Wstęp do cyberbezpieczeństwa (WCYB)

Moduł 1: Kali Linux | Wprowadzenie do testów penetracyjncych - rekonesans

Semestr: 19Z

Plan laboratorium

Z tego laboratorium:

- zapoznasz się z systemem operacyjnym Kali Linux
- zapoznasz się z podstawowymi poleceniami systemu Linux
- dowiesz się w jaki sposób zarzadzać usługami na systemie Linux
- dowiesz się jak pisać skrytpy w Bash'u
- dowiesz się co to jest test penetracyjny
- zapoznasz się z podstawami wykonywania rekonesansu jako elementu przygotowania do testów penetracyjnych
- zapoznasz się z metodami pasywnego zbieranie informacji o celu
- zapoznasz się z metodami aktywnymi zbieranie informacji o celu

1. Korzystanie z Kali Linux

1.1. O Kali Linux

Kali Linux to darmowa dystrybucja systemu Operacyjnego Kali Linux przeznaczona dla adminstratorów IT i specjalistów ds. bezpieczeństwa do audytów bezpieczeństwa. Posiada ponad 300 narzędzi do przeprowadzania testów penetracyjnych i audyty bezpieczeństwa, a dzięki jego zgodności ze standardami rozwoju Debiana zapewnia bardziej znane środowisko dla aministratorów IT. W rezultacie jest to bardziej niezawodne rozwiązanie, które można łatwiej zaktualizować. Użytkownicy mogą również dostosować system operacyjny do własnych potrzeb i preferencji.

Wszystkie programy dostarczone z systemem operacyjnym zostały ocenione pod kątem przydatności i skuteczności. Przykłady ważniejszych narzędzi:

- `Metasploit` zawierający bazę exploitów
- `nmap` do skanowania portów i podatności na zagrożenia
- `Wireshark` do monitorowania ruchu w sieci
- `aircrack-ng` do testowania bezpieczeństwa sieci bezprzewodowych

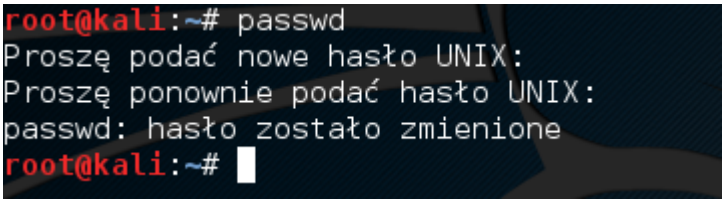
Kali Linux może działać na różnych urządzeniach, jest kompatybilny z wieloma urządzeniami bezprzewodowymi i USB, a także może pracować na urządzeniach z procesorami ARM.

Zanim rozpoczniemy pracę z Kali Linux należy się uwierzytelnić. Domyślnymi poświadczeniami tego systemu dla instalacji z obrazu są:

username: **root**

password: **toor**

Przy pierwszym użyciu nalezy je zmianień za pomocą polecenia `passwd`.



Pamiętaj, aby zawsze zamienić wszelkie domyślne lub słabe hasła dla swoich urządzeń, usług, kont itp. na coś długiego i złożonego. Złożoność może być tu osiągana za pomocą:

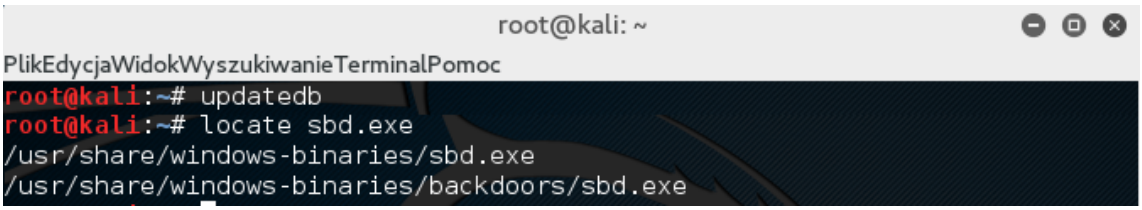
- **mieszanie liter, cyfr, znaków specjalnych, hasło o odpowiedniej długości**
- **hasła składające się z kilku słów, które np. łatwo zapamiętać**

W szczególności warto chronić te konta, które dają wysoki poziom dostępu do działań na maszynie/w usłudze, np. SSH.

1.2 Podstawowe komendy

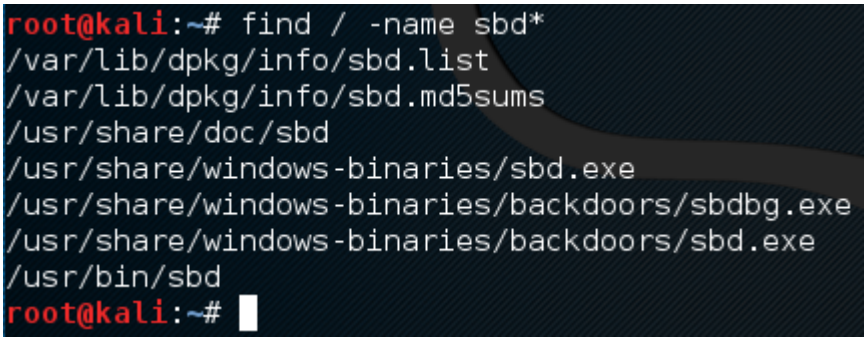
1.2.1 find, locate and which

Istnieje wiele narzędzi systemu Linux, których można użyć do zlokalizowania plików w instalacji systemu Linux, przy czym trzy z nich to: `find`, `locate` i `which`. Wszystkie trzy z tych narzędzi mają podobne funkcje, ale działają i zwracają dane na różne sposoby. Przed użyciem narzędzia **locate** musimy najpierw użyć polecenia `updatedb`, aby zbudować lokalną bazę danych wszystkich plików w systemie plików. Po zbudowaniu bazy danych można użyć **locate** do łatwego przeszukiwania tej bazy danych podczas wyszukiwania plików lokalnych. Przed uruchomieniem **locate** należy zawsze aktualizować lokalną bazę danych za pomocą polecenia `updatedb`.



Komenda **which** przeszukuje katalogi zdefiniowane w zmiennej środowiskowej $PATH dla podanej nazwy pliku. Jeśli zostanie znalezione dopasowanie komenda zwróci pełną ścieżkę do pliku, jak pokazano poniżej.

Polecenie **find** jest bardziej agresywnym narzędziem wyszukiwania niż **locate** lub **which**. Find jest w stanie rekurencyjnie przeszukiwać dowolną ścieżkę w poszukiwaniu różnych plików.



Teraz, gdy znamy podstawowe narzędzia do lokalizowania plików w systemie Kali Linux, przejdźmy do sprawdzenia, jak działają usługi Kali i co jest potrzebne do skutecznego zarządzania nimi.

Ćwiczenia do wykonania

1. Zapoznać się z listą dostępnych narzędzi systemu Kali Linux.
2. Określ lokalizację pliku **plink.exe** w systemie Kali Linux.
3. Znajdź i zapoznaj się z dokumentacją narzędzia dnsenum.

1.3 Wybrane usługi

Kali Linux zawiera kilka niestandardowych funkcji. Domyślna instalacja Kali jest dostarczana z wstępnie zainstalowanymi kilkoma usługami np. SSH, HTTP, MySQL itp. Jeśli nie zostanie to odpowiednio skonfigurowane, to usługi te zostaną automatycznie załadowane podczas rozruchu systemy. Innymi słowy, Kali Linux domyślnie otworzy kilka portów popularnych usług sieciowych. Najczęściej chcemy unikać takich sytuacji - zarówno jako specjaliści cyberbezpieczeństwa, ale też zapewniając bezpieczeństwo np. Klientom. Kali pozwala skonfigurować, które usługi mają być uruchamiane przy starcie systemu.

Ważnymi mechanizmi bezpieczeństwa stosowanym w wielu systemach operacyjnych lub w ogólności - *środowiskach wykonawczych aplikacji/systemów* są:

- mechanizm białej listy - *whitelisting* - domyślenie wykonanie każdej aplikacji czy otwarcie portu jest zabronione w systemie. Aby wykonać aplikację lub otworzyć port należy ją najpierw dodać do listy dopuszczalnych do wykonywania aplikacji/otwarcia portu.
- mechanizm czarnej listy - *blacklisting* - domyślenie wykonanie każdej aplikacji czy otwarcie portu jest dozwolone w systemie. Aby zabronić wykonywania się aplikacji lub otworcia portu należy ją umieścić na liście zabronionych aplikacji/portów.

Whitelisting/blacklisting to jeden z podstawowych składników procesu *hardeningu* systemów i aplikacji. W ogólności stanowi jeden z podstawowych środków zabezpieczania systemów i aplikacji na etapie ich konfiguracji.

Poniżej omówiono niektóre z tych usług, a także sposób ich obsługi i zarządzania nimi.

1.3.1 Usługa SSH

Usługa Secure Shell (`SSH`) jest najczęściej używana do zdalnego dostępu do komputera przy użyciu bezpiecznego, szyfrowanego protokołu. `SSH` jest następcą protokołu

`telnet` - w przeciwieństwie do swojego poprzednika, połączenia zestawiane przez `SSH` są szyfrowane. Jednak protokół SSH ma pewne zaskakujące i przydatne funkcje poza zapewnianiem dostępu do terminala (np. realizacja proxy lub enkapsulacja tunelu SSH w protokole HTTP). Usługa `SSH` jest oparta na protokole `TCP` i domyślnie nasłuchuje na porcie 22. Aby uruchomić usługę `SSH` w Kali, wpisz następujące polecenie w terminalu Kali.

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

Możemy sprawdzić, czy usługa `SSH` działa i nasłuchuje na porcie `TCP` 22, używając polecenia `netstat`. Wykorzystamy do tego mechanizm powłoki Linux polegający na możliwości przekazania danych wynikowych z jednego polecenia jako dane wejściowe do następnego. Przekierowanie odbywa się poprzez stosowanie znaku `|` (*pipe*) między poleceniami, przy czym wynik pierwszego polecenia jest przekazywany do drugiego polecenia. Bardzo popularnym narzędziem systemie Linux jest `grep`, który służy do wyszukiwania w tekście i wyodrębniania linii zawierających ciąg znaków pasujący do podanego wyrażenia, będącego *wyrażeniem regularnym* (*wyrażenia regularne pojawią się w innych modułach przedmiotu*). Poniej zaprezentowano przykład przekierowania wyniku narzędzia `netstat` (tzw. *piping*) do polecenia `grep`, aby wyszukać dane wyjściowe w poszukiwaniu `sshd`.

```
root@kali:~# netstat -antp|grep sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1755/sshd

tcp6       0      0 :::22                   :::*                    LISTEN      1755/sshd
```

Jeśli, podobnie jak wielu użytkowników, chcesz, aby usługa SSH była uruchamiana automatycznie podczas uruchamiania, musisz ją włączyć za pomocą skryptu `update-rc.d` w następujący sposób. Skryptu `update-rc.d` można używać do włączania i wyłączania większości usług w systemie Kali Linux.

```
root@kali:~# update-rc.d ssh enable
```

1.3.2 Usługa HTTP

Usługa `HTTP` może się przydać podczas testu penetracyjnego w celu hostingu witryny lub zapewnienia platformy do pobierania plików na zaatakowany komputer. Usługa `HTTP` jest oparta na protokole `TCP` i domyślnie nasłuchuje na porcie 80. Aby uruchomić usługę `HTTP` w Kali, wpisz następujące polecenie w terminalu.

```
root@kali:~# service apache2 start
```

Podobnie jak w przypadku usługi `SSH`, możemy zweryfikować, czy usługa `HTTP` działa i nasłuchuje na porcie `TCP` 80, używając ponownie poleceń `netstat` i `grep`.

```
root@kali:~# netstat -antp|grep apache
tcp6       0      0 :::80                   :::*                    LISTEN      1997/apache2
```

Aby usługa `HTTP` była uruchamiana w czasie rozruchu, podobnie jak w przypadku usługi `SSH`, musisz jawnie ją włączyć za pomocą **update-rc.d**.

```
root@kali:~# update-rc.d apache2 enable
```

Większość usług w Kali Linux jest obsługiwana w podobny sposób jak demony SSH i HTTP, za pośrednictwem ich skryptów serwisowych lub inicjujących. Aby uzyskać bardziej szczegółową kontrolę nad tymi usługami, możesz użyć narzędzi takich jak **rcconf** lub **sysv-rc-conf**, oba zaprojektowane w celu uproszczenia i zarządzania trwałością rozruchu tych usług

services.

Ćwiczenia do wykonania

1. Jeśli używasz obrazu Kali Linux zmień domyślne hasło na inne, bezpieczniejsze.
   .
2. Przećwicz uruchamianie i zatrzymywanie różnych usług Kali Linux.
3. Umożliwij uruchomienie usługi SSH w momencie rozruchu systemu.

1.4 Podstawy BASHa

Powłoka GNU **B**ourne-**A**gain **SH**ell (`Bash`) zapewnia środowisko do pracy oraz silnik skryptowy, z którego możemy korzystać do automatyzacji procedur przy użyciu istniejących narzędzi Linux. Możliwość szybkiego ulepszenia skryptu Bash w celu zautomatyzowania danego zadania jest niezbędnym wymogiem dla każdego specjalisty cyberbezpieczeństwa. W tej części laboratorium zapoznamy się ze elementami skryptów Bash.

