

Spis treści

O autorze	13
O recenzentach	14
Przedmowa	15

Część I. Środowisko języka Python i narzędzia do programowania systemowego

Rozdział 1. Skrypty w języku Python	21
Wymagania techniczne	21
Wprowadzenie do języka Python	22
Dlaczego warto wybrać język Python?	22
Wielosystemowość i wersje języka	22
Cechy wersji języka Python 3	23
Struktury danych	23
Listy	23
Krotki	26
Słowniki	26
Funkcje, klasy i wyjątki	28
Funkcje	28
Klasy	30
Dziedziczenie cech	31
Obsługa wyjątków	32

Moduły i pakiety	35
Co to jest moduł?	35
Uzyskiwanie informacji o standardowych modułach	36
Różnice pomiędzy modułem a pakietem	36
Indeks modułów	37
Przetwarzanie parametrów	37
Zależności i środowiska wirtualne	39
Zarządzanie zależnościami	39
Tworzenie pliku requirements.txt	39
Środowiska wirtualne	39
Tworzenie środowiska wirtualnego	40
Środowiska programistyczne	40
Przygotowanie środowiska programistycznego	40
PyCharm	41
IDLE	43
Podsumowanie	43
Pytania	44
Dalsza lektura	44
Rozdział 2. Pakiety systemowe	45
Wymagania techniczne	45
Wprowadzenie do modułów systemowych	46
Moduł sys	46
Moduł os	47
Moduł platform	50
Moduł subprocess	50
Korzystanie z systemu plików	53
Operacje na plikach i katalogach	53
Odczytywanie i zapisywanie plików	54
Otwieranie plików za pomocą menedżera kontekstu	56
Odczytywanie archiwów ZIP	57
Zarządzanie wątkami	57
Utworzenie prostego wątku	57
Moduł threading	58
Wielowątkowość i współbieżność	60
Wielowątkowość w języku Python	61
Cechy typowych wątków	62
Współbieżność i klasa ThreadPoolExecutor	62
Uruchamianie wątków za pomocą menedżera kontekstu	63
Moduł socketio	64
Implementacja aplikacji serwerowej przy użyciu modułu socketio	65
Implementacja aplikacji klienckiej	66
Podsumowanie	66
Pytania	67
Dalsza lektura	67

Część II. Komunikacja sieciowa i pozyskiwanie informacji z sieci Tor

Rozdział 3. Programowanie sieciowe	71
Wymagania techniczne	72
Wprowadzenie do programowania sieciowego	72
Gniazda sieciowe	72
Moduł socket	73
Prosta aplikacja kliencka	76
Implementacja serwera HTTP	76
Test serwera HTTP	77
Implementacja odwrotnej powłoki	78
Odwzorowywanie nazw domen na adresy IP i obsługa wyjątków	79
Uzyskiwanie informacji za pomocą modułu socket	80
Odwrotne odwzorowanie nazwy domeny	81
Obsługa wyjątków modułu socket	82
Skanowanie portów	84
Implementacja prostego skanera portów	84
Zaawansowany skaner portów	86
Implementacja prostych programów serwera i klienta TCP	88
Implementacja serwera i klienta	88
Implementacja serwera TCP	89
Implementacja klienta TCP	90
Implementacja prostych programów serwera i klienta UDP	91
Implementacja serwera UDP	91
Implementacja klienta UDP	92
Podsumowanie	93
Pytania	93
Dalsza lektura	94
Rozdział 4. Programowanie komunikacji HTTP	95
Wymagania techniczne	95
Wprowadzenie do protokołu HTTP	96
Kody stanów	96
Tworzenie aplikacji klienckich za pomocą modułu http.client	97
Tworzenie aplikacji klienckich za pomocą modułu urllib.request	98
Przetwarzanie nagłówków żądań i odpowiedzi HTTP	100
Wyodrębnianie adresów e-mail z odpowiedzi	101
Pobieranie plików za pomocą modułu urllib.request	102
Obsługa wyjątków	102
Tworzenie aplikacji klienckich za pomocą modułu requests	103
Wyodrębnienie obrazów i odnośników	105
Wysłanie żądania GET do interfejsu REST API	107
Wysłanie żądania POST do interfejsu REST API	108
Obsługa serwera pośredniczącego	110
Obsługa wyjątków	111

