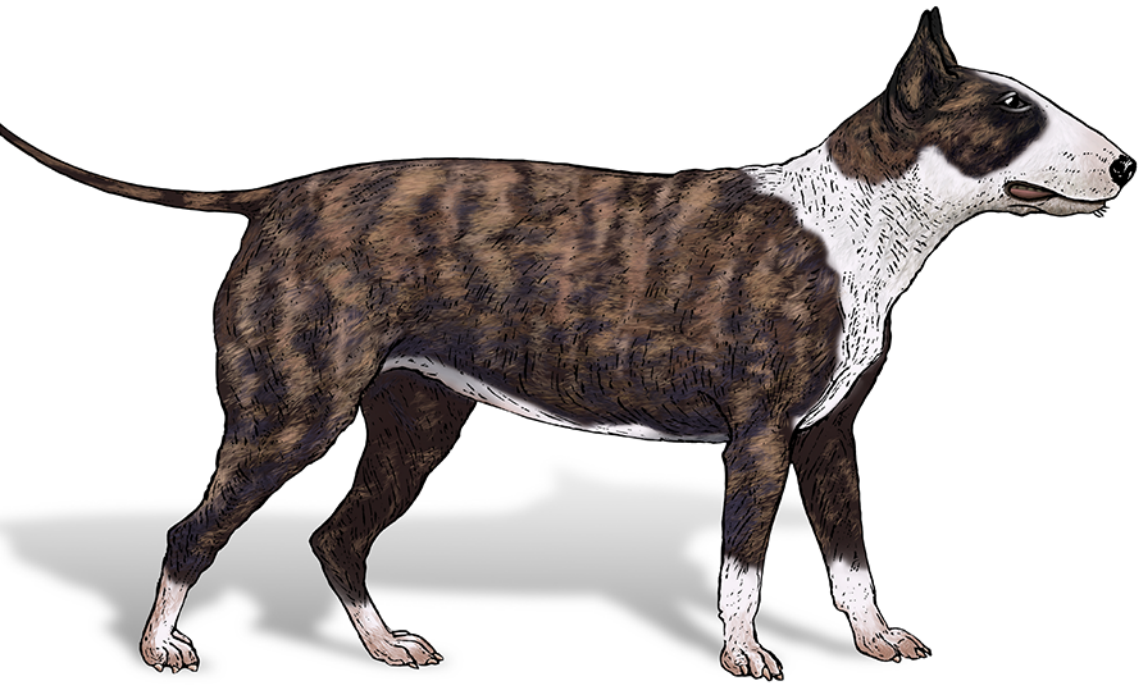
O'REILLY®

Wydanie II

Kali Linux

Testy bezpieczeństwa, testy penetracyjne i etyczne hakowanie



Helion

Ric Messier

Tytuł oryginału: Learning Kali Linux: Security Testing, Penetration Testing and Ethical Hacking, 2nd Edition

Tłumaczenie: Robert Górczyński, z wykorzystaniem fragmentów poprzedniego wydania w przekładzie Andrzeja Watraka

ISBN: 978-83-289-3000-1

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/kalte2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

ROZDZIAŁ 3.

Rekonesans

Testy penetracyjne, ataki etyczne czy weryfikacja bezpieczeństwa systemów zazwyczaj obejmują określony zakres prac. Mogą być w nim ujęte wszystkie obiekty testów, tj. systemy i ludzie, jednak rzadko tak jest. Wtedy obiekty trzeba określić samodzielnie. W tym celu trzeba przeprowadzić tzw. **rekonesans**. Za pomocą narzędzi dostępnych w systemie Kali Linux można uzyskać mnóstwo informacji o firmie i jej pracownikach.

Celami ataków są nie tylko systemy i działające w nich aplikacje, lecz także ludzie. Możesz — choć nie musisz — dostać zlecenie przeprowadzenia ataków socjologicznych. Obecnie w celu włamania się do systemów i zainfekowania ich wykonuje się niemal wyłącznie tego rodzaju ataki. Wprawdzie statystyki różnią się w zależności od badanego roku, ale zgodnie z niektórymi oszacowaniami, na przykład badaniami przeprowadzonymi przez firmy Verizon i Mandiant, przeważająca większość przypadków kradzieży danych jest skutkiem stosowania technik inżynierii społecznej.

Ten rozdział jest poświęcony pozyskiwaniu informacji na odległość, aby tego rodzaju aktywność nie została zauważona. W pewnym momencie jednak trzeba nawiązać kontakt z obiektem testów, aby jeszcze bardziej zbliżyć się do stosowanych w nim systemów. Opisane jest tu jedno z najważniejszych zagadnień: skanowanie portów. Dzięki niemu można uzyskać mnóstwo szczegółowych informacji, a za pomocą dodatkowych narzędzi i technik można określić cele ataku i zawęzić zakres ich poszukiwań.

Czym jest rekonesans?

Zacznijmy od zdefiniowania znaczenia słowa *rekonesans*, abyśmy mówili tym samym językiem. Zgodnie ze słownikiem języka polskiego rekonesans jest to „wstępne rozeznanie w czymś". Definicja sugeruje związki tego terminu z wojskowością. Zważywszy na pojęcia, jakie stosuje się w dziedzinie ochrony informacji, tego rodzaju związek nie powinien wydawać się dziwny. Mówimy o atakach, obronie przed atakami i oczywiście o rekonesansie. Rekonesans polega w tym przypadku na zbieraniu informacji, dzięki którym praca testerów (czyli napastników i agresorów) jest łatwiejsza. Testy można wprawdzie przeprowadzać na oślep, angażując w nie wszystkie siły, ale wtedy ich zasięg jest ograniczony. Trzeba zachowywać ostrożność i mieć

świadomość upływającego czasu. Lepiej jest na początku poświęcić chwilę na sprawdzenie, z czym ma się do czynienia, niż stracić mnóstwo czasu, strzelając na oślep.

Rozpoczynając zbieranie informacji o obiektach ataków, lepiej nie robić wokół tego zbędnego szumu. Rozeznanie trzeba wykonywać z bezpiecznej odległości, nie angażując w nie obiektu bezpośrednio. Oczywiście angażowanie za każdym razem może wyglądać inaczej. Jeżeli pracujesz w firmie, nie musisz siedzieć cicho, ponieważ wszyscy wiedzą, co robisz. Niemniej jednak stosując opisane w tym rozdziale taktyki, możesz dowiedzieć się, jakie ślady zostawia firma w internecie. Może się okazać, że wyciekają informacje, które nigdy nie powinny były zostać upublicznione. Aby obronić firmę przed atakami, możesz wykorzystać bezpłatne narzędzia wywiadowcze i zastosować różne techniki.

OPSPEC

Jednym z ważnych pojęć, o którym warto tu wspomnieć, jest OPSPEC (ang. *operations security*, **bezpieczeństwo operacji**). Zapewne słyszałeś pochodzący z czasów drugiej wojny światowej zwrot „jedno chlapnięcie i po okręcie". Jest on kwintesencją bezpieczeństwa operacji. Krytyczne dla misji informacje muszą pozostać tajne. W przypadku misji militarnych do zachowania tajemnicy zobowiązuje się nawet członków rodziny. Gdyby ktoś z rodziny ujawnił, w którym miejscu świata znajduje się jego bliski i jakie szczególne umiejętności posiada, ktoś postronny mógłby się domyśleć, co się dzieje. Połączyłby fakty niczym puzzle. Jeżeli firma upubliczniłaby zbyt wiele informacji o sobie, nieprzyjaciel (bez względu na jego naturę) mógłby się o niej bardzo wiele dowiedzieć. Przestrzeganie głównych zasad OPSEC jest niezmiernie ważne, aby uchronić się przed intruzami i zapobiec poznaniu przez konkurencję istotnych danych przedsiębiorstwa.

Warto również wiedzieć, jakiego rodzaju agresorzy są najbardziej niebezpieczni dla firmy. Obawy mogą dotyczyć przejęcia własności intelektualnej przez konkurentów lub szeroko pojętych ataków ze strony zorganizowanych grup przestępczych albo państw wyznaniowych. Tego rodzaju rozważania mają pomóc określić, jakie informacje powinny pozostać w firmie, a które można ujawniać.

Dobre praktyki OPSEC mogą pomóc w ochronie przed wybranymi technikami rekonesansu, które zostaną omówione w tym rozdziale.

Gdybyśmy chcieli zabezpieczyć się przed atakami sieciowymi, wystarczyłoby wykonać skanowanie portów i usług. Jednak pełny test bezpieczeństwa nie może polegać na wykonaniu zwykłego, technicznego ataku typu „włamię się przez otwarty port". Test musi obejmować procedury operacyjne, zachowania użytkowników, inżynierię społeczną i wiele innych obszarów. Bezpieczeństwo w biznesie to nie tylko bezpieczeństwo usług udostępnianych zewnętrznemu światu. Dlatego podczas rekonesansu poprzedzającego testy trzeba wykonywać znacznie więcej operacji niż zwykłe skanowanie portów.

Jedną z wielkich zalet internetu jest dostęp do mnóstwa informacji. Jednak im dłużej jesteś połączony z internetem i częściej z niego korzystasz, tym więcej zostawiasz po sobie śladów. To dotyczy zarówno pojedynczych osób, jak i całych organizacji. Pomyśl na przykład o serwisach społecznościowych. W jaki sposób w nich istniejesz? Ile informacji o sobie w nich umieszczasz?

Ile jest w nich informacji o pracowniach Twojej firmy? Oprócz tego internet przechowuje dane niezbędne do tego, aby działał, a użytkownicy mogli z niego korzystać. Są to nazwy domen, kontakty, dane firm, adresy i inne informacje, które można wykorzystać w testach bezpieczeństwa.

Na przestrzeni lat informacja nabrała takiego znaczenia, że powstało wiele narzędzi ułatwiających pozyskiwanie danych z miejsc ich przechowywania. Są to narzędzia stosowane w wierszu poleceń, jak również strony WWW, wtyczki do przeglądarek i różne programy. Miejsc, w których znajdują się informacje, jest mnóstwo, ponieważ coraz więcej ludzi korzysta z internetu. Nie będziemy zajmować się wszystkimi sposobami pozyskiwania informacji za pomocą stron WWW, choć jest ich bardzo wiele. Skupimy się tylko na narzędziach dostępnych w systemie Kali.

Biały wywiad

Jeszcze nie tak dawno temu trudniej było spotkać użytkownika, który wyraźnie istniał w sieci, niż użytkownika, który nie miał pojęcia, czym jest internet. Ale ta sytuacja szybko się odwróciła. Nawet ci, którzy unikają serwisów społecznościowych, takich jak TikTok, Facebook, X (dawniej Twitter), Instagram i wiele innych, są obecni w sieci. Jest tak przede wszystkim dlatego, że istnieją publiczne rejestry danych, w których można znaleźć każdego, kto ma na przykład telefon stacjonarny. Dotyczy to również osób, które nie mają wiele wspólnego z internetem. O ludziach, którzy korzystają z niego od dłuższego czasu, informacji jest znacznie więcej. Moja przygoda z internetem trwa od kilku dekad.

Czym jest **biały wywiad**? Jest to pozyskiwanie informacji we wszelkich ogólnodostępnych źródłach, czy to urzędowych, na przykład dotyczących obrotu nieruchomościami, czy to w upublicznionych archiwach list mailingowych. Termin *biały wywiad* możesz kojarzyć z internetem, ale odnosi się on również do wszelkiego rodzaju informacji gromadzonych w miejscach dostępnych dla każdego (nie są nimi na przykład płatne serwisy internetowe z informacjami o obywatelach).

Zapewne zadajesz sobie pytanie, po co robi się biały wywiad. Jest wiele powodów, aby podczas testów bezpieczeństwa go robić; nie polega on na śledzeniu ludzi. Przede wszystkim celem jest pozyskanie adresów IP i nazw komputerów. Jeżeli Twoim zadaniem jest wykonanie testów całkowicie w tzw. trybie *red team*, czyli spoza instytucji, o której w dodatku nie masz żadnych szczegółowych informacji, musisz wiedzieć, co będziesz atakował. Oznacza to, że powinieneś się dowiedzieć, jakie systemy są stosowane w danej firmie oraz jacy ludzie w niej pracują. Jest to ważne, ponieważ stosując inżynierię społeczną, można w prosty i skuteczny sposób uzyskać dostęp do systemów lub przynajmniej do dodatkowych informacji.

Jeżeli pracujesz w firmie jako specjalista ds. bezpieczeństwa, możesz otrzymać polecenie poszukania śladów pozostawianych w internecie przez firmę i jej najważniejszych pracowników. Firmy starają się zmniejszać prawdopodobieństwo ataku, ograniczając wypływ informacji do otaczającego je świata. Całkowicie powstrzymać tego procesu oczywiście się nie da, jednak udostępniane są przynajmniej nazwy domen, powiązane z nimi adresy IP oraz wpisy w serwerach DNS. Gdyby te informacje nie były publiczne, klienci i inne przedsiębiorstwa, na przykład dostawcy i partnerzy, nie mogliby nawiązać z daną firmą kontaktu.

Wyszukiwarki internetowe dostarczają mnóstwa informacji, dlatego stanowią dobry punkt wyjścia w rekonesansie. Jednak ze względu na ogrom stron istniejących w sieci szybko zostaniesz przytłoczony ilością uzyskiwanych wyników. Aby tego uniknąć, możesz zdefiniować kryteria wyszukiwania. Nie jest to wprawdzie temat ściśle związany z systemem Kali i bardzo wielu użytkowników go zna, jednak jest na tyle ważny, że warto go tu poruszyć. Podczas wykonywania testów bezpieczeństwa będziesz bardzo często wyszukiwał różnego rodzaju informacje. Umiejętnie stosując opisane techniki, zaoszczędzisz sobie mnóstwo czasu, ponieważ nie będziesz musiał przeglądać stron zawierających nieistotne informacje.

Przed przeprowadzeniem ataku wykorzystującego techniki inżynierii społecznej warto dowiedzieć się, kto pracuje w danej instytucji. Można to zrobić na wiele sposobów. Szczególnie przydatne są tu serwisy społecznościowe. Bogatym źródłem informacji o firmach i pracownikach jest LinkedIn. Mnóstwo informacji jest również dostępnych na stronach z ofertami pracy. Jeżeli jakaś firma poszukuje kandydatów znających na przykład rozwiązania Cisco i Microsoft Active Directory, można się domyśleć, jakiego rodzaju infrastruktura jest w niej wykorzystywana. Pewnych informacji dotyczących firm i jej pracowników dostarczają też serwisy takie jak LinkedIn i Facebook.

Miejsc poszukiwania danych firmy jest bardzo dużo. Na szczęście system Kali oferuje narzędzia ułatwiające to zadanie. Są to programy automatycznie pozyskujące mnóstwo informacji z wyszukiwarek i różnych stron internetowych. Narzędzia takie jak theHarvester są proste w użyciu i pozwalają zaoszczędzić mnóstwo czasu. Program Maltego nie tylko automatycznie zbiera informacje, lecz także wyświetla je w sposób ułatwiający ich przeglądanie. Zanim jednak przejdziemy do wymienionych narzędzi, przyjrzyjmy się podstawowym metodom efektywnego zbierania informacji.

Google Hacking

Wyszukiwarki internetowe istniały długo przed pojawieniem się Google. Ta wyszukiwarka jednak funkcjonuje inaczej niż inne i zdominowała popularne serwisy, takie jak AltaVista, InfoSeek czy Inktomi, które ostatecznie zostały przejęte przez inne firmy lub zniknęły z rynku. Podobny los spotkał wiele innych wyszukiwarek. Zespół Google nie tylko zbudował przydatną wyszukiwarkę, lecz także znalazł wyjątkowy sposób, aby wyszukiwanie informacji przekuć na pieniądze, dzięki czemu firma jest dochodowa i istnieje na rynku.

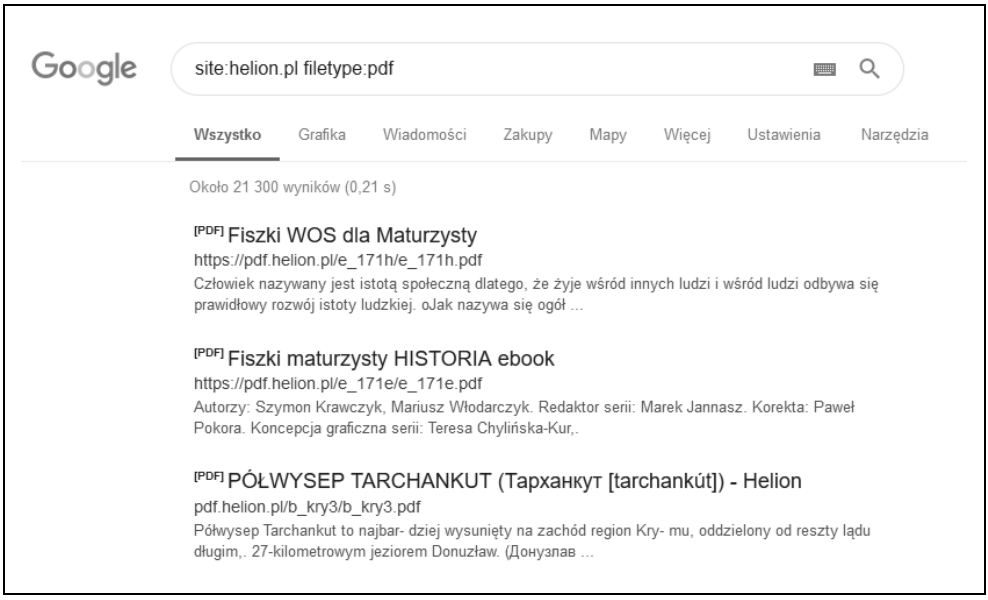
Jedną z nowych funkcjonalności oferowanych przez Google są słowa kluczowe modyfikujące proces wyszukiwania stron. Dzięki temu uzyskuje się mniejsze zbiory wyników. Zapytania wykorzystujące słowa kluczowe określane są mianem **Google Dorks**, a proces wyszukiwania ściśle określonych stron **Google Hacking**. Znajomość tych dwóch terminów może być bardzo pomocna przy wyszukiwaniu informacji o obiekcie ataku.

