Port Scanning with Metasploit

pentestlab.blog/category/information-gathering/page/12

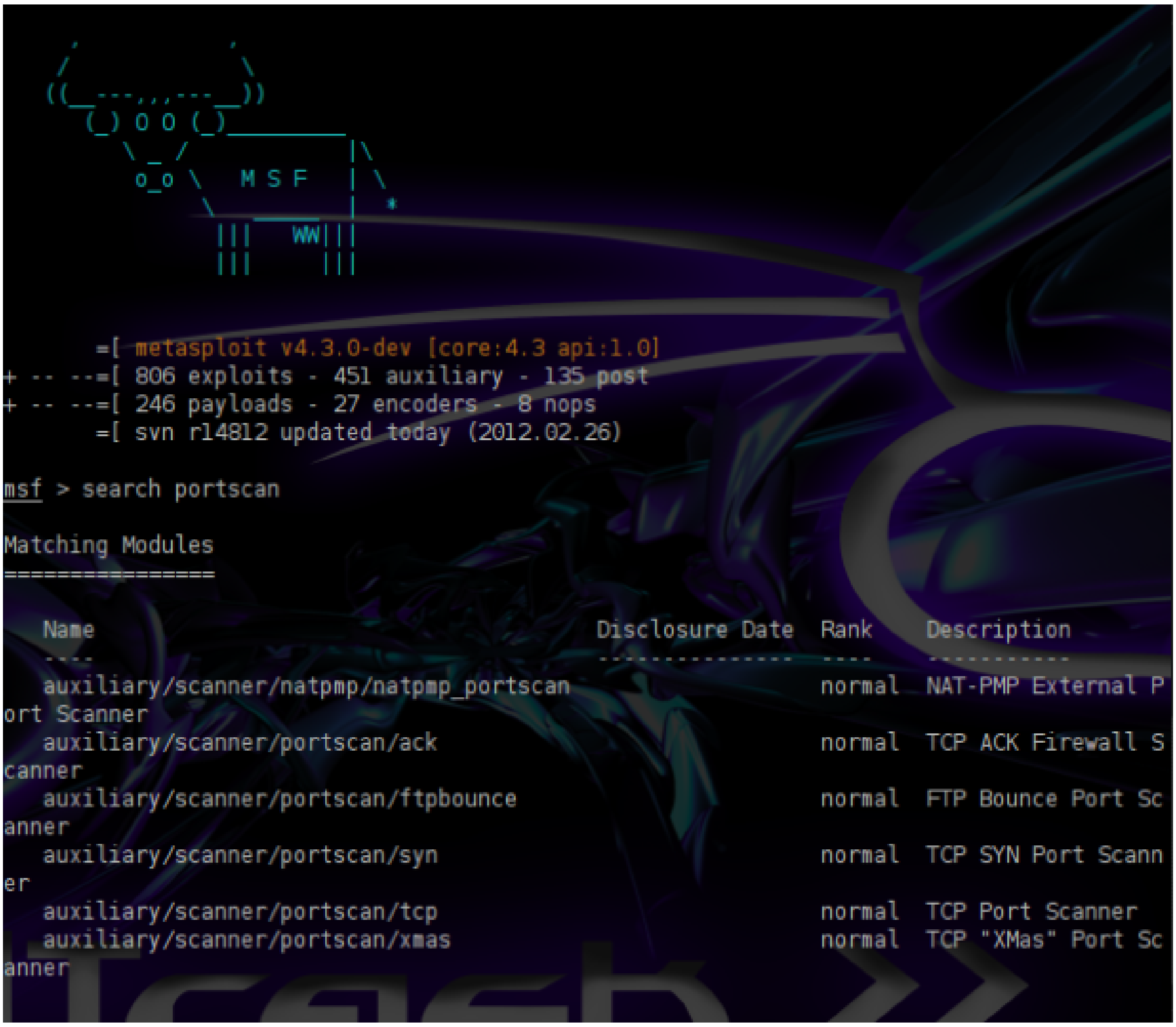
February 26, 2012

Metasploit Framework includes some port scanners that could be used in a situation that we have compromise a system which is behind a NAT Firewall and we want to do a port scan to the rest of the network or we are just performing an internal penetration test.

In this scenario we will see how we can perform a simple TCP and SYN port scan by using the modules of the Metasploit.