Tworzenie aplikacji klienckich za pomocą modułu httpx	112
Mechanizmy uwierzytelniania użytkowników	114
Uwierzytelnianie podstawowe	115
Uwierzytelnianie skrótowe	115
Podsumowanie	117
Pytania	118
Dalsza lektura	118
Rozdział 5. Sieć Tor i ukryte usługi	119
Wymagania techniczne	119
Projekt Tor i ukryte usługi	120
Budowa sieci Tor	120
Trasowanie cebulowe	121
Czym są ukryte usługi?	123
Narzędzia i anonimowość w sieci Tor	124
Łączenie z siecią Tor	124
Typy węzłów w sieci Tor	125
Instalacja usługi Tor	126
Usługi ExoneraTor i Nyx	128
Wykrywanie ukrytych usług za pomocą narzędzi OSINT	130
Wyszukiwarki	130
Badanie adresów stron za pomocą narzędzia onioff	131
Narzędzie OnionScan do głębokiego badania sieci	132
Kontener onion-nmap	133
Moduły i pakiety do komunikacji z siecią Tor	134
Łączenie z siecią Tor	134
Pozyskiwanie informacji z sieci Tor za pomocą modułu stem	137
Narzędzia do wyszukiwania ukrytych usług i automatyzowania procesu indeksowania	143
Pozyskiwanie informacji z sieci Tor za pomocą narzędzi w języku Python	143
Podsumowanie	146
Pytania	146
Część III. Skrypty serwerowe i skanowanie portów	
Rozdział 6. Uzyskiwanie informacji o serwerach	149
Wymagania techniczne	149
Uzyskiwanie informacji o serwerach za pomocą usługi Shodan	150
Korzystanie z usługi Shodan	150
Interfejs REST API usługi Shodan	150
Korzystanie z usługi Shodan w języku Python	152
Filtry Shodan i usługa BinaryEdge	155
Filtry Shodan	155
Usługa BinaryEdge	156
Uzyskiwanie informacji o serwerach za pomocą modułu socket	158
Odczytywanie banerów serwerów	158
Uzyskiwanie informacji o serwerach DNS za pomocą modułu dnspython	161
Usługa DNS	161
Moduł dnspython	162

Wyszukiwanie adresów serwerów podatnych na ataki	165
Fuzer	165
Baza FuzzDB	166
Podsumowanie	169
Pytania	169
Dalsza lektura	170
 Rozdział 7. Usługi FTP, SFTP i SSH	 171
Wymagania techniczne	171
Korzystanie z usługi FTP	172
Moduł ftplib	172
Przeprowadzanie ataków metodą brutalnej siły przy użyciu modułu ftplib	177
Tworzenie testera anonimowego dostępu do usługi FTP	179
Korzystanie z usługi SSH	180
Uruchomienie usługi SSH w systemie Debian	181
Moduł paramiko	181
Instalacja modułu	182
Nawiązywanie połączenia z usługą SSH przy użyciu modułu paramiko	182
Wydawanie poleceń za pomocą modułu paramiko	184
Przeprowadzanie ataków metodą brutalnej siły przy użyciu modułu paramiko	186
Nawiązywanie połączenia z usługą SSH przy użyciu modułu pysftp	187
Implementacja programów serwerowych i klienckich z wykorzystaniem modułów asynssh i asyncio	188
Weryfikacja bezpieczeństwa usługi SSH za pomocą narzędzia ssh-audit	190
Instalacja narzędzia ssh-audit i korzystanie z niego	190
Narzędzie Rebex SSH Check	192
Podsumowanie	192
Pytania	193
Dalsza lektura	193
 Rozdział 8. Skaner Nmap	 195
Wymagania techniczne	195
Skanowanie portów za pomocą narzędzia Nmap	196
Techniki skanowania w narzędziu Nmap	196
Skanowanie portów przy użyciu modułu nmap	198
Tryby skanowania w module nmap	201
Implementacja skanowania synchronicznego	202
Implementacja skanowania asynchronicznego	206
Uruchamianie narzędzia Nmap za pomocą modułów os i subprocess	209
Wykrywanie usług i ich podatności na ataki za pomocą skryptów narzędzia Nmap	210
Uruchamianie skryptów narzędzia Nmap	210
Wykrywanie podatności usług na ataki	213
Podsumowanie	215
Pytania	215
Dalsza lektura	216

Część IV. Podatności serwerów na ataki i bezpieczeństwo modułów języka Python

Rozdział 9. Skanery podatności na ataki

219

Wymagania techniczne

219

Podatność na ataki i szkodliwe oprogramowanie

220

Co to jest szkodliwe oprogramowanie?

220

Baza podatności

221

Skaner Nessus

222

Instalacja i uruchomienie skanera

222

Raporty skanera Nessus

225

Dostęp do interfejsu API skanera

226

Korzystanie ze skanera

226

Skaner OpenVAS

231

Instalacja skanera

231

Interfejs graficzny skanera OpenVAS

233

Skanowanie hostów

235

Korzystanie ze skanera OpenVAS w języku Python

239

Podsumowanie

241

Pytania

242

Dalsza lektura

242

Rozdział 10. Wykrywanie podatności serwerów i aplikacji WWW na ataki

243

Wymagania techniczne

244

Podatności aplikacji internetowych na ataki opisane w projekcie OWASP

244

Skrypty XSS

246

Wykrywanie i analizowanie podatności systemów CMS na ataki

249

Skaner CMSMap

250

Inne skanery systemów CMS

251

Narzędzia do wykrywania podatności stron na wstrzykiwanie zapytań SQL

252

Wstrzykiwanie zapytań SQL

252

Identyfikowanie stron podatnych na wstrzykiwanie zapytań SQL

252

Narzędzie sqlmap

254

Testowanie podatności stron internetowych na wstrzykiwanie zapytań SQL

256

Skaner portów Nmap

259

Wykrywanie zagrożenia Heartbleed i podatności protokołów SSL/TLS

260

Luki w bezpieczeństwie protokołów SSL/TLS

260

Znajdowanie za pomocą wyszukiwarek Shodan i Censys serwerów podatnych na ataki

261

Analiza i wykorzystanie podatności na zagrożenie Heartbleed (OpenSSL CVE-2014-0160)