1.4.1 Praktyczne wykorzystanie Bash'a - przykład 1

Wyobraź sobie, że Twoim zadaniem jest znalezienie wszystkich subdomen wymienionych na stronie `cisco.com`, a następnie znalezienie odpowiadających im adresów IP. Wykonanie tego ręcznie byłoby frustrujące i czasochłonne. Jednak za pomocą kilku prostych poleceń Bash możemy zamienić to w łatwe zadanie. Zaczynamy od pobrania strony indeksu `cisco.com` za pomocą polecenia `wget`.

```
root@kali:~# wget www.cisco.com
--2013-04-02 16:02:56-- http://www.cisco.com/
Resolving www.cisco.com (www.cisco.com)... 23.66.240.170,
Connecting to www.cisco.com (www.cisco.com)|23.66.240.170|:80... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 23419 (23K) [text/html]
Saving to: `index.html' 100%[====================================>]
23,419 --.-K/s in 0.09s
2013-04-02 16:02:57 (267 KB/s) - `index.html' saved [23419/23419]

root@kali:~# ls -l index.html
-rw-r--r-- 1 root root 23419 Apr 2 16:02 index.html
```

Szybko przeglądając ten plik, widzimy wpisy, które zawierają potrzebne informacje, takie jak ta pokazana poniżej:

```
<li><a href="http://newsroom.cisco.com/">Newsroom</a></li>
```

Zaczynamy od użycia polecenia `grep`, aby wyodrębnić wszystkie wiersze w pliku zawierające ciąg „href =”, wskazując, że ten wiersz zawiera URL.

```
root@kali:~# grep "href=" index.html
```

Rezultatem jest nadal mnóstwo niepotrzebnego kodu HTML, ale zauważmy, że większość linii ma podobną strukturę i może być wygodnie podzielona za pomocą znaku „/” jako separatora. Aby odpowiednio wyodrębnić nazwy domen z pliku, możemy spróbować użyć polecenia `cut` z naszym separatorem w 3-cim polu.

```
root@kali:~# grep "href=" index.html | cut -d "/" -f 3
```

Wynik, który otrzymujemy, jest daleki od optymalnego i prawdopodobnie po drodze zostało pominięte sporo linków, ale kontynuujmy. Nasz tekst zawiera teraz następujące wpisy:

```
about
solutions
ordering
siteassets
secure.opinionlab.com
help
```

Następnie wyczyścimy naszą listę, aby uwzględnić tylko nazwy domen. Użyjmy **grep**, aby odfiltrować wszystkie wiersze zawierające kropkę, aby uzyskać przejrzystszy wynik.

```
root@kali:~# grep "href=" index.html | cut -d "/" -f 3 | grep "\."
```

Nasze wyniki są prawie przejrzyste, ale teraz mamy wpisy, które wyglądają następująco.

```
learningnetwork.cisco.com">Learning Network<
```

Możemy je wyczyścić, używając ponownie polecenia `cut` dla pierwszej kolumny.

```
root@kali:~# grep "href=" index.html |cut -d "/" -f 3 | grep "\." | cut -d '"' -f 1
```

Teraz mamy wyczyszczoną listę, ale wiele duplikatów. Możemy je wyczyścić za pomocą polecenia `sort` z opcją *unique*(`-u`).

```
root@kali:~# grep "href=" index.html | cut -d "/" -f 3 | grep "\." | cut -d '"' -f 1 |
sort -u

blogs.cisco.com
communities.cisco.com
csr.cisco.com
developer.cisco.com
grs.cisco.com
home.cisco.com
investor.cisco.com
learningnetwork.cisco.com
newsroom.cisco.com
secure.opinionlab.com
socialmedia.cisco.com
supportforums.cisco.com
tools.cisco.com
www.cisco.com
www.ciscolive.com
```

Jeszcze lepszym sposobem na to byłoby wykorzystanie wyrażeń regularnych do naszego polecenia, przekierowując dane wyjściowe do pliku tekstowego, jak pokazano poniżej:

```
root@kali:~# cat index.html | grep -o 'http://[^"]*' | cut -d "/" -f 3 | sort –u >
list.txt
```

Teraz mamy przejrzystszą listę nazw domen połączonych z główną domeną `cisco.com`. Naszym następnym krokiem będzie użycie polecenia `host` dla każdej nazwy domeny w utworzonym pliku tekstowym, aby znaleźć odpowiedni adres IP. Możemy użyć pętli `for` w formie jednoliniowej, aby zrealizować automatyzację tego zadania.

```
root@kali:~# for url in $(cat list.txt); do host $url; done
```

Polecenie `host` daje nam różnego rodzaju dane wyjściowe, jednak nie wszystkie są istotne. Chcemy wyodrębnić jedynie adresy IP spośród wszystkich informacji, więc kierujemy dane wyjściowe do polecenia `grep`. Poszukiwanym wyrażeniem jest `"has address"`, a następnie wycinamy i sortujemy dane wyjściowe.

```
root@kali-repo:~# for url in $(cat list.txt); do host $url; done |
grep "has address" | cut -d " " -f 4 | sort u

128.30.52.37
136.179.0.2
141.101.112.4
…
206.200.251.19
23.63.101.114
23.63.101.80
23.66.240.170
23.66.251.95
50.56.191.136
64.148.82.50
66.187.208.213
67.192.93.178
```

1.4.2 Praktyczne wykorzystanie Bash'a - przykład 2

Otrzymujemy plik logów serwera `HTTP Apache`, który zawiera dowody przeprowadzenia cyberataku. Naszym zadaniem jest użycie prostych poleceń Bash do sprawdzenia pliku i odkrycia różnych informacji, takich jak:

- kim byli atakujący
- co dokładnie wydarzyło się na serwerze.

Najpierw używamy poleceń `head` i `wc`, aby szybko zapoznać się ze strukturą pliku dziennika.

```
root@kali:~# head access.log
93.241.170.13 - - [22/Apr/2013:07:09:11 -0500] "GET /favicon.ico HTTP/1.1" 404 506 "--"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/537.31 (KHTML, like Gecko)
Chrome/26.0.1410.65 Safari/537.31"
142.96.25.17 - - [22/Apr/2013:07:09:18 -0500] "GET / HTTP/1.1" 200 356 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/536.29.13 (KHTML, like Gecko) Version/6.0.4
Safari/536.29.13"
root@kali:~# wc -l access.log
1788 access.log
```

Zauważmy, że struktura pliku logów jest przyjazna dla polecenia `grep`. Różne pola, takie jak: adres IP, znacznik czasu, żądanie HTTP itp. są oddzielone spacjami. Zaczynamy od przeszukiwania żądań HTTP wysyłanych do serwera, aby znaleźć wszystkie adresy IP zapisane w tym pliku logów. Będziemy przesyłać dane wyjściowe `cat` do poleceń `cut` i `sort`. Może to dać nam wskazówkę co do liczby potencjalnych napastników, z którymi będziemy musieli sobie poradzić.

```
root@kali:~# cat access.log | cut -d " " -f 1 | sort -u
194.25.19.29
202.31.272.117
208.68.234.99
5.16.23.10
88.11.27.23
93.241.170.13
```

Widzimy, że w pliku logów zapisano mniej niż dziesięć adresów IP, choć to wciąż nie mówi nam nic o atakujących. Następnie używamy poleceń `uniq` i `sort` w celu dalszego udoskonalenia naszych wyników i sortowania danych według liczby przypadków, w których każdy adres IP uzyskał dostęp do serwera.

```
root@kali:~# cat access.log | cut -d " " -f 1 | sort | uniq -c | sort -urn
1038 208.68.234.99
445 186.19.15.24
89 194.25.19.29
62 142.96.25.17
56 93.241.170.13
37 10.7.0.52
30 127.0.0.1
13 5.16.23.10
10 88.11.27.23
6 172.16.40.254
13
```

Kilka adresów IP wyróżnia się, ale najpierw skupimy się na adresie, który ma najwyższą częstotliwość dostępu. Aby wyświetlić i policzyć zasoby żądane przez adres IP, można użyć następującej sekwencji poleceń:

```
root@kali:~# cat access.log | grep '208.68.234.99' | cut -d "\"" -f 2 |
uniq -c
1038 GET //admin HTTP/1.1
```

Z tego wyniku wydaje się, że adres IP `208.68.234.99` miał dostęp wyłącznie do katalogu `/admin`. Przyjrzyjmy się temu bliżej:

```
root@kali:~# cat access.log | grep '208.68.234.99' | grep '/admin ' |
sort -u 208.68.234.99 - - [22/Apr/2013:07:51:20 -0500] "GET //admin
HTTP/1.1" 401 742 "--" "Teh Forest Lobster"
...
208.68.234.99 -admin [22/Apr/2013:07:51:25 -0500] "GET //admin HTTP/1.1"
200 575 "-" "Teh Forest Lobster"
...
root@kali:~# cat access.log|grep '208.68.234.99'| grep -v '/admin '
root@kali:~#
```

Wygląda na to, że `208.68.234.99` było zaangażowanych w próbę ataku typu `HTTP Brute Force` przeciwko temu serwerowi WWW. Co więcej, wygląda na to, że po około 1070 próbach atak zakończył się powodzeniem, na co wskazuje komunikat `HTTP 200 OK`.

Ćwiczenia do wykonania

1. Przygotuj skrypt Bash, aby zrealizować zadanie sprawdzające dostępność (aktywność) hostów IP wewnątrz podsieci, w której znajduje się także Twój host. Podpowiedzi:
   - Adresacja/konfiguracja sieciowa Twojego urządzenia jest uzyskiwana za pomocą komeny `ifconfig`
   - Aktywność hosta pod danym adresem IP realizowana jest poleceniem `ping <adres IP>` (podstawowa wersja, w wersji rozszerzonej mamy `ping <opcje> <adres IP>`, gdzie `<opcje>` służą do modyfikacji wysyłanych żądań)
   - Automatyzacja powtarzalnego zadania realizowana jest za pomocą pętli np. `for`
   - Kolejne wartości (sekwencja) można otrzymać z za pomoca polecenia `seq`

1.5 Podsłuchiwanie ruchu sieciowego

Jednym z zagrożeń w sieciach jest podsłuchiwanie ruchu sieciowego i wyciąganie z niego istotnych dla atakującego informacji takich jak: loginy, hasła, prywatne dane itp. Zdarza się też, że atakujący stara się utrudniać lub uniemożliwiać pracę atakowanego serwera np. poprzez blokadę ruchu, rozsyłanie szkodliwych pakietów, tworzenie nadmiernego ruchu sieciowego, wykorzystywanie błędów w zabezpieczeniach serwerów poprzez wstrzykiwanie złośliwych danych (np. SQL Injection) lub przeciążenia serwera - atak typu Denial of Service (DoS).

Podczas podsłuchiwania atakujący monitoruje ruch w sieci, który jest przesyłany przez jego interfejs sieciowy, odczytując w ten sposób nieswoje pakiety. Używa się do tego programów zwanych snifferami. W naszym przypadku skupimy się na dwóch najbardziej popularnych: `Wireshark` i `tcpdump` znajdujących się domyślnie w systemie operacyjnych Kali Linux.

Z punktu widzenia specjalistów cyberbezpieczeństwa podsłuch ruchu sieciowego służy do:

- odkrywania słabych punktów, niezapezpieczionych połączeń i danych, błędnych konfiguracji w ramach badań bezpieczeństwa systemów, aplikacji i infrastruktury (*security assessment*)
- weryfikacjia konfiguracji połączeń oraz ustawień urządzeń sieciowych, w tym urządzeń *security*, takie jak firewalle

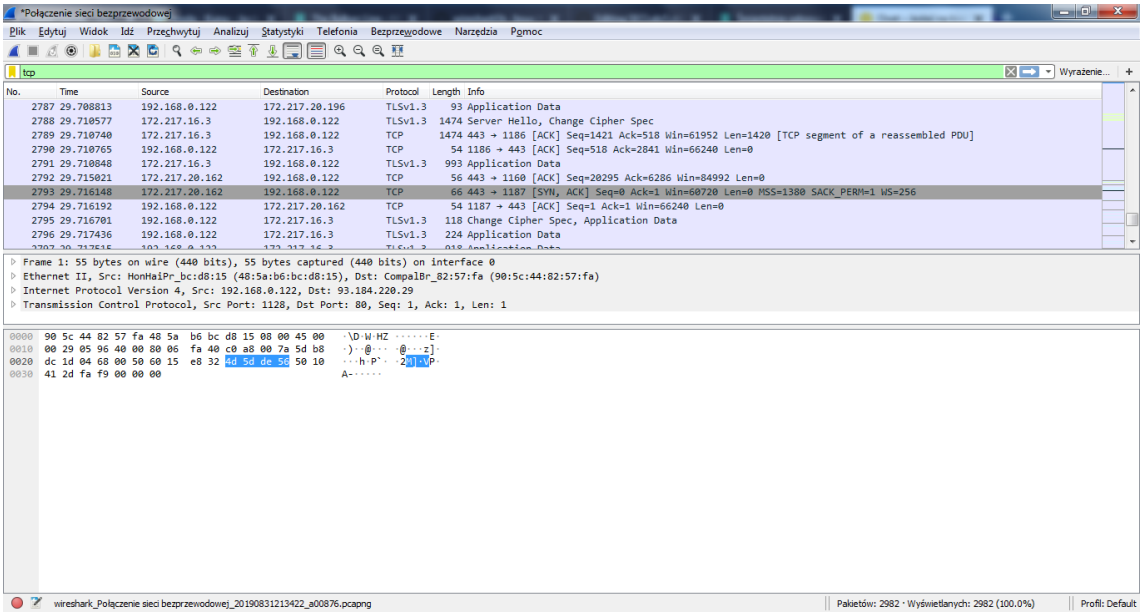
- diagnozowanie problemów
- monitorowania pasywnego i wykorzystywania zebranych logów (zrzutów ruchu sieciowego) do odkrywania wykonywania się ataków czy anomalii sugerujących np. zaszycie się atakującego w naszej sieci.