Jednym z najważniejszych słów kluczowych umożliwiających wyszukiwanie informacji o konkretnym obiekcie jest `site`. Powoduje ono, że Google wyszukuje informacje tylko w obrębie zadanej strony lub domeny. Na przykład wpisanie kryterium `site:helion.pl` sprawia, że w wynikach wyszukiwania umieszczane są jedynie strony z domen kończących się na `helion.pl`.

Można to porównać do korzystania z wewnętrznej wyszukiwarki Google danej firmy. Różnica polega jednak na tym, że przeszukiwane są wszystkie strony należące do danej domeny.

Choć powyższa technika przypomina korzystanie z wewnętrznej wyszukiwarki danej instytucji należy pamiętać, że w wynikach umieszczane są wyłącznie strony dostępne w internecie. Ponadto nie ma wśród nich stron, które wprawdzie istnieją, ale żadne inne strony do nich się nie odwołują. Nie można na przykład przeszukiwać stron intranetowych. Jest to możliwe jedynie, kiedy działa się wewnątrz danej instytucji. Mogą zdarzać się błędne konfiguracje, a roboty indeksujące witryny internetowe musiałyby wiedzieć o witrynach, aby do nich dotrzeć. Istnieje więc możliwość, że ktoś połączy witrynę zewnętrzną z witryną wewnętrzną firmy, do której w ten sposób będzie można uzyskać dostęp spoza organizacji. W takim przypadku wewnętrzna witryna firmy stanie się narażona na jej indeksowanie przez wyszukiwarkę internetową. Jednak z reguły osoba z zewnątrz nie otrzyma żadnych szczegółów dotyczących wewnętrznej witryny organizacji.

Wyniki wyszukiwania można ograniczać do określonych plików, na przykład arkuszy kalkulacyjnych lub dokumentów PDF. W tym celu należy użyć słowa kluczowego `filetype`. Aby uzyskać jeszcze dokładniejsze wyniki, można stosować kilka słów jednocześnie. Rysunek 3.1 przedstawia wyniki wyszukiwania spełniające kryterium `site:helion.pl filetype:pdf`. Zawierają one odnośniki do dokumentów PDF znajdujących się na stronach z domen kończących się na `helion.pl` (patrz dwa ostatnie wyniki).



Rysunek 3.1. Wyniki wyszukiwania przez Google określonych plików na określonych stronach

Są również dwa inne słowa, które można ze sobą łączyć: `inurl` i `intext`. Pierwsze służy do przeszukiwania samych adresów URL, a drugie tylko treści stron. Domyślnie Google analizuje różne elementy stron. Za pomocą powyższych słów można ograniczyć zakres poszukiwań zadanego ciągu. Słowa te szczególnie przydają się do wyszukiwania stron zawierających w adresie na przykład ciąg `/cgi_bin/`. Słowo kluczowe `intext` powoduje, że Google wyszukuje zadany ciąg tylko w treści stron. Zwykle wyszukiwarka zwraca wyniki, które mogą nie zawierać wszystkich zadanych ciągów. Aby temu zapobiec, należy użyć odpowiednio słów `allinurl` i `allintext`.

Od czasu do czasu niektóre słowa kluczowe zmieniają się. Przykładowo: znikło słowo `link`. Słowa opisane wyżej są najczęściej używane. Pamiętaj, że wiele z nich można stosować łącznie. Oprócz tego można wykonywać podstawowe operacje na wynikach. Na przykład operatory logiczne `AND` i `OR` powodują, że wyniki zwracane przez Google spełniają odpowiednio wszystkie kryteria lub przynajmniej jedno z nich. Ponadto do wyszukiwania słów użytych w zadanej kolejności można stosować cudzysłowy. Przykładowo, aby wyszukać wyłącznie strony z informacjami o Jeziorze Głębokim, należy użyć frazy `"Jezioro Głębokie"`. Jeżeli nie użyje się cudzysłowów, pojawią się strony zawierające w treści słowa `jezioro` i `głębokie` (prawdopodobnie większość z nich nie będzie interesująca).

Wartą odnotowania cechą wyszukiwarki Google jest baza przydatnych zapytań. Utworzył ją w 2004 r. Johnny Long, który w 2002 r. zaczął gromadzić różne przydatne kryteria wyszukiwania. Dzisiaj baza jest dostępna pod adresem *www.exploit-db.com*. Zapytania są podzielone na kategorie i zawierają mnóstwo ciekawych słów kluczowych, które można wykorzystać podczas wykonywania testów bezpieczeństwa w firmie. W tym celu należy wpisać wybrane z bazy zapytanie, po nim słowo `site:` i nazwę domeny. W wynikach mogą pojawić się strony potencjalnie podatne na ataki lub zawierające poufne informacje.

Ostatnim słowem kluczowym, które warto znać, aczkolwiek może ono ograniczać wyniki wyszukiwania, jest `cache`. Powoduje ono wybranie z pamięci podręcznej Google danych zebranych podczas ostatniego wyszukiwania. Ponieważ w kryteriach wyszukiwania nie można stosować dat, powyższe słowo nie dostarcza tak przydatnych wyników, jak strona Wayback Machine (*https://archive.org/web*). Jeżeli jednak z jakiegoś powodu wyszukiwana strona nie jest dostępna, można informacje o niej odczytać z bufora Google. Pamiętaj jednak, że nic nie da klikanie odnośników umieszczonych w wynikach. W takim przypadku należałoby ponownie użyć słowa kluczowego `cache`, aby odzyskać tę stronę. To słowo kluczowe dostarcza najnowszą wersję witryny istniejącą w pamięci podręcznej. Wyniki wskazują, kiedy ta najnowsza kopia została utworzona. Podczas testowania wybranej witryny otrzymałem kopię, która miała zaledwie kilka godzin. Niektóre witryny mogą być buforowane rzadziej niż inne. Wprawdzie można używać tego słowa kluczowego, aby odwoływać się do witryny, ale bardziej przydatne może się okazać wykorzystanie go z bezpośrednimi adresami URL. Na przykład jeśli masz adres URL, który odnosi się do określonej strony, możesz go użyć i pobrać zawartość strony z pamięci podręcznej Google.

Automatyczne zbieranie informacji

Ręczne wyszukiwanie informacji może być czasochłonne, szczególnie jeżeli trzeba stosować wiele zapytań, aby zebrać jak najwięcej danych. Na szczęście można wyniki uzyskiwać szybko za pomocą narzędzi oferowanych przez system Kali. Pierwszym narzędziem, któremu się przyjrzymy, jest theHarvester. Program ten wyszukuje szczegółowe informacje w różnych źródłach. Są to nie tylko najpopularniejsze wyszukiwarki Baidu, Bing i DuckDuckGo, lecz także wiele witryn związanych z zapewnianiem bezpieczeństwa, na przykład ThreatMiner, która służy do zbierania informacji dotyczących zagrożeń, i DNSDumpster, używana do wyszukiwania danych DNS.

Wcześniej program theHarvester mógł służyć także do wyszukiwania informacji na temat osób, ponieważ przeszukuje również serwis Pretty Good Privacy (PGP). W tym ostatnim przeszukuje bazę osób szyfrujących lub podpisujących swoje wiadomości e-mail. W ten sposób program może pozyskiwać adresy osób, które kiedykolwiek zarejestrowały swoje klucze. Ponadto przeszukuje biznesowy serwis społecznościowy LinkedIn, w którym użytkownicy zamieszczają swoje CV, poszukują kontaktów biznesowych i pracowników. Wprawdzie jest znacznie mniej przydatny do wyszukiwania adresów e-mail powiązanych z domeną, ale za to bardzo dobrze radzi sobie w identyfikowaniu adresów IP oraz w pełni kwalifikowanych nazw domen (ang. *Fully Qualified Domain Names*, FQDN) powiązanych z daną domeną. Możliwe, iż narzędzie nadal dobrze radzi sobie z wyszukiwaniem adresów e-mail, ponieważ wygenerowane dane wyjściowe sugerują, że ich szuka. Jednak nie udało się znaleźć żadnych adresów e-mail podczas przykładowego wyszukiwania domeny należącej do wydawnictwa O'Reilly.

Listing 3.1 pokazuje wyszukiwanie za pomocą programu theHarvester i opcji `sitedossier` w pełni kwalifikowanych nazw dla domeny *oreilly.com*. Różne źródła danych prowadzą do wygenerowania odmiennych wyników, a niektóre źródła typu otwartego mogą dostarczyć długie listy nazw powiązanych z analizowaną domeną. Wprawdzie w omawianym przykładzie program zwrócił 200 wyników, ale jeśli uruchomisz go w swoim systemie, możesz otrzymać inne wyniki, ponieważ systemy publiczne i powiązane z nimi adresy IP nie są statyczne. Te dane wyjściowe skróciłem, aby pokazać, jak wyglądają poszczególne ich sekcje.

Listing 3.1. Wynik użycia programu theHarvester

```
┌──(kilroy@badmilo)-[~]
└─$ theHarvester -b sitedossier -d oreilly.com
Read proxies.yaml from /home/kilroy/.theHarvester/proxies.yaml
*******************************************************************
*                                                                 *
* | |_| |__   ___     /\  /\__ _ _ ____   _____ ___| |_ ___ _ __  *
* | __| '_ \ / _ \   / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/  / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 4.6.0                                         *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************
```

```
[*] Target: oreilly.com
My current iter_url: http://www.sitedossier.com/parentdomain/oreilly.com/101
My current iter_url: http://www.sitedossier.com/parentdomain/oreilly.com/201
In total found: 200
{'security.oreilly.com', 'ofps3.vz.oreilly.com', 'chimera.labs.oreilly.com',
↪'corporate.oreilly.com', 's.radar.oreilly.com', 'apache.oreilly.com',
↪'access.safari.oreilly.com', 'opengovernment.labs.oreilly.com', 'register.oreilly.com',
↪'ajax.oreilly.com', 'portal.oreilly.com', 'beautifulcode.wiki.oreilly.com',
↪'libraries.oreilly.com', 'ormstore-staging.oreilly.com', 'programming-scala.labs
↪.oreilly.com', 'm.bookworm.oreilly.com', 'news.oreilly.com', 'hacks.oreilly.com.',
↪'macruby.labs.oreilly.com', 'nutshells.oreilly.com', 'etel.wiki.oreilly.com',
↪'mediaservice.oreilly.com', 'stats.oreilly.com', 'cachefly.oreilly.com',
↪'scifoo13.wiki.oreilly.com', 'agiledev.97things.oreilly.com', 'conference.oreilly.com',
↪'community.toc.oreilly.com', 'dev-blogs.oreilly.com', 'ignite.oreilly.com',
↪'bio.oreilly.com', 'rails-nutshell.labs.oreilly.com',
<snip>
[*] Searching Sitedossier.

[*] No IPs found.

[*] No emails found.

[*] Hosts found: 199
---------------------
97things.oreilly.com
academic.oreilly.com
access.safari.oreilly.com
actionscript.oreilly.com
admin.members.oreilly.com
agiledev.97things.oreilly.com
ajax.oreilly.com
akamaicovers.oreilly.com
amazon.oreilly.com
androidcookbook.oreilly.com
animals.oreilly.com
annoyances.oreilly.com
answers.oreilly.com
answers.oreilly.com.
answersstage.oreilly.com
apache.oreilly.com
apprenticeship-patterns.labs.oreilly.com
apprenticeship.oreilly.com
architect.97things.oreilly.com
assets.en.oreilly.com
assets.oreilly.com
atom.oreilly.com
beautifulcode.wiki.oreilly.com
bio.oreilly.com
blogs.oreilly.com
```

W listingu 3.2 zostały pokazane wyniki uzyskane podczas użycia wyszukiwarki internetowej DuckDuckGo, które jednak nie dostarczają tak wielu informacji jak w poprzednim przykładzie. Podczas gdy witryny internetowe, których celem jest rekonesans i zbieranie informacji dostarczają dziesiątki wyników, wyszukiwarka DuckDuckGo twierdzi, że znalazła dziewięć hostów, przy czym wymienia tylko pięć z nich.

Listing 3.2. Wynik programu theHarvester z wyszukiwarką internetową DuckDuckGo

```
┌──(kilroy@badmilo)-[~]
└─$ theHarvester -d oreilly.com -b duckduckgo
*******************************************************************
*                                                                 *
*  _   _                                            _             *
* | |_| |__   ___     /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \   / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/  / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___|  \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.3.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: oreilly.com

[*] Searching Duckduckgo.

[*] No IPs found.
[*] No emails found.
[*] Hosts found: 9

 ---------------------
conferences.oreilly.com
learning.oreilly.com
oreilly.com
radar.oreilly.com
toc.oreilly.com
```

Jest to przykład kolejnej operacji wyszukiwania informacji. Aby ułatwić sobie życie, można utworzyć prosty skrypt. Listing 3.3 przedstawia skrypt napisany w języku Python, wyszukujący w różnych źródłach domenę podaną w argumencie. Kod ten można znacznie usprawnić, aby byli w stanie korzystać z niego użytkownicy, którzy nie wiedzą, na czym dokładnie polega jego działanie. Jednak w moim prywatnym użytku skrypt ten sprawdza się doskonale. W wyniku swego działania wyświetla nazwy plików XML i HTML znalezionych w różnych źródłach.

Listing 3.3. Skrypt wyszukujący informacje za pomocą programu theHarvester

```
#!/usr/bin/python

import sys
import os

if len(sys.argv) < 2:
   sys.exit(-1)

providers = [ 'duckduckgo', 'bing', 'baidu', 'dnsdumpster', 'hunter', 'sitedossier' ]

for a in providers:
    cmd = 'theharvester -d {0} -b {1} -f {2}.html'.format(sys.argv[1], a,  a)
    os.system(cmd)
```

W pętli for jest wielokrotnie uruchamiany program theHarvester, który za każdym przeszukuje inne źródło informacji. Ponieważ program może zapisywać wyniki, skrypt ich nie wyświetla, tylko zapisuje w osobnym pliku dla każdego źródła. Aby dodać lub zmienić źródło, należy odpowiednio zmodyfikować listę. Na przykład można zrezygnować z serwisu *google-profiles* albo dodać Yahoo. Zwykła modyfikacja zmiennej `providers` pozwoli uzyskiwać dodatkowe wyniki w zależności od potrzeb.

LinkedIn może być bardzo cennym źródłem informacji. Do jego przeszukiwania można użyć narzędzia o nazwie Crosslinked, które jednak nie znajduje się w oficjalnym repozytorium. Niniejsza książka jest poświęcona możliwościom dystrybucji Kali Linux, dlatego nie będę w tym miejscu omawiał tego narzędzia. Ponieważ jesteśmy ograniczeni do dostępnych w oficjalnym repozytorium narzędzi do wydobywania informacji z serwisu LinkedIn, jego przeszukanie prowadzi do innego narzędzia, o nazwie `emailharvester`. Listing 3.4 przedstawia wynik użycia tego narzędzia podczas wyszukiwania adresów w LinkedInie. Zwróć uwagę na wielkość wygenerowanych danych wyjściowych, choć tak naprawdę otrzymaliśmy jedynie dwa adresy e-mail. Prawdopodobnie jest to rezultat ograniczeń nakładanych podczas przeprowadzania programowej operacji wyszukiwania. Źródła danych zwracają jedynie ograniczoną ilość danych, co zostało tutaj wskazane, zamiast rzeczywiście znalezionych adresów e-mail.

