

Inhaltsverzeichnis

| | |
|--|----|
| Danksagung..... | 15 |
| Der Autor: T. J. O'Connor | 17 |
| Der Coautor: Rob Frost..... | 19 |
| Der Fachgutachter: Mark Baggett | 21 |
| Einleitung | 23 |
| Zielgruppe..... | 23 |
| Der Aufbau dieses Buches | 23 |
| Kapitel 1: Grundlagen | 24 |
| Kapitel 2: Penetrationstests mit Python | 24 |
| Kapitel 3: Forensische Untersuchungen mit Python | 24 |
| Kapitel 4: Analyse des Netzwerkverkehrs mit Python..... | 25 |
| Kapitel 5: Angriffe auf drahtlose Netzwerke mit Python | 25 |
| Kapitel 6: Webaufklärung mit Python | 25 |
| Kapitel 7: Umgehen von Antivirussoftware mit Python..... | 25 |
| Die Begleitwebsite | 26 |
| 1. Grundlagen | 27 |
| 1.1 Einführung: Ein Penetrationstest mit Python | 27 |
| 1.2 Die Entwicklungsumgebung einrichten..... | 29 |
| 1.2.1 Drittanbieterbibliotheken installieren..... | 29 |
| 1.2.2 Interpretiertes und interaktives Python im Vergleich | 33 |

| | | |
|--------|--|-----------|
| 1.3 | Die Sprache Python | 34 |
| 1.3.1 | Variablen | 35 |
| 1.3.2 | Strings..... | 36 |
| 1.3.3 | Listen..... | 36 |
| 1.3.4 | Dictionarys | 37 |
| 1.3.5 | Netzwerkverbindungen | 38 |
| 1.3.6 | Bedingte Anweisungen..... | 39 |
| 1.3.7 | Ausnahmebehandlung | 40 |
| 1.3.8 | Funktionen | 42 |
| 1.3.9 | Iteration..... | 44 |
| 1.3.10 | Datei-E/A..... | 47 |
| 1.3.11 | Das Modul sys | 48 |
| 1.3.12 | Das Modul os..... | 49 |
| 1.4 | Ihr erstes Python-Programm..... | 52 |
| 1.4.1 | Kuckucksei: Der Hintergrund für Ihr erstes Python- Programm..... | 52 |
| 1.4.2 | Ein UNIX-Passwortknacker | 53 |
| 1.4.3 | Böse Dinge für einen guten Zweck: Der Hintergrund für Ihr zweites Python-Programm | 57 |
| 1.4.4 | Ein Passwortknacker für Zip-Dateien | 58 |
| 1.5 | Zusammenfassung | 64 |
| 1.6 | Literatur..... | 64 |
| 2. | Penetrationstests mit Python | 67 |
| 2.1 | Einführung: Würde der Morris-Wurm heute noch funktionieren? | 67 |
| 2.2 | Einen Portscanner schreiben | 68 |
| 2.2.1 | Vollständiger TCP-Verbindungsscan | 69 |
| 2.2.2 | Das Anwendungsbanner abrufen..... | 72 |

| | | |
|-------|--|-----|
| 2.2.3 | Den Scan auf Threads aufteilen | 74 |
| 2.2.4 | Den Nmap-Portscanner aufnehmen..... | 78 |
| 2.3 | Ein SSH-Botnetz mit Python aufbauen..... | 80 |
| 2.3.1 | Interaktion mit SSH über Pexpect | 82 |
| 2.3.2 | SSH-Passwörter mit Pssh knacken..... | 86 |
| 2.3.3 | SSH über schwache private Schlüssel angreifen | 90 |
| 2.3.4 | Ein SSH-Botnetz aufbauen | 95 |
| 2.4 | Kombinierter Massenangriff über FTP und das Web | 99 |
| 2.4.1 | Einen Scanner für den anonymen FTP-Zugriff mit Python erstellen | 100 |
| 2.4.2 | FTP-Anmeldeinformationen mit Ftplib ermitteln | 102 |
| 2.4.3 | Auf dem FTP-Server nach Webseiten suchen | 104 |
| 2.4.4 | Schadcode auf den Webseiten injizieren | 105 |
| 2.4.5 | Die Einzelteile zusammenfügen..... | 107 |
| 2.5 | Conficker | 113 |
| 2.5.1 | Den Windows-SMB-Dienst mit Metasploit angreifen | 115 |
| 2.5.2 | Python-Code zur interaktiven Nutzung von Metasploit schreiben..... | 117 |
| 2.5.3 | Eine Brute-Force-Methode zur Ausführung von Remoteprozessen..... | 119 |
| 2.5.4 | Endmontage: Einen eigenen Conficker schreiben | 120 |
| 2.6 | Eigenen Zero-Day-Angriffscode schreiben | 124 |
| 2.6.1 | Angriffe mithilfe von Stack-Pufferüberläufen | 124 |
| 2.6.2 | Die Kernelemente des Exploits | 125 |
| 2.6.3 | Den Exploit senden..... | 127 |
| 2.6.4 | Das Exploit-Skript fertigstellen..... | 128 |
| 2.7 | Zusammenfassung des Kapitels | 131 |
| 2.8 | Literatur..... | 131 |