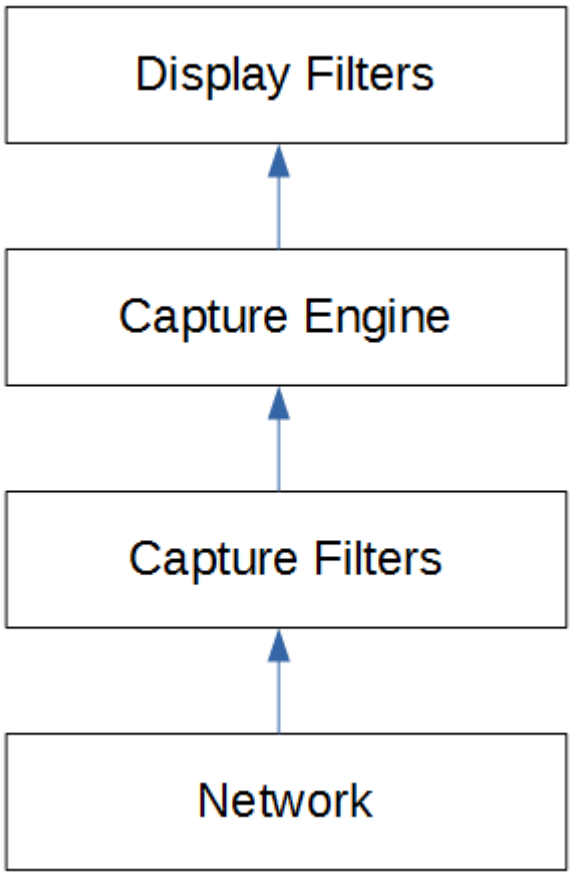
1.5.1 Wireshark

Korzystania z sieciowego snifera pakietów jest bardzo ważne w codziennych operacjach specjalistów cyberbezpieczeństwa - ofensywnych i defensywnych. Niezależnie od tego, czy próbujesz zrozumieć protokół, debugować klienta sieciowego, czy analizować ruch, zawsze będziesz potrzebować sniffera.

1.5.1.1 Podstawy

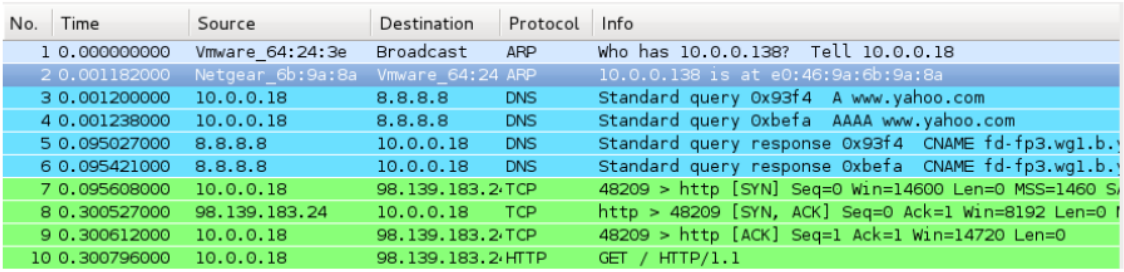
Wireshark korzysta z bibliotek `libpcap` (w systemie Linux) lub `winpcap` (w systemie Windows) w celu przechwytywania pakietów z sieci. Jeśli użytkownik zastosuje filtry przechwytywania (*capture filters*) dla sesji Wireshark, przefiltrowane pakiety zostaną odrzucone i tylko odpowiednie dane zostaną przekazane do silnika przechwytywania. Mechanizm przechwytywania analizuje przychodzące pakiety, a następnie stosuje dodatkowe filtry wyświetlania(*display filters*) przed wyświetleniem danych wyjściowych użytkownikowi. Sekret używania snifferów sieciowych, takich jak `Wireshark`, polega na użyciu filtrów przechwytywania i wyświetlania w celu usunięcia wszystkich informacji, które Cię nie interesują.





1.5.1.2 Analizowanie zrzutów ruchu sieciowego

Przeanalizujmy zrzut ruchu sieciowego w pliku w formacie `pcap` wykonanego podczas przeglądania strony `www.yahoo.com`.

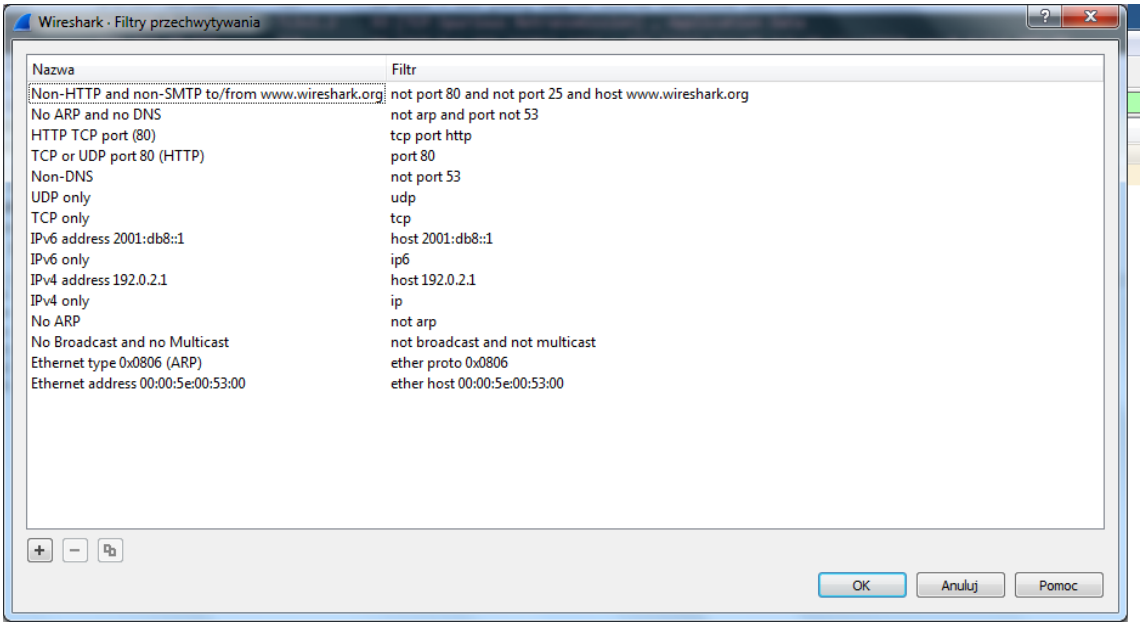


No.	Time	Source	Destination	Protocol	Info
1	0.000000000	Vmware_64:24:3e	Broadcast	ARP	Who has 10.0.0.138? Tell 10.0.0.18
2	0.001182000	Netgear_6b:9a:8a	Vmware_64:24	ARP	10.0.0.138 is at e0:46:9a:6b:9a:8a
3	0.001200000	10.0.0.18	8.8.8.8	DNS	Standard query 0x93f4 A www.yahoo.com
4	0.001238000	10.0.0.18	8.8.8.8	DNS	Standard query 0xbefa AAAA www.yahoo.com
5	0.095027000	8.8.8.8	10.0.0.18	DNS	Standard query response 0x93f4 CNAME fd-fp3.wg1.b.
6	0.095421000	8.8.8.8	10.0.0.18	DNS	Standard query response 0xbefa CNAME fd-fp3.wg1.b.
7	0.095608000	10.0.0.18	98.139.183.2	TCP	48209 > http [SYN] Seq=0 Win=14600 Len=0 MSS=1460 S
8	0.300527000	98.139.183.24	10.0.0.18	TCP	http > 48209 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
9	0.300612000	10.0.0.18	98.139.183.2	TCP	48209 > http [ACK] Seq=1 Ack=1 Win=14720 Len=0
10	0.300796000	10.0.0.18	98.139.183.2	HTTP	GET / HTTP/1.1

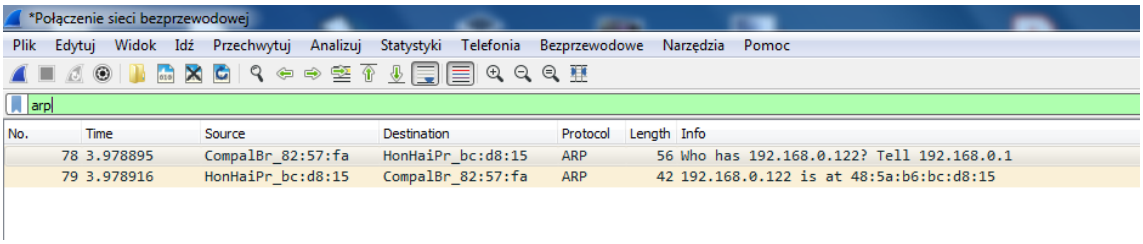
Numer pakietu	**Objaśnienie**
1	Brodcast ARP do domyślnej bramy
2	Odpowiedź unicast ARP zawierajaca adres MAC bramy
3	DNS A (IPv4) wyszukiwanie wprzód dla yahoo.com
4	DNS AAAA (IPv) zapytanie wyszukiwania wprzód
5	DNS A uzyskanie odpowiedzi
6	DNS AAAA uzyskanie odpowiedzi
7-9	Uzgadnianie sesji TCP z portem 80 yahoo.com
10	Początkowa negocjacja protokołu w HTTP. Wysłanie zapytania GET

1.5.1.3 Filtry przechwytywania i wyświetlania

Zrzuty przechwytywania rzadko są tak wyraźne, jak w powyższym przykładzie, ponieważ w sieci zwykle występuje duży ruch tła. Różne transmisje, różne usługi sieciowe i inne działające aplikacje utrudniają analizę ruchu. W tym miejscu pomagają filtry przechwytywania (*capture filters*), które mogą odfiltrować nieistotny ruch jeszcze przed wykonaniem zrzutu. Filtry te znacznie pomagają w określeniu pożądanego ruchu i zmniejszeniu niepożądanego do momentu, w którym możemy bez problemu rozumieć pojawiające się pakiety.

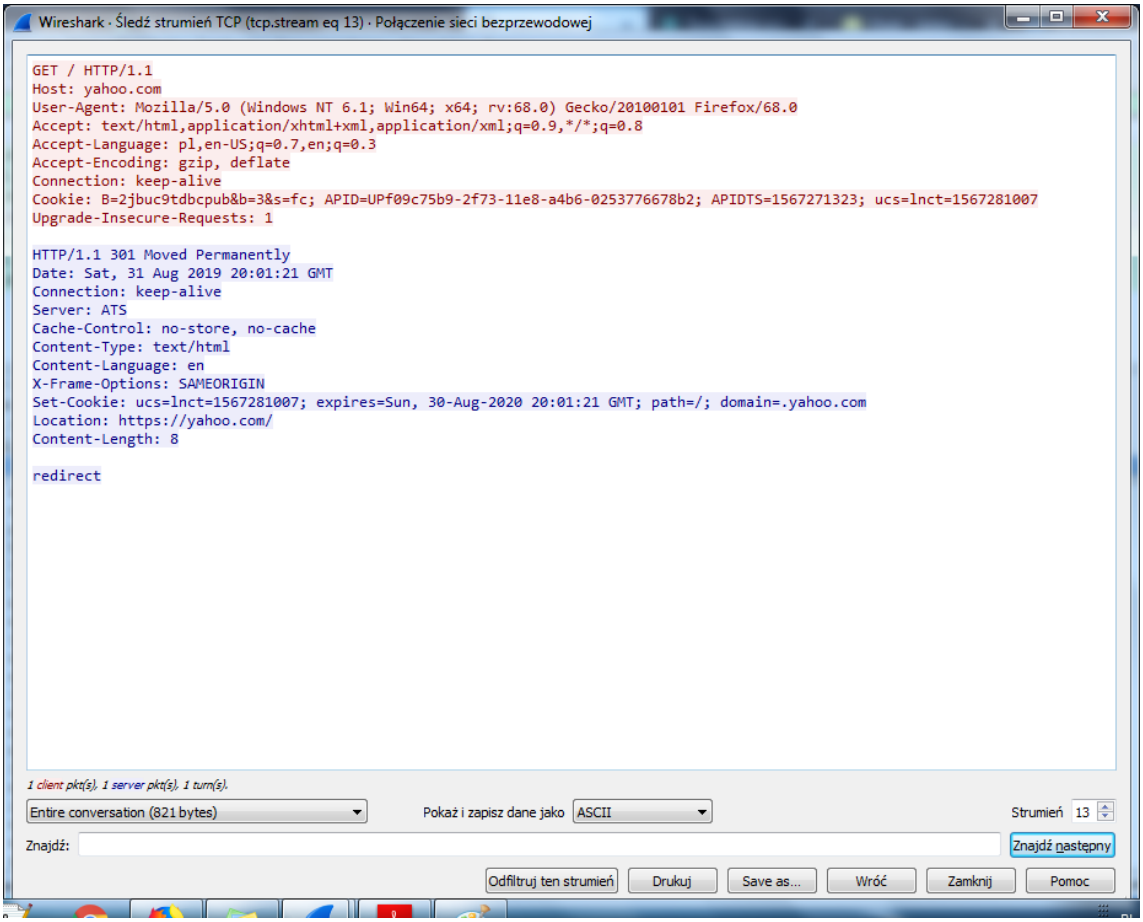


Po przechwyceniu ruchu możemy wybrać ruch, który `Wireshark` ma nam wyświetlać za pomocą filtrów wyświetlania (*display filters*). Poniższy zrzut ekranu pokazuje filtr wyświetlania `arp` zastosowany do naszej sesji przeglądania `yahoo.com`.