SYN Scanning

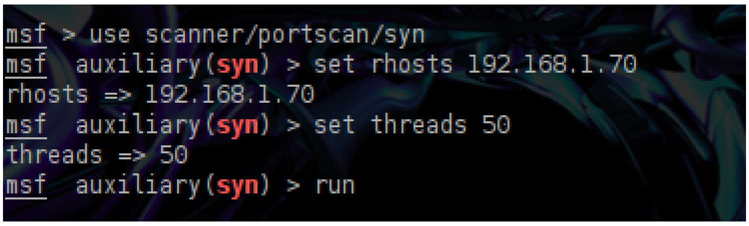
First we open the Metasploit Framework and we type **search portscan** to find the existing scanners.



```
     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


       =[ metasploit v4.3.0-dev [core:4.3 api:1.0]
+ -- --=[ 806 exploits - 451 auxiliary - 135 post
+ -- --=[ 246 payloads - 27 encoders - 8 nops
       =[ svn r14812 updated today (2012.02.26)

msf > search portscan

Matching Modules
================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   auxiliary/scanner/natpmp/natpmp_portscan                     normal  NAT-PMP External P
ort Scanner
   auxiliary/scanner/portscan/ack                               normal  TCP ACK Firewall S
canner
   auxiliary/scanner/portscan/ftpbounce                         normal  FTP Bounce Port Sc
anner
   auxiliary/scanner/portscan/syn                               normal  TCP SYN Port Scann
er
   auxiliary/scanner/portscan/tcp                               normal  TCP Port Scanner
   auxiliary/scanner/portscan/xmas                              normal  TCP "XMas" Port Sc
anner
```

Choosing the Port Scanner

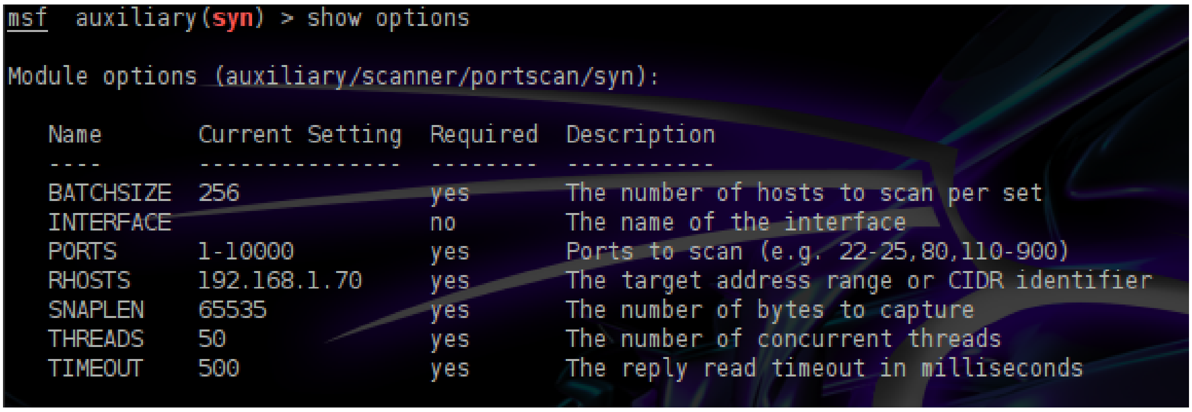
The port scanner that we will use is the syn scanner and we can see the configuration settings in the image below:



```
msf > use scanner/portscan/syn
msf  auxiliary(syn) > set rhosts 192.168.1.70
rhosts => 192.168.1.70
msf  auxiliary(syn) > set threads 50
threads => 50
msf  auxiliary(syn) > run
```

Configuration of SYN Scanner

Before we type the **run** command that it will start the port scanning in the remote host we can use the **show options** command in order to see the available options and to check if all the settings are correct.

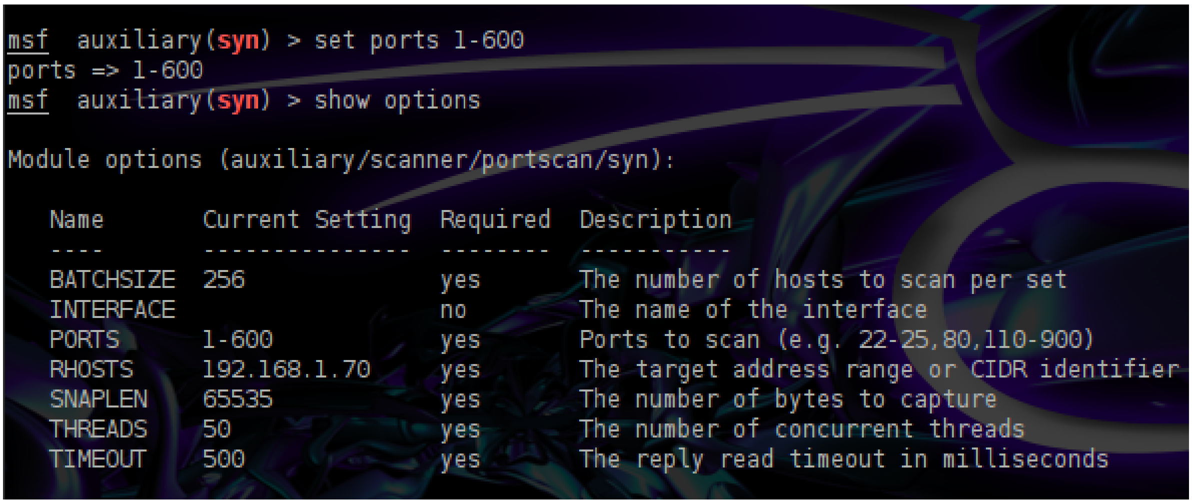


```
msf  auxiliary(syn) > show options

Module options (auxiliary/scanner/portscan/syn):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to scan per set
   INTERFACE                   no        The name of the interface
   PORTS      1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS     192.168.1.70     yes       The target address range or CIDR identifier
   SNAPLEN    65535            yes       The number of bytes to capture
   THREADS    50               yes       The number of concurrent threads
   TIMEOUT    500              yes       The reply read timeout in milliseconds
```