Listing 3.4. Wynik przeszukania serwisu LinkedIn za pomocą programu EmailHarvester

```
┌──(kilroy@badmilo)-[~]
└──$ emailharvester -d oreilly.com -e linkedin
[+] User-Agent in use: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101
↪Firefox/40.1
[+] Searching in Linkedin
[+] Searching in Yahoo + Linkedin: 101 results
[+] Searching in Bing + Linkedin: 50 results
[+] Searching in Bing + Linkedin: 100 results
[+] Searching in Google + Linkedin: 100 results
[+] Searching in Baidu + Linkedin: 10 results
[+] Searching in Baidu + Linkedin: 20 results
[+] Searching in Baidu + Linkedin: 30 results
[+] Searching in Baidu + Linkedin: 40 results
[+] Searching in Baidu + Linkedin: 50 results
[+] Searching in Baidu + Linkedin: 60 results
[+] Searching in Baidu + Linkedin: 70 results
[+] Searching in Baidu + Linkedin: 80 results
[+] Searching in Baidu + Linkedin: 90 results
[+] Searching in Baidu + Linkedin: 100 results
[+] Searching in Exalead + Linkedin: 50 results
[+] Searching in Exalead + Linkedin: 100 results
[+] Emails found: 2
2522@oreilly.com
22@oreilly.com
```

Programu `emailharvester` można użyć z wieloma innymi źródłami. Na listingu 3.5 pokazałem wyniki otrzymane dla wszystkich źródeł dostępnych w programie. Ta lista jest naprawdę obszerna, dlatego ją tutaj skróciłem.

Listing 3.5. Przykład użycia programu EmailHarvester ze wszystkimi dostępnymi źródłami

```
┌──(kilroy@badmilo)-[~]
└─$ emailharvester -d oreilly.com
[+] User-Agent in use: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101
↪Firefox/40.1
[+] Searching everywhere
[+] Searching in Instagram
[+] Searching in Yahoo + Instagram: 101 results
[+] Searching in Bing + Instagram: 50 results
[+] Searching in Bing + Instagram: 100 results
[+] Searching in Google + Instagram: 100 results
[+] Searching in Baidu + Instagram: 10 results
[+] Searching in Baidu + Instagram: 20 results
[+] Searching in Baidu + Instagram: 30 results
[+] Searching in Baidu + Instagram: 40 results
[+] Searching in Baidu + Instagram: 50 results
[+] Searching in Baidu + Instagram: 60 results
[+] Searching in Baidu + Instagram: 70 results
[+] Searching in Baidu + Instagram: 80 results
[+] Searching in Baidu + Instagram: 90 results
[+] Searching in Baidu + Instagram: 100 results
[+] Searching in Exalead + Instagram: 50 results
[+] Searching in Exalead + Instagram: 100 results
[+] Searching in Yahoo: 101 results
[+] Searching in ASK: 10 results
[+] Searching in ASK: 20 results
[+] Searching in ASK: 30 results
[+] Searching in ASK: 40 results
[+] Searching in ASK: 50 results
[+] Searching in ASK: 60 results
[+] Searching in ASK: 70 results
[+] Searching in ASK: 80 results
[+] Searching in ASK: 90 results
[+] Searching in ASK: 100 results
[+] Searching in Baidu: 10 results
[+] Searching in Baidu: 20 results
[+] Searching in Baidu: 30 results
[+] Searching in Baidu: 40 results
[+] Searching in Baidu: 50 results
[+] Searching in Baidu: 60 results
[+] Searching in Baidu: 70 results
[+] Searching in Baidu: 80 results
[+] Searching in Baidu: 90 results
[+] Searching in Baidu: 100 results
[+] Searching in Twitter
[+] Searching in Baidu + Reddit: 30 results
[+] Searching in Baidu + Reddit: 40 results
[+] Searching in Baidu + Reddit: 50 results
[+] Searching in Baidu + Reddit: 60 results
[+] Searching in Baidu + Reddit: 70 results
[+] Searching in Baidu + Reddit: 80 results
[+] Searching in Baidu + Reddit: 90 results
[+] Searching in Baidu + Reddit: 100 results
[+] Searching in Exalead + Reddit: 50 results
[+] Searching in Exalead + Reddit: 100 results
[+] Emails found: 20
mercedes@oreilly.com
```

```
22@oreilly.com
rmendana@oreilly.com
pixel-1687721587520467-web-@oreilly.com
adoption@oreilly.com
santiagocancino@oreilly.com
workwithus@oreilly.com
booktech@oreilly.com
orders@oreilly.com
permissions@oreilly.com
andrewc@oreilly.com
Info@oreilly.com
odewahn@oreilly.com
support@oreilly.com
2522@oreilly.com
pixel-1687721585523016-web-@oreilly.com
corporate@oreilly.com
2B@oreilly.com
bookquestions@oreilly.com
acmsales@oreilly.com
```

Wszystkie źródła dostępne w programie `emailharvester` zwracają jedynie około 20 adresów e-mail, z których większość nie wydaje się powiązana z konkretnymi osobami. Takie wyniki niekoniecznie okażą się użyteczne podczas przeprowadzania testów bezpieczeństwa. To może oznaczać konieczność zastosowania tradycyjnej operacji wyszukiwania, za pomocą interfejsu przeglądarki WWW witryn takich jak LinkedIn, zamiast korzystania z narzędzi Kali Linux.

Recon-NG

Program **Recon-NG** również służy do automatycznego zbierania informacji, ale jest na tyle rozbudowany, że zasługuje na osobny podrozdział. W rzeczywistości jest to platforma składająca się z różnych modułów. Jej przeznaczeniem jest przeprowadzanie rekonesansu, tj. wyszukiwanie w różnych źródłach informacji o firmach i innych obiektach ataku. Przeszukiwanie niektórych z nich wymaga zastosowania metod programistycznych. Dotyczy to między innymi serwisów X (dawniej Twitter), Instagram, Google, Bing i innych. Oprócz tego wiele modułów wymaga uzyskania klucza, który następnie wykorzysta w odwołaniach do interfejsów API. Tylko niektóre moduły, na przykład do przeszukiwania serwisów PGP i `whois`, nie wymagają klucza ani żadnej formy uwierzytelnienia. W większości przypadków jednak dostęp do źródła danych nie jest możliwy bez użycia klucza. Aby go uzyskać, trzeba utworzyć konto i potwierdzić swoją tożsamość. Na przykład w serwisie X trzeba podać numer telefonu komórkowego, który jest następnie weryfikowany. W ten sposób właściciele źródeł rejestrują, kto wysyła zapytania.

Większość modułów, które będą używane podczas rekonesansu, wymaga kluczy API. Jeżeli ponownie skoncentrujemy się na serwisie LinkedIn, Recon-NG użyje podczas wyszukiwania informacji w LinkedInie klucza API pochodzącego z silnika wyszukiwania Bing. Niestety pobranie klucza API do wyszukiwania za pomocą Binga oznacza konieczność pobrania zasobu Azure z serwisu Microsoftu. Zatem musisz mieć konto Azure i otrzymasz rachunek za wykorzystanie zasobów obliczeniowych niezbędnych do przeprowadzenia operacji wyszukiwania.

Warto wiedzieć, że istnieją sposoby na bezpłatne wyszukiwanie informacji. Listing 3.6 przedstawia pobranie modułów przeznaczonych do użycia z Recon-NG.

Listing 3.6. Instalowanie modułów wykorzystywanych przez program Recon-NG

```
[*] No modules enabled/installed.

[recon-ng][default] > marketplace search linkedin
[*] Searching module index for 'linkedin'...

  +-------------------------------------------------------------------------------+
  |   Path                      | Version |     Status     |  Updated   | D | K |
  +-------------------------------------------------------------------------------+
  | recon/companies-contacts/
    bing_linkedin_cache         | 1.0     | not installed | 2019-06-24 |   | * |
  | recon/profiles-contacts/
    bing_linkedin_contacts      | 1.2     | not installed | 2021-08-24 |   | * |
  +-------------------------------------------------------------------------------+

  D = Has dependencies. See info for details.
  K = Requires keys. See info for details.

[recon-ng][default] > marketplace install recon/profiles-contacts/bing_linkedin_contacts
[*] Module installed: recon/profiles-contacts/bing_linkedin_contacts
[*] Reloading modules...
[!] 'bing_api' key not set. bing_linkedin_contacts module will likely fail at runtime.
↪See 'keys add'.
```

Program Recon-NG używa kontekstu do ustalenia dostępnych poleceń oraz znaczenia danego kontekstu. Przed użyciem modułu trzeba go wczytać. Konieczne jest również posiadanie odpowiednich kluczy. W omawianym przykładzie skoncentruję się na module `recon/doma ↪ins-contacts/hunter_io`, ponieważ przedstawiony wcześniej skrypt również wymagał tego klucza API. W listingu 3.7 pokazałem kroki, które trzeba wykonać, aby można było używać tego modułu. Przede wszystkim moduł musi zostać zainstalowany, gdyż domyślnie nie są wczytywane żadne moduły. Następnie trzeba wczytać moduł i dopiero wtedy można przystąpić do jego użycia. Będzie to oznaczało przejście do przestrzeni kontekstu dla danego modułu. Wymagane jest zdefiniowanie zmiennej `SOURCE` (źródło), ponieważ będzie ona zawierała szukane informacje. W omawianym przykładzie jest to nazwa domeny.

Listing 3.7. Przykład użycia programu Recon-NG do wyszukiwania danych kontaktowych dla danej domeny

```
[recon-ng][default] > marketplace install recon/domains-contacts/hunter_io
[*] Module installed: recon/domains-contacts/hunter_io
[*] Reloading modules...
[recon-ng][default] > keys add hunter_io 4ee56b9bffee59a64fa003b7c36f05290ab5ad6c
[*] Key 'hunter_io' added.
[recon-ng][default] > modules load recon/domains-contacts/hunter_io
[recon-ng][default][hunter_io] > info

      Name: Hunter.io Email Address Harvester
    Author: Super Choque (@aplneto)
   Version: 1.3
      Keys: hunter_io

Description:
```

```
  Uses Hunter.io to find email addresses for given domains.

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  COUNT   10             yes       Limit the amount of results returned.
                                   (10 = Free Account)
  SOURCE  default        yes       source of input (see 'info' for details)

Source Options:
  default        SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL
  <string>       string representing a single input
  <path>         path to a file containing a list of inputs
  query <sql>    database query returning one column of inputs

[recon-ng][default][hunter_io] > options set SOURCE oreilly.com
SOURCE => oreilly.com
[recon-ng][default][hunter_io] > run
```

Podczas korzystania z modułów dane są zapisywane w bazie programu Recon-NG. Na przykład nazwiska osób i ich adresy e-mail zbierane przez moduł PGP są zapisywane w bazie kontaktów. Za pomocą polecenia `show` można wyświetlić listę wszystkich dostępnych wyników. Można również użyć modułu raportującego, który umożliwia pobranie dowolnych danych z bazy i zapisanie ich w plikach w formatach XML, HTML, CSV, JSON lub innym. Format zależy wyłącznie od wybranego modułu. Listing 3.8 przedstawia przykład użycia modułu JSON (ang. *JavaScript Object Notation* — zapis obiektów JavaScript). Za pomocą dostępnych opcji można wybierać tabele, z których mają być wyeksportowane dane, jak również wskazywać miejsce zapisu pliku. Po wybraniu opcji, które w poniższym przykładzie mają wartości domyślne, należy uruchomić moduł i wyeksportować dane.

Listing 3.8. Moduł raportujący programu Recon-NG

```
[recon-ng][default] > modules load reporting/json
[recon-ng][default][json] > info
      Name: JSON Report Generator
    Author: Paul (@PaulWebSec)
   Version: 1.0

Description:
  Creates a JSON report.

Options:
  Name      Current Value                  Required  Description
  --------  -------------                  --------  -----------
  FILENAME  /home/kilroy/.recon-ng/        yes       path and filename
            ↪workspaces/                             ↪for report output
            default/results.json
  TABLES    hosts, contacts, credentials   yes       comma delineated
                                                     ↪list of tables

[recon-ng][default][json] > run
[*] 223 records added to '/home/kilroy/.recon-ng/workspaces/default/results.json'.
[recon-ng][default][json] >
```

Program Recon-NG nie oferuje funkcjonalności przestrzeni roboczych, ale jeżeli jest wykorzystywany do wyszukiwania informacji o różnych firmach, można eksportować dane i usuwać je z bazy, co pozwoli uniknąć przecieków. W powyższym przykładzie za pomocą polecenia `delete contacts 1-27` usunąłem z bazy kontaktów rekordy o numerach od 1 do 27. Wcześniej musiałem wysłać zapytanie do bazy, aby wyświetlić wszystkie rekordy i sprawdzić, jakie mają numery. Wysyłanie zapytań jest równie łatwe, jak wyświetlanie kontaktów. Program Recon-NG oferuje mnóstwo funkcjonalności, które zmieniają się w miarę upływu czasu. Można się spodziewać, że w miarę pojawiania się nowych źródeł informacji i znajdowania przez programistów sposobów na pozyskiwanie z nich danych będą powstawać nowe moduły.

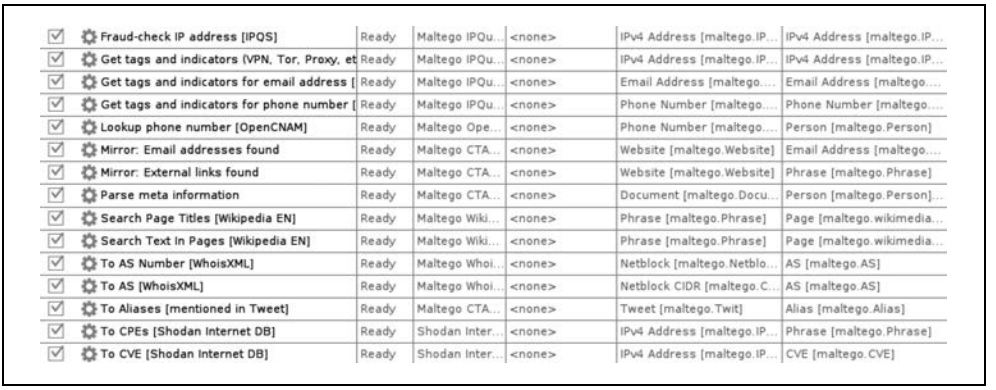
Maltego

Wyrosłem w czasach, gdy graficzny interfejs aplikacji był czymś wyjątkowym. Dlatego jestem entuzjastą wiersza poleceń. Oczywiście system Kali oferuje mnóstwo narzędzi uruchamianych w wierszu poleceń, ale ma również rozwiązania dla użytkowników preferujących graficzne interfejsy. Do tej pory poznałeś wiele narzędzi umożliwiających pozyskiwanie ogromnej ilości danych z różnego rodzaju źródeł. Jedną z rzeczy, których nie da się łatwo zrobić za ich pomocą, jest obraz zależności pomiędzy różnymi jednostkami informacji. Nie można również w szybki i prosty sposób przetwarzać dostępnych danych w celu uzyskania dodatkowych danych. Wyniki dostarczone przez program Recon-NG można wprawdzie przesłać do innego narzędzia lub modułu, ale lepiej byłoby móc wybrać po prostu żądany fragment danych i przetworzyć go za pomocą innego modułu.

W tym miejscu pojawia się program **Maltego**, wyposażony w graficzny interfejs i oferujący funkcjonalności podobne do tych, które miały opisane wcześniej narzędzia. Program wyróżnia się tym, że prezentuje zależności pomiędzy jednostkami danych w postaci czytelnego grafu. Jeżeli pobraliśmy odpowiednie elementy danych, możemy z nich pozyskiwać dodatkowe szczegółowe informacje i na ich podstawie generować następne jednostki.

Zanim zagłębimy się w szczegóły programu Maltego, musisz poznać stosowaną w nim terminologię, abyś wiedział, co ten program prezentuje. Wykorzystywane są w nim **transformaty**, czyli fragmenty kodu napisane w języku MSL (ang. *Maltego Scripting Language* — język skryptowy Maltego), które generują dane na podstawie innych danych. Załóżmy, że mamy nazwę hosta. Za pomocą transformaty można wygenerować nowe dane, tj. adres IP skojarzony z tą nazwą. Jak wspomniałem wcześniej, program Maltego prezentuje dane w postaci grafu, w którym węzłami są jednostki danych.

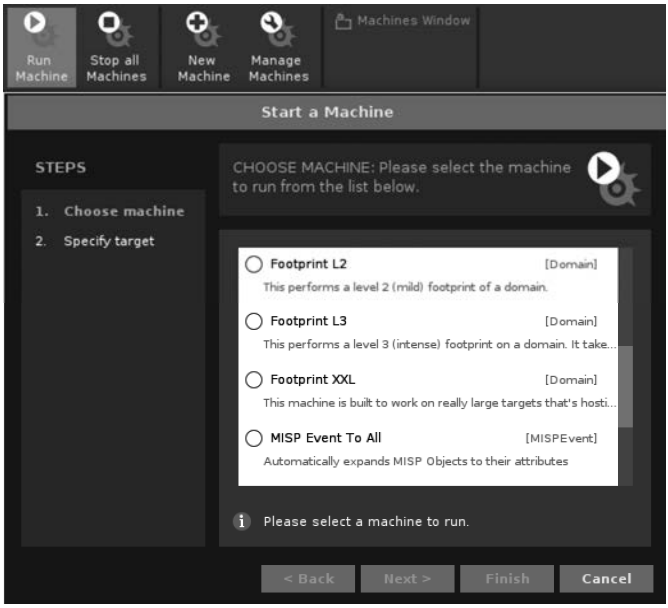
Będziemy korzystać z dostępnej w systemie Kali bezpłatnej wersji programu Maltego, w której można zainstalować ograniczoną liczbę transformat. Firma Paterva oferuje również płatną wersję programu udostępniającą znacznie więcej transformat pochodzących z różnych źródeł. Niemniej jednak będziemy mieć do dyspozycji sporo transformat, których listę przedstawia rysunek 3.2.