262

Wykrywanie zagrożenia Heartbleed za pomocą skanera Nmap

265

Skanowanie konfiguracji protokołów SSL/TLS za pomocą narzędzia SSLyze

265

Podsumowanie

267

Pytania

268

Dalsza lektura

268

Rozdział 11. Luki w bezpieczeństwie modułów języka Python	269
Wymagania techniczne	269
Bezpieczeństwo modułów języka Python	270
Funkcje posiadające luki w bezpieczeństwie	270
Weryfikacja poprawności danych wejściowych	270
Funkcja eval()	271
Kontrola dynamicznego kodu wprowadzanego przez użytkownika	273
Bezpieczeństwo modułu pickle	273
Bezpieczeństwo modułu subprocess	276
Moduł shlex	278
Niebezpieczne pliki tymczasowe	279
Statyczna analiza kodu i wykrywanie podatności na ataki	280
Statyczna analiza kodu	280
Programy Pylint i Dlint	280
Statyczny analizator kodu Bandit	281
Wtyczki narzędzia Bandit	283
Wykrywanie ukrytych wejść i szkodliwego kodu w modułach	285
Niebezpieczne pakiety w repozytorium PyPI	285
Wykrywanie tylnych drzwi	285
Podatność modułu urllib3 na atak typu DoS	286
Bezpieczeństwo aplikacji opartych na platformie Flask	287
Dynamiczne strony internetowe	287
Skrypty XSS	288
Tryb diagnostyczny	289
Przekierowania	289
Dobre praktyki bezpiecznego kodowania w języku Python	291
Zarządzanie pakietami za pomocą pliku __init__.py	291
Aktualizacja wersji środowiska Python	291
Tworzenie wirtualnych środowisk	291
Bezpieczne instalowanie zależności	291
Korzystanie z usług weryfikujących bezpieczeństwo projektów	292
Podsumowanie	294
Pytania	295
Dalsza lektura	295

Część V. Analiza śledcza

Rozdział 12. Narzędzia do analizy śledczej	299
Wymagania techniczne	299
Wyodrębnianie danych z obrazów pamięci i dysków przy użyciu platformy Volatility	300
Instalacja narzędzia Volatility	300
Określenie profilu obrazu	301
Wtyczki	301
Analizowanie bazy danych SQLite	303
Baza danych SQLite	303
Moduł sqlite3	304

Analiza ruchu sieciowego za pomocą narzędzia PcapXray	307
Pozyskiwanie informacji z rejestru systemu Windows	309
Moduł python-registry	310
Rejestrowanie komunikatów	315
Poziomy ważności komunikatów	315
Komponenty modułu logging	315
Podsumowanie	320
Pytania	320
Dalsza lektura	321
 Rozdział 13. Dane geograficzne i metadane w dokumentach, obrazach i przeglądarkach	 323
Wymagania techniczne	324
Uzyskiwanie informacji geolokalizacyjnych	324
Wyodrębnianie metadanych z obrazów	329
Format EXIF i moduł PIL	329
Wyodrębnianie metadanych EXIF z obrazów	330
Wyodrębnianie metadanych z dokumentów PDF	333
Identyfikowanie technologii używanych do tworzenia witryn internetowych	337
Wyodrębnianie metadanych z przeglądarek	339
Wyodrębnianie metadanych z przeglądarki Firefox	339
Wyodrębnianie metadanych z przeglądarki Chrome	342
Podsumowanie	346
Pytania	347
Dalsza lektura	347
 Rozdział 14. Kryptografia i steganografia	 349
Wymagania techniczne	350
Szyfrowanie i deszyfrowanie danych za pomocą modułu pycryptodome	350
Wprowadzenie do kryptografii	350
Moduł pycryptodome	351
Szyfrowanie i deszyfrowanie danych za pomocą modułu cryptography	360
Moduł cryptography	360
Techniki steganograficzne ukrywania informacji w obrazach	363
Wprowadzenie do steganografii	364
Moduł stegim	367
Generowanie kluczy i haseł za pomocą modułów secrets i hashlib	368
Generowanie kluczy za pomocą modułu secrets	368
Generowanie kluczy za pomocą modułu hashlib	370
Podsumowanie	373
Pytania	373
Dalsza lektura	374
 Odpowiedzi	 375