SYN Scanner Options

From the image above we can see that the default setting for the ports that the scanner will scan is from 1-10000.We can change this setting if want the scanner just to check for specific ports or we can give the range that we want.

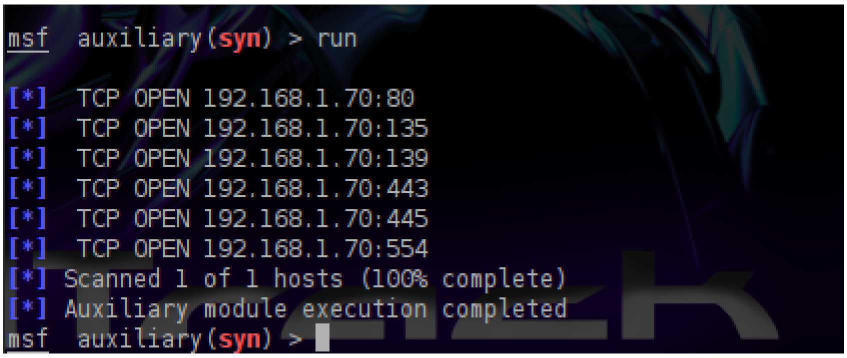
For this example we have chosen to scan the ports from 1 to 600.



```
msf  auxiliary(syn) > set ports 1-600
ports => 1-600
msf  auxiliary(syn) > show options

Module options (auxiliary/scanner/portscan/syn):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to scan per set
   INTERFACE                   no        The name of the interface
   PORTS      1-600            yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS     192.168.1.70     yes       The target address range or CIDR identifier
   SNAPLEN    65535            yes       The number of bytes to capture
   THREADS    50               yes       The number of concurrent threads
   TIMEOUT    500              yes       The reply read timeout in milliseconds
```

Port Range Setting

Now we can type the run command and we can see the results in the image below:



```
msf  auxiliary(syn) > run

[*]  TCP OPEN 192.168.1.70:80
[*]  TCP OPEN 192.168.1.70:135
[*]  TCP OPEN 192.168.1.70:139
[*]  TCP OPEN 192.168.1.70:443
[*]  TCP OPEN 192.168.1.70:445
[*]  TCP OPEN 192.168.1.70:554
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(syn) > 
```