Fraud-check IP address [IPQS]	Ready	Maltego IPQu...	<none>	IPv4 Address [maltego.IP...	IPv4 Address [maltego.IP...
Get tags and indicators (VPN, Tor, Proxy, et	Ready	Maltego IPQu...	<none>	IPv4 Address [maltego.IP...	IPv4 Address [maltego.IP...
Get tags and indicators for email address [	Ready	Maltego IPQu...	<none>	Email Address [maltego....	Email Address [maltego....
Get tags and indicators for phone number [	Ready	Maltego IPQu...	<none>	Phone Number [maltego....	Phone Number [maltego....
Lookup phone number [OpenCNAM]	Ready	Maltego Ope...	<none>	Phone Number [maltego....	Person [maltego.Person]
Mirror: Email addresses found	Ready	Maltego CTA...	<none>	Website [maltego.Website]	Email Address [maltego....
Mirror: External links found	Ready	Maltego CTA...	<none>	Website [maltego.Website]	Phrase [maltego.Phrase]
Parse meta information	Ready	Maltego CTA...	<none>	Document [maltego.Docu...	Person [maltego.Person]...
Search Page Titles [Wikipedia EN]	Ready	Maltego Wiki...	<none>	Phrase [maltego.Phrase]	Page [maltego.wikimedia...
Search Text in Pages [Wikipedia EN]	Ready	Maltego Wiki...	<none>	Phrase [maltego.Phrase]	Page [maltego.wikimedia...
To AS Number [WhoisXML]	Ready	Maltego Whoi...	<none>	Netblock [maltego.Netblo...	AS [maltego.AS]
To AS [WhoisXML]	Ready	Maltego Whoi...	<none>	Netblock CIDR [maltego.C...	AS [maltego.AS]
To Aliases [mentioned in Tweet]	Ready	Maltego CTA...	<none>	Tweet [maltego.Twit]	Alias [maltego.Alias]
To CPEs [Shodan Internet DB]	Ready	Shodan Inter...	<none>	IPv4 Address [maltego.IP...	Phrase [maltego.Phrase]
To CVE [Shodan Internet DB]	Ready	Shodan Inter...	<none>	IPv4 Address [maltego.IP...	CVE [maltego.CVE]

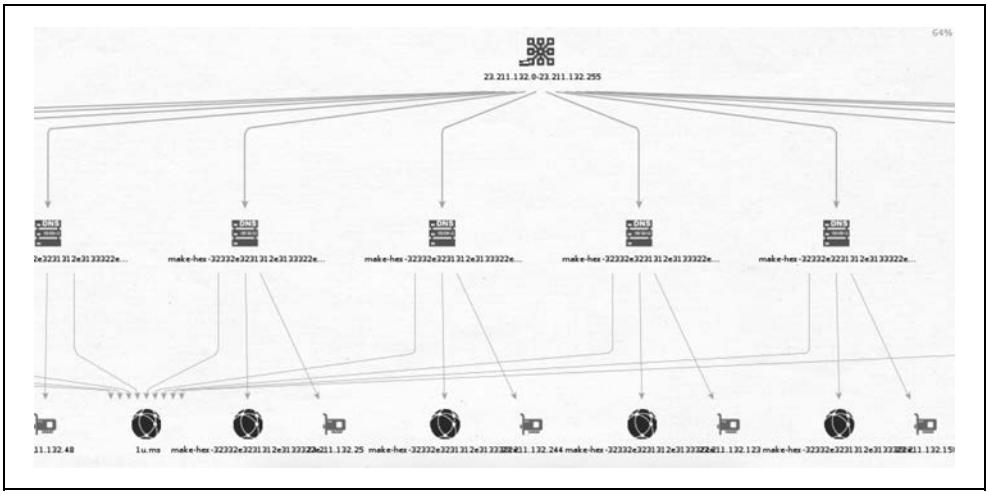
Rysunek 3.2. Transformaty dostępne w bezpłatnej wersji programu Maltego

Siłą programu Maltego są zainstalowane w nim transformaty. Nie trzeba jednak ręcznie stosować jednej transformaty po drugiej. Robi to tzw. **maszyna**. Maszynę tworzy się po to, aby stosować od razu serię transformat. Załóżmy, że chcemy zebrać informacje o wybranej firmie. Maszyna może wykonać tę pracę za nas, stosując transformaty odpytujące serwery DNS i wyszukujące połączenia pomiędzy systemami. Maszyna *Footprint L3* używa transformat przetwarzających rekordy z zadanej domeny, pozyskanych z serwera pocztowego i serwera DNS. Następnie znajduje adresy IP skojarzone z nazwami i szuka powiązanych z nimi innych adresów IP i nazw hostów. Aby uruchomić cały proces, wystarczy kliknąć przycisk *Run Machine*, wybrać żądaną maszynę i podać wymagane przez nią informacje. Rysunek 3.3 przedstawia okno dialogowe do uruchamiania maszyny. W górnej części okna znajduje się pasek z przyciskiem *Run Machine*.



Rysunek 3.3. Uruchamianie maszyny w programie Maltego

Maszyna w trakcie swego działania prosi o określenie, które elementy danych mają znaleźć się w grafie, a które mają zostać wykluczone. Na koniec wyświetla graf. Nie jest to jednak zwykły graf, tylko skierowany, ilustrujący zależności pomiędzy danymi. W jego środku znajduje się podana na początku nazwa domeny, a wokół niej rozmieszczone zostały różnego rodzaju dane. Ikona przy każdej jednostce danych oznacza jej typ. Na przykład ikona z interfejsem sieciowym reprezentuje adres IP. Inna ikona, przedstawiająca stos, reprezentuje w zależności od koloru rekord DNS lub MX. Na rysunku 3.4 znajduje się przykładowy graf.

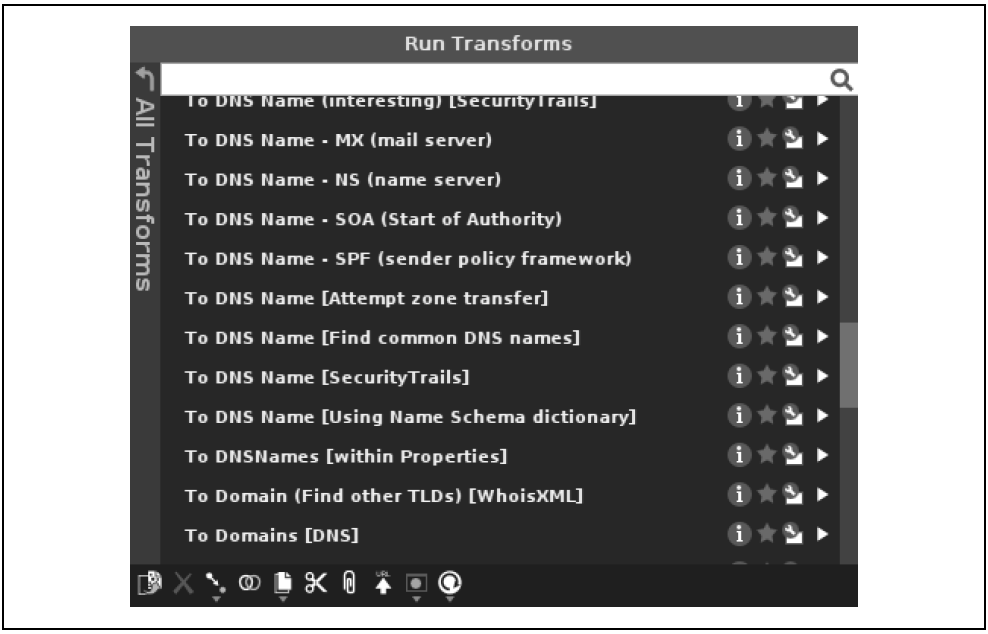


Rysunek 3.4. Graf skierowany utworzony w programie Maltego

Kiedy klikniemy jednostkę prawym przyciskiem myszy, pojawi się menu kontekstowe zawierające listę transformat, które możemy zastosować. Jeżeli dostępna jest nazwa hosta, ale brakuje jego adresu IP, można go poszukać, wybierając odpowiednią transformatę. Ponadto — jak pokazuje rysunek 3.5 — można pozyskiwać informacje z regionalnego rejestru internetowego przypisanego danej jednostce. Służy do tego celu transformata `whois` utworzona przez zespół ThreatMiner.

Po każdorazowym zastosowaniu transformaty graf się powiększa. Im więcej transformat jest dostępnych, tym więcej danych można uzyskać. Zebranie mnóstwa informacji na podstawie początkowej jednostki danych nie trwa długo. Informacje i zależności między nimi są prezentowane w postaci skierowanego grafu, a po kliknięciu dowolnej jednostki można łatwo uzyskiwać dodatkowe szczegółowe informacje, między innymi jednostki skojarzone w kierunku wychodzącym lub przychodzącym. Ponadto graf czytelnie przedstawia źródła, z których pochodzą dane.

Jeżeli należysz do tych użytkowników, którzy preferują graficzną wizualizację rozległych zależności, to na pewno polubisz program Maltego. Oczywiście są również inne sposoby pozyskiwania tych samych informacji, ale są one bardziej pracochłonne i wymagają intensywnego stukania w klawiaturę.



Rysunek 3.5. Transformaty do wykonywania operacji na danych

Rekonesans systemu DNS i usługa whois

Internet w rzeczywistości opiera się na systemie DNS. Gdyby go nie było, musielibyśmy zapamiętywać ogromne tabele adresów IP serwerów, włącznie z tymi, które nieustannie się zmieniają. Dlatego rola tego systemu jest bardzo ważna, a jego bezpieczeństwo jest kluczowe. Zanim powstał system DNS, powiązania adresów IP z nazwami były przechowywane w zwykłym pliku. Gdy do sieci był podłączany nowy komputer (pamiętaj, że było to w czasach, gdy były to duże, współdzielone przez wielu użytkowników komputery), plik trzeba było zmieniać i rozsyłać do wszystkich zainteresowanych. Ten mechanizm nie przetrwał długo i wkrótce narodził się system DNS. Przejął on zadanie przechowywania wszystkich powiązań między hostami i adresami IP.

Działanie systemu sprowadza się do przetwarzania adresów IP przypisanych firmom i instytucjom posiadającym domeny internetowe. W tym miejscu należy wspomnieć o rejestrach RIR (ang. *Regional Internet Registries*, regionalne rejestry internetowe). Podczas pozyskiwania informacji o obiekcie testów należy równolegle z przeprowadzaniem rekonesansu systemu DNS odpytywać rejestry RIR. Do tego celu można użyć na przykład narzędzia `whois`. Choć oba wymienione źródła informacji są przydatne, najpierw zajmiemy się systemem DNS, ponieważ niektóre uzyskiwane z niego dane można wykorzystywać podczas wysyłania zapytań do rejestrów RIR.

Rekonesans systemu DNS

System DNS ma hierarchiczną strukturę. Aby uzyskać z niego informacje, najczęściej wysyła się zapytanie do najbliższego serwera. Zazwyczaj jest to **serwer buforujący**. Nazwa ta oznacza, że serwer przechowuje w buforze zapytania i odpowiedzi na nie. Dzięki temu kolejne zapytania o te same dane są obsługiwane znacznie szybciej. Jeżeli serwer nie znajdzie w buforze żądanej nazwy komputera, zaczyna szukać tej informacji gdzie indziej. W tym celu wykorzystuje kolejne serwery aż do głównych, dla których lokalny serwer DNS powinien mieć adresy IP. W ten sposób uzyskuje adres serwera nazw domen najwyższego poziomu, który zawiera informacje dotyczące domen najwyższego poziomu takich jak *.net*, *.com*, *.org* i setki innych.

Ciągi FQDN (ang. *Fully Qualified Domain Name*, w pełni kwalifikowana nazwa domeny) zawierające nazwy domen (na przykład nazwa *www.helion.pl* składa się z nazwy hosta *www* i domeny *helion.pl*) należy czytać od końca. Pierwszy od prawej człon nazwy jest domeną TLD (ang. *Top Level Domain*, domena najwyższego poziomu). Informacje o domenach TLD są przechowywane w serwerach głównych (ang. *root servers*). Gdy najbliższy serwer DNS poszukuje nazwy *www.helion.pl*, komunikuje się najpierw z serwerem głównym obsługującym domenę TLD o nazwie *pl* w celu uzyskania od niego informacji o serwerze obsługującym domenę *helion.pl*. Proces wysyłania kolejnych zapytań, w którym serwer poszukujący informacji kontaktuje się bezpośrednio z każdym kolejnym serwerem, nosi nazwę **odpytywania iteracyjnego**. To przeciwieństwo zapytania rekurencyjnego, w którym serwery tymczasowe wykonują wybrane zapytania w imieniu systemu inicjującego.

Termin FQDN może być niezrozumiały, ponieważ samo pojęcie nazwy domeny jest dość trudne. Niekiedy nazwie domeny, na przykład *helion.pl*, jest przypisany ten sam adres co nazwie hosta, przykładowo serwera WWW (*www.helion.pl*), ale nie jest to regułą. Nazwa *helion.pl* jest nazwą domeny. Jeżeli serwer DNS zostanie odpowiednio skonfigurowany, można używać osobno na przykład nazw *www* lub *mail*. Aby móc jednoznacznie wskazywać komputer w danej domenie, stosuje się nazwę FQDN, składającą się z nazwy tego komputera (lub jego adresu IP) oraz nazwy jego domeny.

Kiedy serwer DNS nawiąże połączenie z głównym serwerem domeny *pl*, wysyła do niego zapytanie o serwer domeny *helion.pl*. Gdy dowie się, jaki to jest serwer, wysyła do niego zapytanie o nazwę hosta *www.helion.pl*. Odpytywany serwer jest autorytatywnym serwerem danej domeny. Informacja uzyskiwana z najbliższego serwera DNS nie jest odpowiedzią autorytatywną. Jej źródłem jest wprawdzie serwer autorytatywny, ale zanim dotrze ona do Twojego komputera, musi przejść przez lokalny serwer DNS, przez co przestaje być informacją autorytatywną. Dla każdej domeny są skonfigurowane serwery autorytatywne. Na przykład rozważmy domenę o nazwie *oreilly.com* — w jej przypadku zostało skonfigurowanych sześć serwerów autorytatywnych. Jeśli zapytasz inny serwer o informacje w domenie *oreilly.com*, otrzymasz odpowiedź nieautorytatywną.

Jedno z rozwiązań w zakresie pobrania adresu IP polega na wykorzystaniu narzędzia `host`. To prosty i łatwy w użyciu program. Nie dostarcza on zbyt wielu innych informacji ponad te, o które prosi użytkownik. Natomiast inne narzędzia zapewniają większą kontrolę nad serwerami, z których są pobierane informacje. Na listingu 3.9 pokazałem przykład użycia narzędzia `host` do sprawdzenia w pełni kwalifikowanej nazwy domeny *www.oreilly.com*.

Listing 3.9. Przykład użycia narzędzia host

```
┌──(kilroy@badmilo)-[~]
└─$ host www.oreilly.com
www.oreilly.com is an alias for www.oreilly.com.edgekey.net.
www.oreilly.com.edgekey.net is an alias for e4619.g.akamaiedge.net.
e4619.g.akamaiedge.net has address 104.104.104.60
```

Do pobierania informacji z serwerów DNS można używać jeszcze wielu innych serwerów. Te narzędzia oferują znacznie większą kontrolę, którą można wykorzystać w trakcie rekonesansu.

Narzędzia nslookup i dig

Jednym z narzędzi służących do odpytywania serwerów DNS jest `nslookup`. Wysyła ono zapytania do domyślnego serwera DNS lub do innego wskazanego w argumencie. Listing 3.10 przedstawia przykład odpytania lokalnego serwera. Jak widać, odpowiedź nie jest autorytatywna. Wynik zawiera również adres IP serwera DNS.

Listing 3.10. Przykład użycia narzędzia nslookup

```
┌──(kilroy2badmilo)-[~]
└─$ nslookup www.oreilly.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.oreilly.com canonical name = www.oreilly.com.edgekey.net.
www.oreilly.com.edgekey.net     canonical name = e4619.g.akamaiedge.net.
Name:   e4619.g.akamaiedge.net
Address: 104.104.104.60
```

W tym przypadku odpowiedzi udzielił najbliższy serwer DNS z zastrzeżeniem, że nie jest to odpowiedź autorytatywna. Odpowiedź zawiera aliasy i przypisany im adres IP. Aby uzyskać autorytatywną odpowiedź, trzeba odpytać autorytatywny serwer danej domeny. W tym celu należy użyć innego narzędzia. Jest to program `dig` zwracający rekord danych z serwera DNS. Listing 3.11 przedstawia przykład użycia tego programu.

