

Vijay Kumar Velu

Kali Linux

Testy penetracyjne
i bezpieczeństwo sieci
dla zaawansowanych

Wydanie II

Helion 

Packt 

Tytuł oryginału: Mastering Kali Linux for Advanced Penetration Testing - Second Edition

Tłumaczenie: Grzegorz Kowalczyk

ISBN: 978-83-283-4037-4

Copyright © Packt Publishing 2017. First published in the English language under the title 'Mastering Kali Linux for Advanced Penetration Testing - Second Edition - (9781787120235)'

Polish edition copyright © 2018 by Helion SA
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/kalit2>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

O autorze	13
O recenzencie	15
Przedmowa	17
Rozdział 1. Testy penetracyjne ukierunkowane na osiągnięcie celu	23
Koncepcyjny przegląd testów bezpieczeństwa	24
Zmierzch klasycznych testów penetracyjnych, skanowania w poszukiwaniu podatności i działań zespołów Red Team	24
Metodologia przeprowadzania testów	26
Wprowadzenie do systemu Kali Linux — jego historia i przeznaczenie	28
Instalowanie i aktualizowanie systemu Kali Linux	30
Uruchamianie systemu Kali Linux z urządzenia przenośnego	30
Instalowanie systemu Kali Linux w maszynie wirtualnej	31
VMware Workstation Player	32
VirtualBox	34
Instalowanie aplikacji Docker	39
Instalowanie systemu Kali Linux w chmurze — tworzenie instancji AWS	41
Dostosowywanie systemu Kali Linux	43
Konfigurowanie i dostosowywanie systemu Kali Linux	44
Zmiana hasła użytkownika root	44
Dodawanie zwykłego konta użytkownika	44
Przyspieszanie działania systemu Kali Linux	45
Udostępnianie i współużytkowanie folderów z systemem operacyjnym hosta	46
Dostosowywanie systemu Kali Linux do własnych potrzeb przy użyciu skryptów powłoki bash	48
Budowanie środowiska testowego	49
Konfigurowanie sieci wirtualnej z usługą Active Directory	49
Instalowanie zdefiniowanych celów	52

Metasploitable3	52
Mutillidae	53
Zarządzanie testami penetracyjnymi przy użyciu pakietu Faraday	54
Podsumowanie	58
Rozdział 2. OSINT oraz rozpoznanie pasywne	59
Podstawowe zasady przeprowadzania rozpoznania	60
Biały wywiad (OSINT)	61
Ofensywny biały wywiad	62
Maltego	63
CaseFile	66
Usługi Google Cache	67
Scraping	68
Pozyskiwanie nazw kont użytkowników i adresów e-mail	69
Zbieranie informacji o użytkownikach	70
Wyszukiwarki Shodan i censys.io	70
Google Hacking Database	71
Używanie zaawansowanych operatorów Google	72
Serwery szybkiej wymiany danych	73
Zastosowanie skryptów do automatycznego zbierania informacji OSINT	74
Defensywny biały wywiad	75
Profilowanie użytkowników pod kątem przygotowywania listy haseł	78
Tworzenie słowników do łapania haseł	78
Zastosowanie programu CeWL do mapowania witryny internetowej	79
Pozyskiwanie listy słów z serwisu Twitter przy użyciu programu Twofl	80
Podsumowanie	80
Rozdział 3. Aktywne rozpoznawanie zewnętrznych i wewnętrznych środowisk celu	83
Trudne do wykrycia techniki skanowania	84
Modyfikowanie źródłowych adresów IP	
i dostosowywanie ustawień używanych narzędzi	85
Modyfikowanie parametrów pakietów	86
Używanie serwerów proxy i sieci anonimowych	88
Rozpoznanie DNS i mapowanie sieci	91
Polecenie whois	92
Wykorzystywanie kompleksowych aplikacji wspomagających przeprowadzanie rozpoznania	94
Framework recon-ng	94
Protokół IPv6 — wybrane narzędzia	99
Mapowanie trasy do celu	100
Identyfikowanie zewnętrznej infrastruktury sieciowej	103
Mapowanie sieci poza zaporą sieciową	104
Identyfikacja systemów IDS/IPS	105
Wyszukiwanie hostów	107
Wykrywanie aktywnych hostów	108
Wykrywanie otwartych portów, systemu operacyjnego oraz działających usług	109
Skanowanie portów	109

Tworzenie własnego skanera portów przy użyciu programu netcat	110
Identyfikacja systemu operacyjnego zdalnego hosta	111
Wykrywanie usług działających na zdalnych hostach	112
Skanowanie dużych środowisk celu	113
Wykorzystanie danych DHCP	114
Wykrywanie oraz identyfikacja hostów w wewnętrznych sieciach środowiska celu	115
Wbudowane polecenia konsolowe systemu Windows	116
Rozgłoszenia ARP	117
Wykrywanie hostów w sieci za pomocą pakietów ping	117
Zastosowanie skryptów do łączenia skanów z użyciem programów masscan i nmap	119
Wykorzystanie protokołu SNMP	120
Pozyskiwanie informacji o kontaktach użytkowników Windows za pośrednictwem sesji SMB	121
Identyfikacja udziałów sieciowych	123
Rozpoznawanie serwerów w domenie Active Directory	124
Zastosowanie narzędzi złożonych (SPARTA)	125
Przykład konfiguracji pakietu SPARTA	126
Podsumowanie	127
Rozdział 4. Wyszukiwanie podatności i luk w zabezpieczeniach	129
<hr/>	
Trochę nomenklatury	130
Lokalne i sieciowe bazy podatności i luk w zabezpieczeniach	131
Skanowanie w poszukiwaniu podatności przy użyciu programu nmap	135
Wprowadzenie do skryptów LUA	137
Dostosowywanie skryptów NSE do własnych potrzeb	137
Skanery podatności aplikacji sieciowych	139
Wprowadzenie do skanerów Nikto i Vega	140
Dostosowywanie skanerów Nikto i Vega do własnych potrzeb	142
Skanery podatności dla aplikacji mobilnych	146
Skaner podatności OpenVAS	148
Dostosowywanie skanera OpenVAS do własnych potrzeb	150
Specjalizowane skanery podatności	150
Modelowanie zagrożeń	151
Podsumowanie	153
Rozdział 5. Bezpieczeństwo fizyczne i metody socjotechniczne	155
<hr/>	
Metodologia przeprowadzania ataków	157
Ataki z wykorzystaniem komputera	157
Ataki z wykorzystaniem telefonu	158
Ataki z dostępem fizycznym	159
Ataki z dostępem do konsoli systemu	159
Programy samdump2 i chntpw	160
Ułatwienia dostępu — opcja Sticky Keys	163
Ataki na pamięć systemową przy użyciu programu Inception	164
Tworzenie złośliwych urządzeń fizycznych	166
Ataki z wykorzystaniem urządzeń mikroprocesorowych	168

Pakiet SET	170
Ataki na witryny internetowe — atak ze zbieraniem poświadczeń logowania	174
Ataki na witryny internetowe — atak typu tabnabbing	176
Ataki na witryny internetowe — ataki złożone	177
Atak ze wstrzykiwaniem alfanumerycznego kodu shellcode z powłoki Powershell	178
Ataki z wykorzystaniem aplikacji HTA	179
Ukrywanie plików wykonywalnych oraz maskowanie adresu URL napastnika	181
Eskalowanie ataków przy użyciu przekierowań DNS	183
Ataki typu spear phishing	184
Przeprowadzanie kampanii phishingowej z wykorzystaniem pakietu Phishing Frenzy	188
Przeprowadzanie ataku phishingowego	192
Podsumowanie	194
Rozdział 6. Ataki na sieci bezprzewodowe	195
Konfigurowanie systemu Kali Linux do przeprowadzania ataków na sieci bezprzewodowe	196
Przeprowadzanie rozpoznania w sieciach bezprzewodowych	197
Kismet	200
Omijanie zabezpieczenia sieci z ukrytym identyfikatorem SSID	202
Omijanie zabezpieczenia sieci z filtrowaniem adresów MAC oraz otwartym uwierzytelnianiem	204
Atakowanie sieci z szyfrowaniem WPA i WPA2	206
Ataki typu brute-force	207
Atakowanie routerów sieci bezprzewodowych przy użyciu programu Reaver	210
Ataki typu DoS na sieci bezprzewodowe	211
Ataki na sieci WLAN z szyfrowaniem WPA/WPA2-Enterprise	213
Praca z pakietem Ghost Phisher	217
Podsumowanie	218
Rozdział 7. Rozpoznawanie i przełamywanie zabezpieczeń aplikacji internetowych	221
Metodologia	222
Planowanie ataku	224
Przeprowadzanie rozpoznania witryny internetowej	225
Wykrywanie zapór WAF oraz systemów równoważenia obciążenia	227
Tworzenie sygnatur aplikacji internetowych i systemów CMS	228
Tworzenie lustrzanej kopii strony internetowej z poziomu wiersza poleceń	231
Serwery proxy po stronie klienta	232
Burp Proxy	232
Poszerzanie funkcjonalności przeglądarek internetowych	237
Przeszukiwanie sieci i ataki typu brute-force na struktury katalogów	239
Skanery podatności wykrywające podatności określonych usług i aplikacji	239
Ataki specyficzne dla określonych aplikacji	241
Ataki typu brute-force na poświadczenia logowania	241
Wstrzykiwanie poleceń systemu operacyjnego przy użyciu narzędzia commix	241
Ataki ze wstrzykiwaniem danych lub kodu do baz danych	243
Utrzymywanie dostępu za pomocą powłok webshell	245
Podsumowanie	247

Rozdział 8. Ataki na zdalny dostęp	249
Wykorzystywanie luk w zabezpieczeniach protokołów komunikacyjnych	250
Przełamywanie zabezpieczeń protokołu RDP	250
Przełamywanie zabezpieczeń protokołu SSH	253
Przełamywanie zabezpieczeń protokołu VNC	255
Ataki na połączenia SSL	257
Słabe strony i luki w zabezpieczeniach protokołu SSL	257
Praca z programem Testssl	259
Rozpoznawanie połączeń SSL	260
Zastosowanie programu sslstrip do przeprowadzania ataku man-in-the-middle	265
Ataki typu DoS na połączenia SSL	268
Ataki na wirtualne sieci prywatne z protokołem IPSec	269
Skanowanie w poszukiwaniu bramek VPN	270
Tworzenie cyfrowego odcisku palca bramy VPN	271
Przechwytywanie kluczy PSK	272
Łamanie kluczy PSK w trybie offline	272
Identyfikacja domyślnych kont użytkowników	273
Podsumowanie	273
Rozdział 9. Ataki po stronie klienta	275
Backdooring — tworzenie plików wykonywalnych wyposażonych w tylne wejścia	276
Atakowanie systemów przy użyciu złośliwych skryptów	279
Przeprowadzanie ataków za pomocą skryptów w języku VBScript	279
Atakowanie systemów przy użyciu skryptów powłoki PowerShell	282
Pakiet XSS Framework	285
Pakiet BeEF	289
Konfigurowanie pakietu BeEF	290
Praca z pakietem BeEF	293
Integracja pakietów BeEF i Metasploit	296
Używanie pakietu BeEF jako tunelującego serwera proxy	297
Podsumowanie	299
Rozdział 10. Omijanie mechanizmów zabezpieczających	301
Omijanie zabezpieczeń wprowadzanych przez mechanizm NAC	302
Weryfikacja przed uzyskaniem dostępu do sieci	303
Weryfikacja po uzyskaniu dostępu do sieci	305
Omijanie programów antywirusowych przy użyciu różnych narzędzi	305
Korzystanie z pakietu Veil Framework	307
Używanie programu Shellter	312
Omijanie zabezpieczeń działających na poziomie aplikacji	316
Zastosowanie protokołu SSH do tunelowania połączeń przez zapory sieciowe działające po stronie klienta	316
Omijanie białej listy aplikacji	320

Omijanie zabezpieczeń systemu operacyjnego Windows	322
Pakiet EMET (Enhanced Migration Experience Toolkit)	322
UAC — kontrola konta użytkownika	323
Inne zabezpieczenia systemu operacyjnego Windows	328
Podsumowanie	331
Rozdział 11. Wykorzystywanie podatności i luk w zabezpieczeniach	333
Pakiet Metasploit	334
Biblioteki	334
Interfejsy	335
Moduły	336
Tworzenie i konfiguracja bazy danych	337
Atakowanie celów przy użyciu pakietu Metasploit Framework	342
Atakowanie pojedynczych systemów z użyciem odwróconej powłoki	342
Atakowanie pojedynczych systemów z użyciem odwróconej powłoki PowerShell	344
Atakowanie wielu systemów przy użyciu plików zasobów pakietu Metasploit Framework	345
Atakowanie wielu systemów przy użyciu pakietu Armitage	346
Używanie publicznych exploitów	349
Lokalizowanie i weryfikowanie publicznie dostępnych exploitów	349
Kompilowanie i używanie exploitów	351
Tworzenie exploitów dla systemu Windows	353
Identyfikacja podatności i luk w zabezpieczeniach przy użyciu fuzzingu	354
Tworzenie exploita dla systemu Windows	360
Podsumowanie	363
Rozdział 12. Powłamaniowa eksploracja środowiska celu	365
Eksploracja skompromitowanego systemu lokalnego	366
Przeprowadzenie szybkiego rozpoznania skompromitowanego systemu	367
Wyszukiwanie i pobieranie wrażliwych danych — plądrowanie celu	368
Narzędzia wspomagające powłamaniową eksplorację systemu (MSF, framework Veil-Pillage, skrypty)	372
Pakiet Veil-Pillage	375
Eskalacja pozioma i atakowanie innych systemów	379
Kompromitowanie relacji zaufania między domenami oraz udziałów sieciowych	380
PsExec, WMIC i inne narzędzia	381
Eskalacja pozioma z użyciem usług	385
Pivoting i przekierowywanie portów	385
Podsumowanie	388
Rozdział 13. Podnoszenie uprawnień	389
Typowa metodologia podnoszenia uprawnień	390
Podnoszenie uprawnień w systemie lokalnym	391
Podnoszenie uprawnień z poziomu administratora na poziom systemu	392
Wstrzykiwanie bibliotek DLL	393
Narzędzie PowerShell Empire	395

Ataki pozwalające na zbieranie poświadczeń i podnoszenie uprawnień	400
Sniffery haseł	401
Responder	402
Ataki typu SMB relay	405
Podnoszenie uprawnień w Active Directory	405
Ataki typu Golden Ticket na protokół Kerberos	412
Podsumowanie	414
Rozdział 14. Sterowanie i kontrola	415
<hr/>	
Używanie agentów persystencji	416
Używanie programu Netcat jako agenta persystencji	417
Zastosowanie programu shtasks do konfigurowania trwałych zadań	421
Utrzymywanie trwałego dostępu przy użyciu pakietu Metasploit	422
Używanie skryptu persistence	423
Tworzenie samodzielnego trwałego agenta z wykorzystaniem pakietu Metasploit	424
Utrzymywanie trwałego dostępu za pomocą mediów społecznościowych i poczty Gmail	426
Eksfiltracja danych	429
Korzystanie z istniejących usług systemowych (Telnet, RDP i VNC)	430
Eksfiltracja danych z wykorzystaniem protokołu DNS	431
Eksfiltracja danych z wykorzystaniem protokołu DNS	433
Pakiet Data Exfiltration Toolkit (DET)	435
Eksfiltracja danych z wykorzystaniem powłoki PowerShell	437
Ukrywanie śladów ataku	437
Podsumowanie	439
Skorowidz	441
<hr/>	

Testy penetracyjne ukierunkowane na osiągnięcie celu

„Istnieją tylko dwa rodzaje ludzi: ci, którzy hakują, i ci, którzy zostaną zhakowani”

Wszystko rozpoczyna się od wyznaczenia celu, który chcesz osiągnąć. Z tego względu w tym rozdziale będziemy omawiać znaczenie testów penetracyjnych ukierunkowanych na osiągnięcie celu oraz wyjaśnimy, dlaczego bez wyznaczenia takiego celu przeprowadzanie skanowania w poszukiwaniu podatności (ang. *vulnerability scanning*), testy penetracyjne czy inne działania zespołów Red Team¹ mogą zakończyć się spektakularnym niepowodzeniem. W tym rozdziale omówimy również szereg zagadnień związanych z przeprowadzaniem testów bezpieczeństwa, tworzeniem i konfigurowaniem środowiska testowego oraz używaniem systemu Kali Linux do przeprowadzania zaawansowanych testów penetracyjnych. Po przeczytaniu tego rozdziału będziesz wiedział:

- Na czym polegają testy bezpieczeństwa systemów informatycznych.
- Dlaczego skanowanie w poszukiwaniu podatności, testy penetracyjne i inne działania zespołu Red Team mogą zakończyć się niepowodzeniem.
- Jak aktualizować i konfigurować system Kali Linux.
- Jak za pomocą skryptów powłoki *bash* dostosowywać system Kali Linux do własnych potrzeb.

¹ Red Team — zespół czerwony, ofensywny, w scenariuszach testów cyberbezpieczeństwa odpowiedzialny za przeprowadzanie kontrolowanych ataków mających na celu przełamywanie zabezpieczeń środowiska komputerowego chronionego przez zespół defensywny (Blue Team; zespół niebieski) — *przyp. tłum.*

- Jak wyznaczać cele działania.
- Jak zbudować środowisko testowe.

Koncepcyjny przegląd testów bezpieczeństwa

Praktycznie każda firma czy organizacja na świecie może się obawiać różnych zagrożeń związanych z cyberprzestrzenią, takich jak wycieki wrażliwych danych, złośliwe oprogramowanie czy cyberterrorizm. Każda próba przeciwdziałania musi rozpocząć się od przygotowania ogólnego zarysu i koncepcji planowanych zabezpieczeń. Jeżeli stu różnym konsultantom zadasz pytanie: „Czym jest testowanie bezpieczeństwa środowiska teleinformatycznego?”, to z dużą dozą prawdopodobieństwa możesz założyć, że otrzymasz bardzo wiele różnych odpowiedzi. Najprościej mówiąc, testowanie bezpieczeństwa to proces polegający na sprawdzaniu, czy dane środowisko teleinformatyczne i poszczególne działające w nim systemy są odpowiednio chronione i czy mogą funkcjonować zgodnie z oczekiwaniami.