| | | |
|-----------|--|------------|
| 3. | Forensische Untersuchungen mit Python..... | 135 |
| 3.1 | Einführung: Wie die BTK-Morde durch forensische Untersuchungen aufgeklärt wurden | 135 |
| 3.2 | Wo bist du gewesen? – Drahtlose Zugriffspunkte in der Registrierung analysieren | 137 |
| 3.2.1 | Die Windows-Registrierung mit WinReg lesen | 138 |
| 3.2.2 | Die MAC-Adresse mit Mechanize an Wigle übertragen... .. | 140 |
| 3.3 | Gelöschte Elemente im Papierkorb mit Python untersuchen | 146 |
| 3.3.1 | Gelöschte Elemente mithilfe des Moduls OS finden | 146 |
| 3.3.2 | SIDs Benutzern zuordnen | 147 |
| 3.4 | Metadaten | 150 |
| 3.4.1 | PDF-Metadaten mit PyPDF analysieren | 151 |
| 3.4.2 | Exif-Metadaten | 154 |
| 3.4.3 | Bilder mit BeautifulSoup herunterladen | 155 |
| 3.4.4 | Exif-Metadaten von Bildern mit der Python-Bibliothek Imaging lesen | 157 |
| 3.5 | Spuren von Anwendungen mit Python untersuchen | 161 |
| 3.5.1 | Die SQLite-Datenbank von Skype | 161 |
| 3.5.2 | Skype-Datenbankabfragen mit Python und Sqlite3 automatisieren | 163 |
| 3.5.3 | SQLite-Datenbanken von Firefox mit Python untersuchen | 171 |
| 3.6 | iTunes-Backups mit Python untersuchen..... | 181 |
| 3.7 | Zusammenfassung des Kapitels | 189 |
| 3.8 | Literatur..... | 189 |

| | | |
|-----------|---|------------|
| 4. | Analyse des Netzwerkverkehrs mit Python | 191 |
| 4.1 | Einführung: Operation Aurora – Das Offensichtliche übersehen | 192 |
| 4.2 | Wohin geht der Datenverkehr? Python antwortet! | 193 |
| 4.2.1 | IP-Adressen mit PyGeoIP physischen Standorten zuordnen | 194 |
| 4.2.2 | Pakete mit Dpkt analysieren | 195 |
| 4.2.3 | Google-Karten mit Python erstellen | 200 |
| 4.3 | Analyse des LOIC-Datenverkehrs: Ist Anonymous wirklich anonym? | 204 |
| 4.3.1 | LOIC-Downloads mit Dpkt finden | 204 |
| 4.3.2 | IRC-Befehle zum Hive analysieren | 206 |
| 4.3.3 | Laufende DDoS-Angriffe erkennen | 209 |
| 4.4 | Wie H. D. Moore das Problem des Pentagon löste | 215 |
| 4.4.1 | Das TTL-Feld | 216 |
| 4.4.2 | TTL-Felder mit Scapy analysieren | 218 |
| 4.5 | Die Fast-Flux- und Domain-Flux-Techniken von Storm und Conficker | 223 |
| 4.5.1 | Weiß Ihr DNS etwas, das Sie nicht wissen? | 224 |
| 4.5.2 | DNS-Datenverkehr mit Scapy analysieren | 225 |
| 4.5.3 | Fast-Flux-Datenverkehr mit Scapy aufspüren | 226 |
| 4.5.4 | Domain-Flux-Datenverkehr mit Scapy aufspüren | 228 |
| 4.6 | Kevin Mitnick: Vorhersage von TCP-Folgenummern | 230 |
| 4.6.1 | Vorhersage von TCP-Folgenummern selbst gemacht | 231 |
| 4.6.2 | Eine SYN-Flood mit Scapy hervorrufen | 232 |
| 4.6.3 | TCP-Folgenummern berechnen | 233 |
| 4.6.4 | Die TCP-Verbindung fälschen | 236 |
| 4.7 | Intrusion-Detection-Systeme mit Scapy unterlaufen | 240 |
| 4.8 | Zusammenfassung des Kapitels | 249 |
| 4.9 | Literatur | 249 |