1.5.1.4 Śledzenie strumienia TCP

Często zdarza się, że pojedyncze pakiety z analizowanego ruchu sieciowego są trudne do zrozumienia, ponieważ zawierają tylko fragment informacji z całego strumienia do którego należą. Większość dostępnych snifferów, w tym Wireshark umie złożyć pojedyncze pakiety w konkrete sesje i wyświetlić ją w różnych formatach. Aby wyświetlić konkretny strumień `TCP`, kliknij PPM interesujący Cię pakiet, a następnie wybierz "Podążaj (Follow)" i "Strumień TCP" (Follow TCP Stream) z menu kontekstowego. Strumień TCP otworzy nowe okno, jak pokazano poniżej.



Ćwiczenia do wykonania

1. Użyj Wireshark do przechwycenia aktywności sieci podczas próby logowania na swoją uczelnianą skrzynkę pocztową.
2. W strumieniu TCP odnajdź sekwencję logowania.
3. Użyj filtra wyświetlania, aby zobaczyć tylko ruch na porcie 443
4. Uruchom ponownie przechwytywanie, tym razem za pomocą filtra przechwytywania, aby zebrać tylko port 443.

1.5.2 Tcpdump

Czasami możemy nie mieć dostępu do graficznych interfejsów snifferów sieciowych, takich jak `Wireshark`. W takich przypadkach możemy użyć narzędzia `tcpdump` z wiersza poleceń. `tcpdump` jest jednym z najpopularniejszych analizatorów pakietów wiersza poleceń i można go znaleźć w większości systemów operacyjnych Linux. `tcpdump` może przechwytywać pliki z sieci lub czytać istniejące pliki przechwytywania. Spójrzmy na to, co się stało w pliku `pcap` **password_cracking_filtered** (https://www.offensive-security.com/pwk-online/password_cracking_filtered.pcap), który został pobrany na zaporze ogniowej (*firewall*).

```
root@kali:~# tcpdump -r password_cracking_filtered.pcap
```

1.5.2.1 Filtrowanie ruchu

Dane wyjściowe są początkowo nieco przytłaczające, dlatego spróbujmy lepiej zrozumieć adresy IP i porty, używając poleceń `awk` i `sort`.

```
root@kali:~# tcpdump -n -r password_cracking_filtered.pcap | awk -F" " '{print $3}' |
sort -u | head
172.16.40.10.81
208.68.234.99.32768
208.68.234.99.32769
208.68.234.99.32770
208.68.234.99.32771
208.68.234.99.32772
208.68.234.99.32773
...
```

Wygląda na to, że `208.68.234.99` wysłało wiele żądań do `172.16.40.10` na porcie 81. Możemy łatwo filtrować na podstawie adresu IP docelowego lub źródłowych i portów z wykorzystaniem składni podobnej do następującej:

```
tcpdump -n src host 172.16.40.10 -r password_cracking_filtered.pcap
tcpdump -n dst host 172.16.40.10 -r password_cracking_filtered.pcap
tcpdump -n port 81 -r password_cracking_filtered.pcap
```

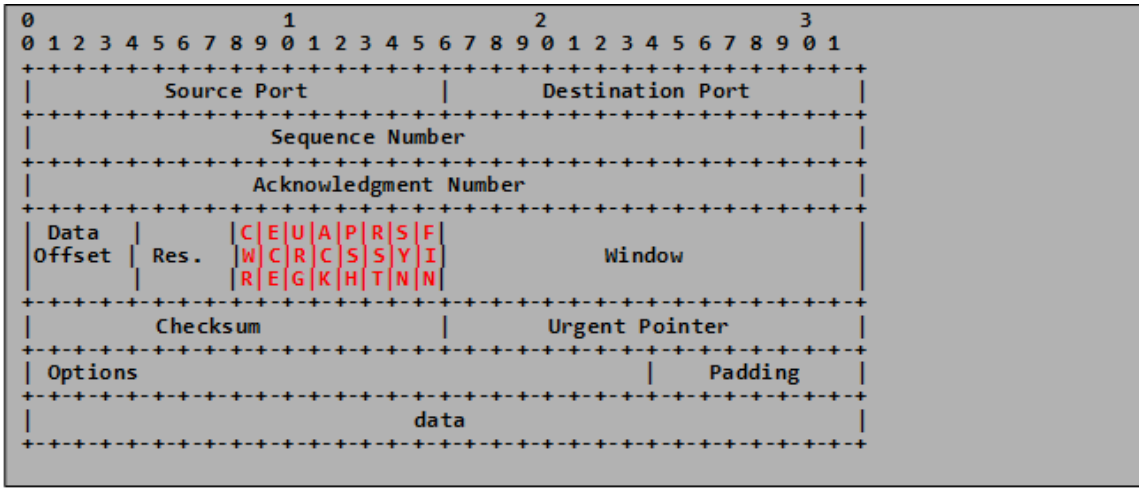
Zacznijmy od analizy ruchu przechwyconego w pliku zrzutu, w formacie szesnastkowym, aby sprawdzić, czy możemy uzyskać dodatkowe informacje z przesłanych danych:

```
root@kali:~# tcpdump -nX -r password_cracking_filtered.pcap
...
08:51:25.043062 IP 208.68.234.99.33313 > 172.16.40.10.81: Flags [P.], seq 1:140,
ack 1, win 115, options [nop,nop,
TS val 25539314 ecr 71431651], length 139
0x0000: 4500 00bf 158c 4000 3906 9cea d044 ea63 E.....@.9....D.c
0x0010: ac10 280a 8221 0051 a726 a77c 6fd8 ee8a ..(..!.Q.&.|...
0x0020: 8018 0073 1c76 0000 0101 080a 0185 b2f2 ...s.v..........
0x0030: 0441 f5e3 4745 5420 2f2f 6164 6d69 6e20 .A..GET.//admin.
0x0040: 4854 5450 2f31 2e31 0d0a 486f 7374 3a20 HTTP/1.1..Host:.
0x0050: 6164 6d69 6e2e 6d65 6761 636f 7270 6f6e admin.megacorpon
0x0060: 652e 636f 6d3a 3831 0d0a 5573 6572 2d41 e.com:81..User--A
0x0070: 6765 6e74 3a20 5465 6820 466f 7265 7374 gent:.Teh.Forest
0x0080: 204c 6f62 7374 6572 0d0a 4175 7468 6f72 .Lobster..Author
0x0090: 697a 6174 696f 6e3a 2042 6173 6963 2059 ization:.Basic.Y
0x00a0: 5752 7461 5734 3662 6d46 7562 3352 6c59 WRtaW46bmFub3RlY
0x00b0: 3268 7562 3278 765a 336b 780d 0a0d 0a 2hub2xvZ3kx....
...
```

Natychmiast możemy zauważyć, że ruch do `172.16.40.10` na porcie 81 wygląda jak `HTTP`. Co więcej, wygląda na to, że te żądania `HTTP` zawierają podstawowe dane uwierzytelniające z nagłówkiem `HTTP User-Agent: "Teh Forest Lobster"`.

1.5.2.2 Zaawansowane filtrowanie nagłówków

`tcpdump` ma kilka zaawansowanych opcji filtrowania nagłówków, które mogą nam pomóc w naszej analizie pcap. Chcielibyśmy odfiltrować i wyświetlić tylko te zrzuty danych, które mają włączone flagi PSH i ACK. Jak widać na poniższym diagramie, flagi `TCP` są zdefiniowane w 14-stym bajcie nagłówka `TCP`.



Aby określić właściwy filtr do użycia, włączamy bity dla konkretnych flag, których potrzebujemy, w tym przykładzie flagi ACK i PSH:

```
CEUAPRSF
00011000 = 24 in decimal
```

Nasze polecenie wyglądałoby podobnie do następującego - podając, że czternasty bajt w wyświetlanych pakietach powinien mieć ustawione flagi ACK lub PSH:

```
root@kali:~# tcpdump -A -n 'tcp[13] = 24' -r password_cracking_filtered.pcap
...
08:51:25.040064 IP 172.16.40.10.81 > 208.68.234.99.33312
A......HTTP/1.1 401 Authorization Required
Date: Mon, 22 Apr 2013 12:51:25 GMT
Server: Apache/2.2.20 (Ubuntu)
WWW-Authenticate: Basic realm="Password Protected Area"
Vary: Accept-Encoding
Content-Length: 488 Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Authorization Required</title>
</head><body>
<h1>Authorization Required</h1>
<p>This server could not verify that you
are authorized to accessthe document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.2.20 (Ubuntu) Server at admin.megacorpone.com Port
81</address>
</body></html>
...
08:51:25.044432 IP 172.16.40.10.81 > 208.68.234.99.33313:
E..s.m@.@..U..(
.D.c.Q.!o....&......^u.....
.A......HTTP/1.1 301 Moved Permanently
Date: Mon, 22 Apr 2013 12:51:25 GMT
Server: Apache/2.2.20 (Ubuntu)
Location: http://admin.megacorpone.com:81/admin/
Vary: Accept-Encoding
Content-Length: 333
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a
href="http://admin.megacorpone.com:81/admin/">here</a>.</p>
<hr>
<address>Apache/2.2.20 (Ubuntu) Server at admin.megacorpone.com Port
81</address>
</body></html>
```

Odtąd historia staje się jaśniejsza. Widzimy znaczną liczbę nieudanych prób uwierzytelnienia w katalogu `/admin`, na które wysłano odpowiedzami HTTP 401, podczas gdy ostatnia próba zalogowania się do katalogu `/admin` wydaje się być udana, ponieważ serwer odpowiedział odpowiedzią HTTP 301 .

Ćwiczenia do wykonania

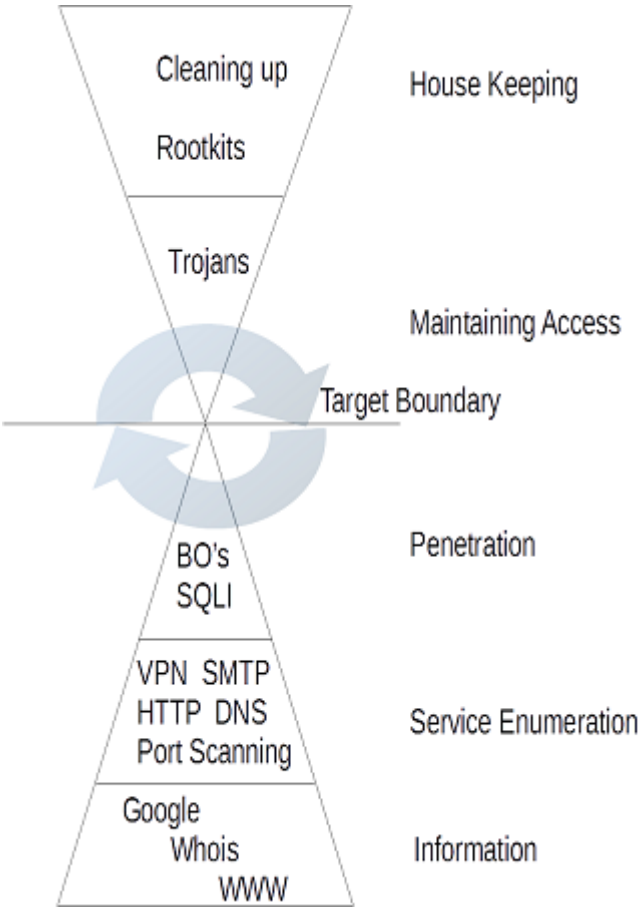
1. Przećwicz działanie tcpdumpa zgodnie z punktem 1.5.2
2. Użyj flagi -X, aby wyświetlić zawartość pakietu. Jeśli dane są obcięte, sprawdź, w jaki sposób flaga --s może pomóc.

2. Cyberbezpieczeństwo ofensywne - podstawy testów penetracyjnych

2.1 O testach penetracyjnych

Testy penetracyjne (*pentesty*) stanowią podstawę współcześnie określanego cyberbezpieczeństwa ofensywnego. Stanowią ciągły cykl sprawdzania podatności (*vulnerability assessments*), ataków na cel (*targeting*) i prób wykorzystania podatności (*exploiting*). Najczęściej proces taki trwa określony czas, więc należy utrzymywać dostęp do testowanych środowisk (*backdoors*). Odpowiada to także modelowi Cyber Kill Chain, w którym pojedyncza operacja testu penetracyjnego symuluje wykonanie każdego z etapów Cyber Kill Chain po kolei. Rezultatem testów penetracyjnych ma być ocena bieżącego stanu systemów, aplikacji i infrastruktury Zamawiającego (w określonym z nim zakresie i zgodnie z umową). Wiąże się to także często z szerszym pojęciem badań bezpieczeństwa (*security assessments*), dla których testy penetracyjne mogą stanowić jeden z etapów, który ma wyznaczyć dziury w systemach i pozwola oszacować ryzyko zagrożeń cyberbezpieczeństwa.