Zmierzch klasycznych testów penetracyjnych, skanowania w poszukiwaniu podatności i działań zespołów Red Team

W tym podrozdziale skoncentrujemy się na omawianiu ograniczeń tradycyjnych bądź, jak kto woli, klasycznych metod przeprowadzania testów penetracyjnych, wyszukiwania podatności i działań zespołów Red Team. Najpierw jednak spróbujemy w prostych słowach objaśnić obecne znaczenie tych trzech zagadnień i pokazać ich ograniczenia.

- **Skanowanie w poszukiwaniu podatności** (ang. *vulnerability scanning*) — jest to proces mający na celu identyfikację podatności na ataki i luk w zabezpieczeniach atakowanego środowiska i działających w nim systemów. Poważnym ograniczeniem takiego skanowania jest fakt, że jego wyniki to lista potencjalnych podatności, z których duża część może być fałszywie pozytywna. Dla właściciela danego biznesu może to stanowić poważny problem z oszacowaniem ryzyka, ponieważ w takiej sytuacji nie mamy jasnego obrazu tego, która z podatności stanowi rzeczywiste zagrożenie dla funkcjonowania środowiska, a która jest tylko fałszywie potencjalnym zagrożeniem niemającym odzwierciedlenia w rzeczywistości.
- **Testy penetracyjne** (ang. *penetration testing*) — jest to proces mający na celu dokonanie próby bezpiecznego wykorzystania wykrytych wcześniej podatności i luk w zabezpieczeniach do przeprowadzenia kontrolowanego ataku na badane

środowisko bez wprowadzania poważnych zagrożeń dla jego normalnego funkcjonowania. Przeprowadzenie takich testów daje znacznie mniejszy odsetek wyników fałszywie pozytywnych, ponieważ pentesterzy próbują użyć każdej ze znalezionych podatności. Poważnym ograniczeniem testów penetracyjnych może być to, że z reguły są one zawężane tylko do publicznie znanych podatności i exploitów. Co więcej, podczas przeprowadzania testów penetracyjnych często słyszymy słowa: „Bingo! Mamy roota!”, ale bardzo rzadko pada pytanie: „Co robimy dalej?”. Dzieje się tak z wielu bardzo różnych powodów, takich jak narzucone z góry ograniczenia zakresu przeprowadzanych testów penetracyjnych, konieczność raportowania tylko podatności wysokiego ryzyka czy ograniczenie przez klienta zakresu testów tylko do wybranych systemów czy segmentów sieci.

- **Ćwiczenia zespołów Red Team** (ang. *Red Team Exercises*) — jest to proces szacowania efektywności ochrony badanego środowiska przed zagrożeniami z cyberprzestrzeni i poprawiania zaimplementowanych zabezpieczeń. Podczas takich ćwiczeń z reguły stosowanych jest wiele różnych sposobów atakowania badanego środowiska, wykorzystujących metody socjotechniczne, kampanie phishingowe, ataki na sieci bezprzewodowe czy fizyczne testy penetracyjne. Ograniczeniem takich ćwiczeń mogą być ramy czasowe, restrykcje budżetowe, postępowanie według z góry ustalonych scenariuszy czy pomijanie niektórych działań, które mogą być zbyt niebezpieczne do przeprowadzenia w rzeczywistym środowisku produkcyjnym.

Bardzo często wszystkie trzy opisane procesy powiązane są z określeniami takimi jak *hacking* czy *łamanie zabezpieczeń*. Możemy co prawda powiedzieć klientowi, że spróbujemy włamać się do jego sieci i pokazać słabe strony jej zabezpieczeń, ale czy tak naprawdę klient czy właściciel środowiska rozumie, na czym takie „hakowanie” czy „łamanie zabezpieczeń” będzie polegało? Jak możemy je zmierzyć? Jakie są kryteria „hakowania”? Skąd możemy wiedzieć, że „hakowanie” czy „łamanie zabezpieczeń sieci” zostało zakończone? Wszystkie tego typu pytania w prostej mierze prowadzą do jednego, fundamentalnego zagadnienia — jaki jest podstawowy cel przeprowadzania danej operacji?

Podstawowym celem przeprowadzania testów penetracyjnych czy ćwiczeń zespołów Red Team jest określenie stopnia podatności danego środowiska na cyberataki, wyznaczenie zagrożeń dla jego poszczególnych elementów składowych (poszczególnych segmentów sieci i działających w nim systemów) oraz oszacowanie ryzyka, jakie stanowią dla funkcjonowania całego środowiska firmy czy organizacji. Z reguły nie jest to jednak kwestia ilości podatności znalezionych na poszczególnych hostach, ale to, jak bardzo eksponowany jest dany system i jak jego działanie jest krytyczne dla funkcjonowania całego środowiska. Nie każda znaleziona podatność i luka w zabezpieczeniach jest istotna i nie zawsze musi się ona wiązać z poważnym zagrożeniem. Na przykład znaleziona podatność na ataki typu *Cross-Site Scripting* (XSS) na odseparowanym, informacyjnym serwerze WWW nie musi stanowić poważnego zagrożenia dla funkcjonowania całej firmy; nie zmienia to jednak faktu, że po wykryciu takiej luki właściciel systemu może podjąć decyzję o minimalizacji zagrożenia poprzez zaimplementowanie takich rozwiązań jak WAF (ang. *Web Application Firewall*), zapobiegających przeprowadzaniu ataków typu XSS.

Metodologia przeprowadzania testów

Niestety metodologia przeprowadzania testów penetracyjnych bardzo często nie obejmuje powodów, dla których klient zlecił przeprowadzenie takich testów, ani nie zawiera listy danych, które są krytyczne dla biznesu i powinny być chronione w szczególny sposób. Pominięcie tak istotnych informacji już na samym początku procesu może spowodować, że właściwy cel przeprowadzania takiego testu penetracyjnego ulegnie rozmyciu.

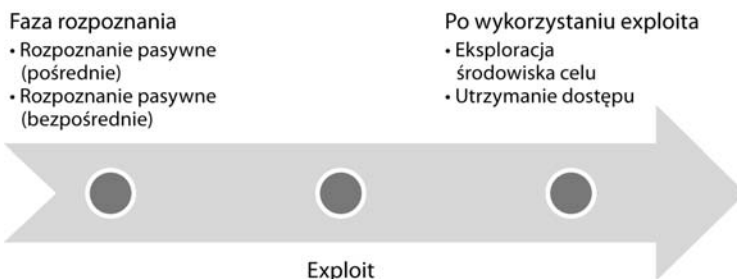
Bardzo wielu pentesterów niechętnie podchodzi do z góry wyznaczonych metodologii postępowania, obawiając się, że może to w jakiś sposób ograniczać ich kreatywność w wyszukiwaniu podatności i luk w zabezpieczeniach, a następnie w ich efektywnym wykorzystywaniu. Testy penetracyjne często nie odzwierciedlają sposobu, w jaki może postępować potencjalny napastnik. Na przykład klient zlecający test chce sprawdzić, czy pentester będzie w stanie w nieautoryzowany sposób uzyskać dostęp na poziomie administratora do atakowanego systemu, podczas gdy intencją prawdziwego napastnika może być skopiowanie z tego systemu wrażliwych danych w sposób, który ani nie wymaga uzyskania takiego dostępu, ani nie spowoduje awarii systemu.

Aby można było poradzić sobie z ograniczeniami wynikającymi z formalnych metod testowania, testy penetracyjne muszą być zintegrowane w ramach, które pozwalają spojrzeć na atakowaną sieć z punktu widzenia napastnika i w uporządkowany sposób przeprowadzić skuteczny atak.

W roku 2009 Mike Cloppert, pracujący w zespole CERT firmy Lockheed Martin, opracował koncepcję znaną obecnie jako atak cybernetyczny (ang. *Attacker Kill Chain*), która opisywała poszczególne kroki, jakie musi podjąć napastnik w celu przeprowadzenia efektywnego cyberataku na system sieciowy. Ataki takie nie zawsze mają przebieg liniowy, ponieważ niektóre z ich faz mogą być przeprowadzane równolegle. W danym okresie ten sam cel może być atakowany wielokrotnie, przy czym niektóre równoległe fazy ataku mogą być przeprowadzane w tym samym czasie.

W tej książce zmodyfikowaliśmy nieco oryginalny koncept cyberataku opracowany przez Mike'a Clopperta, aby jeszcze dokładniej odzwierciedlić poszczególne operacje podejmowane przez potencjalnego napastnika podczas atakowania całych sieci komputerowych oraz funkcjonujących w nich systemów, aplikacji i usług sieciowych.

Na rysunku przedstawionym poniżej pokazano, jak wygląda typowy atak cybernetyczny:



Typowy atak cybernetyczny może wyglądać tak:

- **Faza rozpoznania** — twierdzenie, że „rekonesans nigdy nie jest stratą czasu”, przyjęte i powszechnie stosowane przez wojsko na całym świecie, jest prostym potwierdzeniem tego, że przed rozpoczęciem ataku zawsze dobrze jest zebrać jak największą ilość informacji o nieprzyjacielu. Z tego samego powodu hakerzy przed rozpoczęciem cyberataku przeprowadzają zakrojone na szeroką skalę rozpoznanie celu. W praktyce szacuje się, że na przeprowadzenie odpowiedniego rozpoznania środowiska celu cyberprzestępcy (jak również profesjonalni pentesterzy) zużywają nawet 70 procent czasu i zasobów całego ataku! Ogólnie rzecz biorąc, cyberprzestępcy najczęściej wykorzystują dwa scenariusze fazy rozpoznania:
 - **Rozpoznanie pasywne** — w takim scenariuszu napastnik nie wchodzi w bezpośrednią interakcję ze środowiskiem celu, lecz korzysta na przykład z zawartości publicznie dostępnych stron internetowych, materiałów konferencyjnych, informacji dostępnych w mediach i serwisach sieciowych (a zwłaszcza w serwisach społecznościowych). Bazując na pozyskanych informacjach, stara się wyznaczyć odpowiednią **plaszczynę ataku** (ang. *attack surface*). Jedną z operacji często wykonywanych podczas rozpoznania pasywnego jest przygotowywanie listy obecnych i poprzednich pracowników atakowanej firmy, która może stanowić dobrą bazę do łamania haseł metodą brute-force lub może być wykorzystana do przeprowadzania ataków socjotechnicznych. Rozpoznanie przeprowadzane w sposób pasywny jest bardzo trudne, a często wręcz niemożliwe do wykrycia.
 - **Rozpoznanie aktywne** — aktywność napastnika związana z aktywnym rozpoznawaniem środowiska celu może zostać wykryta, ale w praktyce bardzo trudno ją odróżnić od tego, z czym większość firm czy organizacji styka się na co dzień. Rozpoznanie aktywne może obejmować skanowanie portów, zdalne skanowanie systemów w poszukiwaniu podatności i luk w zabezpieczeniach, a nawet fizyczne odwiedziny w siedzibie firmy pod pozorem udawania klienta, dostawcy pizzy czy serwisanta.
- **Faza dostawy** — w tej fazie dokonywany jest wybór i opracowanie narzędzia, które zostanie użyte do wykorzystania danego exploita podczas przeprowadzania ataku. To, jakie narzędzie zostanie finalnie wybrane, zależy zarówno od intencji atakującego, jak i od planowanego sposobu dostarczenia narzędzia do atakowanego celu (na przykład za pomocą wiadomości poczty elektronicznej, sieci bezprzewodowej czy usługi sieci WWW). Znaczenie fazy dostawy i jej wpływ na przeprowadzanie ataku zostaną szczegółowo omówione w drugiej połowie książki.
- **Faza penetracji (faza ataku)** — jest to faza, w której określony exploit zostaje pomyślnie wykorzystany, co pozwala napastnikowi na osiągnięcie zaplanowanego celu. Skompromitowanie celu mogło wystąpić w jednej fazie (na przykład napastnik wykorzystał dobrze znaną podatność danego systemu na ataki z przepełnianiem bufora), ale równie dobrze mogło wymagać przeprowadzenia operacji wielofazowej (na przykład napastnik, wykorzystując metody socjotechniczne, przedostał się na teren firmy i skradł egzemplarz firmowej książki telefonicznej. Znajdujące się tam informacje zostały wykorzystane do utworzenia słownika danych użytego następnie

do przeprowadzenia ataku typu brute-force na portal firmy. Dodatkowo na pozyskane poprzez kradzież książki adresy poczty elektronicznej pracowników napastnik porożysłał wiadomości mające na celu skłonić użytkowników do kliknięcia osadzonego łącza i pobrania odpowiednio spreparowanego, złośliwego dokumentu PDF, za pomocą którego napastnik mógłby przejąć kontrolę nad zainfekowanymi systemami). Ataki wielofazowe są powszechnie wykorzystywane w scenariuszach, gdzie celem ataku jest określona firma czy organizacja.

- **Faza *post-exploit* (eksploracja środowiska celu)** — faza eksploracji środowiska celu bardzo często bywa niepoprawnie nazwana **fazą eksfiltracji** (ang. *exfiltration phase*), ponieważ dość powszechnie takie ataki postrzegane są niemal wyłącznie jako metody kradzieży poufnych i wrażliwych danych (takich jak listy użytkowników i haseł dostępu, dane osobowe czy informacje finansowe). W praktyce jednak napastnicy mają często zupełnie inne cele. Na przykład nieuczciwa firma może być zainteresowana przeprowadzeniem ataku typu DoS na internetową sieć usług swojego konkurenta, czego efektem będzie potencjalne zwiększenie zainteresowania klientów swoimi rozwiązaniami. Z tego względu faza eksploracji środowiska celu musi skupiać się na wielu potencjalnie możliwych czynnościach napastnika. Jedne z najczęściej obserwowanych działań związanych z exploitami mają miejsce wtedy, gdy napastnicy próbują zwiększyć swoje uprawnienia do najwyższego możliwego poziomu (eskalacja pionowa) i uzyskać dostęp do jak największej liczby kont i systemów (eskalacja pozioma).
- **Faza *post-exploit* (utrzymanie dostępu)** — przełamanie zabezpieczeń i uzyskanie dostępu do atakowanego systemu może być dla napastnika bardzo wartościowe, ale z pewnością ta wartość znacząco wzrośnie, jeżeli napastnikowi uda się utworzyć przyczółek i zachować stały dostęp do skompromitowanego systemu, dzięki czemu będzie mógł w dowolnym momencie powrócić do zaatakowanego systemu. Z punktu widzenia zespołu odpowiedzialnego za ochronę systemu ten element cyberataku jest zazwyczaj najłatwiejszy do wykrycia.

Przebieg ataku cybernetycznego (ang. *kill chain*) jest swego rodzaju metamodelem zachowania napastnika próbującego przełamać zabezpieczenia atakowanego celu i uzyskać do niego nieautoryzowany dostęp. Jako metamodel atak cybernetyczny może zawierać dowolną, otwartą lub komercyjną metodologię przeprowadzania testów penetracyjnych. W przeciwieństwie jednak do metodologii cyberatak pokazuje przebieg ataku na środowisko celu z perspektywy strategicznej. Układ i zawartość naszej książki zostały dobrane tak, aby odzwierciedlić czynności wykonywane przez napastnika podczas przeprowadzania ataku.

Wprowadzenie do systemu Kali Linux — jego historia i przeznaczenie

System Kali Linux jest następcą znanego i bardzo popularnego niegdyś systemu BackTrack. W środowisku pentesterów Kali Linux jest swego rodzaju standardem wśród narzędzi i pakietów wspomagających przeprowadzanie testów penetracyjnych sieci komputerowych. Autorami

systemu są Mati Aharoni oraz Devon Kearns z firmy Offensive Security. Poniżej przedstawiamy krótką historię systemu Kali Linux od momentu jego powstania:

- W marcu 2013 roku system BackTrack został zastąpiony systemem Kali Linux, wykorzystującym nową architekturę opartą na systemie Debian GNU/Linux.
- **Kali 1.1.0 (data wydania: 9 lutego 2015)** — kolejna wersja systemu Kali Linux pojawiła się dopiero po dwóch latach od jego premiery i przyniosła aktualizację jądra systemu do wersji 3.18, poprawki modułów wspomagających ataki ze wstrzykiwaniem ramek do sieci Wi-Fi oraz obsługę nowych sterowników bezprzewodowych kart sieciowych; w sumie w tym wydaniu poprawionych zostało ponad 58 błędów. W następnych wydaniach tej wersji, takich jak Kali 1.1.0a, poprawione zostały również drobne błędy i inne problemy instalatora.
- **Kali 2.0 (data wydania: 11 sierpnia 2015)** — wersja 2.0 była bardzo poważną aktualizacją, zawierającą m.in. duże zmiany w interfejsie użytkownika; stała się tzw. dystrybucją ciągłą systemu Kali Linux (ang. *rolling distribution*). Począwszy od wersji 2.0, można dokonywać aktualizacji systemu Kali Linux ze starszej do nowszej wersji.
- **Kali 2016.1 (data wydania: 21 stycznia 2016)** — pierwsze ciągle wydanie systemu Kali Linux, które przyniosło aktualizację jądra do wersji 4.3 oraz nowe środowisko GNOME w wersji 3.18.
- **Kali 2016.2 (data wydania: 31 sierpnia 2016)** — kolejne ciągle wydanie systemu Kali Linux, wyposażone w jądro 4.6, środowisko GNOME 3.20.2 oraz wiele poprawek i aktualizacji.

