Contenido

Introducción 1

CAPÍTULO 1
Trabajando con *sockets* en Python 3
1.1. Introducción a Python para proyectos de seguridad 3
1.2. Introducción a los *sockets* 4
1.2.1. *Sockets* de red en Python 4
1.2.2. Módulo *socket* en Python 6
1.3. Recopilación de información con *sockets* 8
1.3.1. Obtener información de un servidor y dirección IP. 10
1.4. Implementar un escáner de puertos con *sockets* 13
1.4.1. Escáner de puertos con *sockets* 13
1.4.2. Escáner de puertos avanzado 16
1.4.3. Escáner de puertos a partir de un dominio 18
1.5. Implementar en Python un servidor HTTP 21
1.5.1. Implementación del servidor 21
1.5.2. Implementación del cliente 23
1.6. Aplicaciones clientes-servidor con *sockets* en Python 24
1.6.1. Implementación del cliente 25
1.6.2. Administrar excepciones de *socket* 27
1.6.3. Creando un cliente y un servidor TCP con *sockets* 29
1.6.4. Método para aceptar conexiones 30
1.6.5. Enviar y recibir datos del *socket* 31
1.6.6. Implementando el servidor TCP 31
1.6.7. Implementando el cliente TCP 33
1.7. *Shell* inversa con *sockets* 35
1.8. Conclusiones 41

CAPÍTULO 2
Módulos para realizar peticiones con Python 43
2.1. Introducción 43
2.2. Protocolo HTTP y creación de clientes HTTP en Python 43
2.2.1. Introducción al protocolo HTTP 43
2.2.2. Módulo http.client 45

2.3. Construyendo un cliente HTTP con urllib.request 47
2.3.1. Usando el método urlopen() 48
2.3.2. Objeto respuesta y códigos de estado 49
2.3.3. Comprobación de cabeceras HTTP con urllib.request 52
2.3.4. Personalización de cabeceras HTTP con urllib.request 55
2.3.5. Obtener correos electrónicos y enlaces de una URL 57
2.3.6. Obtener imágenes de una URL con urllib.request 60
2.4. Construyendo un cliente HTTP con *requests* 61
2.4.1. Introducción al módulo *requests* 62
2.4.2. Obtener número de palabras de un fichero 66
2.4.3. Obtener cabeceras con el módulo *requests* 68
2.4.4. Realizar peticiones GET a una API REST 70
2.4.5. Realizar peticiones POST a una API REST 73
2.4.6. Realizar peticiones mediante un *proxy* 75
2.4.7. Gestionar excepciones con el módulo *requests* 76
2.5. Conclusiones 79

CAPÍTULO 3
Recolección de información de servidores con Python 81
3.1. Introducción 81
3.2. Utilizando Shodan para la obtención de información 81
3.2.1. Filtros en Shodan 84
3.3. Utilizando Python para realizar búsquedas en Shodan 85
3.3.1. Acceso a Shodan desde Python 87
3.3.2. Búsquedas de Shodan en Python 90
3.3.3. Obtención de información de un servidor FTP 96
3.4. Utilizando el protocolo WHOIS para obtener información de un servidor 98
3.4.1. Servicio WHOIS 100
3.4.2. Consultas al servicio WHOAPI.com 101
3.4.3. Consultas con el módulo Python-whois 105
3.4.4. Consultas con el módulo ipwhois 108
3.5. Extracción de información de servidores DNS 110
3.5.1. Servidores DNS 112
3.5.2 Módulo DNSPython 112
3.5.3. Otras operaciones con el módulo dnspython 121
3.6. Servicios DNS 123
3.7. Conclusiones 125

CAPÍTULO 4
Extracción de metadatos con Python 127
4.1. Introducción 127
4.2. Obtener información de geolocalización 127

4.3. Módulos de geolocalización en Python 131
4.3.1. Geolocalización con geoip2-python 131
4.3.2. Geolocalización con maxminddb-geolite2 132
4.3.3. Geolocalización con python-geoip-python3 139
4.4. Extracción de metadatos en documentos PDF 140
4.4.1. Obtención de metadatos con PdfReader 142
4.4.2. Extraer texto e imágenes de documentos PDF 143
4.5. Extracción de metadatos en imágenes 146
4.5.1. Extracción de metadatos con el módulo PIL.ExifTags 148
4.6. Conclusiones 153