Listing 3.11. Przykład użycia narzędzia dig

```
┌──(kilroy@badmilo)-[~]
└─$ dig ns oreilly.com

; <<>> DiG 9.18.13-1-Debian <<>> ns oreilly.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58242
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 13
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;oreilly.com.                   IN      NS

;; ANSWER SECTION:
oreilly.com.            3600    IN      NS      a3-67.akam.net.
oreilly.com.            3600    IN      NS      a1-225.akam.net.
oreilly.com.            3600    IN      NS      a4-64.akam.net.
oreilly.com.            3600    IN      NS      a20-66.akam.net.
oreilly.com.            3600    IN      NS      a16-65.akam.net.
oreilly.com.            3600    IN      NS      a13-64.akam.net.

;; ADDITIONAL SECTION:
a3-67.akam.net.         5997    IN      A       96.7.49.67
a3-67.akam.net.         59586   IN      AAAA    2600:1408:1c::43
a1-225.akam.net.        89047   IN      A       193.108.91.225
a1-225.akam.net.        89488   IN      AAAA    2600:1401:2::e1
a4-64.akam.net.         9596    IN      A       72.246.46.64
a4-64.akam.net.         13503   IN      AAAA    2600:1480:9000::40
a20-66.akam.net.        7359    IN      A       95.100.175.66
a20-66.akam.net.        8658    IN      AAAA    2a02:26f0:67::42
a16-65.akam.net.        6725    IN      A       23.211.132.65
a16-65.akam.net.        59139   IN      AAAA    2600:1406:1b::41
a13-64.akam.net.        88094   IN      A       2.22.230.64
a13-64.akam.net.        55852   IN      AAAA    2600:1480:800::40

;; Query time: 56 msec
;; SERVER: 192.168.1.1#53(192.168.1.1) (UDP)
;; WHEN: Mon Jun 26 18:31:23 EDT 2023
;; MSG SIZE  rcvd: 436
```

Na dobrą sprawę można korzystać wyłącznie z programu `dig`. Niemiej jednak wróćmy do programu `nslookup`, aby dokładnie poznać różnice w uzyskiwanych wynikach. Tym razem określimy serwer DNS, który ma zostać odpytany. Będzie to jeden z serwerów wymienionych w listingu 3.10. Jego nazwę należy wpisać na końcu użytego wcześniej polecenia. Listing 3.12 przedstawia wynik.

Listing 3.12. Przykład użycia narzędzia nslookup z nazwą serwera DNS

```
┌──(kilroy@badmilo)-[~]
└─$ nslookup www.oreilly.com a3-67.akam.net
Server:         a3-67.akam.net
Address:        2600:1408:1c::43#53

www.oreilly.com canonical name = www.oreilly.com.edgekey.net.
www.oreilly.com.edgekey.net     canonical name = e4619.g.akamaiedge.net.
┌──(kilroy@badmilo)-[~]
└─$ nslookup e4619.g.akamaiedge.net.
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   e4619.g.akamaiedge.net
Address: 104.104.104.60
```

Wynikiem pierwotnego zapytania jest alias, więc aby otrzymać adres IP, konieczne jest jego wyszukanie w ostatecznej wartości w pełni kwalifikowanej nazwy domeny. W tym miejscu można przeprowadzić dodatkowe operacje, które miałyby na celu ustalenie serwera autorytatywnego dla domeny *g.akamaiedge.net*, ale w omawianym przykładzie to prawdopodobnie spowoduje otrzymanie dokładnie takiej samej odpowiedzi jak w przypadku serwera nieautorytatywnego. Zwróć uwagę na to, że *g.akamaiedge.net* obejmuje nie tylko domenę *akamaiedge.net*, ale również poddomenę *g*. Zbiór znaków, które znajdują się przed pierwszą kropką od lewej strony, to nazwa hosta. Natomiast wszystko pozostałe to domena. Zatem w omawianym przykładzie *g* jest poddomeną dla *akamaiedge.net*. Jeżeli mamy adres IP w pełni kwalifikowanej nazwy domeny, możemy go wykorzystać do pozyskiwania innych adresów należących do tego samego obiektu testów. W tym celu musimy przenieść się na wyższy poziom hierarchii systemu DNS. Dlatego użyjemy programu `whois` do uzyskania bardziej szczegółowych informacji o obiekcie.

Automatyzacja rekonesansu systemu DNS

Narzędzia takie jak `host` i `nslookup` dostarczają wielu szczegółowych informacji, ale ich zbieranie zajmuje dużo czasu. Zamiast ręcznie uruchamiać poszczególne programy, można użyć innego narzędzia dostarczającego obszernych informacji. Jedną z cech tego typu narzędzi jest możliwość wykonywania **transferu stref**. Termin ten oznacza pobieranie z serwera DNS wszystkich rekordów przypisanych danej strefie. **Strefa** z kolei oznacza zbiór powiązanych ze sobą informacji w serwerze DNS. Dla domeny *oreilly.com* jest prawdopodobnie utworzona osobna strefa zawierająca wszystkie należące do tej domeny rekordy, na przykład z adresem serwera WWW i serwera pocztowego.

Ponieważ transfer strefy może być skutecznym sposobem pozyskania informacji o firmie, zazwyczaj jego przeprowadzenie jest blokowane. Czasami transfer wykonuje się w celu zsynchronizowania baz danych na serwerach podstawowym i zapasowym. Jednak w większości przypadków nie będziesz w stanie wykonać transferu strefy, chyba że uzyskasz specjalne uprawnienia do wykonania tej operacji. Nie martw się jednak. Niektóre narzędzia wymagają wprawdzie uprawnień do wykonywania transferu, jednak są też inne, które dostarczają szczegółowych informacji o hostach. Jednym z nich jest program `dnsrecon`, który nie tylko transferuje strefy, lecz także testuje hosty z zadanej listy. Aby go użyć, należy utworzyć plik tekstowy z nazwami hostów, które będą uzupełniane o zadaną nazwę domeny. Mogą to być popularne nazwy opisujące różne usługi, na przykład *www*, *mail*, *smtp* lub *FTP*. Dostępny jest również gotowy plik zawierający ponad 1900 nazw. Dzięki liście da się wykrywać za pomocą programu `dnsrecon` serwery, o których istnieniu można nawet nie wiedzieć.

W powyższym opisie przyjęte zostało założenie, że nazwy hostów posiadanych przez obiekt testów zostały zapisane w zewnętrznym serwerze DNS. Serwery DNS mają tę ważną cechę, że tworzą wprawdzie hierarchiczną strukturę, ale są **rozdzielne**. Dlatego niektóre instytucje korzystają z tzw. **podzielonego systemu DNS**. Oznacza to, że wewnętrzny system instytucji odwołuje się do autorytatywnych serwerów DNS danej domeny. Wewnętrzny system zawiera informacje o hostach, których nazw nie mogą znać zewnętrzne podmioty. Ponieważ główne

serwery DNS nie „wiedzą” o istnieniu takich wewnętrznych serwerów, osoby z zewnątrz nie mogą z nich korzystać. Aby pozyskać informacje z takiego serwera, wymagane jest połączenie bezpośrednio z nim, co zazwyczaj nie jest możliwe dla osób spoza instytucji.

Mimo wszystko nie powinieneś zniechęcać się do programu `dnsrecon`, ponieważ i tak dostarcza on mnóstwa przydatnych informacji. Listing 3.13 przedstawia fragment wyników uzyskanych dla mojej domeny, w której korzystam z usługi Google Apps for Business. Widoczny jest w nim rekord TXT zawierający informację, że domena została zarejestrowana na mnie i ja zarządzam rekordami w serwerze DNS. W przypadku domen utrzymywanych przez zewnętrznych dostawców ten właśnie rekord może być dostępny. Z wyników można również się dowiedzieć, jak nazywa się serwer domeny. Jest to tylko fragment danych, których to narzędzie dostarcza bardzo dużo. Aby uzyskać przedstawiony wynik, użyłem polecenia `dnsrecon -d cloudroy.com -D /usr/share/dnsrecon/namelist.txt`.

Listing 3.13. Przykład użycia narzędzia dnsrecon do pozyskiwania informacji z systemu DNS

```
┌──(kilroy@badmilo)-[~]
└─$ dnsrecon -d cloudroy.com -D /usr/share/dnsrecon/namelist.txt
[*] std: Performing General Enumeration against: cloudroy.com...
[!] Wildcard resolution is enabled on this domain
[!] It is resolving to 208.91.197.39
[!] All queries will resolve to this list of addresses!!
[*] DNSSEC is configured for cloudroy.com
[*] DNSKEYs:
[*]     NSEC3 KSk RSASHA256 03010001930e1f6af3a8c51cfa966fcf
        b1694c4af2533242a0797c55edb22276 a870275ca3a5dc330e48b7a46212e3a1
        4fdf05e1f7fdee5c0b8ef75be967b3ec da1563e3b7087beee282ce007e5e52c9
        fdd41dd7a3208f798244e906a34c0409 4691d0dd59d463e0fd051e4a9156657d
        201f21ebd48836302a0e08d907320702 376d462806d3792bad565ab8bf8718dc
        297b64f585f797a3de682051f8fe07e2 9964c86cdcba96d0827b534c575a0c1d
        b3baf3e3644058886f481e7452b7477a 17371d761126a9aa43218abfa1d754f5
        937fcecf865950e1dd316a51c503d0da 1e0a6e2f9ccb8d52740a4e19df7c308a
        d96e12e03989f05a63d5803e6f20da49 e79bd583
[*]     NSEC3 ZSK RSASHA256 03010001b8b61cdba5b05ba524f92a36
        c4a46d8a925fe65f0db41c3de586956b 03d9f37a82b5c78095bd6651c22c2f0d
        c44be348f04b31e33175fbc7c1dac2b3 3bf1e520d8bc40fed15faef8ce162537
        f2ea00bb587f2353a5586285cd619a9b 65545fc0db75dd8a3e5e5bba9de57cb3
        02adf14002dc005539a1139b366ae52a 1ba30a05
[*]      SOA ns-cloud-a1.googledomains.com 216.239.32.106
[*]      SOA ns-cloud-a1.googledomains.com 2001:4860:4802:32::6a
[*]      NS ns-cloud-a1.googledomains.com 216.239.32.106
[*]      NS ns-cloud-a1.googledomains.com 2001:4860:4802:32::6a
[*]      NS ns-cloud-a2.googledomains.com 216.239.34.106
[*]      NS ns-cloud-a2.googledomains.com 2001:4860:4802:34::6a
[*]      NS ns-cloud-a3.googledomains.com 216.239.36.106
[*]      NS ns-cloud-a3.googledomains.com 2001:4860:4802:36::6a
[*]      NS ns-cloud-a4.googledomains.com 216.239.38.106
[*]      NS ns-cloud-a4.googledomains.com 2001:4860:4802:38::6a
[*]      MX alt3.aspmx.l.google.com 142.250.27.27
[*]      MX alt4.aspmx.l.google.com 142.250.153.26
[*]      MX aspmx.l.google.com 172.253.122.26
[*]      MX alt1.aspmx.l.google.com 209.85.202.26
[*]      MX alt2.aspmx.l.google.com 64.233.184.27
[*]      MX alt3.aspmx.l.google.com 2a00:1450:4025:401::1a
[*]      MX alt4.aspmx.l.google.com 2a00:1450:4013:c16::1a
```

```
[*]      MX aspmx.l.google.com 2607:f8b0:4004:c08::1a
[*]      MX alt1.aspmx.l.google.com 2a00:1450:400b:c00::1b
[*]      MX alt2.aspmx.l.google.com 2a00:1450:400c:c0b::1b
[*]      A cloudroy.com 208.91.197.39
[*]      TXT cloudroy.com google-site-verification=
         ↪rq3wZzkl6pdKp1nwWX_BItql6r1qKt34QmMcqE8jqCg
[*]      TXT cloudroy.com v=spf1 include:_spf.google.com ~all
[*] Enumerating SRV Records
[+] 0 Records Found
```

W mojej domenie usługi są świadczone przez Google, o czym jasno informują rekordy MX oraz rekord TXT. Nie oznacza to jednak, że znalezienie rekordu TXT kończy pracę. Czasami instytucje zmieniają operatorów hostingowych albo nie korzystają z usług, które nie wymagają rekordów TXT. Często zdarza się, że rekord, który przestaje być wykorzystywany w instytucji, pozostaje w serwerze, ponieważ nie jest to nic złego. Jednak informacja, że jakieś usługi były kiedyś wykorzystywane, może być cenna. Dlatego podczas testów warto jest korzystać z narzędzi takich jak `dnsrecon` w celu pozyskiwania jak największej ilości informacji.

Dane wyjściowe zamieszczone na listingu 3.13 pokazują, że w przypadku tej domeny są zastosowane zabezpieczenia dla poczty elektronicznej, co może okazać się jednym z najłatwiejszych sposobów identyfikowania najczęściej wybieranych zabezpieczeń dla poczty. Jednym z tych typowych zabezpieczeń jest Sender Policy Framework (SPF). Widać to doskonale w drugim rekordzie TXT, który wskazuje na użycie SPF w wersji 1. Zwróć uwagę na odwołanie do *spf.google.com* w zakresie obsługi SPF. Można również zobaczyć, że w analizowanej domenie obowiązują zabezpieczenia DNS, które obejmują także klucze dla danej domeny.

Rejestry RIR

Internet ma hierarchiczną strukturę. Wszystkimi stosowanymi w nim numerami, takimi jak zarejestrowane numery portów, bloki adresów IP i numery AS (ang. *autonomous system*, system autonomiczny), zarządza instytucja ICANN (ang. *Internet Corporation for Assigned Names and Numbers* — Internetowa Korporacja ds. Nadawanych Nazw i Numerów). ICANN z kolei deleguje zarządzanie częścią numerów do rejestrów RIR obsługujących różne części świata. Obecnie działają następujące rejestry:

- AfriNIC (ang. *African Network Information Center* — Afrykańskie Centrum Informacji Sieciowej) obejmujący Afrykę.
- ARIN (ang. *American Registry for Internet Numbers* — Amerykański Rejestr Numerów Internetowych) obejmujący Amerykę Północną, Antarktykę i część Karaibów.
- APNIC (ang. *Asia Pacific Network Information Centre* — Centrum Informacyjne Sieci Azji i Pacyfiku) obejmujący Azję, Australię, Nową Zelandię i sąsiednie kraje.
- LACNIC (ang. *Latin America and Caribbean Network Information Centre* — Centrum Informacyjne Sieci Ameryki Łacińskiej i Karaibów) obejmujący Amerykę Łacińską i część Karaibów.
- RIPE NCC (fr./ang. *Réseaux IP Européens Network Coordination Centre* — Centrum Koordynacyjne Europejskiej Sieci IP) obejmujący Europę, Rosję, Środkowy Wschód i centralną Azję.

Rejestry RIR zarządzają adresami IP i numerami AS. Numery AS są wykorzystywane w instytucjach do kierowania transmisją danych przez sieć. Są one przypisywane operatorom i instytucjom posiadającym duże sieci i wykorzystywane w protokole BGP (Border Gateway Protocol — protokół bram granicznych), powszechnie stosowanym w internecie. Wewnątrz instytucji stosowane są inne protokoły, na przykład OSPF (ang. *Open Shortest Path First*, najpierw najkrótsza ścieżka), natomiast BGP jest wykorzystywany do przesyłania danych pomiędzy sieciami z różnymi numerami AS.

whois

Do pozyskiwania informacji z dowolnego rejestru RIR służy narzędzie `whois`. Jest to program stosowany w wierszu poleceń, dostępny we wszystkich dystrybucjach systemu Linux. Za jego pomocą można sprawdzać, kto jest właścicielem zadanej puli adresów. Listing 3.14 przedstawia przykład zapytania o właściciela puli adresów 8.9.10.0. Wynik zawiera informację o instytucji, której został przydzielony podany blok adresów. W tym przypadku jest on bardzo duży. Tej wielkości bloki posiadają operatorzy lub instytucje, którym były przydzielane pierwsze adresy IP.

Listing 3.14. Przykład użycia narzędzia whois z blokiem adresów

```
┌──(kilroy@badmilo)-[~]
└─$ whois 8.9.10.0

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
#

NetRange:       8.0.0.0 - 8.127.255.255
CIDR:           8.0.0.0/9
NetName:        LVLT-ORG-8-8
NetHandle:      NET-8-0-0-0-1
Parent:         NET8 (NET-8-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Level 3 Parent, LLC (LPL-141)
RegDate:        1992-12-01
Updated:        2018-04-23
Ref:            https://rdap.arin.net/registry/ip/8.0.0.0

OrgName:        Level 3 Parent, LLC
OrgId:          LPL-141
Address:        100 CenturyLink Drive
City:           Monroe
StateProv:      LA
PostalCode:     71203
Country:        US
```

```
RegDate:        2018-02-06
Updated:        2023-04-07
Comment:        USAGE OF IP SPACE MUST COMPLY WITH OUR ACCEPTABLE USE POLICY:
Comment:        https://www.lumen.com/en-us/about/legal/acceptable-use-policy.html
```

Jeżeli wskazana pula adresów jest częścią większego bloku, narzędzie `whois` wyświetli nazwy właścicieli obu bloków. Przeanalizujmy fragment zakresu adresów 8.0.0.0 – 8.255.255.255. Listing 3.15 przedstawia wynik uzyskany dla jednego z podzakresów. Jak widać, należy on do Google. Jednak przed przedstawionym niżej wynikiem znajduje się zawartość poprzedniego listingu, w którym widoczna jest nazwa Level 3 Parent właściciela całego bloku rozpoczynającego się od cyfry 8.