System Kali Linux w wersji 2.0 (i w kolejnych aktualizacjach) posiada między innymi następujące cechy:

- Ponad 300 narzędzi wspomagających przeprowadzanie testów penetracyjnych oraz ekspertyz z informatyki śledczej. W kolejnych wydaniach systemu poszczególne narzędzia są sukcesywnie aktualizowane bądź zastępowane nowymi wersjami. Wiele z tych narzędzi współpracuje z bezprzewodowymi kartami sieciowymi i pozwala na przeprowadzanie testów penetracyjnych w sieciach Wi-Fi.
- Obsługa wielu środowisk desktopowych, takich jak KDE, GNOME, Xfce, Mate, e17, lxde czy i3wm.
- Narzędzia zgodne z dystrybucją Debian są synchronizowane z repozytoriami co najmniej cztery razy dziennie, co pozwala na ich aktualizację oraz instalację poprawek niemal natychmiast po ich opublikowaniu.
- Obsługa środowiska Secure Development Environment oraz pakietów i repozytoriów podpisanych kluczem GPG.
- Wsparcie dla użytkowników chcących przygotować indywidualne, dostosowane do własnych potrzeb dystrybucje ISO systemu Kali Linux. Funkcja bootstrap znakomicie ułatwia instalowanie systemu również w dużych środowiskach korporacyjnych, gdzie proces instalacji może być zautomatyzowany przy użyciu predefiniowanych plików konfiguracyjnych (ang. *Linux preseed files*).

- Ze względu na rosnącą popularność (i spadające ceny) systemów z procesorami ARM w systemie zaimplementowana została również obsługa architektur ARMEL i ARMHF, dzięki czemu Kali Linux można instalować na takich urządzeniach jak rk3306 mk/ss808, Raspberry Pi, ODROID U2/X2, Samsung Chromebook, EfikaMX, Beaglebone Black, CuBox czy Galaxy Note 10.1.
- Mimo rosnącej popularności Kali Linux nadal pozostaje systemem klasy open source, dostępnym bezpłatnie dla każdego i co najważniejsze — aktywnie wspieranym przez ogromną społeczność użytkowników.

Celem systemu Kali Linux jest zebranie wszystkich najważniejszych narzędzi wspomagających przeprowadzanie testów bezpieczeństwa i testów penetracyjnych w jednej, elastycznej i bardzo uniwersalnej platformie systemowej.

Instalowanie i aktualizowanie systemu Kali Linux

W poprzednich wydaniach tej książki koncentrowaliśmy się głównie na procesie instalacji systemu Kali Linux w maszynach wirtualnych VMware, zatem tym razem postaramy się nieco bardziej zagłębić w różne inne techniki instalowania i aktualizowania systemu Kali Linux.

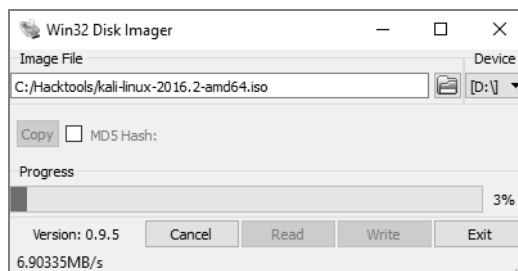
Uruchamianie systemu Kali Linux z urządzenia przenośnego

Proces instalowania systemu Kali Linux na urządzeniu przenośnym jest całkiem prosty. W niektórych sytuacjach klienci nie zezwalają na podłączenie i używanie „obcych” laptopów w ich sieci — klient dostarcza wtedy pentesterowi odpowiednie komputery, przy użyciu których ma on przeprowadzić testowanie. Przeprowadzanie testów penetracyjnych przy użyciu systemu Kali Linux uruchomionego z urządzenia przenośnego ma wiele zalet, na przykład:

- Wszystko zmieścisz w kieszeni (pamięci USB czy zewnętrzne dyski twarde mają małe rozmiary).
- System Kali Linux możesz uruchomić na komputerze bez konieczności wprowadzania jakichkolwiek modyfikacji w zainstalowanym systemie operacyjnym.
- Korzystając z urządzeń przenośnych, możesz utworzyć kilka osobnych wersji systemu Kali Linux, dostosowanych do określonych scenariuszy testowania i wymagań klienta.

Jeżeli pracujesz na komputerze działającym pod kontrolą systemu Windows, zainstalowanie systemu Kali Linux na urządzeniu przenośnym będzie wymagało wykonania zaledwie trzech kroków:

1. Pobierz oficjalny obraz systemu Kali Linux ze strony internetowej o adresie:
<https://docs.kali.org/introduction/download-official-kali-linux-images>
2. Pobierz program Win32 Disk Imager ze strony internetowej
<https://sourceforge.net/projects/win32diskimager/>.
3. Uruchom program Win32 Disk Imager jako administrator. Podłącz przenośną pamięć USB do dowolnego portu USB komputera. Na ekranie powinno pojawić się okno podobne do przedstawionego poniżej. Wybierz urządzenie, na którym chcesz zainstalować system Kali Linux, i naciśnij przycisk *Write* (zapisz):



Po zakończeniu instalowania systemu zamknij program Win32 Disk Imager i bezpiecznie odłącz pamięć USB. Kali Linux zainstalowany na tym urządzeniu jest już gotowy do użycia; możesz je na przykład podłączyć do laptopa i uruchomić z niego system Kali Linux. Jeżeli chcesz zainstalować system Kali Linux z poziomu komputera działającego pod kontrolą systemu Linux, proces instalacji jest jeszcze prostszy i wymaga wykonania z poziomu konsoli tylko dwóch poleceń. Pierwsze to `sudo fdisk -l`, które wyświetla listę wszystkich zamontowanych napędów dyskowych, a drugie to `dd if=kali linux.iso of=/dev/nazwa_urzadzenia bs=512k`. To wszystko. Polecenie `dd` pobiera instalacyjny plik ISO i zapisuje jego zawartość na podanym urządzeniu. Polecenie pobiera kilka argumentów wywołania, gdzie `if` to nazwa wejściowego pliku ISO, `of` to nazwa urządzenia docelowego, a `bs` to rozmiar zapisywanych bloków.

Instalowanie systemu Kali Linux w maszynie wirtualnej

W tej sekcji szczegółowo omówimy sposób instalowania systemu Kali Linux w maszynach wirtualnych działających pod kontrolą oprogramowania VMware Workstation Player oraz Oracle VirtualBox.

VMware Workstation Player

VMware Workstation Player (starsze wersje nosiły nazwę VMware Player) to oprogramowanie wirtualizacyjne, pozwalające na uruchamianie maszyn wirtualnych działających pod kontrolą innych systemów operacyjnych, które jest bezpłatne dla zastosowań osobistych. Dla zastosowań komercyjnych wymagane jest zakupienie odpowiedniej licencji. Oprogramowanie można pobrać ze strony internetowej o następującym adresie URL:

<https://www.vmware.com/products/player/playerpro-evaluation.html>

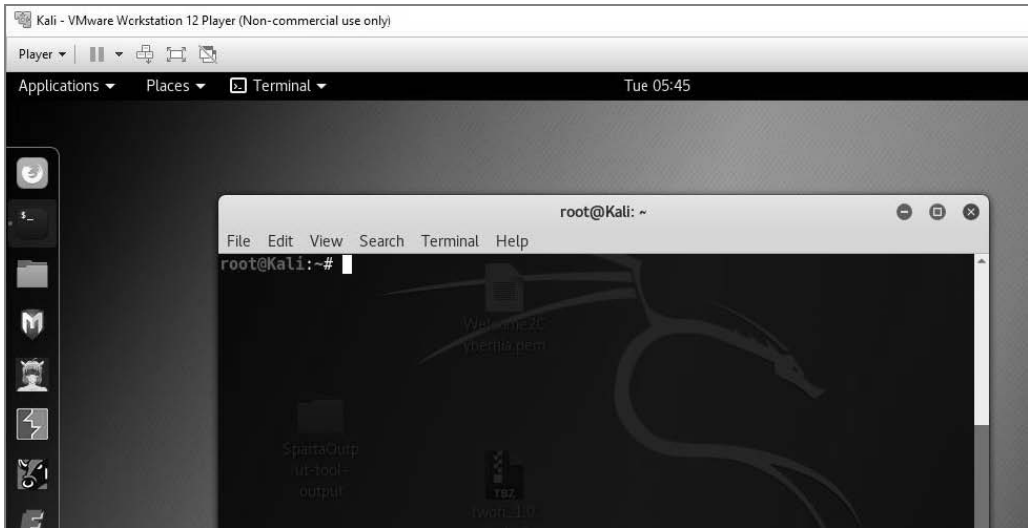
Poniżej przedstawiamy krok po kroku procedurę instalacji systemu Kali Linux w maszynie wirtualnej VMware Workstation Player.

Po pobraniu pliku instalacyjnego VMware Workstation Player uruchom go, a na ekranie pojawi się pierwszy ekran kreatora instalacji, tak jak to zostało pokazane poniżej. Naciśnij przycisk *Next* (dalej).



Na ekranie pojawi się okno z warunkami licencji. Naciśnij przycisk *Accept* (akceptuję), a następnie na kolejnych ekranach naciskaj przycisk *Next* — aż do momentu, kiedy pojawi się ekran przedstawiony na pierwszym rysunku na następnej stronie, sygnalizujący, że proces instalacji programu VMware Workstation Player został zakończony pomyślnie.

Kolejnym krokiem będzie utworzenie maszyny wirtualnej i zainstalowanie w niej systemu Kali Linux, którego instalacyjny obraz ISO pobraliśmy już wcześniej z oficjalnej strony internetowej. Naciśnij przycisk *Create a New Virtual machine* (utwórz nową maszynę wirtualną), a następnie wybierz opcję *Installer disc image file (iso)* (plik obrazu ISO dysku instalacyjnego), wskaż odpowiedni plik ISO i naciśnij przycisk *Next*. Wpisz nazwę tworzonej maszyny wirtualnej (na przykład *HackBox*) i wybierz miejsce, w którym będą przechowywane jej pliki. Naciśnij przycisk *Next*, określ rozmiar dysku maszyny wirtualnej dla systemu Kali Linux (rekomendowany rozmiar to 10 GB), naciskaj przycisk *Next* na kolejnych ekranach kreatora aż do utworzenia maszyny wirtualnej. Po jej uruchomieniu powinieneś zobaczyć następujący ekran (patrz drugi rysunek na następnej stronie).



Począwszy od wersji 2016.2, z systemu Kali Linux zostały usunięte repozytoria Sana. *Sana* to nazwa kodowa wersji systemu Kali Linux, która używała repozytoriów zawierających pakiety. Z tego powodu pierwszym, zdecydowanie zalecanym krokiem, jaki powinieneś wykonać po zainstalowaniu i uruchomieniu systemu Kali Linux, powinno być wykonanie polecenia `apt-get update`, dzięki któremu zaktualizowana zostanie zawartość pliku *sources.lst*.

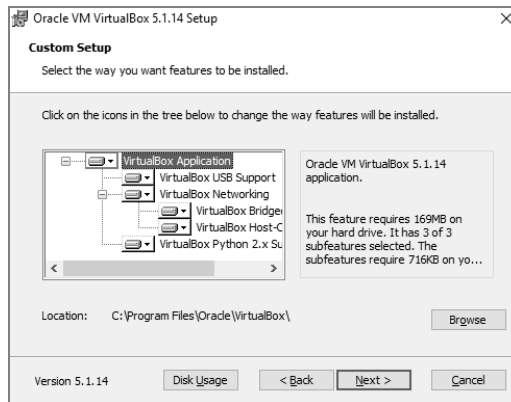
VirtualBox

Program VirtualBox jest rozwiązaniem bardzo podobnym do VMware Workstation Player. Jest to oprogramowanie pozwalające na tworzenie i uruchamianie maszyn wirtualnych, które jest bezpłatnie udostępniane w wersji open source. Pakiet VirtualBox możesz pobrać z następującej strony internetowej:

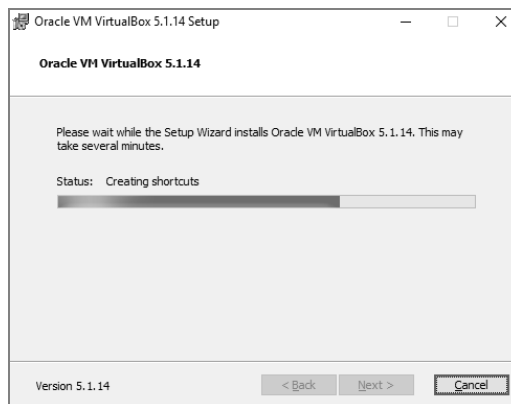
<https://www.virtualbox.org/wiki/Downloads>

Teraz pokażemy sposób instalacji systemu Kali Linux w maszynie wirtualnej działającej pod kontrolą oprogramowania VirtualBox. Podobnie jak to miało miejsce w przypadku VMware, najpierw musimy zainstalować sam pakiet VirtualBox. Aby to zrobić, uruchom pobrany program instalacyjny; na ekranie powinieneś zobaczyć pierwsze okno kreatora instalacji (patrz pierwszy rysunek na następnej stronie).

Naciśnij przycisk *Next* (dalej), a na ekranie pojawi się lista opcji konfiguracyjnych pakietu. W naszym przypadku wybierzemy domyślną opcję *VirtualBox Application* (patrz drugi rysunek na następnej stronie).



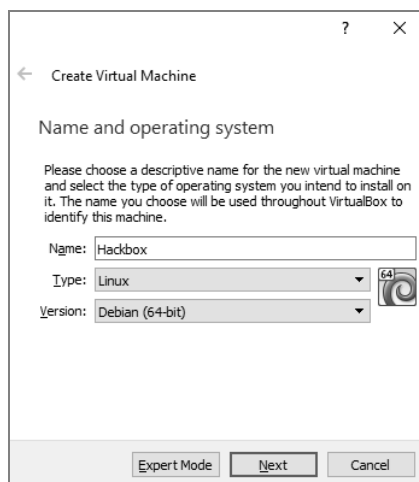
Naciśnij przycisk *Next*, a na ekranie pojawi się kolejne okno kreatora, pokazujące postęp procesu instalacji, tak jak to zostało pokazane na rysunku poniżej:



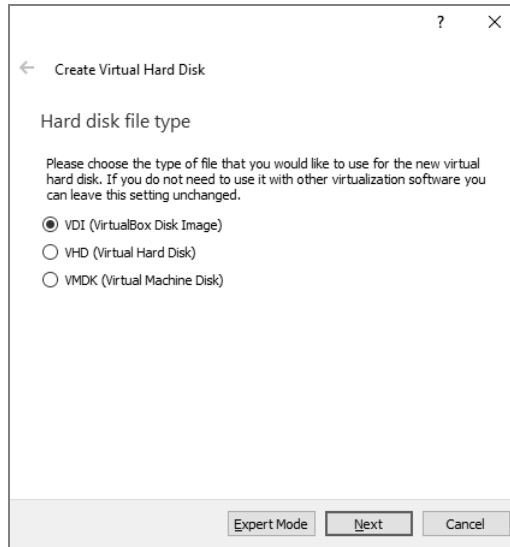
Jeżeli proces instalacji pakietu Oracle VirtualBox zakończy się pomyślnie, po uruchomieniu programu ekran powinien wyglądać tak, jak to zostało pokazane poniżej:



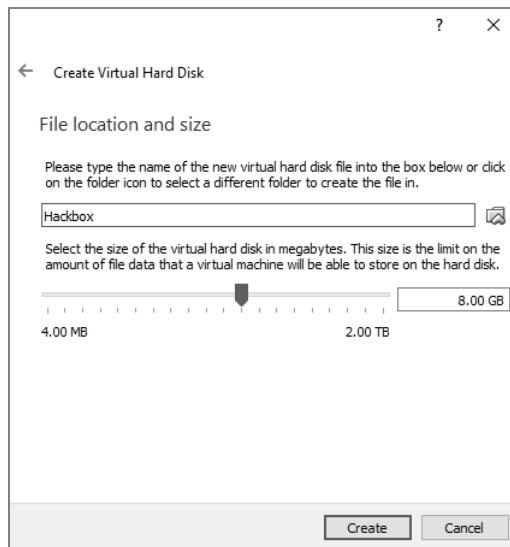
Kolejnym krokiem będzie utworzenie maszyny wirtualnej i zainstalowanie w niej systemu Kali Linux. Aby to zrobić, naciśnij przycisk *New* (nowy), znajdujący się na pasku narzędzi programu VirtualBox. Na ekranie powinno się pojawić okno kreatora tworzenia nowej maszyny wirtualnej. W polu *Name* (nazwa) wpisz żądaną nazwę maszyny (na przykład *HackBox*), a następnie w zależności od pobranej wersji systemu Kali Linux wybierz odpowiedni typ maszyny wirtualnej, na przykład *Debian (64-bit)* albo *Debian (32-bit)* (zobacz rysunek poniżej).



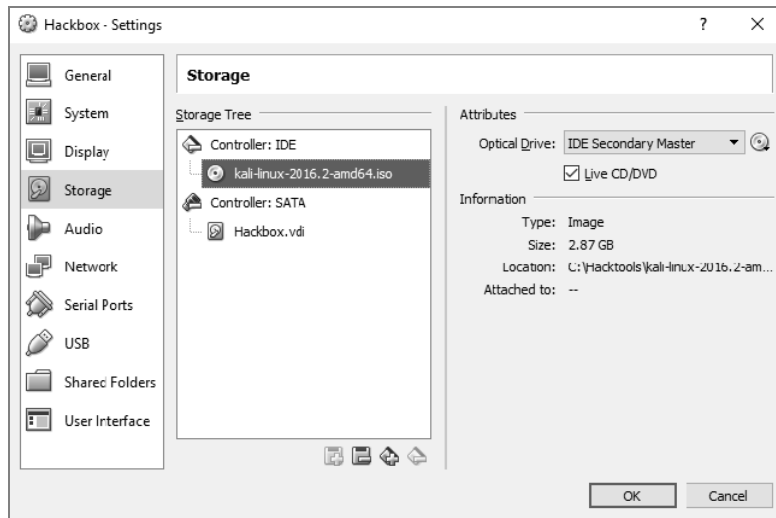
Naciśnij przycisk *Next* i podaj żądany rozmiar pamięci RAM dla systemu Kali Linux. Rekomendowana wielkość to co najmniej 1 GB. W kolejnym kroku utworzymy wirtualny dysk twardej dla instalowanego systemu. Naciśnij przycisk *Next* i wybierz typ dysku; w większości przypadków będziemy korzystać z domyślnej opcji *VDI (VirtualBox Disk Image)*, tak jak to zostało pokazane na pierwszym rysunku na następnej stronie.