CAPÍTULO 5
***Web Scraping* con Python** 155
5.1. Introducción 155
5.2. *Parsers* XML y HTML 156
5.2.1. Extraer etiquetas de un sitio web 159
5.2.2. Extracción de documentos PDF con el módulo lxml 162
5.3. Extraer contenido y etiquetas con BeautifulSoup 165
5.3.1. Extraer nombres de dominio con BeautifulSoup 168
5.3.2. Extracción de contenido mediante expresiones regulares 169
5.3.3. Extracción de imágenes y enlaces con el módulo bs4 170
5.4. *Web Scraping* con Scrapy 176
5.4.1. Características de Scrapy 176
5.4.2. Arquitectura de Scrapy 177
5.4.3. Instalación y comandos de Scrapy 178
5.4.4. Extrayendo información mediante Scrapy Shell 179
5.4.5. Scrapy como *framework* de desarrollo de *spyders* 182
5.4.6. Fichero de configuración settings.py 188
5.4.7. Exportación de resultados en formatos json, csv, xml 190
5.5. Conclusiones 191

CAPÍTULO 6
Escaneo de puertos y redes con Python 193
6.1. Introducción 193
6.2. Nmap como herramienta de escáner de puertos 194
6.2.1. Tipos de escaneo con nmap 194
6.3. Escaneo de puertos con Python-nmap 199
6.3.1. Escaneo síncrono con Python-nmap 204
6.3.2. Guardar el resultado del escaneo en un fichero JSON 206

6.3.3. Usando PortScanner Yield 209
6.3.4. Usando nmap con otros módulos de Python..... 210
6.3.5. Escaneo asíncrono 212
6.4. Ejecutar *scripts* de nmap para detectar servicios y vulnerabilidades..... 217
6.4.1. Ejecutar *scripts* de nmap 218
6.4.2. Obtener subdominios con *script* de nmap 224
6.4.3. Analizar el servicio FTP con *scripts* de nmap..... 226
6.5. Obtener las máquinas activas de un segmento de red 231
6.6. Scanless 237
6.7. Conclusiones..... 241

CAPÍTULO 7
Conexión con servidores FTP, SFTP y SSH desde Python..... 243
7.1. Introducción 243
7.2. Conexiones con servidores FTP utilizando el módulo ftpLib 243
7.2.1. Conexiones con servidores FTP..... 244
7.2.2. Descarga de ficheros de servidores FTP..... 249
7.2.3. Comprobar conexión FTP anónima 251
7.2.4. Proceso de fuerza bruta para conectarnos con un servidor FTP 254
7.3. Conexión con servidores SSH utilizando paramiko 261
7.3.1. Ejecutar comandos con paramiko..... 272
7.3.2. Conexión con la clase Transport 273
7.3.3. Ejecutar comandos con la clase Transport..... 278
7.3.4. Obtener algoritmos de cifrado 281
7.3.5. Operaciones sobre archivos mediante el cliente SFTP 285
7.3.6. Descarga de ficheros con el cliente SFTP..... 290
7.4. Conexión con servidores SFTP utilizando PYSFTP..... 292
7.4.1. Descarga de ficheros utilizando PYSFTP 295
7.5. Conclusiones..... 296

CAPÍTULO 8
Análisis de vulnerabilidades en aplicaciones web con Python..... 299
8.1. Introducción..... 299
8.2. Introducción a la metodología OWASP..... 299
8.2.1. Inyección de comandos 300
8.2.2. SQL Injection..... 300
8.2.3. Cross-Site Scripting (XSS) 301
8.3. *Scripts* en Python para detectar vulnerabilidades en sitios web..... 303
8.3.1. *Script* en Python para detectar SQL Injection 304
8.3.2. *Script* en Python para detectar Cross-Site Scripting XSS..... 307

