib3 368
Security in Python web applications with the Flask framework 370
Rendering an HTML page with Flask 370
Cross-site scripting (XSS) in Flask 371
Disabling debug mode in the Flask app 372
Security redirections with Flask 373

Python security best practices 374
Using packages with the __init__.py interface 374
Updating your Python version 375
Installing virtualenv 375
Installing dependencies 375
Using services to check security in Python projects 376
Summary 379
Questions 379
Further reading 380

Section 5: Python Forensics

12

Python Tools for Forensics Analysis

Technical requirements 384
Volatility framework for extracting data from memory and disk images 384
Installing Volatility 385
Identifying the image profile 385
Volatility plugins 386
Connecting and analyzing SQLite databases 390
SQLite databases 390
The sqlite3 module 391
Network forensics with PcapXray 396
Getting information from the Windows registry 399
Introducing python-registry 399
Logging in Python 406
Logging levels 407
Logging module components 407
Summary 414
Questions 414
Further reading 415

13

Extracting Geolocation and Metadata from Documents, Images, and Browsers

Technical requirements 418
Extracting geolocation information 418
Extracting metadata from images 426
Introduction to EXIF and the PIL module 426
Getting the EXIF data from an image 427
Extracting metadata from PDF documents 432

Identifying the technology used by a website 437
Extracting metadata from web browsers 441
Firefox forensics with Python 441
Chrome forensics with Python 445
Summary 452
Questions 452
Further reading 452

14
Cryptography and Steganography

Technical requirements 456
Encrypting and decrypting information with pycryptodome 456
Introduction to cryptography 456
Introduction to pycryptodome 457
Encrypting and decrypting information with cryptography 469
Introduction to the cryptography module 469
Steganography techniques for hiding information in images 474
Introduction to steganography 474
Steganography with Stepic 479
Generating keys securely with the secrets and hashlib modules 481
Generating keys securely with the secrets module 481
Generating keys securely with the hashlib module 483
Summary 488
Questions 488
Further reading 489

Assessments

Chapter 1 – Working with Python Scripting 491
Chapter 2 – System Programming Packages 491
Chapter 3 – Socket Programming 492
Chapter 4 – HTTP Programming 492
Chapter 5 – Connecting to the Tor Network and Discovering Hidden Services 492
Chapter 6 – Gathering Information from Servers 493
Chapter 7 – Interacting with FTP, SFTP, and SSH Servers 493
Chapter 8 – Working with Nmap Scanner 494
Chapter 9 – Interacting with Vulnerability Scanners 494

Chapter 10 – Identifying Server Vulnerabilities in Web Applications 494

Chapter 11 – Security and Vulnerabilities in Python Modules 495

Chapter 12 – Python Tools for Forensics Analysis 495

Chapter 13 – Extracting Geolocation and Metadata from Documents, Images, and Browsers 495

Chapter 14 – Cryptography and Steganography 496

Other Books You May Enjoy

Index