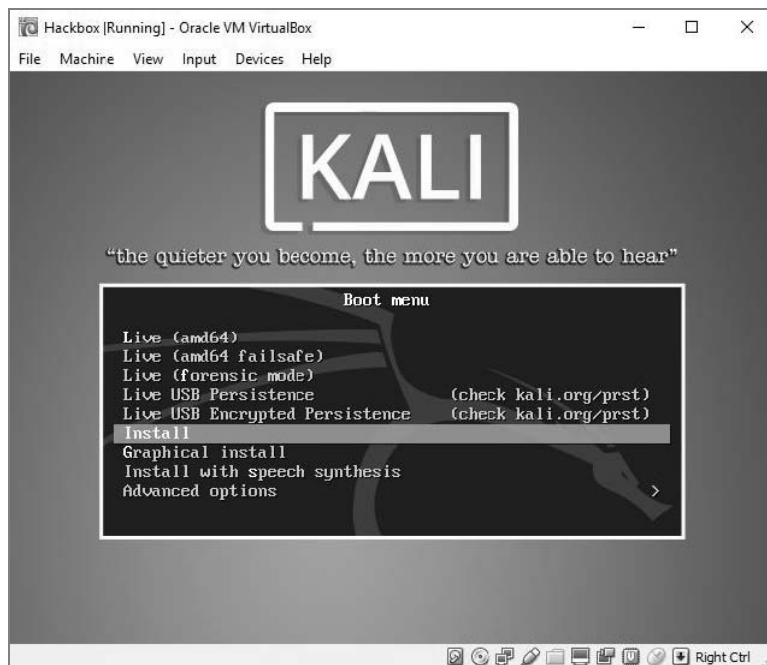
Naciśnij przycisk *Next* i w kolejnym oknie kreatora wybierz rozmiar dysku twardego, który zostanie utworzony (zobacz rysunek poniżej):



Na koniec musimy przejść do ustawień maszyny wirtualnej *HackBox* (opcja *Settings*) i podłączyć instalacyjny obraz ISO systemu Kali Linux jako dysk zewnętrzny, tak jak to zostało pokazane na pierwszym rysunku na następnej stronie.



Gotowe. Po uruchomieniu maszyny wirtualnej na ekranie powinno się pojawić okno, w którym z menu uruchomieniowego możesz wybrać opcję uruchomienia wersji Live lub zainstalowania systemu Kali Linux na dysku (zobacz rysunek poniżej):



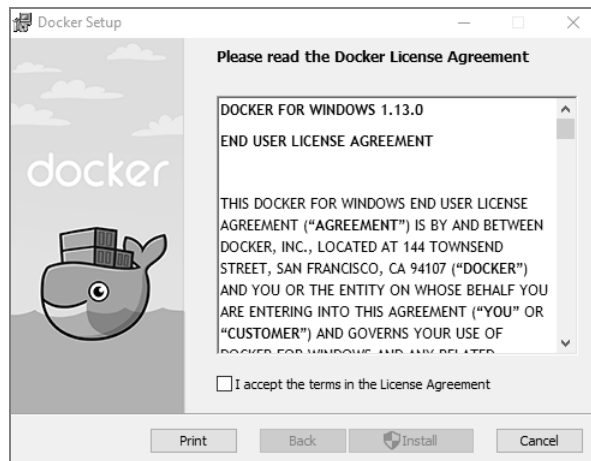
Instalowanie aplikacji Docker

Docker to aplikacja typu *open source* pozwalająca na tworzenie, dostarczanie i uruchamianie aplikacji w tzw. kontenerach. Docker zapewnia również dodatkową warstwę abstrakcji i automatyzacji na poziomie wirtualizacji systemu operacyjnego.

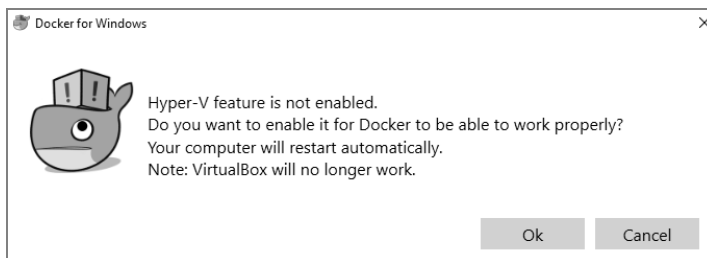
Docker jest dostępny w wersjach dla systemów Windows, macOS, AWS (ang. *Amazon Web Services*) oraz Azure. Dla systemu Windows pakiet Docker możesz pobrać z następującego adresu URL:

<https://download.docker.com/win/stable/InstallDocker.msi>

Poniżej pokażemy, w jaki sposób możesz zainstalować pakiet Docker na komputerze działającym pod kontrolą systemu Windows 10:



Docker wykorzystuje mechanizm Hyper-V systemu Microsoft Windows. Jeżeli Hyper-V nie jest włączony, to najprawdopodobniej na ekranie pojawi się następujący komunikat:



Po naciśnięciu przycisku *Ok* mechanizm Hyper-V zostanie włączony i będziesz mógł sprawdzić działanie programu Docker, uruchamiając konsolę i wykonując polecenie `docker`, tak jak to zostało pokazane na rysunku:

```

C:\Windows\system32\cmd.exe
C:\Hacktools>docker

Usage:  docker COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "C:\Users\EISC\.docker")
  -D, --debug          Enable debug mode
  --help              Print usage
  -H, --host list      Daemon socket(s) to connect to (default [])
  -l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")

  --tls               Use TLS; implied by --tlsverify
  --tlscert string    Trust certs signed only by this CA (default "C:\Users\EISC\.docker\ca.pem")
  --tlscert string    Path to TLS certificate file (default "C:\Users\EISC\.docker\cert.pem")
  --tlskey string     Path to TLS key file (default "C:\Users\EISC\.docker\key.pem")
  --tlsverify         Use TLS and verify the remote
  -v, --version       Print version information and quit

Management Commands:
  checkpoint  Manage checkpoints
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
    
```

Na tym proces instalowania aplikacji Docker zostaje zakończony. Teraz musimy zainstalować system Kali Linux. Aby to zrobić, powinieneś z poziomu konsoli wykonać polecenie `docker pull kalilinux/kali-linux-docker`, tak jak to zostało pokazane na rysunku poniżej:

```

C:\Windows\system32\cmd.exe
C:\Hacktools>docker pull kalilinux/kali-linux-docker
Using default tag: latest
latest: Pulling from kalilinux/kali-linux-docker
Digest: sha256:b89e91e9e08cbcf1a0cb825522bee556fa4b50891fffd27f1d56292e7667dcc
Status: Image is up to date for kalilinux/kali-linux-docker:latest

C:\Hacktools>
    
```

Po załadowaniu systemu Kali Linux do aplikacji Docker powinieneś być w stanie uruchomić powłokę bash z pobranego kontenera Kali Docker poprzez wykonanie polecenia `run -t -i kalilinux/kali-linux-docker /bin/bash`, tak jak to zostało pokazane na rysunku poniżej:

```

C:\Windows\system32\cmd.exe - docker run -t -i kalilinux/kali-linux-docker /bin/bash
C:\Hacktools>docker run -t -i kalilinux/kali-linux-docker /bin/bash
root@87b94bd8d4d4:/# ls
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
root@87b94bd8d4d4:/#
    
```

System Kali Linux możesz teraz uruchomić bezpośrednio z poziomu kontenera Docker. Zwróć uwagę, że Docker wykorzystuje pracujące w tle środowisko VirtualBox, zatem w takiej sytuacji nasza maszyna wirtualna z systemem Kali Linux działa w środowisku VirtualBox za pośrednictwem kontenera Docker.

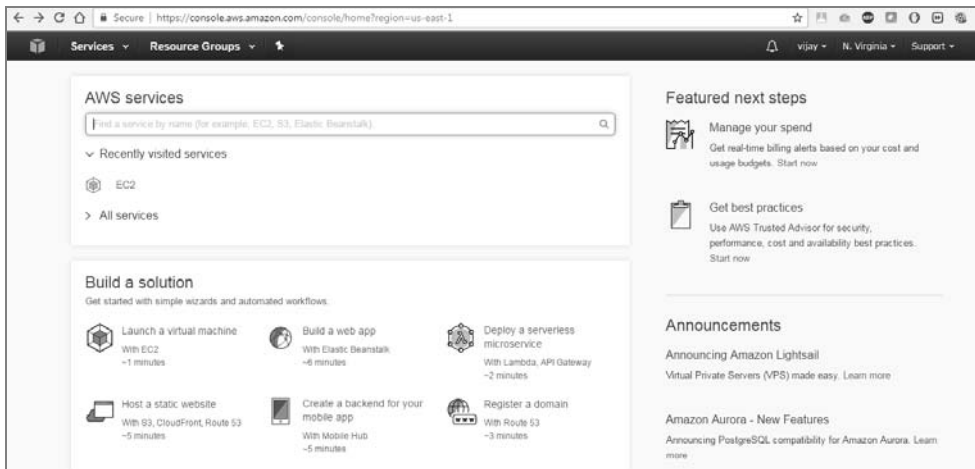
Instalowanie systemu Kali Linux w chmurze — tworzenie instancji AWS

AWS to opracowana przez firmę Amazon platforma oparta na technologii chmury, której podstawowym przeznaczeniem jest udostępnianie użytkownikom mocy obliczeniowej, przestrzeni dyskowej oraz dostarczanie treści. Jako pentester możesz wykorzystać AWS do przeprowadzania testów penetracyjnych. W tym podrozdziale pokażemy najłatwiejszą metodę zainstalowania systemu Kali Linux w chmurze AWS, co może być bardzo użyteczne w przypadku korzystania z zewnętrznych serwerów C2 (ang. *command and control*).

Najpierw musimy utworzyć swoje konto na platformie AWS. Aby to zrobić, powinieneś odwiedzić następującą stronę internetową:

<https://console.aws.amazon.com/console/home>

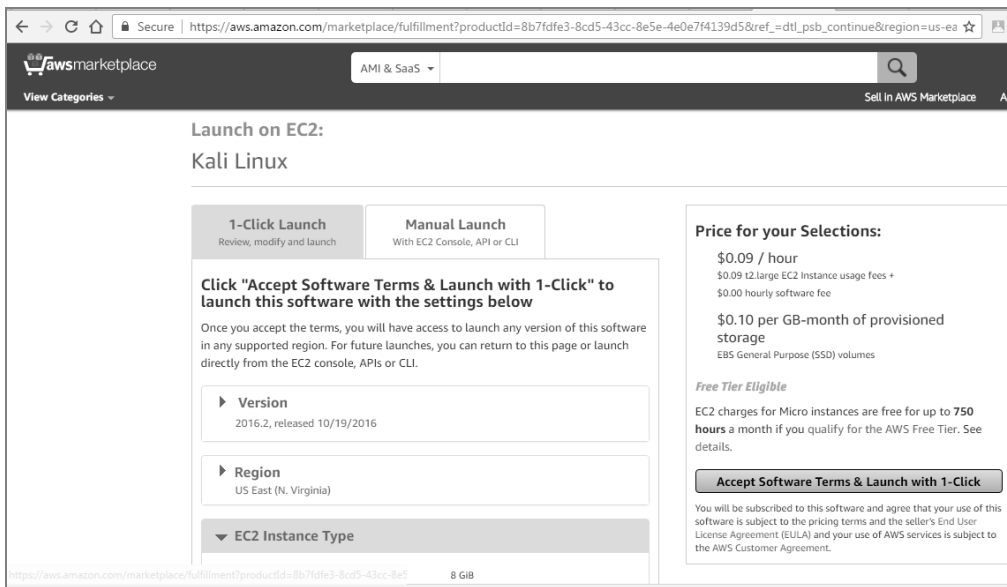
Po zalogowaniu się na konto AWS możesz zobaczyć listę wszystkich dostępnych usług, tak jak to zostało pokazane na rysunku poniżej:



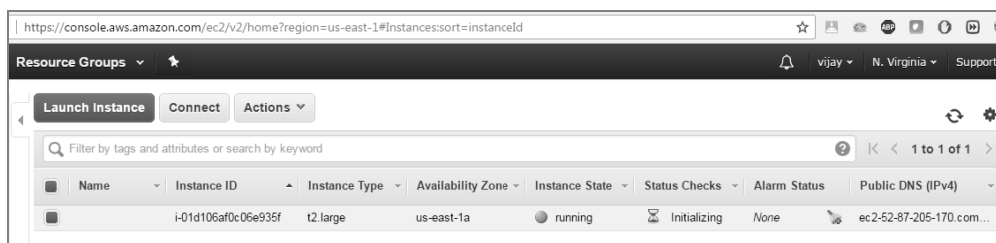
Kolejnym krokiem będzie uruchomienie systemu Kali Linux na platformie AWS. Zrobimy to, korzystając z odpowiednio dostosowanej wersji systemu Kali Linux bazującej na dystrybucji Debian. Dzięki społeczności programistów open source możemy w bardzo prosty sposób uruchomić prekonfigurowaną wersję systemu Kali Linux 2016.2 za pośrednictwem platformy Amazon Marketplace. Adres URL przedstawiony poniżej pozwoli Ci na szybkie uruchomienie systemu Kali Linux:

<https://aws.amazon.com/marketplace/pp/B01M26MMTT>

Po jego uruchomieniu w oknie przeglądarki powinna się pojawić strona, która będzie wyglądała mniej więcej tak:



Naciśnij przycisk *Accept Software Terms & Launch with 1-Click* (zaakceptuj warunki licencji i uruchom jednym kliknięciem), a następnie przejdź do konsoli AWS, odwiedzając stronę <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1>. Teraz powinieneś być już w stanie uruchomić wybraną instancję systemu Kali Linux, wybierając odpowiedni wiersz lub identyfikator instancji (ang. *Instance ID*) i naciskając przycisk *Launch Instance* (uruchom instancję), tak jak to zostało pokazane na rysunku poniżej:



Teraz musisz utworzyć parę kluczy, co zagwarantuje, że tylko Ty będziesz miał dostęp do tej instancji systemu Kali Linux. Po utworzeniu będziesz mógł zalogować się do konta w chmurze AWS za pomocą swojego wygenerowanego wcześniej klucza prywatnego. Aby to zrobić, powinieneś z poziomu wiersza poleceń powłoki wykonać następującą komendę:

```
ssh -i klucz_prywatny.pem ec2-user@ip_serwera_DNS_Amazon
```

Pojawienie się ekranu pokazanego na następnej stronie będzie świadczyło o tym, że pomyślnie zainstalowałeś system Kali Linux w chmurze AWS:

Konfigurowanie i dostosowywanie systemu Kali Linux

Kali Linux to system wspomagający przeprowadzanie testów penetracyjnych. Jednak aby taki system był efektywny, pentester nie może się czuć w żaden sposób związany czy ograniczony przez domyślnie zainstalowane narzędzia czy domyślną konfigurację interfejsu użytkownika i pulpitu. Dostosowując system Kali Linux do własnych potrzeb i wymagań, pentester może polepszyć poziom zabezpieczenia danych na temat środowiska celu, zbieranych podczas testów penetracyjnych, a także znacząco ułatwić ich przeprowadzanie.

Najczęściej podczas dostosowywania systemu Kali Linux wykonywane są następujące operacje:

- Resetowanie i zmiana hasła użytkownika *root*.
- Dodawanie innych kont użytkowników, którzy nie posiadają uprawnień użytkownika *root*.
- Optymalizacja i przyspieszanie działania systemu Kali Linux.
- Udostępnianie i współużytkowanie wybranych folderów z systemem Windows.
- Tworzenie zaszyfrowanych folderów.