Listing 3.15. Przykład użycia narzędzia whois z mniejszym blokiem adresów

```
NetRange:       8.8.8.0 - 8.8.8.255
CIDR:           8.8.8.0/24
NetName:        LVLT-GOGL-8-8-8
NetHandle:      NET-8-8-8-0-1
Parent:         LVLT-ORG-8-8 (NET-8-0-0-0-1)
NetType:        Reallocated
OriginAS:
Organization:   Google LLC (GOGL)
RegDate:        2014-03-14
Updated:        2014-03-14
Ref:            https://rdap.arin.net/registry/ip/8.8.8.0

OrgName:        Google LLC
OrgId:          GOGL
Address:        1600 Amphitheatre Parkway
City:           Mountain View
StateProv:      CA
PostalCode:     94043
Country:        US
RegDate:        2000-03-30
Updated:        2019-10-31
Comment:        Please note that the recommended way to file abuse complaints are
                ↪located in the following links.
Comment:
Comment:        To report abuse and illegal activity: https://www.google.com/contact/
Comment:
Comment:        For legal requests: http://support.google.com/legal
Comment:
Comment:        Regards,
Comment:        The Google Team
Ref:            https://rdap.arin.net/registry/entity/GOGL

OrgTechHandle: ZG39-ARIN
OrgTechName:   Google LLC
OrgTechPhone:  +1-650-253-0000
OrgTechEmail:  arin-contact@google.com
OrgTechRef:    https://rdap.arin.net/registry/entity/ZG39-ARIN

OrgAbuseHandle: ABUSE5250-ARIN
OrgAbuseName:   Abuse
OrgAbusePhone:  +1-650-253-0000
OrgAbuseEmail:  network-abuse@google.com
OrgAbuseRef:    https://rdap.arin.net/registry/entity/ABUSE5250-ARIN
```

W ten sposób można na podstawie chociażby adresu IP serwera WWW lub serwera pocztowego dowiedzieć się, kto jest właścicielem całej puli adresów. Często blok zawierający podany adres IP, na przykład serwera WWW wydawnictwa Helion, należy do operatora telekomunikacyjnego i nie można poznać adresów innych potencjalnych obiektów testów. Jeżeli jednak blok należy do jednej instytucji, można poznać wiele adresów. Taki blok jeszcze przyda się później, podczas bardziej aktywnego rekonesansu. Za pomocą narzędzi `dig` i `nslookup` da się także określać nazwy przypisane znalezionym adresom IP.

Aby można było określić nazwę przypisaną adresowi IP, w serwerze DNS musi być dla danej instytucji zdefiniowana strefa odwrotna. W tej strefie dla każdego adresu IP musi być zdefiniowany rekord PTR (ang. *pointer*, wskaźnik). Pamiętaj jednak, że nie zawsze istnieją zależności pomiędzy strefą podstawową i odwrotną. Jeżeli na przykład nazwa *www.foo.com* jest skojarzona z adresem 1.2.3.42, nie oznacza to, że adres 1.2.3.42 jest skojarzony z nazwą *www.foo.com*. Adresy IP mogą być przypisywane systemom o różnym przeznaczeniu, o nazwach opisujących ich funkcje.

Rekonesans pasywny

Często podczas rekonesansu trzeba szperać wokół infrastruktury obiektu testów. Nie oznacza to jednak aktywnego sondowania jego sieci. Aktywności takie jak skanowanie portów, o czym będzie mowa później, czynią wiele hałasu i zwracają uwagę administratorów, co jest niepożądanym zjawiskiem, gdy przymierzamy się do prawdziwego ataku. Informacje można zbierać w sposób pasywny poprzez zwykłe korzystanie z udostępnionych systemów. Można na przykład przeglądać stronę WWW obiektu i zbierać dostępne na niej informacje. Oprócz tego można korzystać z programu `p0f`.

Program `p0f` obserwuje przesyłane pakiety i wyodrębnia z nich potencjalnie interesujące informacje. Są to dane zapisane w nagłówkach pakietów, między innymi adresy oraz porty źródłowe i docelowe. Listing 3.16 prezentuje informacje, jakie program `p0f` uzyskał o serwerach WWW i systemach operacyjnych. Pierwszy blok przedstawia zapytanie HTTP zawierające szczegółowe dane klienta, między innymi dane hosta i agenta. Informacje te zostały pozyskane z nagłówków HTTP. Jest to możliwe, ponieważ komunikacja nie jest szyfrowana. Program `p0f` nie potrafi wyodrębniać danych z zaszyfrowanych strumieni komunikacji.

Listing 3.16. Wynik użycia programu p0f

```
.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (syn) ]-
|
| client   = 192.168.1.253/54072
| os       = Linux 2.2.x-3.x
| dist     = 0
| params   = generic
| raw_sig  = 4:64+0:0:1460:mss*44,7:mss,sok,ts,nop,ws:df,id+:0
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (syn+ack) ]-
|
```

```
| server   = 192.168.1.15/80
| os       = ???
| dist     = 0
| params   = none
| raw_sig  = 4:64+0:0:1460:mss*45,7:mss,sok,ts,nop,ws:df:0
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (mtu) ]-
|
| server   = 192.168.1.15/80
| link     = Ethernet or modem
| raw_mtu  = 1500
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (uptime) ]-
|
| client   = 192.168.1.253/54072
| uptime   = 31 days 10 hrs 49 min (modulo 49 days)
| raw_freq = 1001.18 Hz
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (uptime) ]-
|
| server   = 192.168.1.15/80
| uptime   = 14 days 3 hrs 45 min (modulo 49 days)
| raw_freq = 1000.11 Hz
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (http request) ]-
|
| client   = 192.168.1.253/54072
| app      = ???
| lang     = none
| params   = anonymous
| raw_sig  = 1:Host:User-Agent,Connection,Accept,Accept-Encoding,Accept-Language,
  ↪Accept-Charset,Keep-Alive:
|
`----

.-[ 192.168.1.253/54072 -> 192.168.1.15/80 (http response) ]-
|
| server   = 192.168.1.15/80
| app      = Apache 2.x
| lang     = none
| params   = none
| raw_sig  = 1:Date,Server,?Set-Cookie,?Set-Cookie,?Expires,?Cache-Control,
  ↪?Pragma,?Location,?Content-Length,Content-Type:Connection,Keep-Alive,Accept-
  ↪Ranges:Apache/2.4.55 (Ubuntu)
|
`----
```

Jednym z warunków korzystania z programu p0f jest posiadanie dostępu do pakietów przesyłanych przez obserwowany system. Ponadto konieczne jest komunikowanie się z systemem objętym pasywnym rekonesansem. Ponieważ są to publicznie dostępne serwery, jest mało prawdopodobne, że zwrócisz na siebie uwagę, a sam system nie będzie „wiedział", że używasz programu p0f, ponieważ nie wchodzi on w żadne aktywne interakcje z systemem w celu zebrania o nim szczegółowych informacji. Korzysta tylko z tego, co udostępniają usługi.

Najlepszym miejscem do pozyskiwania informacji jest lokalny koniec połączenia, dlatego że na podstawie adresów MAC można dowiedzieć się szczegółów o producentach komunikujących się urządzeń i ich typach. Tak jak w przypadku innych programów przechwytujących pakiety trzeba uzyskać dostęp do danych, które nie są wysyłane do Twojego systemu. W tym celu należy użyć huba, przełącznika z portem kopiującym pakiety lub podszyć się pod inne urządzenie. Adres MAC znajduje się w nagłówku dodawanym w warstwie danych (warstwie drugiej). Nagłówek ten jest modyfikowany, gdy pakiet przechodzi przez urządzenie obsługujące warstwę sieciową (trzecią), na przykład router.

Choć informacje, jakie można pozyskać, wykonując pasywny rekonesans za pomocą takich narzędzi jak p0f, są ograniczone do tego, co udostępnia dana usługa lub dany system, to w ten sposób oszczędza się mnóstwo manualnej pracy, jaką należało wykonać w celu uzyskania informacji o wymaganym stopniu szczegółowości. Największą zaletą programu p0f jest szybkie wyodrębnianie danych bez angażowania użytkownika i bez aktywnego sondowania systemu obiektu. Dzięki temu, stosując ten program, nie zwrócisz na siebie uwagi systemu monitorującego ani zespołu administratorów w obiekcie testów.

Skanowanie portów

Gdy już zbierzesz wszystkie informacje, które da się uzyskać bez aktywnego i hałaśliwego sondowania sieci obiektu, możesz zacząć robić szum, skanując porty. Zazwyczaj używa się do tego celu skanerów portów. Samo skanowanie nie musi być jednak hałaśliwe i generować dużego ruchu w sieci. Polega ono na wykorzystaniu protokołów sieciowych i uzyskaniu informacji o otwartych portach i aplikacjach działających w testowanym systemie. Otwarte porty mogą bardzo wiele powiedzieć o tych aplikacjach. Ostatecznym celem testów jest znalezienie wejść do systemu. Takimi wejściami mogą być otwarte porty.

Otwarty port jest to port, na którym aplikacja nasłuchuje zapytań wysyłanych przez klientów. Jeżeli żadna aplikacja niczego nie nasłuchuje, wszystkie porty są zamknięte. Porty, które nas interesują, znajdują się w warstwie transportowej. Aplikacja w zależności od potrzeb wykorzystuje protokół TCP lub UDP. Wspólną cechą obu protokołów są porty, które mają swoje numery od 0 do 65 535 (w sumie jest to 65 356 portów).

Skanowanie nie daje odpowiedzi, które porty są otwarte po stronie klienta. Na przykład nie można określić, czy są nawiązane połączenia z serwerem WWW, serwerem pocztowym i innymi usługami. Można określić jedynie porty otwarte do nasłuchu. Jeżeli komputer klienta nawiąże połączenie z innym systemem, jego port nie będzie otwarty w trybie nasłuchu. Komunikację pomiędzy klientem a serwerem definiują cztery parametry: para adresów IP (źródłowy i docelowy) oraz para portów.

Ponieważ wymienione wcześniej protokoły transportowe różnią się między sobą, skanowanie wykorzystywanych przez nie portów również przebiega inaczej. Zadanie polega oczywiście na znalezieniu otwartych portów, ale w tym celu trzeba stosować różne metody. System Kali jest wyposażony w dwa narzędzia. Nieformalnym standardem jest program `nmap`, więc zajmiemy się nim w pierwszej kolejności. Później przyjrzymy się narzędziom przeznaczonym do szybkiego skanowania dużych sieci.

Skanowanie portów TCP

Protokół TCP jest protokołem połączeniowym, tzn. obie komunikujące się strony kontrolują połączenie. Dzięki temu uzyskuje się gwarantowaną transmisję danych. Jeżeli coś złego stanie się z siecią łączącą oba urządzenia, żadna ze stron nie może zagwarantować przesłania danych, ale obie dowiadują się o problemie. Ponadto jeśli jedna ze stron nie odbierze danych, to druga też się o tym dowiaduje.

Podczas nawiązywania połączenia TCP dokonywany jest tzw. **potrójny uścisk dłoni** (ang. *three-way handshake*). Skanowanie i wykrywanie otwartych portów polega właśnie na wymienianiu takich uścisków. Najpierw klient wysyła komunikat SYN na zadany port serwera. Jeżeli port jest otwarty, serwer odpowiada komunikatem z flagami SYN i ACK. W przeciwnym razie wysyła komunikat RST (ang. *reset*) informujący klienta, aby nie wysyłał więcej komunikatów na ten port. W ten sposób nadawca dowiaduje się, że dany port nie jest otwarty.

Przeszkodami w skanowaniu portów są zapory sieciowe i mechanizmy chroniące porty. Zapora lub lista dostępu może nie przepuścić komunikatu. W takim przypadku nadawca znajduje się w stanie niepewności, ponieważ brak odpowiedzi z serwera nie oznacza, że port jest zamknięty.

Skanując porty, należy ponadto pamiętać, że nagłówki datagramów TCP zawierają kilka innych flag oprócz SYN i ACK. Oznacza to, że można wysyłać do systemu komunikaty z różnymi flagami i obserwować jego reakcje. Różne systemy reagują w różny sposób na różne ustawienia flag.

Skanowanie portów UDP

Protokół UDP jest prosty. Nie są w nim nawiązywane połączenia, nie ma też gwarancji przesyłania danych ani wzajemnego powiadamiania stron. Dlatego skanowanie portów UDP jest trudniejsze. Zważywszy na prostotę protokołu może to wydawać się sprzecznością.

Protokół TCP definiuje interakcje między stronami. Klient musi wysłać komunikat z ustawioną w nagłówku flagą SYN. Gdy serwer odbierze komunikat, odsyła odpowiedź z ustawionymi flagami SYN i ACK. Następnie klient odpowiada komunikatem ACK. W ten sposób obie strony dowiadują się o wzajemnym istnieniu. Klient „wie”, że serwer reaguje, ponieważ otrzymał od niego komunikat z flagami SYN i ACK. Z kolei serwer „wie”, że nikt się nie podszył pod klienta, ponieważ otrzymał od niego komunikat ACK.

Interakcje w protokole UDP nie są określone, a w nagłówku nie ma żadnych pól zawierających informacje o stanie połączenia. Gdy klient wyśle komunikat do serwera, odesłanie odpowiedzi zależy wyłącznie od działającej na nim aplikacji. Ponieważ nie ma tu flag SYN i ACK, klient nie ma możliwości, aby „dowiedzieć się", czy port na serwerze jest otwarty czy zamknięty. Brak odpowiedzi może również oznaczać, że serwer „nie rozumie" komunikatu klienta albo że aplikacja uległa awarii. Podczas skanowania portów UDP nie sposób określić, czy brak odpowiedzi z serwera oznacza, że dany port jest zamknięty. Dlatego program skanujący porty zazwyczaj wysyła komunikaty wielokrotnie. Ponieważ protokół UDP ma z reguły niższy priorytet niż TCP, przesłanie komunikatu w obie strony może zająć nieco więcej czasu. Dlatego programy skanujące zazwyczaj z niewielkim opóźnieniem ponownie wysyłają komunikat. Robią tak kilka razy, aby uzyskać pewność, że dany port jest niedostępny. Jest to taka sama reakcja jak w przypadku braku odpowiedzi podczas skanowania portów TCP. Zapora może blokować komunikaty. Program skanujący nie otrzymuje komunikatu RST ani ICMP, więc przyjmuje, że wysłany przez niego komunikat został utracony, i ponawia próbę. Wielokrotne wysyłanie komunikatów jest czasochłonne, szczególnie jeżeli trzeba przeskanować ponad 65 000 portów. Do tego każdy komunikat trzeba wysłać kilka razy. Dlatego złożoność procesu skanowania portów UDP wynika z niepewności będącej skutkiem braku odpowiedzi.

Skanowanie portów za pomocą programu nmap

Dzisiaj praktycznie jedynym programem do skanowania portów, który zresztą na samym początku zyskał popularność, jest `nmap`. Narzędzie to istnieje już ponad 20 lat i stało się tak ważne, że funkcje niektórych jego argumentów zostały wykorzystane w innych programach skanujących. Oprócz skanowania portów program ten oferuje mnóstwo innych funkcjonalności.

Zacznijmy od przyjrzenia się temu, jak `nmap` skanuje porty TCP. Zanim jednak tym się zajmiemy, musisz wiedzieć, że istnieją różne metody skanowania. Nawet komunikaty SYN można wysyłać na różne sposoby. Pierwszy z nich jest prosty: program wysyła komunikat i ocenia, czy port jest zamknięty czy otwarty. Jeżeli w odpowiedzi otrzyma komunikat RST, uznaje, że port jest zamknięty, i skanuje kolejny port. Jeżeli natomiast program uzyska komunikat z flagami SYN i ACK, wysyła do serwera komunikat RST kończący połączenie. Ten sposób jest niekiedy nazywany **skanowaniem połowicznym**.

W trybie **skanowania pełnego** program `nmap` przeprowadza cały proces potrójnego uścisku dłoni i dopiero potem kończy połączenie. W trybie tym istnieje mniejsze prawdopodobieństwo, że skanowanie zostanie wykryte przez system monitoringu lub zespół administratorów. Informacje uzyskanie w wyniku skanowania połowicznego i pełnego są jednak takie same. Wybór metody zależy jedynie od tego, która z nich jest mniej agresywna i z mniejszym prawdopodobieństwem zostanie wykryta.