| | | |
|-----------|--|------------|
| 5. | Angriffe auf drahtlose Netzwerke mit Python | 251 |
| 5.1 | Einführung: (Un-) Sicherheit von WLANs und der Eismann..... | 252 |
| 5.2 | Die Umgebung für WLAN-Angriffe einrichten | 252 |
| 5.2.1 | Die Erfassung von WLAN-Datenverkehr mit Scapy testen..... | 253 |
| 5.2.2 | Bluetooth-Pakete für Python installieren..... | 255 |
| 5.3 | Wall of Sheep – WLAN-Geheimnisse passiv belauschen..... | 256 |
| 5.3.1 | Kreditkarteninformationen mit regulären Ausdrücken ausspähen..... | 256 |
| 5.3.2 | Hotelgäste ausspionieren..... | 261 |
| 5.3.3 | Einen WLAN-Keylogger für Google-Abfragen schreiben..... | 264 |
| 5.3.4 | FTP-Anmeldeinformationen ausspionieren..... | 269 |
| 5.4 | Wo ist Ihr Laptop gewesen? Python antwortet!..... | 271 |
| 5.4.1 | Auf 802.11-Suchanfragen lauschen..... | 272 |
| 5.4.2 | 802.11-Signal von verborgenen Netzwerken finden | 273 |
| 5.4.3 | Verborgene 802.11-Netzwerke enttarnen..... | 274 |
| 5.5 | Drohnen mit Python übernehmen und ausspionieren... | 275 |
| 5.5.1 | Datenverkehr abfangen und das Protokoll analysieren..... | 276 |
| 5.5.2 | 802.11-Pakete mit Scapy gestalten | 279 |
| 5.5.3 | Die Drohne zum Absturz bringen | 283 |
| 5.6 | Firesheep erkennen..... | 285 |
| 5.6.1 | Wordpress-Sitzungsscookies | 286 |
| 5.6.2 | Schafe hüten: Die Wiederverwendung von Wordpress-Cookies erkennen..... | 288 |
| 5.7 | Spionieren mit Bluetooth und Python..... | 291 |
| 5.7.1 | Drahtlosen Datenverkehr zum Ermitteln von Bluetooth-Adressen abfangen..... | 293 |

| | | |
|-----------|--|------------|
| 5.7.2 | RFCOMM-Kanäle suchen..... | 297 |
| 5.7.3 | Service Discovery Protocol | 299 |
| 5.7.4 | Einen Drucker mit Python ObexFTP übernehmen..... | 300 |
| 5.7.5 | BlueBug-Angriffe mit Python durchführen | 301 |
| 5.8 | Zusammenfassung des Kapitels | 303 |
| 5.9 | Literatur..... | 304 |
| 6. | Webaufklärung mit Python | 307 |
| 6.1 | Einführung: Social Engineering heute..... | 307 |
| 6.1.1 | Aufklärung vor dem Angriff..... | 308 |
| 6.2 | Mit der Bibliothek Mechanize im Internet surfen | 309 |
| 6.2.1 | Anonymität: Proxys, Benutzeragenten und Cookies | 311 |
| 6.2.2 | Eine Python-Klasse für den anonymen Browser schreiben..... | 315 |
| 6.3 | Webseiten mit anonBrowser untersuchen | 318 |
| 6.3.1 | HREF-Links mit BeautifulSoup abschöpfen..... | 318 |
| 6.3.2 | Bilder mit BeautifulSoup spiegeln | 321 |
| 6.4 | Recherche, Untersuchung und Aufdeckung | 323 |
| 6.4.1 | Mit Python auf die Google-API zugreifen | 324 |
| 6.4.2 | Tweets mit Python analysieren | 328 |
| 6.4.3 | Ortsangaben aus Tweets entnehmen..... | 331 |
| 6.4.4 | Interessen auf Twitter mithilfe regulärer Ausdrücke bestimmen | 334 |
| 6.5 | Anonyme E-Mail..... | 340 |
| 6.6 | Social Engineering als Massenangriff | 341 |
| 6.6.1 | E-Mails mit Smtplib verschicken | 342 |
| 6.6.2 | Spear Phishing mit Smtplib | 344 |
| 6.7 | Zusammenfassung des Kapitels | 348 |
| 6.8 | Literatur..... | 349 |

| | | |
|-----------|---|------------|
| 7. | Umgehen von Antivirussoftware mit Python | 351 |
| 7.1 | Einführung: Flame | 351 |
| 7.2 | Antivirusprogrammen ausweichen | 352 |
| 7.3 | Die Umgehung der Antivirussoftware bestätigen..... | 357 |
| 7.4 | Zusammenfassung | 365 |
| 7.5 | Literatur..... | 365 |
| | Stichwortverzeichnis | 367 |