Zmiana hasła użytkownika root

Aby zmienić hasło użytkownika *root*, powinieneś wykonać następujące polecenie:

```
passwd root
```

System poprosi o wpisanie nowego hasła, tak jak to zostało pokazane na rysunku poniżej:

```
root@kali:~# passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Dodawanie zwykłego konta użytkownika

Bardzo wiele narzędzi dostępnych w systemie Kali Linux musi być uruchamianych na prawach użytkownika *root*. Ciągła praca z uprawnieniami użytkownika *root* niesie jednak ze sobą poważne zagrożenia, na przykład prosta pomyłka czy błąd podczas wpisywania nazwy polecenia może skutkować awarią danej aplikacji czy nawet uszkodzeniem testowanego systemu. W niektórych przypadkach preferowanym rozwiązaniem jest wcześniejsze sprawdzenie działania polecenia na prawach zwykłego użytkownika. Warto zauważyć, że niektóre aplikacje wręcz wymuszają uruchamianie z poziomu użytkownika o niższych uprawnieniach niż administrator systemu.

Aby utworzyć konto zwykłego użytkownika, możesz po prostu z poziomu konsoli użyć polecenia `adduser`, a następnie postępować według poleceń pojawiających się na ekranie, tak jak to zostało pokazane na rysunku poniżej:

```

root@Kali:~# adduser noroot
Adding user `noroot' ...
Adding new group `noroot' (1000) ...
Adding new user `noroot' (1000) with group `noroot' ...
Creating home directory `/home/noroot' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for noroot
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []: 007
  Home Phone []: 007
  Other []:
Is the information correct? [Y/n] y

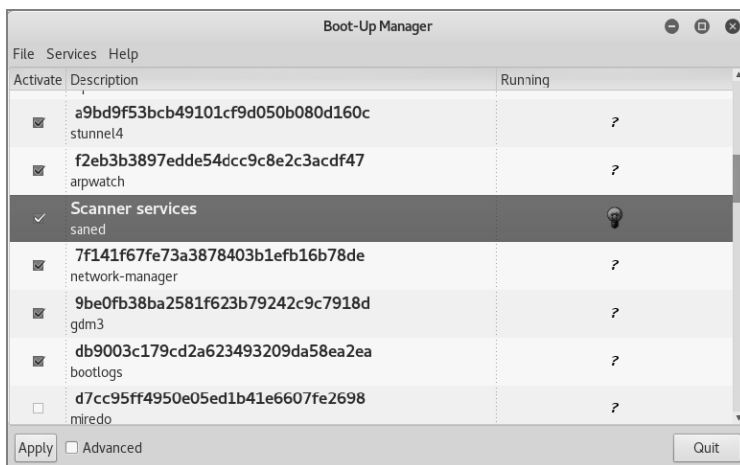
```

Przyspieszanie działania systemu Kali Linux

Istnieje co najmniej kilka narzędzi, których możesz używać do optymalizacji i przyspieszania działania systemu Kali Linux:

- Jeżeli korzystasz z systemu Kali Linux działającego w maszynie wirtualnej, powinieneś dodatkowo zainstalować w niej pakiet Guest Additions (VirtualBox) lub VMware Tools (VMware). Pamiętaj, aby przed instalacją dodatkowo wykonać polecenie `apt-get update`.
- Kiedy stworzysz nową maszynę wirtualną, powinieneś zamiast dysku alokowanego dynamicznie wybrać dysk twardy o stałej wielkości. Takie rozwiązanie powoduje szybsze zapisywanie plików na dysku i zmniejsza ich fragmentację.
- Aplikacja `preload` identyfikuje najczęściej wykorzystywane przez użytkownika programy i przyspiesza działanie systemu poprzez wcześniejsze buforowanie odpowiednich plików wykonywalnych i zależności. Aby ją zainstalować, powinieneś wykonać polecenie `apt-get install preload`. Po zainstalowaniu i zrestartowaniu systemu aplikacja działa całkowicie automatycznie.
- `BleachBit` (`apt-get install bleachbit`) to aplikacja, która optymalizuje działanie systemu poprzez zwalnianie niepotrzebnie zajętych zasobów dyskowych, usuwanie ciasteczek, czyszczenie historii przeglądania sieci internet, usuwanie plików tymczasowych, kasowanie logów oraz innych niepotrzebnych plików. Oprócz tego aplikacja posiada kilka dodatkowych, przydatnych funkcji, takich jak usuwanie plików metodą nadpisywania — co uniemożliwia ich odzyskanie — czy nadpisywanie niealokowanej przestrzeni dyskowej — co powoduje definitywne usunięcie resztek danych pozostawionych przez skasowane wcześniej pliki.

- Domyślnie Kali Linux nie wyświetla wszystkich aplikacji uruchamianych podczas ładowania systemu. Każda aplikacja ładowana i uruchamiana podczas bootowania systemu wydłuża ten proces, może zabierać cenne zasoby i wpływać na zmniejszenie wydajności działania systemu. Aby przeglądać listę takich programów i zablokować aplikacje niepotrzebnie ładowane podczas uruchamiania systemu, powinieneś zainstalować program *Boot Up Manager (BUM)*, pokazany na rysunku poniżej. Aby to zrobić, powinieneś wykonać polecenie `apt-get install bum`.



- Zainstaluj program `gnome-do` (`apt-get install gnome-do`), który pozwala na uruchamianie aplikacji bezpośrednio z poziomu klawiatury. Aby ją skonfigurować, musisz ją uruchomić z menu *Applications/Accessories* (aplikacje/akcesoria), a następnie wybrać menu *Preferences* (właściwości), aktywować opcję *Quiet Launch* (ciche uruchamianie), wybrać uruchamiającą kombinację klawiszy (na przykład `Ctrl+Shift`) i wpisać wiersz polecenia, który powinien zostać wykonany po naciśnięciu tej kombinacji.
- Pamiętaj, że do wykonywania bardziej złożonych operacji możesz używać nie tylko skrótów klawiszowych, ale też odpowiednio przygotowanych skryptów powłoki.

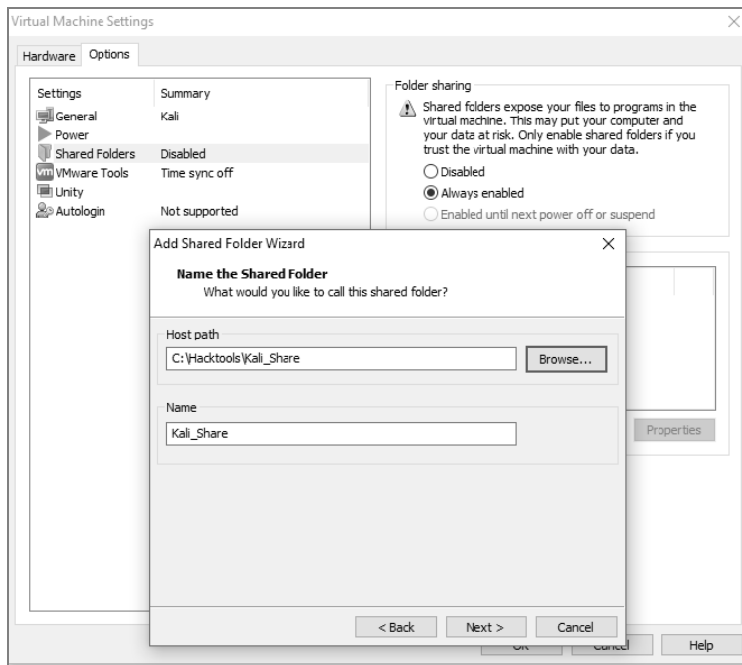
Udostępnianie i współużytkowanie folderów z systemem operacyjnym hosta

System Kali Linux pozwala na udostępnianie plików i danych aplikacjom działającym pod kontrolą innych systemów operacyjnych, a zwłaszcza systemu Microsoft Windows. Najbardziej efektywnym sposobem udostępniania danych jest utworzenie dedykowanego foldera, który będzie dostępny zarówno z poziomu systemu operacyjnego hosta, jak i z poziomu systemu Kali Linux działającego w maszynie wirtualnej.

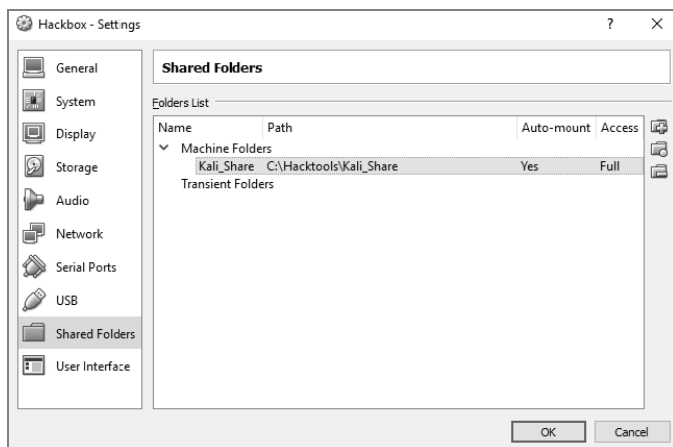
Kiedy w takim współdzielonym folderze zostaną umieszczone jakieś pliki, stają się one natychmiast dostępne dla wszystkich innych systemów korzystających z tego foldera.

Aby utworzyć taki współdzielony folder, powinieneś wykonać następujące polecenia:

1. Utwórz folder w systemie plików hosta. W naszym przypadku utworzymy folder o nazwie *Kali_Share*.
2. Kliknij nowo utworzony folder prawym przyciskiem myszy i wybierz polecenie *Sharing* (udostępnianie).
3. Włącz udostępnianie foldera i upewnij się, że dostęp będą mieli wszyscy użytkownicy (*Everyone*), a poziom uprawnień ustawiony jest na prawo odczytu i zapisu (*Read/Write*).
4. Jeżeli jeszcze tego nie zrobiłeś, zainstaluj w maszynie wirtualnej dodatki Guest Additions (dla VirtualBox) lub VMware Tools (dla VMware); więcej szczegółowych informacji na ten temat znajdziesz w dodatku, w sekcji „Instalowanie systemu Kali Linux”.
5. Po zakończeniu instalacji przejdź do menu głównego VMware, wybierz polecenie *Manage/Virtual Machine Settings* (zarządzaj/ustawienia maszyny wirtualnej), przejdź na kartę *Options* (opcje), kliknij kategorię *Shared Folders* (foldery współdzielone) i zaznacz opcję *Always Enabled* (zawsze włączone). Wybierz katalog w systemie plików hosta, który ma być współdzielony, tak jak to zostało pokazane na rysunku poniżej:



6. Jeżeli używasz programu Oracle VirtualBox, zaznacz maszynę wirtualną, wybierz polecenie *Settings* (ustawienia), zaznacz opcję *Shared Folders* (foldery współdzielone) i wybierz katalog w systemie plików hosta, który ma być współdzielony, tak jak to zostało pokazane na rysunku poniżej:



W starszych wersjach programu VMWare Player menu wygląda nieco inaczej.

- Przejdź do systemu Kali Linux w maszynie wirtualnej i uruchom przeglądarkę plików. Wybrany współdzielony katalog powinien być widoczny w folderze *mnt* (lub w jego subfolderze *hgfs*).
- Przeciągnij ikonę współdzielonego foldera na pulpit systemu Kali Linux, aby utworzyć na nim skrót do tego foldera.
- Od tej chwili wszystkie pliki, jakie umieścisz we współdzielonym folderze, będą dostępne zarówno dla systemu Kali Linux, jak i z poziomu systemu operacyjnego hosta maszyny wirtualnej.

Pamiętaj, że jeżeli we współdzielonym folderze chcesz przechowywać wrażliwe dane pozyskane w trakcie przeprowadzania testu penetracyjnego, to zawartość takiego foldera powinna być zaszyfrowana. Dzięki temu wrażliwe dane klienta będą lepiej zabezpieczone i chronione przed przypadkowym wyciekiem, gdyby zawierający je dysk został zagubiony lub skradziony.

Dostosowywanie systemu Kali Linux do własnych potrzeb przy użyciu skryptów powłoki bash

W systemie Linux dostępnych jest wiele różnych rodzajów powłok, za pomocą których możemy korzystać z systemu z poziomu wiersza poleceń konsoli. Najczęściej możemy spotkać powłoki *sh*, *bash*, *csch*, *tcsh* oraz *ksh*.

W zależności od celu przeprowadzanego testu penetracyjnego do dostosowania systemu Kali Linux do własnych potrzeb możemy użyć jednego z następujących skryptów powłoki *bash*:

- <https://github.com/leebaird/discover/blob/master/update.sh>
- <https://code.google.com/archive/p/lazykali/downloads>

Budowanie środowiska testowego

Każdy pentester powinien utworzyć swoje własne środowisko testowe, w którym będzie można testować różne podatności i luki w zabezpieczeniach przed dokonaniem próby ich wykorzystania podczas rzeczywistego testu penetracyjnego w środowisku klienta.

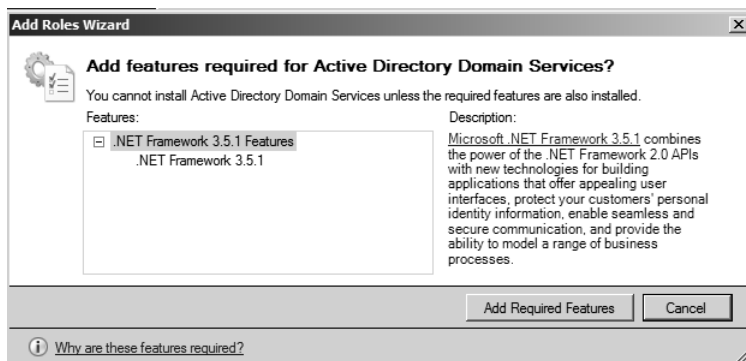
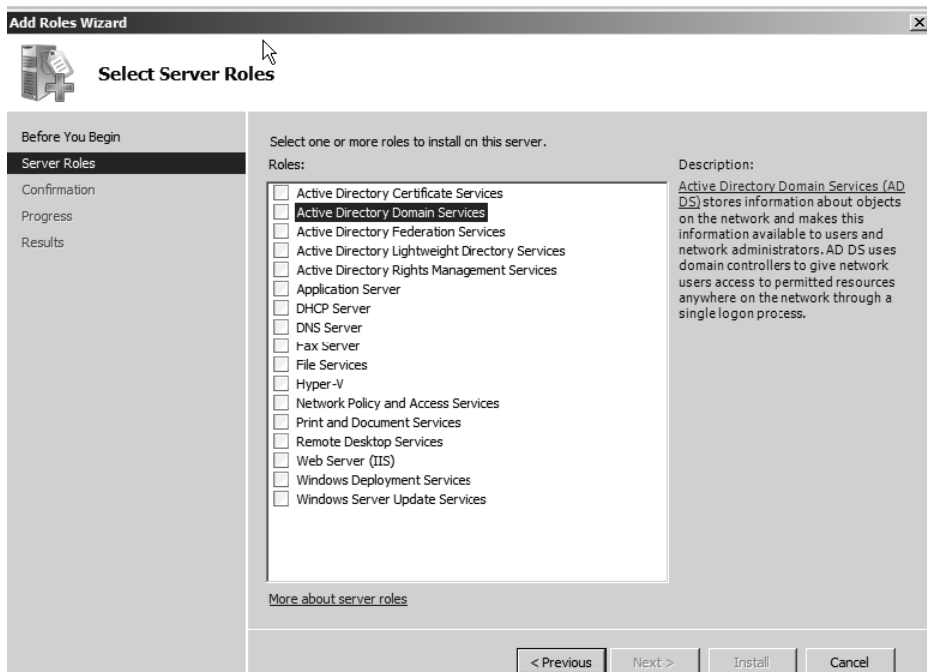
Konfigurowanie sieci wirtualnej z usługą Active Directory

Zdecydowana większość dużych firm i organizacji do zarządzania kontami użytkowników, zasobami plikowymi, czy drukarkami sieciowymi wykorzystuje w swoich środowiskach usługę Microsoft Active Directory. Potencjalni napastnicy obecnie nie są już zainteresowani wyłącznie możliwością wykonywania poleceń na skompromitowanym serwerze, ponieważ znacznie bardziej atrakcyjnym dla nich celem może być przejęcie kontroli nad kontrolerem domeny, który spełnia rolę swego rodzaju podwójnej helisy DNA dla całej firmy. W dalszej części książki będziemy opisywać niektóre zaawansowane ataki na usługę Active Directory oraz serwery DNS danego środowiska. W tej sekcji pokażemy, jak zainstalować usługę Active Directory na serwerze Windows 2008 R2.

Najpierw musimy w naszym środowisku testowym zainstalować serwer Windows 2008 R2. Aby to zrobić, musimy wykonać takie same kroki, jakie wykonywaliśmy podczas instalowania systemu Kali Linux.

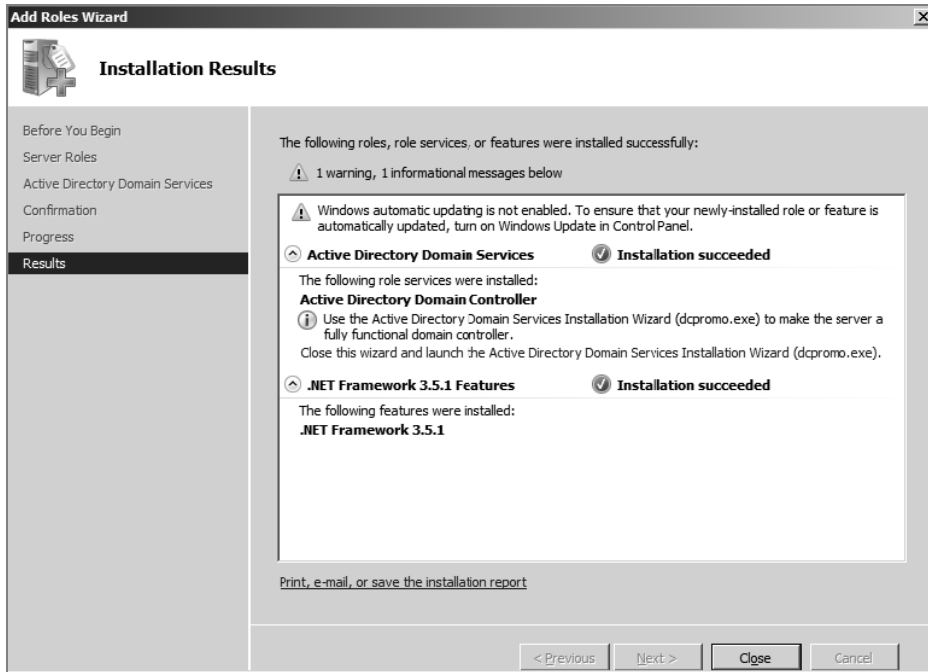
Następnie na ekranie pojawi się kreator, który krok po kroku przeprowadzi nas przez proces instalowania usług Active Directory Domain Services. Załóżmy zatem, że serwer Windows 2008 R2 mamy już zainstalowany i na ekranie pojawił się kreator instalowania usług Active Directory. Kliknij polecenie *Server Manager* (zarządzanie serwerem), przejdź do opcji *Roles* (role), a następnie wybierz polecenie *Add Roles* (dodaj role). Na ekranie pojawi się pierwszy krok kreatora, *Before you begin* (zanim rozpocznieš). Naciśnij przycisk *Next* (dalej), a na ekranie zobaczysz okno przedstawione na pierwszym rysunku na następnej stronie.

Zaznacz opcję *Active Directory Domain Services* (domenowe usługi Active Directory). Istnieje duże prawdopodobieństwo, że po wybraniu tej opcji na ekranie pojawi się komunikat z prośbą o zainstalowanie pakietu Microsoft .NET Framework 3.5.1, który jest niezbędny do poprawnego działania usług Active Directory. Naciśnij przycisk *Add Required Features* (dodaj wymagane komponenty), tak jak to zostało pokazane na drugim rysunku na następnej stronie.



Aby rozpocząć instalowanie, naciśnij przycisk *Install* (instaluj). Wybrane komponenty zostaną zainstalowane i na ekranie pojawi się okno z informacją o pomyślnym zakończeniu procesu instalacji usług Active Directory, tak jak to zostało pokazane na pierwszym rysunku na następnej stronie.

Po zakończeniu instalacji usług Active Directory musimy się upewnić, że wszystko działa poprawnie. Aby to zrobić, musimy kliknąć przycisk uruchamiający kreatora usług *Active Directory Domain Services* i utworzyć nowy las AD (ang. *Active Directory Forest*). W naszym przypadku będziemy tworzyć nowy las, którego pełna nazwa domenowa FQDN (ang. *Fully Qualified Domain Name*) to *Secure.kali.com*. Nazwę domeny NetBIOS ustawiamy na *Secure*, następnie



ustawiamy poziom funkcjonalności lasu (ang. *forest functional level*) na Windows 2003 lub Windows 2008 R2, co spowoduje uruchomienie konfiguracji serwera DNS (ang. *Domain Name Server*). Ponieważ w naszym przypadku mamy zupełnie nową instancję domeny, musimy dopiero zainstalować serwer DNS i następnie przy użyciu kreatora ustawić nową domenę na *secure.kali.com*, tak jak to zostało pokazane na rysunku poniżej:



Instalowanie zdefiniowanych celów

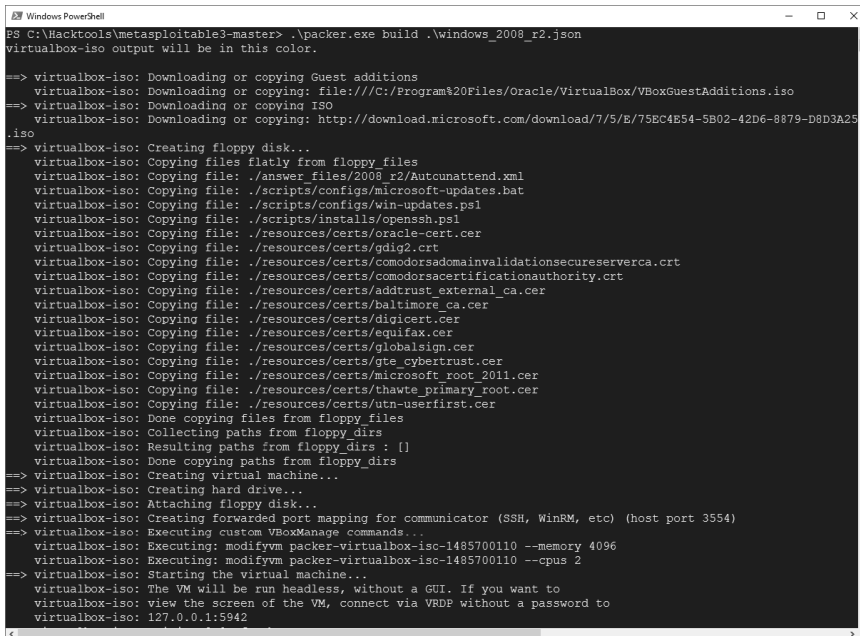
To praktyka sprawia, że człowiek staje się doskonały. Aby zatem wprawiać się w sztuce przełamania zabezpieczeń, powinieneś użyć oprogramowania, które posiada szereg dobrze znanych i opisanych podatności i luk w zabezpieczeniach. W tej sekcji pokażemy, jak możesz zainstalować pakiet metasploitable3 (platforma Windows) oraz Mutillidae (aplikacja sieciowa PHP).

Metasploitable3

Metasploitable3 to maszyna wirtualna działająca pod kontrolą systemu Windows i celowo wyposażona w szereg podatności i luk w zabezpieczeniach, przeznaczona do wspomagania ćwiczeń w wykrywaniu i wykorzystywaniu luk w zabezpieczeniach przy użyciu pakietu Metasploit. Pakiet jest udostępniany na licencji opartej na licencji BSD. Maszynę wirtualną Metasploitable3 można pobrać ze strony <https://github.com/rapid7/metasploitable3>. W zależności od potrzeb możesz pobrać archiwum ZIP i rozpakować je do wybranego foldera w swoim systemie Windows (w naszym przypadku zazwyczaj umieszczamy pliki w folderze `C:\HackTools`) lub z poziomu konsoli użyć polecenia `git clone`.

Powinieneś również zainstalować wszystkie dostępne pomocnicze pakiety oprogramowania, takie jak Packer, Vagrant, VirtualBox czy wtyczka Vagrant.

Na rysunku zamieszczonym poniżej przedstawiamy proces ręcznej instalacji pakietu Metasploitable3 na platformie Windows 10.