Listing 3.17 przedstawia fragment wyniku skanowania pełnego. Program `nmap` został użyty do przeskanowania całej sieci. Zapis `/24` oznacza, że skanowane będą wszystkie komputery o adresach z zakresu 192.168.86.0 – 192.168.86.255. Jest to jeden ze sposób określania zbioru adresów. Można również podać listę adresów, jeżeli zajdzie taka potrzeba. Zwróć również

uwagę na użycie parametru `-T 5`, który powoduje zdefiniowanie szybkości skanowania. Wartość 5 oznacza tutaj, że program `nmap` powinien działać z maksymalną szybkością. Jeżeli temu parametrowi przypiszesz wartość 1, to żądania będą wykonywane bardzo wolno, co może pomóc w uniknięciu wykrycia podejmowanych działań.

Listing 3.17. Pełne skanowanie za pomocą programu nmap

```
┌──(kilroy@badmilo)-[~]
└─$ sudo nmap -sT -T 5 192.168.1.0/24
Nmap scan report for 192.168.1.1
Host is up (0.0056s latency).
Not shown: 987 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
22/tcp    filtered ssh
23/tcp    filtered telnet
53/tcp    open     domain
80/tcp    open     http
111/tcp   filtered rpcbind
139/tcp   open     netbios-ssn
443/tcp   open     https
445/tcp   open     microsoft-ds
5000/tcp  open     upnp
8200/tcp  open     trivnet1
20005/tcp open     btx
49152/tcp open     unknown
49153/tcp open     unknown
MAC Address: 80:CC:9C:DD:71:F2 (Netgear)

Nmap scan report for 192.168.1.15
Host is up (0.0053s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
111/tcp open  rpcbind
443/tcp open  https
MAC Address: 1C:69:7A:66:64:2A (EliteGroup Computer Systems)

Nmap scan report for 192.168.1.78
Host is up (0.016s latency).
Not shown: 979 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
6/tcp     filtered unknown
9/tcp     filtered discard
425/tcp   filtered icad-el
544/tcp   filtered kshell
1175/tcp  filtered dossier
1556/tcp  filtered veritas_pbx
2106/tcp  filtered ekshell
2121/tcp  filtered ccproxy-ftp
2144/tcp  filtered lv-ffx
2602/tcp  filtered ripd
2604/tcp  filtered ospfd
2920/tcp  filtered roboeda
3052/tcp  filtered powerchute
3690/tcp  filtered svn
```

```
5633/tcp  filtered beorl
6580/tcp  filtered parsec-master
10566/tcp filtered unknown
13782/tcp filtered netbackup
32780/tcp filtered sometimes-rpc23
49158/tcp filtered unknown
49165/tcp filtered unknown
MAC Address: E2:43:49:15:DF:19 (Unknown)
```

Program w wynikach podaje nie tylko numery portów, lecz także odpowiadające im nazwy usług. Nazwy te pochodzą z listy używanej przez `nmap`, jednak mogą nie mieć nic wspólnego z rzeczywistymi usługami wykorzystującymi wyświetlone porty. Program może wykrywać usługi na podstawie otrzymywanych od nich odpowiedzi. Ponadto wyświetla nazwę producenta karty sieciowej określoną na podstawie adresu MAC. Nazwa ta dostarcza informacji o sondowanym urządzeniu. Na przykład w powyższym przykładzie producentem jest firma Tp-Link Technologies dostarczająca urządzenia sieciowe, między innymi routery bezprzewodowe.

Zapewne zauważyłeś, że w powyższym przykładzie nie został określony zakres portów do przeskanowania. Domyślnie program `nmap` skanuje 1000 najczęściej wykorzystywanych portów. Dzięki temu skanowanie trwa krócej niż w przypadku 65 536 portów, z których ogromna większość nie jest wykorzystywana. Porty do przeskanowania można określić w formie zakresu lub listy. Aby przeskanować wszystkie porty, należy użyć argumentu `-p-`. Ponadto program skanuje porty z pewną domyślną szybkością, zależną od interwału pomiędzy kolejnymi wysyłanymi komunikatami. Aby zmienić szybkość, trzeba użyć argumentu `-T` i wartości od 0 do 5. Domyślną wartością jest 3. Jeżeli nie chcesz zajmować ograniczonego pasma transmisyjnego łącza albo jesteś ostrożny i chcesz zminimalizować prawdopodobieństwo wykrycia swojej aktywności, zmniejsz wartość argumentu. Jeśli się tym nie przejmujesz i chcesz przyspieszyć skanowanie, możesz użyć większej wartości.

Choć istnieją też inne metody skanowania portów TCP, wyżej opisana daje w większości przypadków dobre wyniki. Inne metody obejmują omijanie bądź testowanie zapory sieciowej, jednak są one dobrze znane od wielu lat i systemy zostały przed nimi zabezpieczone.

Zajmijmy się teraz skanowaniem portów UDP. Można zmieniać szybkość skanowania podobnie jak w przypadku portów TCP. Jednak nawet przy większych szybkościach istnieje problem z retransmitowaniem komunikatów. Przy większych szybkościach skanowanie trwa krócej niż zwykle, ale dłużej niż w przypadku portów TCP. Listing 3.18 przedstawia wynik skanowania portów UDP.

Listing 3.18. Skanowanie portów UDP za pomocą programu nmap

```
┌──(kilroy@badmilo)-[~]
└─$ sudo nmap -sU 192.168.1.0/24
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-26 19:08 EDT

Nmap scan report for 192.168.1.1
Host is up (0.0049s latency).
Not shown: 500 closed udp ports (port-unreach), 496 open|filtered udp ports (no-response)
PORT     STATE SERVICE
53/udp   open  domain
67/udp   open  dhcps
```

```
137/udp  open  netbios-ns
1900/udp open  upnp
MAC Address: 80:CC:9C:DD:71:F2 (Netgear)

Nmap scan report for 192.168.1.10
Host is up (0.0044s latency).
Not shown: 801 open|filtered udp ports (no-response), 197 closed udp ports (port-unreach)
PORT    STATE SERVICE
111/udp open  rpcbind
137/udp open  netbios-ns
MAC Address: 1C:69:7A:F1:2A:E0 (EliteGroup Computer Systems)
```

Skanowanie portów TCP wszystkich systemów w mojej sieci trwało poniżej 5 minut. Skanowanie portów UDP natomiast zajęło prawie dwie godziny, mimo że obejmowało sieć lokalną. Szybsze zakończenie skanowania portów UDP wymaga znacznego przyśpieszenia operacji skanowania.

Program `nmap` oprócz skanowania portów oferuje też inne funkcjonalności. Na przykład wykrywa systemy operacyjne na podstawie charakterystycznych cech wysyłanych przez nie pakietów. Ponadto w zależności od wykrytych otwartych portów program może uruchamiać różne skrypty napisane w języku Lua. Choć skrypty dostarczane razem z programem oferują mnóstwo funkcjonalności, można również tworzyć własne. Aby uruchomić skrypt, należy podać jego nazwę w argumencie programu. Oprócz tego, jak pokazuje listing 3.19, można uruchamiać całe kolekcje skryptów. W tym przykładzie zostały uruchomione wszystkie skrypty o nazwach rozpoczynających się od *http*. W sumie było ich 138.

Listing 3.19. Uruchamianie skryptów za pomocą programu nmap

```
┌──(kilroy@badmilo)-[~]
└─$ sudo nmap -sS -T 3 -p 80 -oN nse-out.txt --script http* 192.168.1.15
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-27 19:40 EDT
NSE: Warning: Could not load 'http-brute.nse': no path to file/directory: http-brute.nse
Pre-scan script results:
|_http-robtex-shared-ns: *TEMPORARILY DISABLED* due to changes in Robtex's API. See
↪https://www.robtex.com/api/
NSE: [http-form-brute] usernames: Time limit 10m00s exceeded.
NSE: [http-form-brute] usernames: Time limit 10m00s exceeded.
NSE: [http-form-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.1.15
Host is up (0.0097s latency).

PORT   STATE SERVICE
80/tcp open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-form-brute:
|   Accounts: No valid accounts found
|_  Statistics: Performed 3980 guesses in 600 seconds, average tps: 6.3
|_http-slowloris: false
|_http-favicon: Unknown favicon MD5: 69C728902A3F1DF75CF9EAC73BD55556
| http-vhosts:
|_128 names had status 302
| http-form-fuzzer:
|   Path: / Action: login.php
|     username
|       string lengths that caused errors:
```

```
|         309106
|       integer lengths that caused errors:
|_        304092, 308800
|_http-devframework: Couldn't determine the underlying framework or CMS. Try increasing
↪'httpspider.maxpagecount' value to spider more pages.
| http-headers:
|   Date: Wed, 28 Jun 2023 10:57:57 GMT
|   Server: Apache/2.4.55 (Ubuntu)
|   Set-Cookie: security=low; path=/
|   Set-Cookie: PHPSESSID=6p5qtnekffolip8bek4akmqk3p; expires=Thu, 29-Jun-2023 10:57:57
↪GMT; Max-Age=86400; path=/; HttpOnly; SameSite=1
|   Expires: Tue, 23 Jun 2009 12:00:00 GMT
|   Cache-Control: no-cache, must-revalidate
|   Pragma: no-cache
|   Connection: close
|   Content-Type: text/html;charset=utf-8
|
|_  (Request type: HEAD)
|_http-config-backup: ERROR: Script execution failed (use -d to debug)
|_http-malware-host: Host appears to be clean
|_http-date: Wed, 28 Jun 2023 10:57:58 GMT; -1s from local time.
MAC Address: 1C:69:7A:66:64:2A (EliteGroup Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 49765.40 seconds
```

Powyższy wynik pokazuje, że został przeskanowany jeden port na jednym urządzeniu. Jeżeli trzeba uruchamiać skrypty wykorzystujące protokół HTTP, można skanowanie ograniczyć tylko do portów wykorzystywanych przez ten protokół. Oczywiście skrypty takie jak powyższy można uruchamiać podczas zwykłego skanowania 1000 portów. Jednak wtedy uzyska się wyniki, które trzeba będzie odpowiednio zinterpretować, między innymi wyszukać w nich tylko te informacje, które dotyczą serwerów WWW.

Program `nmap` nie tylko uruchamia skrypty i skanuje porty, lecz także zbiera informacje o obiekcie testów i udostępnianych przez niego usługach. Argument `-A` powoduje włączenie funkcjonalności wykrywania systemu operacyjnego i jego wersji. Uruchamianie są również skrypty odpowiednie dla wykrytych portów. Ponadto program wykonuje polecenie `traceroute`, które dostarcza informacji o ścieżce sieciowej pomiędzy Twoim komputerem a testowanym obiektem.

Skanowanie szybkie

Program `nmap` jest niewątpliwie najpopularniejszym skanerem portów, ale nie jedynym. Jest bardzo skuteczny, ale nie jest zoptymalizowany pod kątem skanowania dużych sieci. Gdy pojawi się potrzeba wykonania takiego skanowania, należy użyć przystosowanego do tego celu narzędzia, na przykład `masscan`. Program ten różni się od `nmap` głównie tym, że skanuje sieć asynchronicznie, tj. wysyła kolejne komunikaty, nie czekając na odpowiedzi. Przetwarzaniem odpowiedzi, które nadchodzą, zajmuje się osobna część programu. Dzięki tej funkcjonalności przeskanowanie naprawdę dużych sieci zajmuje kilka minut. Szybkości działania obu programów można porównać, skanując lokalną sieć z adresacją /24 składającą się z 254 komputerów.

Program masscan ma własne argumenty, ale można stosować w nim też argumenty typowe dla programu nmap. Jeżeli znasz nmap, szybko nauczysz się posługiwać programem masscan. Jedyna różnica pomiędzy obydwoma narzędziami polega na tym, że w tym drugim trzeba określać porty do przeskanowania, ponieważ nie jest stosowany domyślny zakres portów. Ilustruje to listing 3.20, który przedstawia wynik przeskanowania zakresu portów 0 – 1500. Gdyby trzeba było wyszukać systemy wykorzystujące tylko port numer 443, tj. serwery WWW używające protokołu TLS, należałoby w argumencie programu podać tylko ten jeden port. Dzięki możliwości wykluczania nieistotnych portów oszczędza się mnóstwo czasu. Warto zauważyć, że uruchamianie programu masscan w tej samej sieci, która wcześniej była skanowana za pomocą nmap, trwało znacznie dłużej podczas skanowania SYN niż w przypadku nmap.

Listing 3.20. Wynik szybkiego skanowania portów za pomocą programu masscan

```
┌──(kilroy@badmilo)-[~]
└─$ sudo masscan -sS --ports 1-1500 192.168.1.0/24
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-06-27 22:30:33 GMT
Initiating SYN Stealth Scan
Scanning 256 hosts [1500 ports/host]
Discovered open port 80/tcp on 192.168.1.22
Discovered open port 853/tcp on 192.168.1.118
Discovered open port 443/tcp on 192.168.1.1
Discovered open port 22/tcp on 192.168.1.15
Discovered open port 445/tcp on 192.168.1.144
Discovered open port 445/tcp on 192.168.1.1
Discovered open port 53/tcp on 192.168.1.1
Discovered open port 53/tcp on 192.168.1.194
Discovered open port 853/tcp on 192.168.1.113
Discovered open port 853/tcp on 192.168.1.55
Discovered open port 53/tcp on 192.168.1.55
Discovered open port 53/tcp on 192.168.1.236
Discovered open port 443/tcp on 192.168.1.15
```

Dostępne jest również uniwersalne narzędzie, które pozwala kontrolować interwał pomiędzy kolejnymi wysyłanymi komunikatami. Program masscan jest szybki, ponieważ działa w trybie asynchronicznym, natomiast hping3 pozawala określać odstęp czasu pomiędzy następującymi po sobie komunikatami, mimo to nie jest jednak w stanie osiągnąć naprawdę dużych szybkości skanowania. Oferuje za to wiele innych funkcjonalności. Pozwala między innymi za pomocą argumentów definiować zawartość pakietów. W rzeczywistości hping3 jest niezwykle rozbudowanym programem ping, a nie kopią programu nmap czy następnym skanerem. Jeżeli jednak trzeba przeskanować pojedynczy komputer i zebrać o nim informacje, hping3 okazuje się niezastąpiony. Listing 3.21 przedstawia wynik skanowania 10 portów. Argument -S oznacza wysyłanie komunikatów SYN, a -p określa numer portu. Prefiks ++ powoduje rozpoczęcie odliczania kolejnych numerów portów od wskazanego portu. Liczbę portów określa się za pomocą argumentu -c. Tutaj program hping3 przeskanował 10 portów, zaczynając od portu numer 80. Dodatkowo argument -s określa numer postu źródłowego. W tym przypadku nie ma on znaczenia, ale czasami może mieć.

Listing 3.21. Wynik skanowania portów za pomocą programu hping3

```
root@rosebud:~# hping3 -S -p ++80 -s 1657 -c 10 192.168.86.1
HPING 192.168.86.1 (eth0 192.168.86.1): S set, 40 headers + 0 data bytes
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=0 win=29200 rtt=7.8 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15522 sport=81 flags=RA seq=1 win=0 rtt=7.6 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15523 sport=82 flags=RA seq=2 win=0 rtt=7.3 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15524 sport=83 flags=RA seq=3 win=0 rtt=7.0 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15525 sport=84 flags=RA seq=4 win=0 rtt=6.7 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15526 sport=85 flags=RA seq=5 win=0 rtt=6.5 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15527 sport=86 flags=RA seq=6 win=0 rtt=6.2 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15528 sport=87 flags=RA seq=7 win=0 rtt=5.9 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15529 sport=88 flags=RA seq=8 win=0 rtt=5.6 ms
len=46 ip=192.168.86.1 ttl=64 DF id=15530 sport=89 flags=RA seq=9 win=0 rtt=5.3 ms

--- 192.168.86.1 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 5.3/6.6/7.8 ms
```

Program `hping3` nie dostarcza wprost informacji o otwartych portach, tak jak robią to inne skanery, dlatego trzeba umiejętnie interpretować uzyskiwane wyniki. W każdym wyświetlonym wierszu znajduje się opcja `flags`. Pierwszy odebrany komunikat z odpowiednią zawierał flagi SYN i ACK oznaczające, że skanowany port był otwarty. Jego numer określa opcja `sport` (ang. *source port*, port źródłowy), w tym przypadku jest on równy 80. Zapewne dziwisz się, dlaczego jest to port źródłowy. Pamiętaj, że jest to komunikat z odpowiedzią. Komunikat wysyłany przez program zawiera port docelowy numer 80, który w odpowiedzi jest używany jako port źródłowy.