Ataki powinny być ustrukturyzowane i obliczone na cel, a jeśli to możliwe, zweryfikowany w laboratorium przed wdrożeniem go na żywym celu. Oto jak wizualizujemy proces testu penetracyjnego:



Jak sugeruje model, im więcej informacji zbierzemy, tym większe prawdopodobieństwo udanego ataku. Po przekroczeniu początkowej granicy celu zwykle rozpoczynamy cykl ponownie - na przykład zbierając informacje o sieci wewnętrznej w celu jej głębszej penetracji. To w jaki sposób przeprowadzać dane etapy zależy od przyjętej metodologii. Istnieje kilka metodologii przeprowadzania testów bezpieczeństwa (np. OWASP czy OSSTMM) jednak specjaliści ds. bezpieczeństwa opracowują często dedykowane metodologie - na własny użytek, na potrzeby Klienta, wykorzystując w nich istniejące standardy.

2.2 Rekonesans

Rekonesans jest pierwszym etapem testu penetracyjnego. Jego celem jest nieagresywne zbieranie informacji (czyli takie, które nie powoduje naruszeń polityki bezpieczeństwa) na temat badanej organizacji.

Zbieranie informacji można może być realizowane na dwa sposoby: aktywny i pasywny. Aktywne zbieranie informacji ma miejsce podczas wprowadzania ruchu sieciowego do sieci organizacji podlegającej badaniu. Innymi słowy oznacza to aktywne bezpośrednie oddziaływanie na sieć badanej organizacji, co już same w sobie może być uznane jako atak hakerski. W przypadku techniki pasywnej, informacje są gromadzone poprzez wykorzystanie usług firm trzecich, takich jak np. różne wyszukiwarki, m.in. Google. Podczas pasywnego rozpoznania nie są wysłane dane do systemów docelowych. Źródłem wiedzy potencjalnego napastnika są ogólnodostępne zasoby. Określa się to także mianiem *OSINT* - **o**pen-**s**ource **int**elligence.

2.3 Pasywne zbieranie informacji

Pasywne zbieranie informacji to proces gromadzenia informacji o celach za pomocą publicznie dostępnych informacji. Może to obejmować usługi takie jak wyniki wyszukiwania, informacje Whois, informacje pochodzące z usług, informacje o spółkach publicznych itp. Innymi słowy, czynność gromadzenia informacji o celu bez bezpośredniej komunikacji z nimi można uznać za "pasywny". Im więcej informacji uda nam się zebrać na temat naszego celu przed atakiem, tym większe prawdopodobieństwo, że odniesiemy sukces.

Dobrym przykładem pasywnego gromadzenia informacji jest przypadek podczas testu penetracyjnego w małej firmie kilka lat temu. Firma ta praktycznie nie była obecna w Internecie i miała mało zewnętrznych usług, które okazały się bezpieczne. Po wielu godzianch przeszukiwania Google w końcu udało się znaleźć post na forum kolekcjonerów znaczków napisany przez jednego z pracowników:

```
Hi!

I am looking for rare stamps from the 1950's -  for sale or trade.
Please contact me at david@company-address.com
Cell: 999-9999999
```

To były wszystkie informacje, które były potrzebne, aby przeprowadzić częściowo zaawansowany atak po stronie klienta. Szybko zarejestrowano domenę, taką jak `rare-stamps-trade.com` i zaprojektowano stronę docelową, która wyświetlała różne rzadkie znaczki z lat 50. XX wieku, które można znaleźć za pomocą wyszukiwarki grafiki Google. Zarówno nazwa domeny, jak i projekt strony doprowadziły do zwiększenia postrzeganej wiarygodności strony internetowej z znaczkami. Następnie przystąpiono do osadzania złośliwego kodu HTML w kodzie witryny, zawierającego kod wykorzystujący najnowszą lukę w zabezpieczeniach programu Internet Explorer (w tym czasie MS05-001) i zadzwoniono do Davida na jego telefon komórkowy. Powiedziano mu, że dziadek atakującego dał mu ogromną kolekcję rzadkich znaczków, z której możliwa jest wymiana kilku znaczków. Zadbano o to, aby zadzwonić w ciągu dnia roboczego, aby zwiększyć szanse atakującego na dotarcie do Davida w biurze. David był bardzo szczęśliwy, że otrzymał takie wezwanie i bez wahania odwiedził złośliwą stronę internetową, aby zobaczyć „znaczki”, które atakujący miał do zaoferowania. Podczas przeglądania strony kod exploita na

stronie internetowej pobrał i wykonał "ładunek podobny do Netcata" na swojej lokalnej maszynie, odsyłując atakującemyu powłokę zwrotną (*reverse shell*). Jest to dobry przykład tego, jak niektóre nieszkodliwe informacje, takie jak pracownik łączący swoje życie osobiste z firmową pocztą e-mail, mogą doprowadzić do udanego ataku. Ponadto przedstawiony przykład to praktyczne zastosowanie technik inżynierii społecznej - *spear phishing* - na etapie rekonesansu oraz dostarczenia narzędzi ataku do celu (*Delivery - Cyber Kill Chain*). Opisany sposób oszukiwania ludzi i w ten sposób dostarczanie złośliwych aplikacji do organizacji stanowi jeden z największych problemów współczesnych systemów teleinformatycznych i komputerowych.

Zbieranie iformacji jest najważniejszym etapem testu penetracyjnego. Znajomość celu przed atakiem to sprawdzony przepis na sukces. Nawet przyziemne posty na forum mogą dostarczyć przydatnych informacji.

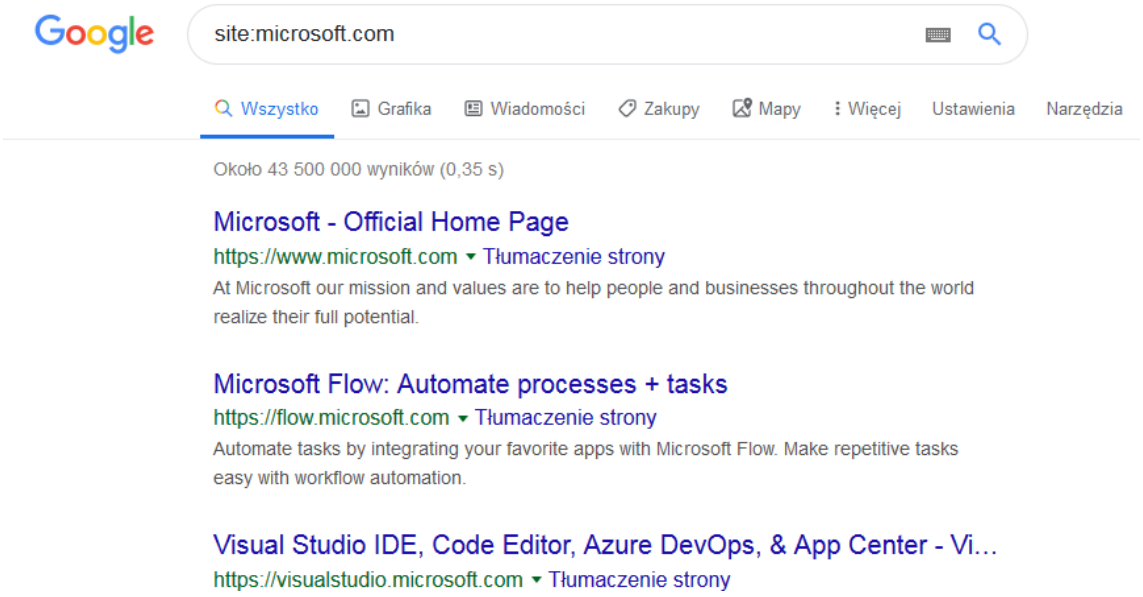
2.3.1 Zbieranie informacji dostępnych w Internecie

Na początku pozyskiwania informacji najpierw poświęcić trochę czasu na przeglądanie sieci, szukając dodatkowych informacji o organizacji docelowej. Czym się zajmuje? Jak wygląda punkt styky ze swiatem? Czy mają dział sprzedaży? Czy sami zatrudniają? Przeglądaj witrynę organizacji i poszukaj ogólnych informacji, takich jak dane kontaktowe, numery telefonu i faksu, e-maile, struktura firmy i tak dalej. Pamiętaj też, aby poszukać witryn, które prowadzą do strony docelowej lub e-maili firmowych krążących się w Internecie. Czasami są to najdrobniejsze szczegóły, które dają najwięcej informacji: jak dobrze zaprojektowana jest docelowa witryna? Jak czysty jest ich kod HTML? Może to dać wskazówkę co do ich budżetu na tworzenie stron internetowych, co może wpłynąć na budżet bezpieczeństwa.

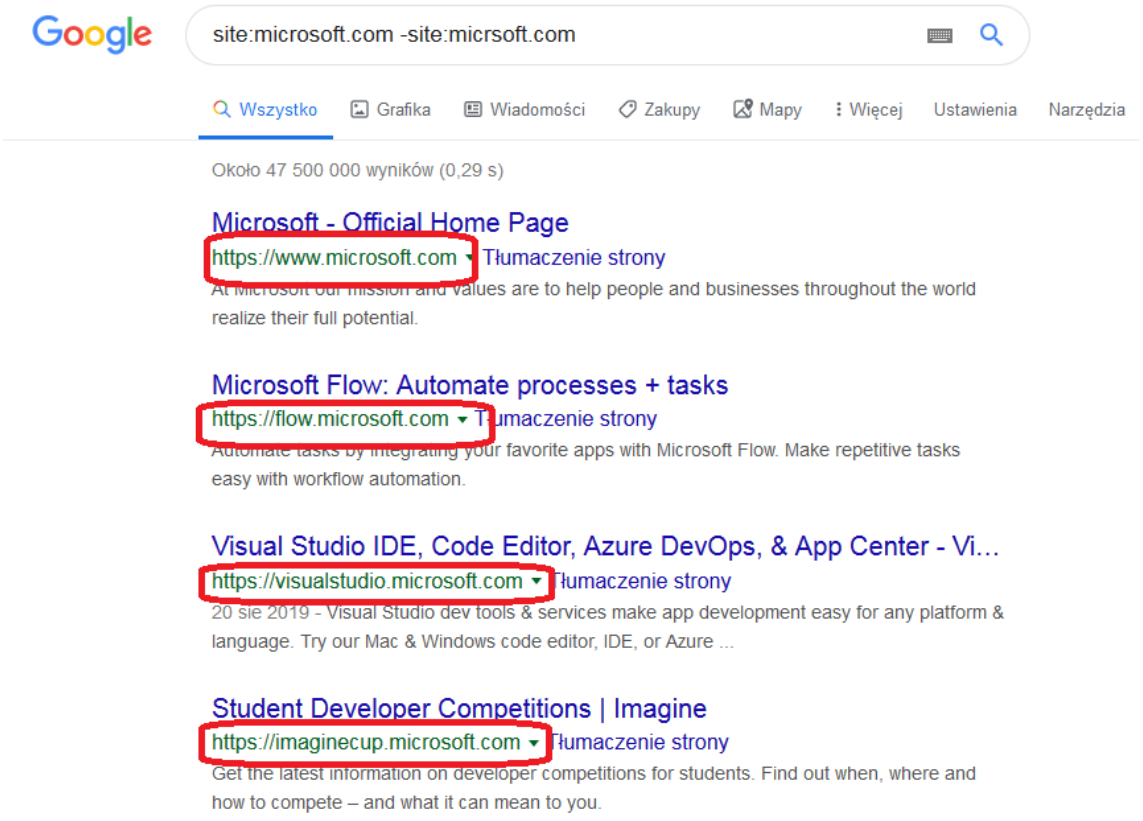
2.3.1.1 Enumeracja z Google

Wyszukiwarka Google jest najlepszym przyjacielem audytora bezpieczeństwa, szczególnie jeśli chodzi o zbieranie informacji.

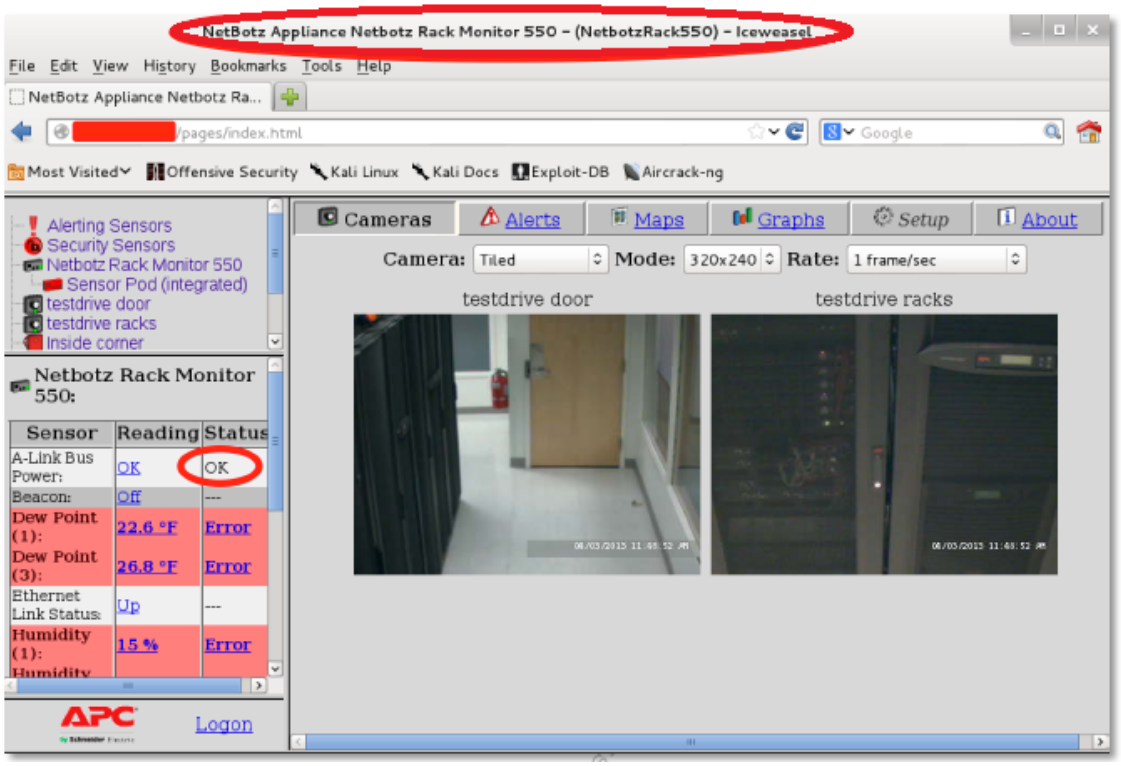
oogle obsługuje korzystanie z różnych operatorów wyszukiwania, które pozwalają użytkownikowi zawęzić i wskazać wyniki wyszukiwania. Na przykład operator **site** ograniczy wyniki wyszukiwania Google do jednej domeny. Prosty operator wyszukiwania taki jak ten dostarcza nam przydatnych informacji. Powiedzmy na przykład, że chcemy poznać przybliżoną obecność organizacji w sieci przed rozpoczęciem zaangażowania.