```

PS C:\HackTools\metasploitable3-master> .\packer.exe build .\windows_2008_r2.json
virtualbox-iso output will be in this color.

=> virtualbox-iso: Downloading or copying Guest additions
   virtualbox-iso: Downloading or copying: file:///C:/Program%20Files/Oracle/VirtualBox/VBoxGuestAdditions.iso
=> virtualbox-iso: Downloading or copying ISO
   virtualbox-iso: Downloading or copying: http://download.microsoft.com/download/7/5/E/75E4E54-5B02-42D6-8879-D8D3A25
   iso
=> virtualbox-iso: Creating floppy disk...
   virtualbox-iso: Copying files flatly from floppy files
   virtualbox-iso: Copying file: ./answer_files/2008_r2/Autounattend.xml
   virtualbox-iso: Copying file: ./scripts/configs/microsoft-updates.bat
   virtualbox-iso: Copying file: ./scripts/configs/win-updates.ps1
   virtualbox-iso: Copying file: ./scripts/installs/openssh.ps1
   virtualbox-iso: Copying file: ./resources/certs/oracle-cert.cer
   virtualbox-iso: Copying file: ./resources/certs/gdig2.cer
   virtualbox-iso: Copying file: ./resources/certs/comodorsadomainvalidationsecureserverca.crt
   virtualbox-iso: Copying file: ./resources/certs/comodorsacertificationauthority.crt
   virtualbox-iso: Copying file: ./resources/certs/addtrust_external_ca.cer
   virtualbox-iso: Copying file: ./resources/certs/baltimore ca.cer
   virtualbox-iso: Copying file: ./resources/certs/digicert.cer
   virtualbox-iso: Copying file: ./resources/certs/equifax.cer
   virtualbox-iso: Copying file: ./resources/certs/globalsign.cer
   virtualbox-iso: Copying file: ./resources/certs/gte_cybertrust.cer
   virtualbox-iso: Copying file: ./resources/certs/microsoft root 2011.cer
   virtualbox-iso: Copying file: ./resources/certs/thawte_primary_root.cer
   virtualbox-iso: Copying file: ./resources/certs/utn-userfirst.cer
   virtualbox-iso: Done copying files from floppy files
   virtualbox-iso: Collecting paths from floppy dirs
   virtualbox-iso: Resulting paths from floppy dirs : []
   virtualbox-iso: Done copying paths from floppy_dirs
=> virtualbox-iso: Creating virtual machine...
=> virtualbox-iso: Creating hard drive...
=> virtualbox-iso: Attaching floppy disk...
=> virtualbox-iso: Creating forwarded port mapping for communicator (SSH, WinRM, etc) (host port 3554)
=> virtualbox-iso: Executing custom VBoxManage commands...
   virtualbox-iso: Executing: modifyvm packer-virtualbox-isc-1485700110 --memory 4096
   virtualbox-iso: Executing: modifyvm packer-virtualbox-isc-1485700110 --cpus 2
=> virtualbox-iso: Starting the virtual machine...
   virtualbox-iso: The VM will be run headless, without a GUI. If you want to
   virtualbox-iso: view the screen of the VM, connect via VRDP without a password to
   virtualbox-iso: 127.0.0.1:5942
  
```


Po zakończeniu pobierania pliku ISO zawierającego obraz maszyny wirtualnej w oknie konsoli zostanie wyświetlony komunikat przedstawiony na rysunku poniżej:

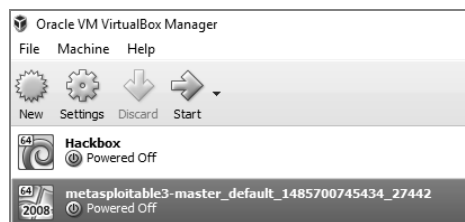
```

Select Windows PowerShell
virtualbox-iso: C:\Users\vagrant>cmd /c certutil -addstore -f "Root" A:\microsoft_root_2011.cer
virtualbox-iso: Root
virtualbox-iso: Signature matches Public Key
virtualbox-iso: Certificate "CN=Microsoft Root Certificate Authority 2011, O=Microsoft Corporation, L=Redmond, S=Washi
virtualbox-iso: CertUtil: -addstore command completed successfully.
virtualbox-iso:
virtualbox-iso: C:\Users\vagrant>cmd /c certutil -addstore -f "Root" A:\thawte_primary_root.cer
virtualbox-iso: Root
virtualbox-iso: Signature matches Public Key
virtualbox-iso: Certificate "CN=thawte Primary Root CA - G3, OU="(c) 2008 thawte, Inc. - For authorized use only", OU=
e.
virtualbox-iso: CertUtil: -addstore command completed successfully.
virtualbox-iso:
virtualbox-iso: C:\Users\vagrant>cmd /c certutil -addstore -f "Root" A:\utn-userfirst.cer
virtualbox-iso: Root
virtualbox-iso: Signature matches Public Key
virtualbox-iso: Certificate "CN=UTW-USERFirst-Object, OU=http://www.usertrust.com, O=The USERTRUST Network, L=Salt Lak
virtualbox-iso: CertUtil: -addstore command completed successfully.
> virtualbox-iso: Gracefully halting virtual machine...
virtualbox-iso: Removing floppy drive...
> virtualbox-iso: Preparing to export machine...
virtualbox-iso: Deleting forwarded port mapping for the communicator (SSH, WinRM, etc) (host port 3554)
> virtualbox-iso: Exporting virtual machine...
virtualbox-iso: Executing: export packer-virtualbox-iso-1485700110 --output output-virtualbox-iso\packer-virtualbox-iso
> virtualbox-iso: Unregistering and deleting virtual machine...
virtualbox-iso: Running post-processor: vagrant
> virtualbox-iso (vagrant): Creating Vagrant box for 'virtualbox' provider
virtualbox-iso (vagrant): Copying from artifact: output-virtualbox-iso\packer-virtualbox-iso-1485700110-disk1.vmdk
virtualbox-iso (vagrant): Copying from artifact: output-virtualbox-iso\packer-virtualbox-iso-1485700110.ovf
virtualbox-iso (vagrant): Renaming the OVF to box.ovf...
virtualbox-iso (vagrant): Using custom Vagrantfile: vagrantfile-windows_2008_r2.template
virtualbox-iso (vagrant): Compressing: Vagrantfile
virtualbox-iso (vagrant): Compressing: box.ovf
virtualbox-iso (vagrant): Compressing: metadata.json
virtualbox-iso (vagrant): Compressing: packer-virtualbox-iso-1485700110-disk1.vmdk
ild 'virtualbox-iso' finished.

> Builds finished. The artifacts of successful builds are:
> virtualbox-iso: 'virtualbox' provider box: windows_2008_r2_virtualbox.box
C:\Hacktools\metasploitable3-master>

```

Po zakończeniu pobierania pliku maszyny wirtualnej musisz jeszcze tylko z poziomu tej samej konsoli PowerShell uruchomić polecenie `vagrant up`, które powinno dodać nową maszynę wirtualną do konsoli VirtualBox, tak jak to zostało pokazane na rysunku poniżej:

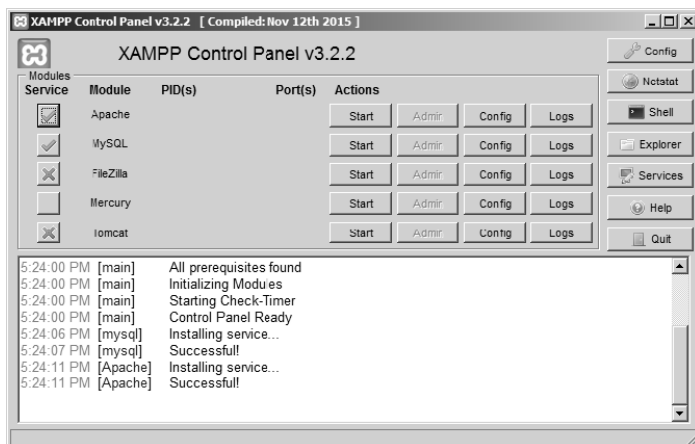


Mutillidae

Mutillidae to wyposażona w szereg podatności i luk w zabezpieczeniach aplikacja sieciowa typu open source, która jest przeznaczona do wspomagania ćwiczeń w wykrywaniu i wykorzystywaniu luk w zabezpieczeniach aplikacji sieciowych.

XAMPP to inna bezpłatna aplikacja typu open source, opracowana przez zespół programistów Apache Friends, którą możesz pobrać ze strony <https://www.apachefriends.org/download.html>.

Teraz zainstalujemy aplikację Mutillidae na naszym nowo zainstalowanym serwerze Microsoft Windows 2008 R2. Po zakończeniu pobierania pakietu XAMPP uruchamiamy program instalacyjny i postępujemy zgodnie z poleceniami kreatora. Po zakończeniu instalacji uruchamiamy XAMPP; na ekranie powinno się pojawić okno przedstawione na rysunku poniżej:



Pakiet Mutillidae można pobrać ze strony <https://sourceforge.net/projects/mutillidae/files/latest/download>.

Po zakończeniu pobierania rozpakuj archiwum ZIP i skopiuj pliki do katalogu `c:\Lokalizacja XAMPP\htdocs\<mutillidae>`.

Aby sprawdzić, czy aplikacja została zainstalowana pomyślnie, powinieneś uruchomić przeglądarkę sieciową i wpisać w niej adres przedstawiony na pierwszym rysunku na następnej stronie:

<http://localhost/mutillidae/>

Zarządzanie testami penetracyjnymi przy użyciu pakietu Faraday

Do najtrudniejszych aspektów przeprowadzania rozbudowanych testów penetracyjnych należą: konieczność przetestowania wszystkich istotnych elementów sieci i systemów środowiska celu, umiejętność zapamiętywania, które elementy systemu zostały już sprawdzone, oraz umiejętność określenia po zakończeniu testu, jakie operacje zostały przeprowadzone w trakcie jego trwania. W niektórych przypadkach klient może zażądać przeprowadzenia rozbudowanych



testów penetracyjnych, wymagających zatrudnienia wielu pentesterów operujących z różnych lokalizacji, a kierownictwo chce mieć możliwość zbiorczego monitorowania i koordynowania ich działań z jednego miejsca. Pakiet Faraday zapewnia pentesterom możliwość ujednoliconego widoku na testowane środowisko, przy założeniu, że mogą się oni ze sobą komunikować w danej sieci lokalnej lub sieci internet (w przypadku pentestów zewnętrznych).

Faraday to wieloużytkownikowe, zintegrowane środowisko IDE (ang. *Integrated Development Environment*) wspomagające przeprowadzanie testów penetracyjnych, które pozwala pentesterom na współużytkowanie, dystrybucję, indeksowanie i analizowanie wszelkich danych wygenerowanych lub pozyskanych podczas przeprowadzania testów penetracyjnych oraz przeprowadzanie audytów bezpieczeństwa i tworzenie raportów.

Środowisko Faraday IDE zostało zaprojektowane i napisane w języku Python przez programistów firmy Infobyte. Aplikację można pobrać ze strony <https://github.com/infobyte/faraday/wiki> lub bezpośrednio za pomocą polecenia `git clone`, tak jak to zostało pokazane na pierwszym rysunku na następnej stronie.

Po zakończeniu klonowania powinieneś wykonać polecenie `./install.sh`, które spowoduje również zainstalowanie wszystkich niezbędnych zależności. Nie zapomnij również uruchomić usługi CouchDB, ponieważ platforma Faraday wykorzystuje bazę danych CouchDB do przechowywania swoich danych. Na koniec powinieneś wykonać polecenie `faraday-server.py`, które spowoduje uruchomienie serwera Faraday. Aby uruchomić klienta środowiska, powinieneś z poziomu konsoli wykonać polecenie `faraday.py`, tak jak to zostało pokazane na drugim rysunku na następnej stronie.