8.4. Introducción a la herramienta SQLmap para detectar vulnerabilidades del tipo SQL Injection 311
8.4.1. Ejecutar SQLmap sobre un dominio vulnerable..... 312
8.4.2. Extracción de tablas y columnas de una base de datos 314
8.4.3. Acceder a información de una tabla..... 317
8.5. Introducción a la herramienta Bandit para detectar vulnerabilidades en proyectos de Python..... 318
8.5.1. Instalar y ejecutar Bandit 318
8.5.2. Análisis de vulnerabilidades con Bandit .. 322
8.5.3. *Plug-ins* de Bandit para el análisis de código estático 325
8.5.4. Plugin SQL Injection 330
8.6. Otras herramientas de análisis y detección de vulnerabilidades en Python ... 332
8.6.1. Safety 333
8.6.2. Ejemplo de código para detectar XSS 334
8.6.3. Escáner de vulnerabilidades XSS para Python 336
8.7. Conclusiones..... 338

CAPÍTULO 9
Análisis del tráfico de red y rastreo de paquetes con scapy..... 341
9.1. Introducción..... 341
9.2. Captura e inyección de paquetes con scapy 341
9.3. Envío y recepción de paquetes con scapy 352
9.3.1. Enviar y recibir paquetes con scapy 356
9.4. Descubrimiento de redes con scapy..... 363
9.4.1. Escaneo de puertos con scapy 366
9.4.2. Comando traceroute con scapy 370
9.5. Lectura de ficheros PCAP con scapy..... 376
9.6. Rastreo de paquetes con scapy 380
9.7. Conclusiones..... 391

CAPÍTULO 10
Recopilación de información con herramientas OSINT..... 393
10.1. Introducción 393
10.2. Introducción a la Inteligencia de Fuentes Abiertas (OSINT)..... 394
10.2.1. Google Dorks y la base de datos de Google Hacking 395
10.2.2. Maltego..... 396
10.2.3. Photon..... 399
10.2.4. The Harvester 400
10.2.5. Censys..... 400
10.2.6. crt.sh 401
10.2.7. DnsDumpster 402
10.2.8. Web Check..... 402
10.2.9. WaybackMachine..... 403
10.2.10. OSINT framework 404
10.2.11. El motor de búsqueda Shodan..... 405

10.2.12. El motor de búsqueda BinaryEdge 406
10.3. Obtener información con Google Dorks 408
10.3.1. Google Dorks 409
10.3.2. Katana: una herramienta Python para Google Hacking 410
10.3.3. Dorks Hunter 411
10.4. Obtener información con SpiderFoot 414
10.4.1. Módulos de SpiderFoot 419
10.5. Obtener información sobre servidores DNS con DNSPython y DNSRecon 420
10.5.1. Protocolo DNS 420
10.5.2. Módulo DNSPython 421
10.5.3. DNSRecon 427
10.6. Obtención de servidores vulnerables con *fuzzing* 433
10.6.1. El proceso de *fuzzing* 433
10.6.2. *Web fuzzing* 434
10.6.3. Introducción del proyecto FuzzDB 435
10.6.4. Identificación de páginas de inicio de sesión predecibles con el proyecto FuzzDB 440
10.6.5. Identificación de inyección SQL con el proyecto FuzzDB 442
10.7. Conclusiones 445

CAPÍTULO 11
Criptografía y ofuscación de código 447
11.1. Introducción 447
11.2. Introducción a la criptografía 447
11.3. Cifrar y descifrar información con pycryptodome 448
11.3.1. Cifrar y descifrar con el algoritmo DES 452
11.3.2. Cifrar y descifrar con el algoritmo AES 454
11.3.3. Cifrado de archivos con el algoritmo AES 458
11.3.4. Generación de firmas RSA con pycryptodome 463
11.4. Cifrar y descifrar información con cryptography 468
11.4.1. Cifrado simétrico con el paquete fernet 469
11.5. Generación segura de claves con los módulos secrets y hashlib 474
11.5.1. Generar claves de forma segura con el módulo hashlib 477
11.5.2. Comprobar la integridad de un fichero con el módulo hashlib 483
11.6. Herramientas de Python para la ofuscación de código 485
11.6.1. Ofuscación de código con pyarmor 486
11.7. Conclusiones 491

Glosario 493