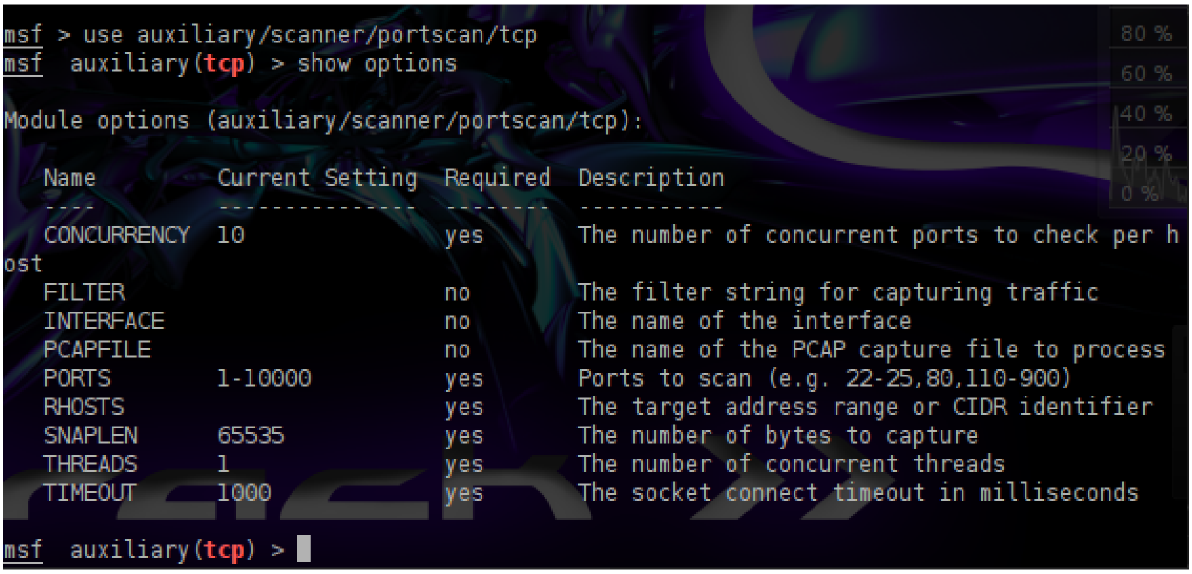
Scanning the target with SYN scan

We can see that some common ports are open on the remote host like port 80,139 and 445.This is giving us also an indication for the operating system of the target.It is definitely Windows because ports 139 and 445 belongs to the netbios service in Windows environments.

TCP Scanning

Metasploit Framework has also and a TCP Scanner.We have used this scanner as well into the same remote host.

We can see that we have slightly different options from the SYN scanner.For example we can set a filter string for capturing traffic or we can process a packet capture file.



```
msf > use auxiliary/scanner/portscan/tcp
msf  auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per h
ost
   FILTER                        no        The filter string for capturing traffic
   INTERFACE                     no        The name of the interface
   PCAPFILE                      no        The name of the PCAP capture file to process
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target address range or CIDR identifier
   SNAPLEN      65535            yes       The number of bytes to capture
   THREADS      1                yes       The number of concurrent threads
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf  auxiliary(tcp) > 
```

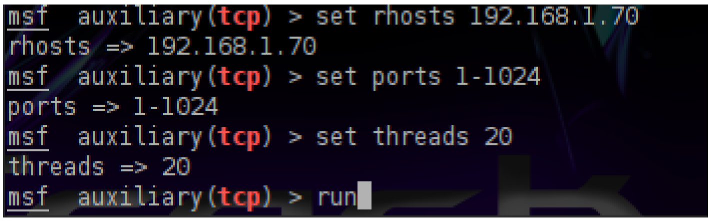
Available options of TCP Scanner

For the TCP scan we have set the following parameters:

The screenshot below is showing us the output of the scan:

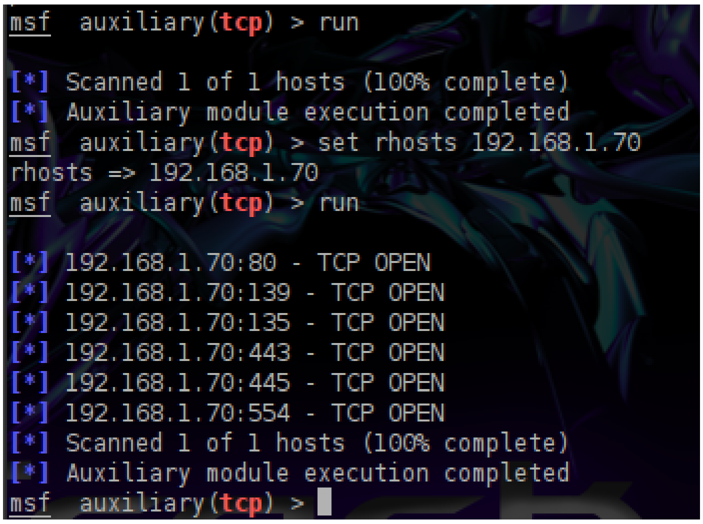
Conclusions

From these two scans we have noticed that the TCP scan is much faster however it is not as stealth as a SYN scan and it could be identified by the IDS (Intrusion Detection System). From the other hand a SYN scan is slower but less intrusive because it sends the RST packet to the remote host before the connection is established.



```
msf  auxiliary(tcp) > set rhosts 192.168.1.70
rhosts => 192.168.1.70
msf  auxiliary(tcp) > set ports 1-1024
ports => 1-1024
msf  auxiliary(tcp) > set threads 20
threads => 20
msf  auxiliary(tcp) > run
```

TCP Scanner Settings



```
msf  auxiliary(tcp) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(tcp) > set rhosts 192.168.1.70
rhosts => 192.168.1.70
msf  auxiliary(tcp) > run

[*] 192.168.1.70:80 - TCP OPEN
[*] 192.168.1.70:139 - TCP OPEN
[*] 192.168.1.70:135 - TCP OPEN
[*] 192.168.1.70:443 - TCP OPEN
[*] 192.168.1.70:445 - TCP OPEN
[*] 192.168.1.70:554 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(tcp) > 
```

TCP Scan Results