```

root@kali: /faraday-dev
File Edit View Search Terminal Help

root@kali:~# git clone https://github.com/infobyte/faraday.git faraday-dev
Cloning into 'faraday-dev'...
remote: Counting objects: 25366, done.
remote: Total 25366 (delta 0), reused 0 (delta 0), pack-reused 25366
Receiving objects: 100% (25366/25366), 7.69 MiB | 215.00 KiB/s, done.
Resolving deltas: 100% (15579/15579), done.
Checking connectivity... done.
root@kali:~# cd faraday-dev/
root@kali:~/faraday-dev# ls
apis                faraday.py          persistence         tests_web
AUTHORS             faraday-server.py  plugins            updates
backup              faraday-terminal.zsh README.md          utils
bin                 gui                 RELEASE.md         VERSION
config              helpers             requirements_server.txt
controllers         __init__.py        requirements.txt   zsh
data                install.sh          scripts
doc                 managers           server
exporters           model              test_cases
root@kali:~/faraday-dev# ./install.sh
[+] Install Kali GNU/Linux Rolling x86 64
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main Sources [11.1 MB]
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/contrib Sources [67.7 kB]

```

```

root@kali: /faraday-dev
File Edit View Search Terminal Help

root@kali:~/faraday-dev# ./faraday.py
[+] Open Source Penetration Test IDE
Where pwnage goes multiplayer

2017-01-30 15:47:32,942 - faraday.launcher - INFO - Starting Faraday IDE., line
2017-01-30 15:47:32,942 - faraday.launcher - INFO - Dependencies met.
2017-01-30 15:47:32,943 - faraday.launcher - INFO - Checking configuration.
2017-01-30 15:47:32,943 - faraday.launcher - INFO - Setting up plugins.
2017-01-30 15:47:32,943 - faraday.launcher - INFO - Removing old plugins folder.
2017-01-30 15:47:32,952 - faraday.launcher - INFO - Setting up ZSH integration.
2017-01-30 15:47:32,953 - faraday.launcher - INFO - Setting up user configuratio
n.
2017-01-30 15:47:32,953 - faraday.launcher - INFO - Copying default configuratio
n from project.
2017-01-30 15:47:32,953 - faraday.launcher - INFO - Setting up icons for GTK int
erface.

```

Uruchomienie klienta powinno spowodować otwarcie konsoli środowiska Faraday, tak jak to zostało pokazane na pierwszym rysunku na następnej stronie.

Jedną z wielkich zalet środowiska Faraday jest to, że po kliknięciu odpowiedniej opcji możemy od razu zobaczyć wizualizację wszystkich skanów i innych operacji wykonywanych przez Ciebie lub innych pentesterów pracujących nad tym projektem, tak jak to zostało pokazane na drugim rysunku na następnej stronie.

Choć pełna wersja pakietu Faraday jest produktem komercyjnym, to jednak nadal dostępna jest również bezpłatna wersja tego środowiska, która ma nieco ograniczone możliwości, ale nadal pozwala na wyświetlanie w jednym miejscu wszystkich informacji o wynikach przeprowadzanego testu penetracyjnego.

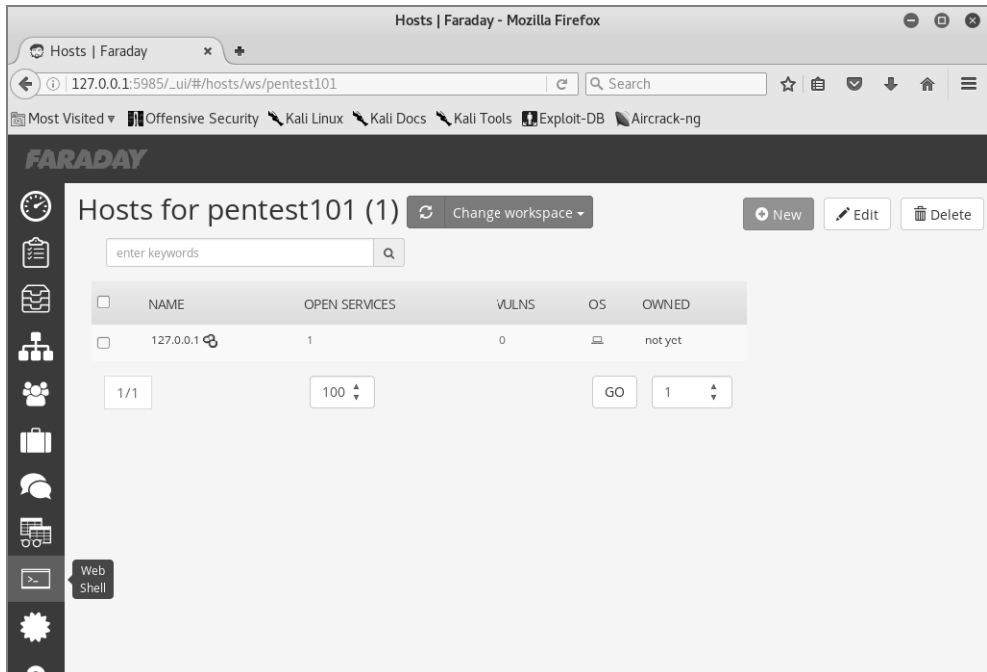
```

Faraday 2.2.0
1
>>> WELCOME TO FARADAY
[+] Current Workspace: untitled
[+] API: OK
[faraday](untitled) kali# nmap -oX /root/.faraday/data/pentest101_Nmap_output-6.00999157877.xml localhost 2>&1 | tee -a tmp.qrK54rnyJcC85NnPgAqrQfvgMLagp

Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-01-30 15:49 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed ports
PORT      STATE SERVICE
9876/tcp  open  sd

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
[faraday](pentest101) kali# msfconsole
[*] Starting the Metasploit Framework console...

Welcome to Faraday!
[ERROR ]- 2017-01-30 15:47:36,167 - faraday.GTK - Workspace untitled wasn't found
[INFO ]- 2017-01-30 15:48:12,869 - faraday - Creating workspace 'pentest101'
[INFO ]- 2017-01-30 15:49:20,151 - faraday.ModelController - Plugin Started: Nmap
[INFO ]- 2017-01-30 15:49:20,593 - faraday.ModelController - Plugin Ended: Nmap
    