Pozostałe komunikaty zawierają flagi RST i ACK. Pierwsza oznacza, że dany port jest zamknięty. Program `hping3` pozwala określać wartości dowolnych flag. Przy jego użyciu można wykonać tzw. skanowanie choinkowe, w którym ustawione są flagi FIN, PSH i URG. Nazwa wzięła się stąd, że komunikat z ustawionymi wszystkimi flagami przypomina świąteczną choinkę z zapalonymi lampkami. Przy odrobinie wyobraźni ustawienie jednej flagi odpowiada zapaleniu lampki na choince. Aby wykonać tego rodzaju skanowanie należy użyć argumentów `-F -P -U`. W takim przypadku flagi RST i ACK byłyby ustawione w komunikatach wysłanych przez serwer z portów źródłowych 81 – 89. Odpowiedź z portu numer 80 nie nadeszłaby w ogóle, ponieważ zgodnie z zaleceniem RFC 793 zapytania z ustawionymi wszystkim flagami powinny być odrzucane bez odsyłania odpowiedzi.

Jak już wspomniałem, programu `hping3` potrafi wysyłać komunikaty z dużą szybkością. Robi to na dwa sposoby. Pierwszy wymaga użycia argumentu `-i` oraz podania liczby określającej czas zwłoki w sekundach. Można też wykorzystać argumenty `-i u1` wprowadzające odstęp jednej milisekundy. Prefiks `u` oznacza, że umieszczona po nim wartość jest liczbą milisekund. Drugi sposób szybkiego wysyłania komunikatów polega na użyciu argumentu `--flood`. Program `hping3` wysyła wtedy komunikaty z maksymalną szybkością bez oczekiwania na odpowiedzi.

Skanowanie usług

Ostatnią informacją, jaką można uzyskać w wyniku skanowania, są rodzaje usług wykorzystujących otwarte porty. Same numery portów niosą wiele informacji, ale nie zawsze tak jest. Czasami usługi wykorzystują niestandardowe numery portów. Na przykład usługa SSH zazwyczaj wykorzystuje port TCP numer 22. Gdy program `nmap` wykryje otwarty port 22, informuje, że wykrył usługę SSH. Natomiast jeżeli wykryje port numer 2222, to „nie wie", jaka to usługa, chyba że wykorzysta inną swoją funkcjonalność — wykrywania wersji usługi na podstawie generowanych przez nią informacji.

Program `amap` nie wiąże usługi z numerem portu. Zamiast tego analizuje uzyskiwane odpowiedzi. W tym celu wysyła zapytanie do usługi i porównuje odpowiedź z informacjami zapisanymi w swojej bazie danych.

Listing 3.22 przedstawia wynik dwukrotnego uruchomienia programu `amap`. Za pierwszym razem program skanował serwer WWW wykorzystujący domyślny numer portu. Zgodne z oczekiwaniami serwer odpowiedział, że usługa wykorzystuje protokół HTTP. Za drugim razem skanowany był port numer 8383, który pojawił się podczas skanowania portów docelowych. W trakcie tej operacji został uznany za należący do usługi serwera WWW, ponieważ zalicza się do tzw. dobrze znanych portów wykorzystywanych przez określone protokoły lub przynajmniej ta usługa była wcześniej zarejestrowana dla danego portu. Jednak to nie jest powszechnie używany port, a sama rejestracja usługi działającej na tym porcie nie oznacza, że aplikacja będzie go nasłuchiwała. Tym razem program musiał wykonać nieco więcej pracy, aby określić stosowany protokół.

Listing 3.22. Pozyskiwanie informacji o usłudze za pomocą programu amap

```
┌──(kilroy@badmilo)-[~]
└─$ sudo amap 192.168.1.219 80
amap v5.4 (www.thc.org/thc-amap) started at 2023-06-28 18:16:39 - APPLICATION MAPPING mode

Protocol on 192.168.1.219:80/tcp matches http
Protocol on 192.168.1.219:80/tcp matches http-apache-2
Protocol on 192.168.1.219:80/tcp matches http-iis

Unidentified ports: none.

amap v5.4 finished at 2023-06-28 18:16:40
┌──(kilroy@badmilo)-[~]
└─$ sudo amap 192.168.1.219 8383
amap v5.4 (www.thc.org/thc-amap) started at 2023-06-28 18:18:14 - APPLICATION MAPPING mode

Protocol on 192.168.1.219:8383/tcp matches http
Protocol on 192.168.1.219:8383/tcp matches http-apache-2
Protocol on 192.168.1.219:8383/tcp matches ssl

Unidentified ports: none.

amap v5.4 finished at 2023-06-28 18:18:20
```

Do pozyskiwania informacji o obiekcie testów można wykorzystywać niektóre protokoły. Jednym z nich jest SMB (ang. *Server Message Block*, blok komunikatów serwera) używany w systemie Windows do udostępnia plików i zarządzania zewnętrznymi komputerami

z zainstalowanym tym systemem. Do skanowania systemów wykorzystujących powyższy protokół służy kilka narzędzi. Jednym z nich jest program `smbmap` wyświetlający listę wszystkich udostępnianych przez dany system udziałów. Listing 3.23 przedstawia wynik przeskanowania za pomocą programu `smbmap` komputera z systemem macOS udostępniającym pliki za pomocą protokołu SMB. Zazwyczaj dostęp do udziałów nie jest możliwy bez wcześniejszego uwierzytelnienia. Dlatego konieczne jest podanie loginu i hasła, co jest pewną sprzecznością, ponieważ jeżeli ma się te dane, nie trzeba korzystać z narzędzia takiego jak `smbmap`.

Listing 3.23. Lista udostępnianych zasobów uzyskana za pomocą programu smbmap

```
┌──(kilroy@badmilo)-[~]
└─$ smbmap -u kilroy -p obScurePW123! -H 192.168.1.10
[+] IP: 192.168.1.10:445   Name: 192.168.1.10
        Disk                    Permissions Comment
        ----                    ----------- -------
        homes                   READ, WRITE Home Directories
        media                   READ ONLY   Media directory
        print$                  READ ONLY   Printer Drivers
        IPC$                    NO ACCESS   IPC Service (bobbie server (Samba, Ubuntu))
        kilroy                  READ, WRITE
        Home Directories        READ, WRITE
```

Innym narzędziem do wyszukiwania udziałów i różnych danych udostępnianych za pomocą protokołu SMB jest `enum4linux`. Jest to skrypt wywołujący programy zawarte w pakiecie Samba implementującym protokół SMB w systemie Linux. Z aplikacji tych można również korzystać bezpośrednio. Na przykład (zob. listing 3.24) za pomocą programu `enum4linux` można komunikować się z zewnętrznymi systemami i uzyskiwać listę udziałów podobną do przedstawionej w listingu 3.23.

Listing 3.24. Przykład użycia programu enum4linux

```
┌──(kilroy@badmilo)-[~]
└─$ sudo enum4linux -U 192.168.1.10
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/ enum4linux/ ) on
↪Wed Jun 28 18:25:51 2023

 ====================( Target Information)=====================
Target ........... 192.168.1.10
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

 =======( Enumerating Workgroup/Domain on 192.168.1.10)=========
[+] Got domain/workgroup name: WASHERE
 ========( Getting domain SID for 192.168.1.10 )================
Domain Name: WASHERE
Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup
===============( Users on 192.168.1.10 )======================
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: kilroy   Name: Ric Messier   Desc:
user:[kilroy] rid:[0x3e8]
```

To tylko niewielki przykład pokazujący możliwości programu `enum4linux`. Jak widać, program ten pozwala wyodrębniać ogromne ilości informacji za pomocą protokołu SMB. Wynika to stąd, że do pewnego stopnia protokół SMB udostępnia informacje w sieci lokalnej, ponieważ pomagają one innym systemom w korzystaniu z usług oferowanych przez dany serwer. Przykładem takich informacji jest lista tzw. udziałów.

Ręczne interakcje

Narzędzia do automatycznego pozyskiwania informacji są bardzo przydatne, jednak niekiedy trzeba zakasać rękawy i samodzielnie wykonywać niektóre operacje, takie jak nawiązywanie połączenia z określonym portem usługi i wysyłanie poleceń. Jednym z programów służących do tego celu jest `telnet`. Nie należy go mylić z protokołem Telnet ani usługą o tej samej nazwie. Program ten pozwala komunikować się z usługą Telnet, ale jego rolą jest po prostu nawiązywanie połączenia TCP z serwerem. Wystarczy w tym celu podać w argumencie numer portu. Listing 3.25 przedstawia wynik nawiązania połączenia z serwerem SMTP (ang. *Simple Mail Transfer Protocol*, prosty protokół przesyłania poczty).

Listing 3.25. Przykład interakcji z serwerem pocztowym za pomocą programu telnet

```
┌──(kilroy@badmilo)-[~]
└─$ telnet 192.168.1.15 25
Trying 192.168.1.15...
Connected to 192.168.1.15.
Escape character is '^]'.
220 bobbie ESMTP Postfix (Ubuntu)
EHLO wubble.com
250-bobbie
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
MAIL FROM: foo@foo.com
250 2.1.0 Ok
RCPT TO: root@localhost
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Hi,
This is me.

.
250 2.0.0 Ok: queued as 43AB04343C14
```

Domyślnie program `telnet` nawiązuje połączenie z portem numer 23, typowym dla usługi Telnet. Jednak podając w argumencie numer portu, na przykład 25 jak w powyższym listingu, można zestawiać dowolne połączenia. Po nawiązaniu połączenia, co jest sygnalizowane

czytelnym komunikatem, można wpisywać polecenia. W tym przykładzie wykorzystany został serwer pocztowy używający protokołu ESMTP (ang. *Extended SMTP*, rozszerzony protokół SMTP). Pozyskane podczas tego połączenia informacje to typ serwera (Postfix) oraz lista dostępnych poleceń. Nie oznacza to jednak, że wszystkie są obsługiwane przez serwer. Widoczne jest na przykład polecenie `VRFY` służące do weryfikowania adresów użytkowników. Umiejętnie je wykorzystując, można poznać adresy pocztowe. Instytucje nie chcą udostępniać osobom trzecim tego rodzaju informacji (mogłyby być one wykorzystane przez hakerów), dlatego blokują to polecenie.

Pierwszym komunikatem odebranym z serwera jest zgłoszenie usługi. Czasami zawiera on dość szczegółowe informacje o usłudze, więc jest wykorzystywany przez narzędzia takie jak `nmap`. Nie wszystkie usługi wyświetlają zgłoszenia zawierające informacje o protokołach i serwerach.

Program `telnet` nie jest jedynym narzędziem do komunikowania się z serwerami. Do tego celu można zastosować również program `netcat`, który zazwyczaj wywołuje się w terminalu za pomocą polecenia `nc`. Korzysta się z niego w taki sam sposób, jak z programu `telnet`. Listing 3.26 przedstawia wynik nawiązania połączenia z serwerem WWW o adresie 192.168.1.219. Program `netcat` nie wyświetla jednak informacji o pomyślnym nawiązaniu połączenia. Jedynie w przypadku, gdy wybrany port jest zamknięty, pojawia się komunikat *Connection refused* (połączenie odrzucone). Jeżeli komunikat ten się nie pojawi, można przyjąć, że połączenie zostało nawiązane i można wpisywać polecenia. Listing przedstawia zapytanie wysłane do serwera za pomocą protokołu HTTP/1.1. Pusty wiersz informuje serwer, że wszystkie nagłówki zostały już przesłane i może on zacząć przetwarzać zapytanie.

Listing 3.26. Przykład interakcji z serwerem WWW za pomocą programu netcat

```
┌──(kilroy@badmilo)-[~]
└─$ nc 192.168.1.219 80
GET / HTTP/1.1
Host: 192.168.1.219

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Sun, 19 Mar 2023 09:10:48 GMT
Accept-Ranges: bytes
ETag: "bccfeab1425ad91:0"
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 28 Jun 2023 22:37:59 GMT
Content-Length: 1116928

<html>
<head>
        <style>
                body {
                        background:url('hahaha.jpg') no-repeat center center;
                        min-height: 100%;
                        background-color: black;
                }
        </style>
</head>
<body>
```

Powyższy wynik przedstawia jedynie zawartość nagłówka odpowiedzi. Kod HTML strony został pominięty. Jedną z zalet programu `nc` w porównaniu z programem `telnet` jest możliwość uruchamiania go w trybie nasłuchu. Oznacza to, że przy jego użyciu można utworzyć miejsce docelowe dla pakietów przesyłanych przez sieć. W ten sposób zbiera się informacje o intruzach nawiązujących połączenia ze wszystkimi otwartymi portami. Ponadto program `telnet` wykorzystuje tylko protokół TCP, a `nc` TCP i UDP, dzięki czemu może komunikować się z usługami wykorzystującymi protokół UDP.

Podsumowanie

Zebranie informacji o obiekcie testów ułatwia późniejszą pracę. Na podstawie ujawnianych danych można wykrywać potencjalne luki w bezpieczeństwie systemów. Warto, abyś poświęcił czas na pozyskanie takich informacji, nawet jeżeli bardzo Ci się śpieszy do rozpoczęcia ataku.

Oto najważniejsze wnioski płynące z lektury tego rozdziału:

- Do zbierania informacji o obiekcie testów można wykorzystywać publicznie dostępne źródła.
- Do automatycznego pozyskiwania informacji z dostępnych źródeł służy program Maltego.
- Program theHarvester można wykorzystać do automatycznego pozyskiwania szczegółowych informacji o ludziach i ich adresach e-mail.
- System DNS może zawierać mnóstwo szczegółowych informacji o instytucji będącej celem testów.
- Rejestry RIR są źródłami mnóstwa szczegółowych informacji o adresach IP i ich właścicielach.
- Program `nmap` służy do skanowania portów, jak również do zbierania informacji o systemach operacyjnych i wersjach aplikacji.
- Skanowanie portów pozwala jednoznacznie określać porty wykorzystywane przez usługi.
- Narzędzia do wiązania numeru portu z rodzajem usługi przydają się do pozyskiwania informacji o wersji aplikacji.
- Za pomocą programów `telnet` i `nc` można, między innymi na podstawie zgłoszeń usług, zbierać szczegółowe informacje o zewnętrznych systemach.

Przydatne materiały

- Blog Camerona Colquhouna *A Brief History of Open Source Intelligence* (*https://www.bellingcat.com/resources/articles/2016/07/14/a-brief-history-of-open-source-intelligence/*).
- Blog Sudhanshu Chauhana, *Tools For Open Source Intelligence* (*https://www.slideshare.net/slideshow/tools-for-open-source-intelligence-osint-61284325/61284325*).
- Robert Layton, Paul Watters, *Automating Open Source Intelligence*, Elsevier, 2015 (*https://shop.elsevier.com/books/automating-open-source-intelligence/layton/978-0-12-802916-9*).
- Sudhanshu Chauhan, Nutan Kumar Panda, *Hacking Web Intelligence*, Syngress, 2015 (*https://www.oreilly.com/library/view/hacking-web-intelligence/9780128018675/*).

Skorowidz

A

B

D

C

E

F

G

H

I

J

K

L

Ł

M

N

O

P

R

S

Ś

T

U

V

W

X

Y

Z

Kali Linux to specjalistyczna dystrybucja Linuksa oparta na Debianie. System został zaprojektowany z myślą o specjalistach do spraw cyberbezpieczeństwa. Umożliwia testowanie zabezpieczeń, tworzenie eksploitów, analizę kodu aplikacji i wykrywanie nadużyć. Zapewnia wszystkie potrzebne narzędzia, a także setki dodatkowych pakietów przeznaczonych do badania bezpieczeństwa.

Oddajemy Czytelnikowi nowe, zaktualizowane i uzupełnione wydanie książki prezentującej możliwości Kali Linux w zakresie testowania bezpieczeństwa oprogramowania. Poza opisem samego systemu i dostępnych narzędzi w wyczerpujący sposób przedstawiono tu szczegóły prowadzenia testów bezpieczeństwa, w tym sieci bezprzewodowych. Omówiono zasady testowania aplikacji WWW, techniki łamania haseł i korzystania z eksploitów. W tym wydaniu znalazły się również bezcenne wskazówki dotyczące analizy oprogramowania z wykorzystaniem inżynierii wstecznej i śledzenia nadużyć cyfrowych. Nie zabrakło też kluczowych dla każdego pentestera kwestii etyki i legalności podejmowanych działań.

Zwięzła, przejrzysta i poparta doświadczeniem — ta pozycja nie ma sobie równych! Jest przystępnym i cennym źródłem wiedzy dla każdego.

Alexander Arlt, główny konsultant do spraw bezpieczeństwa w Google

W książce:

- narzędzia dostępne w Kali Linux i znaczenie testów bezpieczeństwa
- podstawy prowadzenia testów penetracyjnych
- instalowanie Kali Linux na różnych platformach
- struktura testu bezpieczeństwa prowadzonego za pomocą Kali Linux
- Kali Linux i zaawansowane techniki ataku
- opracowanie raportu po zakończeniu testów

Ric Messier jest autorem publikacji, trenerem i wykładowcą. Posiada certyfikaty: GCE-ACE, CCSP, GCIH, GSEC, CEH, CISSP i MS. Od kilkudziesięciu lat zajmuje się zagadnieniami cyberbezpieczeństwa. Obecnie jest głównym konsultantem w firmie Mandiant, będącej częścią Google Cloud.

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-3000-1
9 788328 930001
Cena: 119,00 zł