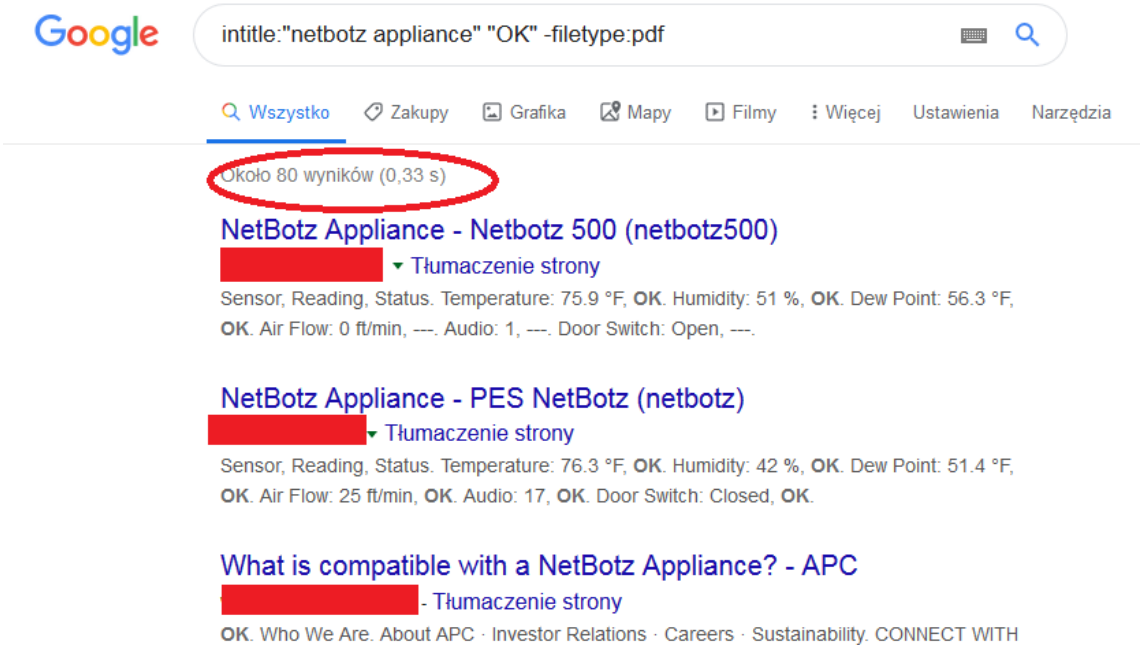
W powyższym przykładzie użyliśmy parametru **site**, aby ograniczyć wyniki wyświetlane przez Google tylko do domeny *microsoft.com*. Tego dnia Google zaindeksował około 67 milionów stron z domeny *microsoft.com*. Zauważ, że większość wyników, które do nas wracają, pochodzi z subdomeny www.microsoft.com. Odfiltrujmy je, aby zobaczyć, jakie inne subdomeny mogą istnieć na *microsoft.com*.



Te dwa proste zapytania ujawniły sporo podstawowych informacji o domenie *microsoft.com*, takich jak daned na temat ich obecności w Internecie i lista ich poddomen dostępnych w Internecie. Oczywiście to tylko jeden operator wyszukiwania, a jest ich znacznie więcej. Przykładami innych operatorów wyszukiwania są **filetype**, **inurl** i **intitle** . Na przykład wspólny system wideo w serwerowni ma następującą stronę domyślną.



Zwróć uwagę, w jaki sposób to urządzenie wideo zapewnia unikalny ozbaczenie tytułu - urządzenie Netbotz, a także numer modelu. Za pomocą kilku prostych wyszukiwań w Google możemy zawęzić wyniki wyszukiwania, tak aby obejmowały tylko te urządzenia.

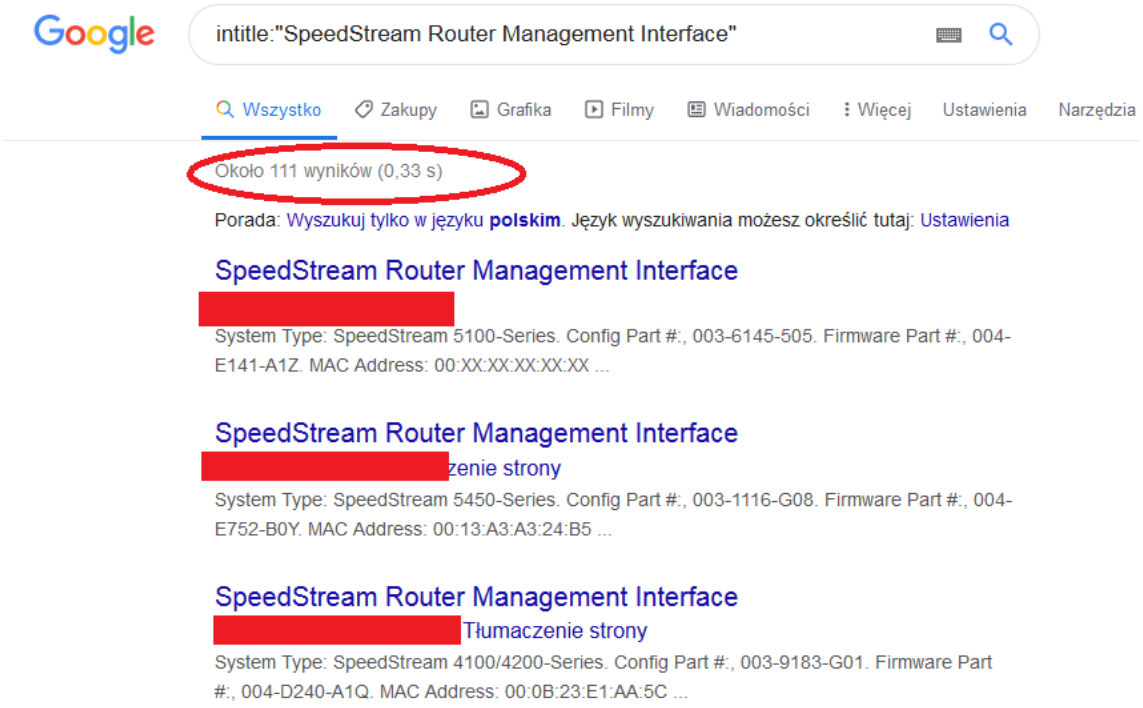


Przykłady specyficzne dla produktu, takie jak te, są z natury dynamiczne i mogą nie dać żadnych wyników dla tego konkretnego urządzenia w ciągu najbliższych kilku miesięcy. Jednak koncepcja tego typu wyszukiwań jest taka sama. Jeśli wiesz, jak efektywnie korzystać z operatorów wyszukiwania Google i wiesz dokładnie, czego szukasz, możesz znaleźć prawie wszystko.

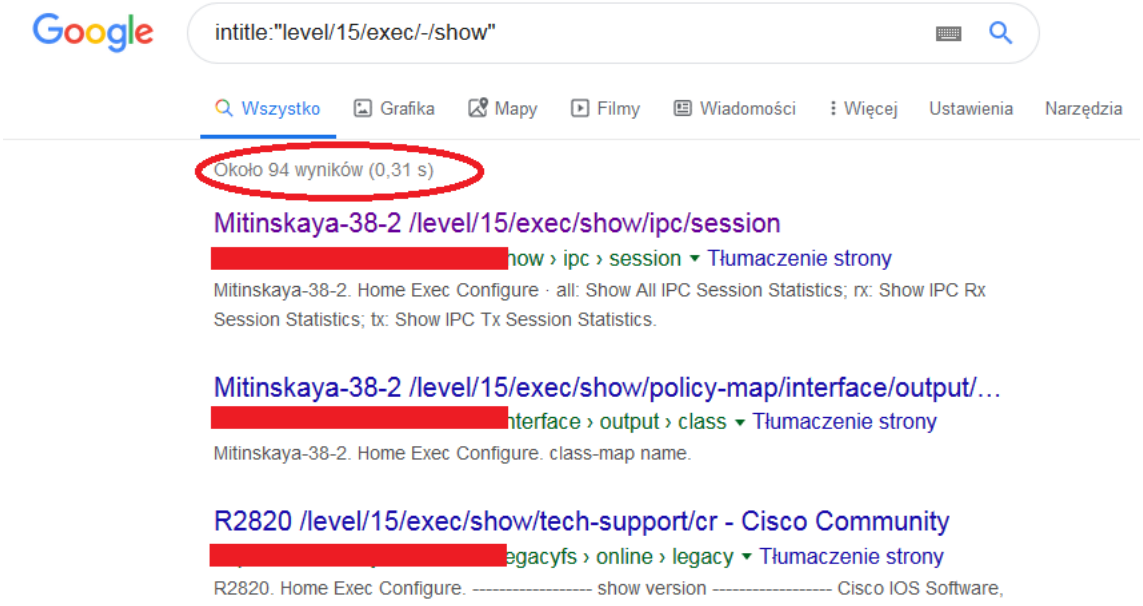
2.3.1.2 Google Hacking

Używanie Google do znajdowania ciekawych informacji, luk w zabezpieczeniach lub źle skonfigurowanych witryn zostało publicznie wprowadzone przez Johnny'ego Longa w 2001 roku. Od tego czasu opracowano bazę danych interesujących wyszukiwań, aby umożliwić audytorom bezpieczeństwa (i hakerom) szybkie wykrycie licznych nieprawidłowych konfiguracji w obrębie danej domeny. Kolejne zrzuty ekranu pokazują takie wyszukiwania.

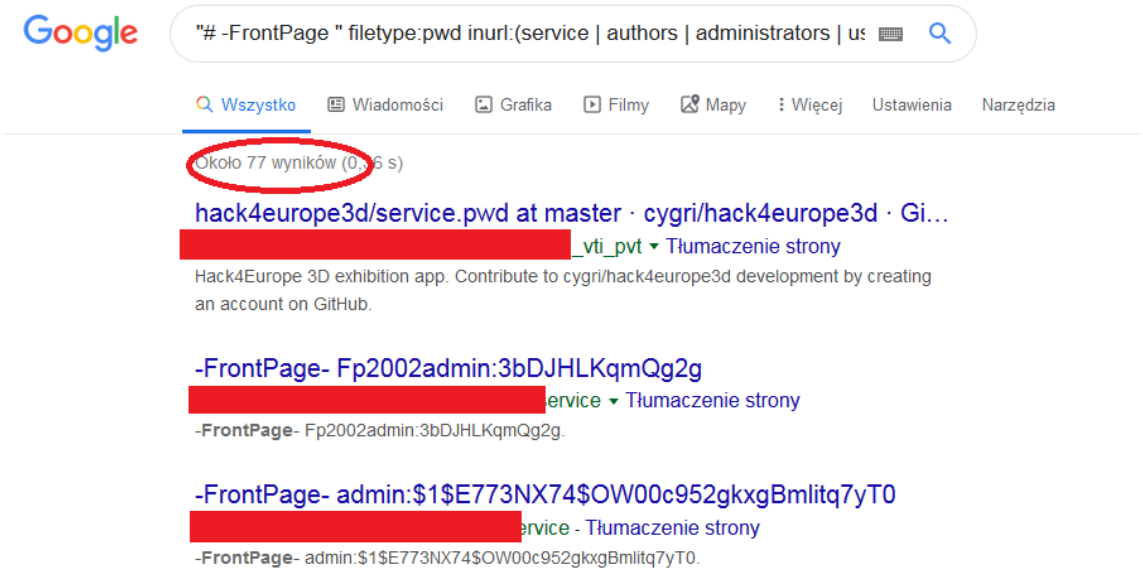
2.3.1.2.1 Sprzęt ze znanymi podatnściami