```



Podsumowanie

W tym rozdziale omawialiśmy różne metodologie testów penetracyjnych ukierunkowanych na osiągnięcie celu, które pozwalają firmom i organizacjom na sprawdzanie i szacowanie odporności ich środowiska na pełnowymiarowe ataki cybernetyczne. Pokazywaliśmy, jak pentesterzy mogą używać systemu Kali Linux do przeprowadzania testów bezpieczeństwa sieci i systemów w środowisku celu. Szczegółowo omówiliśmy proces instalowania systemu Kali Linux na różnych platformach wirtualizacyjnych i pokazywaliśmy, jak przy użyciu aplikacji Docker można szybko uruchomić system Linux na platformie Windows.

Zbudowaliśmy również nasze własne środowisko testowe, zainstalowaliśmy usługi Active Directory Domain Services oraz tak skonfigurowaliśmy dwie różne maszyny wirtualne, aby działały w tej samej sieci. Dowiedziałeś się również, jak można dostosować system Kali Linux do własnych, indywidualnych wymagań oraz jak zwiększyć bezpieczeństwo używanych narzędzi i pozyskiwanych przez nie danych. Pracowaliśmy nad osiągnięciem celu, jakim jest przygotowanie narzędzi wspierających nasz proces, a nie odwrotnie!

W kolejnym rozdziale dowiesz się, jak efektywnie używać narzędzi *Open Source Intelligence* (OSINT) do identyfikacji podatnej na atak powierzchni środowiska oraz jak tworzyć niestandardowe listy nazw użytkowników i haseł ułatwiających przeprowadzenie bardziej skoncentrowanych ataków oraz wykorzystywanie znalezionych podatności i luk w zabezpieczeniach.

Skorowidz

A

- Active Directory
 - instalowanie usługi, 49
 - podnoszenie uprawnień, 405
 - rozpoznawanie serwerów, 124
- administratorzy
 - delegowani, 390
 - domeny, 390
 - lokalni, 390
 - przedsiębiorstwa, 390
 - schematu, 391
- adres
 - e-mail, 69
 - IP, 85, 97
 - MAC, 204
 - URL, 181
- agent persystencji, 416
 - Netcat, 417
 - tworzenie, 424
- agenty, 397
 - PowerShell, 395
- algorytm NT LanMan, 371
- algorytmy szyfrowania, 261
 - cipher suites, 261
- analiza zagrożeń, 76
- aplikacja
 - alive6, 108
 - Arachnid, 240
 - Bettercap, 401
 - bleachBit, 45
 - CaseFile, 66
 - CENSYS.IO, 104
 - CeWL, 79
 - chntpw, 160
 - commix, 241
 - covert_send6, 99
 - denial6, 99
 - detect-new-ip6, 99, 108
 - detect-sniffer6, 99
 - dnmap, 108
 - dnsdict6, 99
 - dnsenum, 97
 - dnsmap, 97
 - dnsrecon, 97
 - dnsrevenue6, 99
 - dnstracer, 97
 - dnswalk, 97
 - Docker, 39
 - exploit6, 99
 - fake_dhcps6, 99
 - Faraday, 54
 - fierce, 97
 - fping, 118
 - fragroute, 105
 - GoLismero, 240
 - hping3, 102
 - ike-scan, 272
 - Inception, 164
 - intrace, 102
 - lbd, 104
 - Metasploitable3, 52
 - miranda.py, 104
 - Mutillidae, 52, 53
 - netcat, 110, 417
 - Nikto, 240
 - nmap, 86, 104, 122, 135, 270
 - PowerShell Empire, 395
 - preload, 45
 - PsExec, 381
 - Reaver, 210
 - samdump2, 160
 - Shellter, 312
 - Shodan, 104
 - Skipfish, 240
 - ssllaudit, 263
 - ssldump, 263
 - sslsbiff, 263
 - sslsan, 262, 263
 - sslsplit, 263
 - sslstrip, 263, 265
 - sslyze, 263, 264
 - Testssl, 259
 - theHarvester, 69
 - tlssled, 263
 - trace6, 102
 - traceroute, 101
 - Twofi, 80
 - Vega, 240
 - Veil-Evasion, 187
 - VirtualBox, 34
 - VMware Workstation Player, 32
 - w3af, 240
 - Wapiti, 240
 - Webscarab, 240
 - Webshag, 240
 - Websploit, 240
 - Windows Credential Editor, 380
 - Windows Defender, 314
 - Windows Task Scheduler, 421
 - WMIC, 381
- aplikacje
 - biała lista, 320
 - internetowe, 221
 - mobilne, 146
 - sieciowe, 139
 - wspomagające rozpoznanie, 94
- Armitage, 346
- ARP, 117
- ataki
 - cybernetyczne
 - faza dostawy, 27
 - faza penetracji, 27
 - faza post-exploit, 28
 - faza rozpoznania, 27
 - na aplikacje internetowe, 222
 - analizowanie przebiegu komunikacji, 232
 - planowanie, 224
 - poświadczenia logowania, 241

ataki

- na aplikacje internetowe, 222
 - rozpoznanie witryny internetowej, 225
 - tworzenie sygnatur aplikacji, 228
 - utrzymywanie dostępu, 245
 - wstrzykiwanie poleceń, 241
- na pamięć systemową, 164
- na połączenia SSL, 257
- na protokół Kerberos, 412
- na sieci bezprzewodowe, 195–219
- na sieci z szyfrowaniem WPA, 213
- na strony internetowe, 171
- na wirtualne sieci prywatne, 269
- na witryny internetowe, 174–177
- na zdalny dostęp, 249
- phishingowe, 192
- po stronie klienta, 275
- pozwalające
 - na podnoszenie uprawnień, 400
 - na zbieranie poświadczeń, 400
- socjotechniczne, 157
- typu
 - BEAST, 258
 - BREACH, 258
 - brute-force, 207, 239
 - brute-force na poświadczenia logowania, 241
 - CRIME, 258
 - DoS, 211, 268
 - Golden Ticket, 412
 - FREAK, 258
 - Heartbleed, 258
 - Logjam, 259
 - man-in-the-middle, 263, 265, 401
 - NetBIOS null session, 123
 - POODLE, 259
 - SMB relay, 405
 - SQL Injection, 289
 - spear phishing, 184
 - tabnabbing, 176
 - web jacking, 172
 - XSS, 25, 285
- ukrywanie śladów, 437
- w procesie renegotjacji sesji, 258
- z dostępem
 - do konsoli systemu, 159
 - fizycznym, 159
- z modyfikacją zawartości kart przeglądarki, 171
- z wykorzystaniem
 - apletów Java, 171
 - aplikacji HTA, 172, 179
 - beprzewodowych punktów dostępowych, 173

- exploitów przeglądarki sieciowej, 171
- falszywych wiadomości SMS, 173
- komputera, 157
- pakietu Metasploit, 342
- powłoki Powershell, 173
- telefonu, 158
- trybu pełnoekranowego, 172
- urządzeń Arduino, 172
- urządzeń
 - mikroprocesorowych, 168
- z wyludzeniem poświadczeń logowania, 171
- ze wstrzykiwaniem kodu, 178, 243, 289
- ze zbieraniem poświadczeń logowania, 174
- złożone, 172, 177
- atakowanie
 - kolejnych systemów, 345, 379
 - pojedynczych systemów, 342, 344
 - routerów sieci bezprzewodowych, 210
- awaria serwera, 357
- AWS, 41

B

- backdooring, 276
- backdoory, 416
- baiting, 158
- baza danych
 - konfiguracja, 337
 - SAM, 369
 - tworzenie, 337
- bazy
 - luk w zabezpieczeniach, 131
 - podatności, 131
- BeEF, The Browser Exploitation Framework, 289
- Bettercap, 401
- bezpieczeństwo
 - fizyczne, 155
 - systemu Windows, 330
- biała lista aplikacji, 320
- biały wywiad, 61
- automatyczne zbieranie informacji, 74
- defensywny, 75
- ofensywny, 62
- zbieranie informacji, 69
- biblioteki Metasploit
 - Framework-base, 335
 - Framework-core, 335

- REX, 335
- bramka VPN, 270
- Burp Proxy, 232

C

- CaseFile, 66
- cele ataku
 - drugorzędne, 152
 - główne, 152
 - trzeciorzędne, 152
- CeWL, 79
- chmura, 41
- cipher suites, 261
- commix, 241
- cyfrowy odcisk, 271

Ć

- ćwiczenia zespołów Red Team, 25

D

- Dark Web, 75
- DET, Data Exfiltration Toolkit, 435
- DHCP, 114
- Docker, 39
- dodawanie konta użytkownika, 44
- DoS, Denial of Service, 211
- dostęp
 - do kolejnych systemów, 379
 - do konsoli systemu, 163
 - do sieci
 - weryfikacja po uzyskaniu dostępu, 305
 - weryfikacja przed uzyskaniem dostępu, 303
 - do systemu, 166
 - poziomy, 390
 - stały do skompromitowanego systemu, 371

E

- eksfiltracja danych, 429
- wykorzystanie powłoki PowerShell, 437
- wykorzystanie protokołu DNS, 431
- wykorzystanie protokołu ICMP, 433
- wykorzystanie usług systemowych, 430

eksploracja

- skompromitowanego systemu
 - eskalacja pozioma, 379
 - narzędzia wspomagające, 372
 - narzędzie PowerShell Empire, 395
 - pakiet Veil-Pillage, 375
 - pobieranie wrażliwych danych, 368
 - rozpoznanie, 367
 - systemu lokalnego, 366
- email phishing, 158
- EMET, Enhanced Migration Experience Toolkit, 322
- enkapsulacja protokołu, 297
- enkodery, 337
- eskalacja pozioma, 379
 - z użyciem usług, 385
- exploit, 133, 307–311, 336, 343
 - Exploit-DB, 349
- exploity
 - dla systemu Windows, 353, 360
 - publiczne, 349
 - kompilowanie, 351
 - lokalizowanie, 349
 - używanie, 351
 - weryfikowanie, 349
 - tworzenie, 353

F

- Faraday
 - zarządzanie testami, 54
- faza
 - dostawy, 27
 - penetracji, 27
 - post-exploit, 28
 - rozpoznania, 27, 131
 - sterowania, kontroli i komunikacji, 415
- filtrowanie adresów
 - MAC, 204
 - URL, 317
- foldery
 - udostępnianie, 46
 - współużytkowanie, 46
- framework
 - MobSF, 146
 - recon-ng, 94
 - Veil, 375
 - Websploit, 212
- fuzzing, 354–357

G

- Ghost Phisher, 217
- Google Cache, 67
- Google Hacking Database, 71, 72
- Guest Additions, 45

H

- hasła
 - łamanie słownikowe, 78
 - przygotowywanie listy, 78
- hasło użytkownika root, 44
- hosty
 - aktywne, 108
 - identyfikacja, 115
 - identyfikacja systemu operacyjnego, 111
 - w wewnętrznych sieciach, 115
 - wykrywanie, 108, 115, 117
 - wykrywanie usług, 112

I

- IDE, Integrated Development Environment, 55
- identyfikacja
 - domyślnych kont użytkowników, 273
 - hostów, 115
 - infrastruktury sieciowej, 103
 - luk w zabezpieczeniach, 354
 - podatności, 354
 - systemów IDS/IPS, 105
 - systemu operacyjnego, 111
 - udziałów sieciowych, 123
- identyfikator
 - OUI, 204
 - SID, 124
 - SSID, 202
- ike-scan, 270, 272
- Inception, 164
 - moduły, 165
- incydenty bezpieczeństwa, 76
- informacje
 - o kampanii
 - Email, 192
 - Phishing, 192
 - SMTP, 192
 - Template, 192
 - o kontaktach użytkowników, 121
 - o lukach w zabezpieczeniach, 131
 - o podatnościach, 131
 - o sieci bezprzewodowej, 199

- o środowisku celu, 83
- o użytkownikach, 70
- o włamaniach, 76
- iniekcje DLL, 393
- instalowanie
 - aplikacji Docker, 39
 - pakietu Metasploitable3, 52
 - systemu
 - w chmurze, 41
 - w maszynie wirtualnej, 31, 34
 - usługi Active Directory, 49

J

- język
 - LUA, 137
 - PowerShell, 382
 - VBScript, 279

K

- Kali Linux, 28
 - aktualizowanie, 30
 - dostosowywanie, 43, 48
 - instalowanie, 30
 - na urządzeniu przenośnym, 30
 - w chmurze, 41
 - w maszynie wirtualnej, 31, 34
 - konfigurowanie, 44, 196
 - przyspieszanie działania, 45
 - uruchamianie, 30
- kampania phishingowa, 192
- Kerberos, 412
- Kismet, 200
- klucze
 - PSK, 206, 272
 - łamanie, 272
 - przechwytywanie, 272
 - publiczne/prywatne, 369
- kompilowanie exploitów, 351
- konfigurowanie
 - bazy danych, 337
 - pakietu
 - Phishing Frenzy, 192
 - SPARTA, 126
 - BeEF, 290
 - sieci wirtualnej, 49
 - systemu Kali Linux, 196
 - trwałych zadań, 421
- konto użytkownika, 44, 69, 121
 - identyfikacja, 273
 - mechanizm kontroli UAC, 323
- kontrola konta użytkownika, 323
- kopia strony internetowej, 231

L

lista
 hasel, 78
 słów, 78, 80
 luki w zabezpieczeniach, 131, 250

Ł

ładunki, 336
 łamanie
 hasel, 78
 kluczy PSK, 272
 łączenie skanów, 119

M

Maltego, 63
 Mantra OWASP, 237
 mapowanie
 sieci, 62, 91
 sieci poza zaporą sieciową, 104
 trasy do celu, 100
 witryny internetowe, 79
 maskowanie adresu URL, 181
 maszyna wirtualna
 Metasploitable3, 52
 VirtualBox, 34
 VMware Workstation Player, 32
 mechanizm
 NAC, 302
 zabezpieczający, 301
 pakiet EMET, 322
 Metasploit, 334
 Metasploitable3
 instalowanie pakietu, 52
 metodologia
 atakowania aplikacji
 internetowych, 222
 podnoszenia uprawnień, 390
 przeprowadzania ataków, 157
 przeprowadzania testów, 26
 model BYOD, 195
 modelowanie zagrożeń, 151
 moduły BeEF
 Confirm close, 298
 iFrame keylogger, 299
 Man-in-the-browser, 299
 NOP, 337
 Pop-under module, 298
 Veil-Pillage, 376
 modyfikowanie
 adresów IP, 85
 parametrów pakietów, 86
 Mutillidae, 53

N

NAC
 dostęp do sieci, 303
 omijanie zabezpieczeń, 302
 Post-Admission NAC, 305
 Pre-Admission NAC, 303
 wyjątki, 304
 narzędzia
 bezpieczeństwa, 237
 do eksploracji systemu, 367, 372
 narzędzie, *Patrz* aplikacja, polecenie
 NAT, Network Address Translation,
 303
 Netcat, 417
 Nikto, 140
 dostosowywanie skanera, 142
 nmap, 86, 104, 122, 135, 270
 moduły NSE, 261
 poszukiwanie bramek VPN, 270
 NSE, Nmap Scripting Engine, 135

O

odcisk cyfrowy, 271
 odwrócona powłoka, 342
 PowerShell, 344
 omijanie
 białej listy aplikacji, 320
 mechanizmów filtrowania
 adresów URL, 317
 programów antywirusowych, 305
 zabezpieczeń, 302
 działających na poziomie
 aplikacji, 316
 Windows, 322
 OpenVAS, 148
 dostosowywanie skanera, 150
 operatory Google, 72
 OSINT, *Patrz* biały wywiad

P

pakiet
 Armitage, 346
 atakowanie wielu systemów,
 346
 BeEF, 289–297
 integracja z Metasploit, 296
 konfigurowanie, 290
 moduły, 298
 tunelowanie serwera proxy, 297
 wstrzykiwanie kodu, 289
 DET, 435
 Docker, 39

EMET, 322
 Faraday, 54
 Ghost Phisher, 217
 Guest Additions, 45
 Impacket, 410
 Kismet, 200
 Maltego, 63
 Metasploit, 334
 atakowanie pojedynczych
 systemów, 342, 344
 atakowanie wielu systemów, 345
 autonomiczny plik
 wykonywalny, 424
 biblioteki, 334
 iniekcje DLL, 393
 interfejsy, 335
 moduł PSEXEC, 408
 moduły, 336
 odwrócona powłoka, 344
 tworzenie agenta persystencji,
 424
 utrzymywanie trwałego
 dostępu, 422
 Metasploitable3, 52
 Mutillidae, 54
 Phishing Frenzy, 188, 193
 SET, 170
 Social Engineering Toolkit, 178
 SPARTA, 125
 SPIKE, 355
 Veil Framework, 307
 Veil-Pillage, 375
 moduły, 376
 VirtualBox, 34
 Websploit, 212
 XAMPP, 54
 XSS Framework, 285
 persystencja, 416
 phishing, 188, 192
 Phishing Frenzy, 188, 193
 pivoting, 385
 plik
 dns.conf, 184
 SAM, 161
 pliki
 .hta, 179
 danych, 369
 hasel, 369
 kluczy, 369
 konfiguracyjne, 369
 rejestru systemowego, 369
 skrzynek poczty elektronicznej, 369
 systemowe, 367
 pobieranie wrażliwych danych, 368

- podatności, 131
 - aplikacji sieciowych, 139
 - określonych usług i aplikacji, 239
- podnoszenie uprawnień, 389, 391, 400
 - w Active Directory, 405
- polecenia
 - cmdlet, 383
 - do eksploracji systemu, 367, 383
 - inwazyjne, 371
 - konsolowe Windows, 116
- polecenie
 - ./empire, 396
 - ./incept, 164
 - ./install.sh, 55
 - ./testssl.sh, 259
 - ./update.py, 376
 - adduser, 45
 - agents, 397
 - aircrack, 208
 - airmon-ng, 198
 - airodump, 199, 202
 - apt-get, 45, 88, 169
 - apt-get update, 34
 - armitage, 346
 - arp, 116
 - arp-scan, 117
 - at, 392
 - atk6-alive6, 100
 - background, 391, 394
 - bettercap, 402
 - burpsuite, 233
 - bypassuac http, 400
 - chmod, 120
 - chntpw, 161
 - clearv, 437
 - commix, 242
 - copy, 367
 - creds, 397, 408
 - crunch, 209
 - db_import, 338
 - db_status, 338
 - dd, 31
 - detect-new-ip6, 108
 - dir, 246
 - dnsrecon, 98
 - docker, 39
 - dpkg, 367
 - execute, 399
 - exit, 394, 397
 - exploit, 341
 - faraday.py, 55
 - fierce, 97
 - for, 117
 - fragroute, 106
 - generate, 309
 - getsystem, 326
 - git, 376, 395
 - grep, 132
 - help, 95, 397
 - hping3, 103
 - htrack, 232
 - ifconfig, 115, 202, 367
 - impersonate_token, 375
 - info, 308, 340
 - interact, 397
 - ipconfig /all, 367
 - ipconfig /displaydns, 367
 - iptables, 268, 367
 - iwconfig, 197
 - kismet, 201
 - lbd, 104
 - list, 308, 397
 - listeners, 397
 - load, 96, 397
 - load xssf, 286
 - loot, 342
 - maltegoce, 63
 - migrate, 423
 - mimikatz, 400
 - msfconsole, 256, 335, 345
 - msfinit, 337
 - msfvenom, 276, 310, 361, 424
 - nbtstat, 116
 - nc, 420
 - net share, 116
 - net use, 116
 - net user, 116
 - net view, 116, 368
 - netcat, 229, 354
 - netsh, 304, 418
 - netstat, 116
 - nmap, 86, 104, 122, 135, 270
 - nslookup, 116
 - openvas-setup, 148
 - openvas-start, 149
 - openwav-check-setup, 148
 - passwd root, 44
 - proxychains, 90
 - ps, 372, 394
 - queryval, 418
 - recon-ng, 95
 - reg, 117, 371, 424
 - reload, 397
 - reset, 397
 - responder, 402
 - route, 116
 - run, 96, 256, 373, 378
 - sc, 385
 - schtask, 284
 - search, 339
 - searchmodule, 397
 - searchsploit, 132
 - sessions, 278
 - set, 96, 309, 377, 397
 - setg, 377
 - shell, 380
 - shellter, 312
 - show, 96, 397
 - sysinfo, 343
 - tepdump, 433
 - timestomp, 439
 - traceroute, 100–104
 - tracert, 101
 - tshark, 434
 - twofi, 80
 - upload, 394
 - use, 377
 - usemodule, 397
 - usestager, 397
 - vncviewer, 256
 - websploit, 212
 - whoami, 257, 367
 - whois, 92
 - wmic, 117, 382
 - workspace, 338
 - xssf_victims, 288
- połączenia SSL, 257, 260
- połączenie trwale, 415,
 - Patrz także* utrzymywanie trwałego dostępu, 426
- porty, 109
 - skanowanie, 109
- poszukiwanie podatności, 24
- poświadczenia logowania, 174, 241
- PowerShell, 382
- PowerShell Empire, 395
- powłoka
 - Meterpreter, 372, 380
 - PowerShell, 282, 344, 437
 - webshell, 245, 246
- profilowanie użytkowników, 78
- program, *Patrz* aplikacja
- programy antywirusowe, 305
- protokoły
 - kryptograficzne, 257
 - zdalnego dostępu, 250
- protokół
 - AH, 269
 - DHCP, 114
 - DNS, 431
 - ESP, 269
 - ICMP, 433
 - IKE, 269
 - IPSec, 269
 - IPv4, 97
 - IPv6, 98, 99
 - ISAKMP, 269

- Kerberos, 412
 - RDP
 - przelamywanie zabezpieczeń, 250
 - SA, 269
 - SNMP, 120
 - SSH
 - przelamywanie zabezpieczeń, 253
 - tunelowanie połączeń
 - przez zapory sieciowe, 316
 - SSL, 257
 - analizowanie połączeń, 260
 - atak typu DoS, 268
 - przelamywanie zabezpieczeń, 257
 - TLS, 257
 - VNC
 - przelamywanie zabezpieczeń, 255
 - przechwytywanie
 - hasła, 404
 - kluczy PSK, 272
 - przeglądarki internetowe
 - narzędzia bezpieczeństwa, 237
 - przekierowanie
 - DNS, 183
 - portów, 385
 - przeszukiwanie sieci, 239
 - PsExec, 381
- R**
- Reaver, 210
 - relacje zaufania domen, 380
 - Responder, 402
 - root
 - zmiana hasła, 44
 - rozgłoszenia ARP, 117
 - rozpoznanie
 - aktywne, 27, 61, 83
 - DNS, 62, 91, 97
 - pasywne, 27, 59, 60
 - skompromitowanego systemu, 367
 - środowisk celu, 83
 - w sieciach bezprzewodowych, 197
 - witryny internetowej, 225
- S**
- SAM, Security Accounts Manager, 161
 - schtasks, 421
 - scraping, 68
 - searchsploit, 350
 - SecurityFocus, 350
 - serwer
 - Burp Proxy, 232
 - proxy, 88, 232
 - sesja SMB, 121
 - Shellter, 312
 - sieci
 - anonimowe, 88
 - bezczprzewodowe, 196
 - ataki typu DoS, 211
 - atakowanie routerów, 210
 - filtrowanie adresów MAC, 204
 - omijanie zabezpieczeń, 202
 - otwarte uwierzytelnianie, 204
 - przeprowadzanie rozpoznania, 197
 - szyfrowanie WPA i WPA2, 206
 - szyfrowanie WPA/WPA2-Enterprise, 213
 - ukryty identyfikator SSID, 202
 - wirtualne, 49
 - sieć
 - Dark Web, 75
 - Tor, 88
 - skaner
 - Nikto, 140
 - Vega, 140
 - skanery
 - DNS, 97
 - podatności, 130, 135, 239
 - aplikacji mobilnych, 146
 - aplikacji sieciowych, 139
 - OpenVAS, 148
 - specjalizowane, 150
 - skanowanie, 24, 84
 - dużych środowisk celu, 113
 - NetBIOS, 387
 - portów, 109
 - poszukiwanie bramek VPN, 270
 - skrypty, 279
 - do automatycznego zbierania informacji, 74
 - do łączenia skanów, 119
 - LUA, 137
 - NSE, 135, 137, 261
 - persistence, 423
 - powłoki PowerShell, 282
 - VBScript, 279
 - słowniki do łamania hasła, 78
 - SMSishing, 158
 - sniffer, 250
 - sniffery hasła, 401
 - SNMP, 120
 - socjotechnika, 155
 - SPARTA, 125
 - konfiguracja pakietu, 126
 - Sticky Keys, 163
 - sygnatury aplikacji, 228
 - system
 - CMS, 228
 - IDS/IPS, 105
 - równoważenia obciążenia, 227
 - szybka wymiana danych, 73
 - szyfrowanie, 260
 - WPA/WPA2-Enterprise, 206, 213
- Ś**
- środowisko
 - celu
 - powłamaniowa eksploracja, 365
 - Faraday IDE, 55
 - testowe, 49
- T**
- techniki skanowania, 84
 - technologia NAC, 302
 - testowanie połączeń SSL, 261
 - Testssl, 259
 - testy penetracyjne, 24
 - metodologia, 26
 - pakiet Faraday, 54
 - Tor, 88
 - transformacja, 63
 - trwałe połączenie, *Patrz*
 - utrzymywanie trwałego dostępu, 426
 - tunelowanie
 - połączeń przez zapory sieciowe, 316
 - serwera proxy, 297
 - Twofish, 80
 - tworzenie
 - agenta persystencji, 424
 - autonomicznego pliku wykonywalnego, 424
 - bazy danych, 337
 - dodatkowych kont, 371
 - exploitów, 353, 360
 - instancji AWS, 41
 - lustrzanej kopii strony, 231
 - skanera portów, 110
 - sygnatur aplikacji, 228
 - złośliwego pliku wykonywalnego, 312
 - tylne wejścia, 276

U

UAC, User Account Control, 324
 udostępnianie folderów, 46
 udziały sieciowe, 123, 380
 ukrywanie

- plików wykonywalnych, 181
- śladów ataku, 437

 uprawnienia, 389

- poziom administratora, 392
- poziom systemu, 392
- w systemie lokalnym, 391

 urządzenia fizyczne, 166
 usługa

- Active Directory, 49
- DHCP, 114
- FTP, 109
- LSA, 371
- Tor, 88
- Windows Instrumentation, 382

 usługi

- Google Cache, 67
- systemowe, 430

 utrzymanie połączenia, 298
 utrzymywanie trwałego dostępu, 426

- agent persystencji, 424
- media społecznościowe, 426
- pakiet Metasploit, 422
- poczta Gmail, 426

 użytkownik

- konto, 44, 69, 121, 323
- root, 44
- zmiana hasła, 44

 używanie agentów persystencji, 416

V

Vega, 140

- dostosowywanie skanera, 142

 Veil Framework, 307
 Veil-Pillage, 375
 VirtualBox, 34
 ViShing, 158

VMware Tools, 45
 VMware Workstation Player, 32
 VoIP, Voice over IP, 257
 VPN, Virtual Private Network, 269

W

wiersz poleceń

- kopiowanie strony internetowej, 231

 Wi-Fi phishing, 158
 Windows

- exploity, 353, 360
- zabezpieczenia systemu, 322

 Windows Credential Editor, 380
 Windows Defender, 314
 Windows Task Scheduler, 421
 Wireshark, 250
 wirtualizacja, 31
 WMIC, 381, 382
 WPA, Wi-Fi Protected Access, 206
 WPS, Wi-Fi Protected Setup, 210
 wrażliwe dane, 368
 współużytkowanie folderów, 46
 wstrzykiwanie

- bibliotek DLL, 393
- kodu, 289

 wykrywanie

- hostów, 108, 115, 117
- otwartych portów, 109
- usług, 112
- zapór WAF, 227

 wyszukiwanie

- luk w zabezpieczeniach, 129
- podatności, 129
- wrażliwych danych, 368

 wyszukiwarka

- censys.io, 70, 104
- Shodan, 70, 104

X

XAMPP, 54
 XSS, Cross-Site Scripting, 285

Z

zabezpieczenia

- aplikacji internetowych, 221
- działające na poziomie aplikacji, 316
- mechanizm NAC, 302
- protokołu
 - RDP, 250
 - SSH, 253
 - SSL, 257
 - VNC, 255
- punktów końcowych, 304
- sieci bezprzewodowej
 - filtrowanie adresów MAC, 204
 - ukryty identyfikator SSID, 202
- systemu Windows
 - audyt, 331
 - bezpieczeństwo komunikacji, 331
 - działające na poziomie systemu, 330
 - kontrola konta użytkownika, 323
 - logowanie, 331
 - pakiet EMET, 322
 - szyfrowanie, 330
 - uwierzytelnianie, 329
- zapobieganie wyciekom danych, 435
- zapora sieciowa, 104, 316
 - WAF, 227
- zarządzanie testami penetracyjnymi, 54
- zaufanie między domenami, 380
- zbieranie
 - informacji
 - o adresach e-mail, 69
 - o nazwach kont, 69
 - o użytkownikach, 70
 - stosowanie skryptów, 74
 - poświadczeń, 400
- zdalny dostęp, 249
- zmiana hasła użytkownika root, 44

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>

Kali Linux — subtelne narzędzie pentestera!

Kali Linux jest dystrybucją BackTrack systemu Linux służącą do zaawansowanego badania zabezpieczeń systemów teleinformatycznych, również poprzez testy penetracyjne. Naturalnie, praca pentestera wiąże się również z przeprowadzaniem rozpoznania, skanowaniem w poszukiwaniu słabych stron zabezpieczeń, wykorzystywaniem exploitów, omijaniem zabezpieczeń i późniejszą eksploracją skompromitowanych systemów. W ten sposób diagnozuje się słabe strony systemu, co z kolei umożliwia usunięcie usterek i osiągnięcie wysokiego stopnia bezpieczeństwa. Realizacja tak ambitnego celu wymaga jednak znakomitego przygotowania i szerokiej wiedzy pentestera.

Dzięki tej książce poznasz sprawdzone techniki pokonywania mechanizmów obronnych różnych systemów za pomocą narzędzi dostępnych w Kali Linux. Dowiesz się, jak wybrać najbardziej efektywne rozwiązania, nauczysz się szybkiego skanowania sieci w poszukiwaniu luk w systemie zabezpieczeń, aż w końcu będziesz mógł przeprowadzić atak i powłamaniową eksplorację środowiska, przy czym będziesz wiedzieć, jakie techniki zminimalizują ryzyko wykrycia. Zapoznasz się ze specyfiką ataków na sieci bezprzewodowe, aplikacje internetowe i systemy wykorzystujące zdalny dostęp. W książce przedstawiono również zagadnienia związane z bezpieczeństwem fizycznym infrastruktury i z metodami socjotechnicznymi stosowanymi przez hakerów.

Najważniejsze zagadnienia:

- zarys metodologii testów penetracyjnych
- aktywne i pasywne rozpoznanie celu przed atakiem
- rozpoznawanie i przełamywanie zabezpieczeń
- powłamaniowa eksploracja celu i pozioma eskalacja ataku
- przejmowanie kontroli nad skompromitowanym systemem

Vijay Kumar Velu jest zapalonym praktykiem bezpieczeństwa teleinformatycznego.

Ma ponad 11-letnie doświadczenie w branży IT. Zdobył wiele certyfikatów bezpieczeństwa, w tym Certified Ethical Hacker, EC-Council Certified Security Analyst i Computer Hacking Forensics Investigator. Jest członkiem zarządu Cloud Security Alliance (CSA) w Kuala Lumpur oraz członkiem National Cyber Defense and Research Center (NCDRC) w Indiach. Fanatyk technologii, kocha muzykę i chętnie angażuje się w działalność charytatywną.

 helion.pl	<i>Sprawdź nasze szkolenia!</i> SZKOLENIA  AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	KOD KORZYŚCI <i>Ślepnij po więcej!</i>   ISBN 978-83-283-4037-4  9 788328 340374
 helion.pl		
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl		
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 69,00 zł

Packt