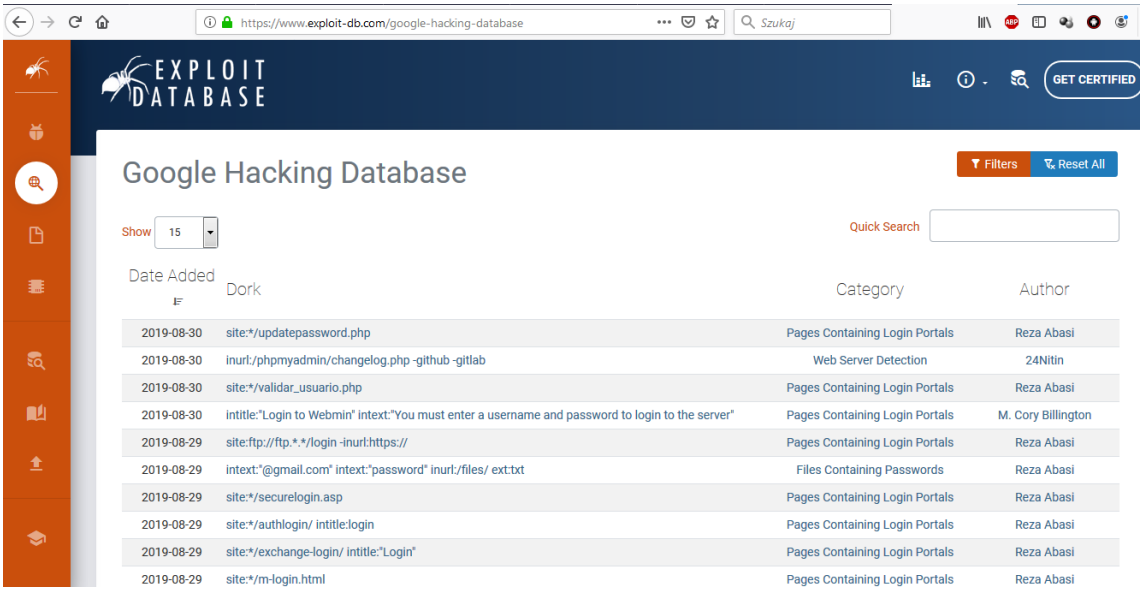
2.3.1.2.2 Dostępne z sieci routery Cisco



2.3.1.2.3 Ujawnione poświadczenia do logowania



Istnieją setki ciekawych wyszukiwań, z których wiele można znaleźć w bazie Google Hacking (GHDB)



2.3.1.2 Pozyskiwanie e-mail

Zbieranie wiadomości e-mail to skuteczny sposób znajdowania wiadomości e-mail i ewentualnie nazw użytkowników należących do organizacji. Wiadomości e-mail są przydatne na wiele sposobów, na przykład dostarczając nam potencjalną listę ataków po stronie klienta, ujawniając konwencję nazewnictwa używaną w organizacji lub mapując użytkowników w organizacji. Jednym z narzędzi w Kali Linux, które może wykonać to zadanie, jest **theharvester**. Narzędzie to może wyszukiwać adresy e-mail w Google, Bing i innych witrynach, korzystając ze składni przedstawionej poniżej

```
root@kali:~# theharvester -d cisco.com -b google >google.txt
root@kali:~# theharvester -d cisco.com -l 10 -b bing >bing.txt
```

2.3.1.2.1 Ćwiczenia do wykonania

1.Użyj **theharvester**, aby wylistować adresy e-mail należące do organizacji wybranej w poprzednich ćwiczeniach. 2. Eksperymentuj z różnymi źródłami danych (**-b**). Która jest dla Ciebie najlepsze?

2.3.1.3 Enumeracja za pomocą Whois

Whois to nazwa usługi TCP, narzędzia i rodzaju bazy danych. Bazy danych Whois zawierają serwer nazw, rejestr oraz, w niektórych przypadkach, pełne dane kontaktowe dotyczące nazwy domeny. Każdy rejestr musi prowadzić bazę danych Whois zawierającą wszystkie dane kontaktowe dla domen, które prowadzą. Centralna baza danych Whois rejestru jest prowadzona przez InterNIC. Te bazy danych są zwykle publikowane przez serwer Whois za pośrednictwem portu TCP 43 i są dostępne za pomocą programu klienta **whois**.

```
root@kali:~# whois megacorpone.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Domain Name: MEGACORPONE.COM
Registrar: GANDI SAS
Whois Server: whois.gandi.net
Referral URL: http://www.gandi.net
Name Server: NS1.MEGACORPONE.COM
Name Server: NS2.MEGACORPONE.COM
Name Server: NS3.MEGACORPONE.COM
Status: clientTransferProhibited
Updated Date:
12-apr-2013
Creation Date: 22-jan-2013
Expiration Date: 22-jan-2016


>>> Last update of whois database: Tue, 11 Jun 2013 18:02:59 UTC <<<
...
domain: megacorpone.com
reg_created: 2013-01-22 23:01:00
expires: 2016-01-22 23:01:00
created: 2013-01-23 00:01:00
changed: 2013-04-12 13:03:56
transfer-prohibited: yes
ns0: ns1.megacorpone.com 50.7.67.186
ns1: ns2.megacorpone.com 50.7.67.154
ns2: ns3.megacorpone.com 50.7.67.170
owner-c:
nic-hdl: AG6848-GANDI
owner-name: MegaCorpOne
organisation: MegaCorpOne
person: Alan Grofield
address: 2 Old Mill St
zipcode: 89001
city: Rachel
state: Nevada
country: United States of America
phone: +1.9038836342
fax: ~
email: 01a71d89e668eea3c82b4a33d851dfd2-1696395@contact.gandi.net
lastupdated: 2013-06-11 19:58:30
```

Klient Whois może także wykonywać wyszukiwania odwrotne. Zamiast wpisywać nazwę domeny, możesz podać adres IP, jak pokazano poniżej:

```
root@kali:~# whois 50.7.67.186
...
NetRange: 50.7.64.0 - 50.7.67.255
CIDR:
50.7.64.0/22
OriginAS: AS30058
NetName: FDCSERVERS-MIAMI
...
OrgName: FDCservers.net
OrgId: FDCSE-8
Address: 200 SE 1st St
City: Miami
StateProv: FL
PostalCode: 33131
Country: US
RegDate: 2013-05-05
Updated: 2013-05-05
...
root@kali:~#
```

Zwróć uwagę, w jaki sposób rejestr i dostawca hostingu są pokazani w wynikach zapytań Whois.

2.3.1.4 Recon-ng

`Recon-ng` to w pełni funkcjonalny program do rozpoznawania stron internetowych napisany w języku Python. W połączeniu z niezależnymi modułami, interakcją z bazą danych, interaktywną pomocą i uzupełnianiem poleceń, Recon-ng zapewnia potężne środowisko, w którym rozpoznanie oparte na sieci open source może być przeprowadzone szybko i dokładnie. Recon-ng wygląda podobnie do Metasploit Framework. Zaczniemy od użycia modułu `whois_poc`, aby wymyślić nazwiska pracowników i adresy e-mail w Cisco.

```
root@kali:~# recon-ng
[recon-ng][default] > use recon/contacts/gather/http/api/whois_pocs
[recon-ng][default][whois_pocs] > show options
Name         Current Value         Req  Description
------       --------------        ---  -----------
DOMAIN                             yes  target domain
[recon-ng][default][whois_pocs] > set DOMAIN cisco.com
DOMAIN => cisco.comrecon-ng
[recon-ng][default][whois_pocs] > run
[*] URL: http://whois.arin.net/rest/pocs;domain=cisco.com
[*] URL: http://whois.arin.net/rest/poc/GAB42-ARIN
[*] Gary Abbott (gabbott@cisco.com) - Whois contact (Concord, TN - United
States)
...
```

Następnie możemy użyć `recon-ng` do wyszukiwania źródeł, takich jak `xssed`, w poszukiwaniu istniejących luk w zabezpieczeniach XSS, które zostały zgłoszone, ale jeszcze nie zostały naprawione, w domenie cisco.com.

```
recon-ng > use recon/hosts/enum/http/web/xssed
recon-ng [xssed] > set DOMAIN cisco.com
DOMAIN => cisco.com
recon-ng [xssed] > run
[*] URL: http://xssed.com/search?key=cisco.com
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/76478/
[*] Domain: www.cisco.com
[*] URL: http://www.cisco.com/survey/exit.html?http://xssed.com/
[*] Date submitted: 16/02/2012
[*] Date published: 16/02/2012
[*] Category: Redirect
[*] Status: UNFIXED
--------------------------------------------------
```

Możemy również użyć modułu `google_site`, aby wyszukać dodatkowe subdomeny cisco.com za pośrednictwem wyszukiwarki Google.

```
recon-ng > use recon/hosts/gather/http/web/google_site
recon-ng [google_site] > set DOMAIN cisco.com
DOMAIN => cisco.com
recon-ng [google_site] > run
[*] URL: http://www.google.com/search?start=0&filter=0&q=site%3Acisco.com
[*] www.cisco.com
[*] supportforums.cisco.com
[*] learningnetwork.cisco.com
[*] newsroom.cisco.com
[*] connectedlearningexchange.cisco.com
[*] blogs.cisco.com
[*] socialmedia.cisco.com
[*] socialviewing.cisco.com
[*] meraki.cisco.com
```

Innym przydatnym przykładem jest moduł `ip_neighbour`, który próbuje wykryć sąsiednie adresy IP domeny docelowej, ewentualnie odkrywając inne domeny w tym procesie.

```
recon-ng > use recon/hosts/gather/http/web/ip_neighbor
recon-ng [ip_neighbor] > set SOURCE cisco.com
SOURCE => cisco.com
recon-ng [ip_neighbor] > run
[*] URL: http://www.my-ip-neighbors.com/?domain=cisco.com
[*] 72.163.4.161
[*] allegrosys.com
[*] apps.cisco.com
[*] broadware.com
[*] cisco-returns.com
[*] cisco.ag
[*] cisco.com
[*] cisco.com.akadns.net
[*] cisco.com.az
[*] cisco.com.do
[*] cisco.com.kz
[*] cisco.com.ru
[*] cisco.hm
```

Ćwiczenia do wykonania

1. Za pomocą narzędzia `recon-ng` systemu operacyjnego Kali Linux sprawdź czego możesz się dowiedzieć o domenie `www.pw.edu.pl` w kontekście omówionych modułów.

2.4 Aktywne zbieranie informacji

Po zebraniu wystarczającej ilości informacji o celu, z wykorzystaniem otwartych zasobów internetowych i innych pasywnych technik gromadzenia informacji, możesz dalej zbierać odpowiednie informacje z innych, bardziej szczegółowych źródeł.

2.4.1 Enumeracja DNS

System DNS (*Domain Name System*) jest jednym z częstych źródeł aktywnego gromadzenia informacji. DNS oferuje wiele informacji na temat publicznych (a czasem prywatnych!) serwerów organizacji, takich jak adresy IP, nazwy serwerów czy ich funkcje.

2.4.1.1 Interakcja z serwerem DNS

Serwer DNS zwykle ujawnia informacje o DNS i serwerze poczty dla domeny, nad którą ma uprawnienia. Jest to konieczne, ponieważ publiczne żądania poczty i adresów serwerów DNS stanowią podstawową funkcjonalność Internetu. Na przykład przyjrzyjmy się domenie `megacorpone.com`, fałszywej domenie w Internecie, którą stworzyliśmy na potrzeby tego ćwiczenia. Użyjemy polecenia `host` wraz z parametrem `-t` (*typ*), aby wykryć zarówno DNS, jak i serwery poczty dla domeny `megacorpone.com`.

```
root@kali:~# host -t ns megacorpone.com
megacorpone.com name server ns2.megacorpone.com.
megacorpone.com name server ns1.megacorpone.com.
megacorpone.com name server ns3.megacorpone.com.
root@kali:~# host -t mx megacorpone.com
megacorpone.com mail is handled by 60 mail.megacorpone.com.
megacorpone.com mail is handled by 50 mail2.megacorpone.com.
```

Domyślnie każda skonfigurowana domena powinna zapewniać przynajmniej DNS i serwery poczty odpowiedzialne za domenę.

2.4.1.2 Automatyzacja wyszukiwania

Teraz, gdy mamy pewne wstępne dane z domeny `megacorpone.com`, możemy nadal korzystać z dodatkowych zapytań DNS, aby znaleźć więcej nazw hostów i adresów IP należących do megacorpone.com. Na przykład możemy założyć, że domena megacorpone.com ma serwer WWW, prawdopodobnie o nazwie hosta www. Możemy przetestować tę teorię za pomocą polecenia `host` jeszcze raz:

```
root@kali:~# host www.megacorpone.com
www.megacorpone.com has address 50.7.67.162
```

Teraz sprawdźmy, czy `megacorpone.com` ma również serwer z nazwą hosta *idontexist*. Zwróć uwagę na różnicę między wynikami zapytania.

```
root@kali:~# host idontexist.megacorpone.com
Host idontexist.megacorpone.com not found: 3(NXDOMAIN)
```

2.4.1.3 Brute force *wyszukiwania wprzód*

Kontynuując poprzednią koncepcję, możemy zautomatyzować wyszukiwanie wprzód DNS dla popularnych nazw hostów za pomocą polecenia `host` i skryptu Bash. Ideą tej techniki jest odgadnięcie prawidłowych nazw serwerów, próbując rozwiązać daną nazwę. Jeśli

nazwa, którą odgadłeś, rozwiązuje się (pozytywne odpowiedzi z DNS), wyniki mogą wskazywać na obecność, a nawet funkcjonalność serwera. Możemy utworzyć krótką (lub długą) listę możliwych nazw hostów i zapętlić polecenie `host`, aby wypróbować każdą z nich.

```
root@kali:~# echo www > list.txt
root@kali:~# echo ftp >> list.txt
root@kali:~# echo mail >> list.txt
root@kali:~# echo owa >> list.txt
root@kali:~# echo proxy >> list.txt
root@kali:~# echo router >> list.txt
root@kali:~# for ip in $(cat list.txt);do host $ip.megacorpone.com;done
www.megacorpone.com has address 50.7.67.162
Host ftp.megacorpone.com not found: 3(NXDOMAIN)
mail.megacorpone.com has address 50.7.67.155
Host owa.megacorpone.com not found: 3(NXDOMAIN)
Host proxy.megacorpone.com not found: 3(NXDOMAIN)
router.megacorpone.com has address 50.7.67.190
root@kali:~#
```

W tym uproszczonym przykładzie zauważamy, że nazwy hostów `www`, `router` i `mail` zostały odkryte w wyniku tego ataku siłowego. Nazwy hostów `owa`, `ftp` i `proxy` nie zostały jednak znalezione.

2.4.1.4 Brute force *wyszukiwania w tył*

Wyszukiwanie wwprzód DNS ujawniło zestaw rozproszonych adresów IP. Jeśli administrator DNS `megacorpone.com` skonfigurował rekordy PTR dla domeny, moglibyśmy znaleźć więcej nazw domen, które zostały pominięte podczas fazy *brute force* wyszukiwania do przodu, sprawdzając zakres tych znalezionych adresów w pętli.

```
root@kali:~# for ip in $(seq 155 190);do host 50.7.67.$ip;done |grep -v "not found"
155.67.7.50.in-addr.arpa domain name pointer mail.megacorpone.com.
162.67.7.50.in-addr.arpa domain name pointer www.megacorpone.com.
163.67.7.50.in-addr.arpa domain name pointer mail2.megacorpone.com.
164.67.7.50.in-addr.arpa domain name pointer www2.megacorpone.com.
165.67.7.50.in-addr.arpa domain name pointer beta.megacorpone.com.
...
```

2.4.1.5 Transfer stref DNS

Transfer strefy jest podobny do czynności replikacji bazy danych między powiązanymi serwerami DNS. Proces ten obejmuje kopiowanie pliku strefy z głównego serwera DNS na serwer podrzędny. Plik strefy zawiera listę wszystkich nazw DNS skonfigurowanych dla tej strefy. Transfery stref powinny zwykle ograniczać się do autoryzowanych podrzędnych serwerów DNS. Niestety wielu administratorów źle konfiguruje swoje serwery DNS, w wyniku czego każdy, kto poprosi o kopię strefy serwera DNS, otrzyma taki serwer. Jest to równoważne z przekazaniem bezpośrednio hakerowi układu sieci korporacyjnej. Wiele organizacji posiada serwery DNS które są źle skonfigurowane:

- nie podzielono wewnętrznej przestrzeni nazw DNS i zewnętrznej przestrzeni nazw DNS na osobne
- występują niepowiązane strefy

W wyniku tego powstaje kompletny obraz struktury sieciowej organzacji. Pomyślne przesłanie strefy nie powoduje bezpośrednio naruszenia sieci. Ułatwia to jednak

proces. Składnia polecenia `host` służąca do wstępnego wykonania transferu strefy jest następująca.

```
host -l <domain name> <dns server address>
```

Z naszego poprzedniego polecenia `host` zauważyliśmy, że dwa serwery DNS obsługują domenę `megacorpone.com`: ns1 i ns2. Spróbujmy przenieść strefę na każdym z nich.

```
root@kali:~# host -l megacorpone.com ns1.megacorpone.com
; Transfer failed.
Using domain server:
Name: ns1.megacorpone.com
Address: 50.7.67.186#53
Aliases:

Host megacorpone.com not found: 5(REFUSED)
; Transfer failed.
root@kali:~# host -l megacorpone.com ns2.megacorpone.com
Using domain server:
Name: ns2.megacorpone.com
Address: 50.7.67.154#53
Aliases:

megacorpone.com name server ns1.megacorpone.com.
megacorpone.com name server ns2.megacorpone.com.
megacorpone.com name server ns3.megacorpone.com.
admin.megacorpone.com has address 50.7.67.187
beta.megacorpone.com has address 50.7.67.165
fs1.megacorpone.com has address 50.7.67.166
intranet.megacorpone.com has address 50.7.67.188
mail.megacorpone.com has address 50.7.67.155
mail2.megacorpone.com has address 50.7.67.163
ns1.megacorpone.com has address 50.7.67.186
ns2.megacorpone.com has address 50.7.67.154
ns3.megacorpone.com has address 50.7.67.170
router.megacorpone.com has address 50.7.67.190
router.megacorpone.com has address 10.7.0.1
siem.megacorpone.com has address 50.7.67.180
snmp.megacorpone.com has address 50.7.67.181
syslog.megacorpone.com has address 50.7.67.178
test.megacorpone.com has address 50.7.67.182
vpn.megacorpone.com has address 50.7.67.189
www.megacorpone.com has address 50.7.67.162
www2.megacorpone.com has address 50.7.67.164
root@kali:~#
```

W tym przypadku *ns1* odrzucił nam naszą prośbę o przeniesienie strefy, podczas gdy *ns2* na to zezwolił. Rezultatem jest pełny zrzut pliku strefy dla domeny `megacorpone.com`, zapewniający nam wygodną listę adresów IP i nazw DNS dla domeny `megacorpone.com`. Domena `megacorpone.com` ma tylko dwa serwery DNS do sprawdzenia. Jednak niektóre większe organizacje mogą mieć wiele serwerów DNS lub możesz próbować przesłać żądania strefy dla danej domeny. Tutaj zaczyna się gra skryptów Bash. Aby wykonać transfer strefy za pomocą polecenia `host`, potrzebujemy dwóch parametrów: analizowanej nazwy domeny i adresu serwera nazw. Aby uzyskać serwery nazw dla danej domeny w czystym formacie, możemy wydać następujące polecenie.

```
root@kali:~# host -t ns megacorpone.com | cut -d " " -f 4
ns2.megacorpone.com.
ns1.megacorpone.com.
```

Idąc krok dalej, możemy napisać następujący prosty skrypt Bash, aby zautomatyzować procedurę wykrywania i próby transferu strefy na każdym znalezionym serwerze DNS.

```
#/bin/bash
# Simple Zone Transfer Bash Script
# $1 is the first argument given after the bash script
# Check if argument was given, if not, print usage
if [ -z "$1" ]; then
echo "[*] Simple Zone transfer script"
echo "[*] Usage : $0 <domain name> "
exit 0
fi
# if argument was given, identify the DNS servers for the domain
for server in $(host -t ns $1 |cut -d" " -f4);do
# For each of these servers, attempt a zone transfer
host -l $1 $server |grep "has address"
done
```

Uruchomienie tego skryptu na megacorpone.com powinno automatycznie zidentyfikować oba serwery nazw i podjąć próbę przeniesienia strefy na każdym z nich.

```
root@kali:~# chmod 755 dns-axfr.sh
root@kali:~# ./dns-axfr.sh megacorpone.com
admin.megacorpone.com has address 50.7.67.187
beta.megacorpone.com has address 50.7.67.165
fs1.megacorpone.com has address 50.7.67.166
intranet.megacorpone.com has address 50.7.67.188
mail.megacorpone.com has address 50.7.67.155
mail2.megacorpone.com has address 50.7.67.163
ns1.megacorpone.com has address 50.7.67.186
ns2.megacorpone.com has address 50.7.67.154
ns3.megacorpone.com has address 50.7.67.170
router.megacorpone.com has address 50.7.67.190
siem.megacorpone.com has address 50.7.67.180
snmp.megacorpone.com has address 50.7.67.181
syslog.megacorpone.com has address 50.7.67.178
test.megacorpone.com has address 50.7.67.182
vpn.megacorpone.com has address 50.7.67.189
www.megacorpone.com has address 50.7.67.162
www2.megacorpone.com has address 50.7.67.164
root@kali:~#
```

2.4.1.6 Odpowiednie narzedzia w Kali Linux

W Kali Linux istnieje kilka narzędzi, które pomagają nam w enumeracji DNS, a większość z nich wykonuje te same zadania, które już omówiliśmy wcześniej. Dwa znaczące narzędzia to DNSrecon i DNSenum. Każde z tych narzędzi ma przydatne opcje. Poniższe wyniki pokazują użycie tych narzędzi przy minimalnych parametrach.

2.4.1.6.1 DNSrecon

DNSRecon to zaawansowany, nowoczesny skrypt enumeracji DNS napisany w języku Python. Uruchomienie skryptu **dnsrecon** w domenie megacorpone.com daje następujące wyniki:

```
root@kali:~# dnsrecon -d megacorpone.com -t axfr
[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for megacorpone.com name servers
[*] Resolving SOA Record
[*] SOA ns1.megacorpone.com 50.7.67.186
[*] Resolving NS Records
[*] NS Servers found:
[*] NS ns2.megacorpone.com 50.7.67.154
[*] NS ns1.megacorpone.com 50.7.67.186
[*] NS ns3.megacorpone.com 50.7.67.170
[*] Removing any duplicate NS server IP  Addresses ...
[*]
[*] Trying NS server 50.7.67.154
[*] 50.7.67.154 Has port 53 TCP Open
[*] Zone Transfer was successful!!
[*] MX @.megacorpone.com fb.mail.gandi.net 217.70.184.163
[*] MX @.megacorpone.com fb.mail.gandi.net 217.70.184.162
[*] MX @.megacorpone.com spool.mail.gandi.net 217.70.184.6
[*] MX @.megacorpone.com spool.mail.gandi.net 2001:4b98:c:521::6
[*] A admin.megacorpone.com 50.7.67.187
[*] A fs1.megacorpone.com 50.7.67.166
[*] A www2.megacorpone.com 50.7.67.164
[*] A test.megacorpone.com 50.7.67.182
[*] A ns1.megacorpone.com 50.7.67.186
[*] A ns2.megacorpone.com 50.7.67.154
[*] A ns3.megacorpone.com 50.7.67.170
[*] A www.megacorpone.com 50.7.67.162
[*] A siem.megacorpone.com 50.7.67.180
[*] A mail2.megacorpone.com 50.7.67.163
[*] A router.megacorpone.com 50.7.67.190
[*] A mail.megacorpone.com 50.7.67.155
[*] A vpn.megacorpone.com 50.7.67.189
[*] A snmp.megacorpone.com 50.7.67.181
[*] A syslog.megacorpone.com 50.7.67.178
[*] A beta.megacorpone.com 50.7.67.165
[*] A intranet.megacorpone.com 50.7.67.188
```

2.4.1.6.2 DNSenum

DNSEnum to kolejne popularne narzędzie do wyliczania DNS. Uruchomienie tego skryptu w domenie **zonetransfer.me**, która w szczególności zezwala na transfery stref, daje następujące wyniki:

```
root@kali:~# dnsenum zonetransfer.me
dnsenum.pl VERSION:1.2.2
----- zonetransfer.me -----

Host's addresses:
__________________

zonetransfer.me 7200 IN A
217.147.180.162

Name Servers:
______________

ns12.zoneedit.com 3653 IN A 209.62.64.46
ns16.zoneedit.com 6975 IN A 69.64.68.41
Mail (MX) Servers:
____________________

ASPMX5.GOOGLEMAIL.COM 293 IN A 173.194.69.26
ASPMX.L.GOOGLE.COM 293 IN A 173.194.74.26
ALT1.ASPMX.L.GOOGLE.COM 293 IN A 173.194.66.26
ALT2.ASPMX.L.GOOGLE.COM 293 IN A 173.194.65.26
ASPMX2.GOOGLEMAIL.COM 293 IN A 173.194.78.26
ASPMX3.GOOGLEMAIL.COM 293 IN A 173.194.65.26
ASPMX4.GOOGLEMAIL.COM 293 IN A 173.194.70.26


Trying Zone Transfers and getting Bind Versions:
________________________________________________



Trying Zone Transfer for zonetransfer.me on ns12.zoneedit.com ...
zonetransfer.me 7200 IN SOA
zonetransfer.me 7200 IN NS
zonetransfer.me 7200 IN NS
zonetransfer.me 7200 IN MX
zonetransfer.me 7200 IN MX
zonetransfer.me 7200 IN MX
zonetransfer.me 7200 IN MX
zonetransfer.me 7200 IN MX
zonetransfer.me 7200 IN MX
...
office.zonetransfer.me 7200 INA 4.23.39.254
owa.zonetransfer.me 7200 IN A 207.46.197.32
info.zonetransfer.me 7200 IN TXT
asfdbbox.zonetransfer.me 7200 IN A 127.0.0.1
canberra_office.zonetransfer.me 7200 IN A 202.14.81.230
asfdbvolume.zonetransfer.me 7800 IN  AFSDB
email.zonetransfer.me 2222 IN NAPTR
dzc.zonetransfer.me 7200 IN TXT
robinwood.zonetransfer.me 302 IN TXT
vpn.zonetransfer.me 4000 IN A 174.36.59.154
_sip._tcp.zonetransfer.me 14000 IN SRV
dc_office.zonetransfer.me 7200 IN A  143.228.181.132


ns16.zoneedit.com Bind Version: 8.4.X


brute force file not specified, bay.
root@kali:~#
```