NMAP
Firewall Evasion Techniques

By: Dimas Febriawan, CEH

Firewall Evasion Techniques

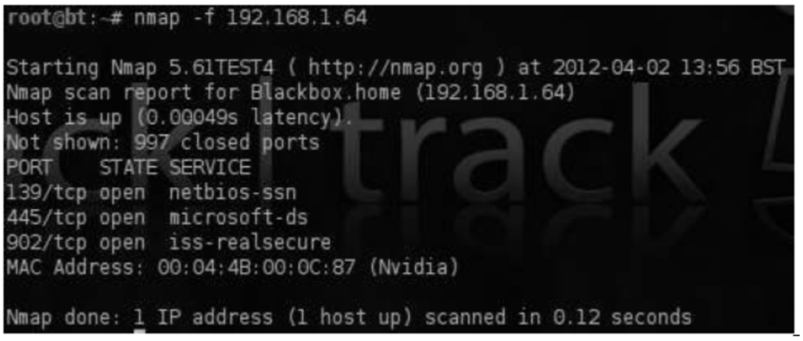
- As a penetration tester you will come across with systems that are behind firewalls.
- So you will need to avoid the firewall rules that are in place and to discover information about a host.
- This step in a penetration testing called Firewall Evasion Rules.
- Nmap is offering a lot of options about Firewall evasion.

Fragmentation scanning

- Instead of just sending the probe packet, you break it into a couple of small IP fragments.
- You are splitting up the TCP header over several packets to make it harder for packet filters and so forth to detect what you are doing.
- The -f switch instructs the specified SYN or FIN scan to use tiny fragmented packets.
- This technique was very effective especially in the old days however you can still use it if you found a firewall that is not properly configured.

Fragmentation scanning

- Nmap –f ip_target



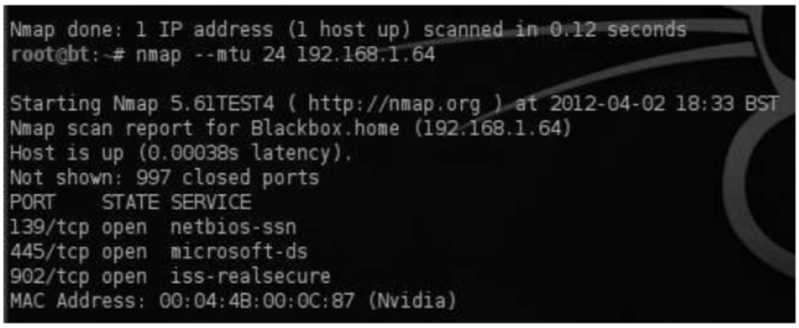
```
root@bt:~# nmap -f 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 13:56 BST
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00049s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Specify Specific MTU

- Nmap is giving the option to the user to set a specific MTU (Maximum Transmission Unit) to the packet.
- This is similar to the packet fragmentation technique.
- During the scan, nmap will create packets with size based on the number that we will give.
- In this example we gave the number 24, so the nmap will create 24-byte packets causing a confusion to the firewall.
- Have in mind that the MTU number must be a multiple of 8 (8,16,24,32 etc).

Specify Specific MTU

- Command: nmap --mtu number target_ip

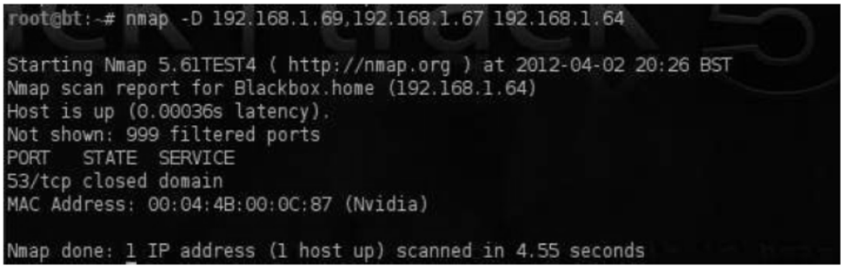


```
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
root@bt:~# nmap --mtu 24 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 18:33 BST
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00038s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure
MAC Address: 00:04:4B:00:0C:87 (Nvidia)
```

Use Decoy Addresses

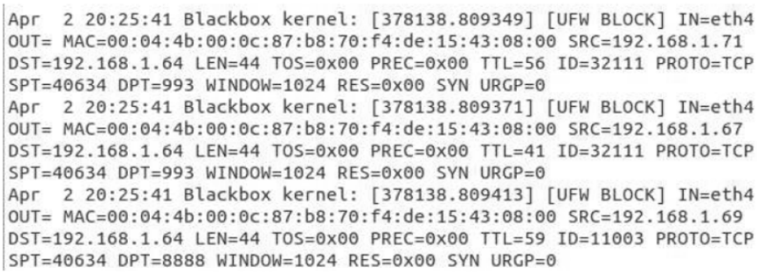
- In this type of scan you can instruct Nmap to spoof packets from other hosts.
- In the firewall logs it will be not only our IP address, but also the IP addresses of the decoys.
- So it will be much harder to determine from which system the scan started.
- There are two options that you can use in this type of scan:
  - Generates 10 random number of decoys:
    nmap -D RND:10 [target_ip]
  - Manually specify the IP addresses of the decoys:
    nmap -D decoy1,decoy2,decoy3, ...

Use Decoy Addresses

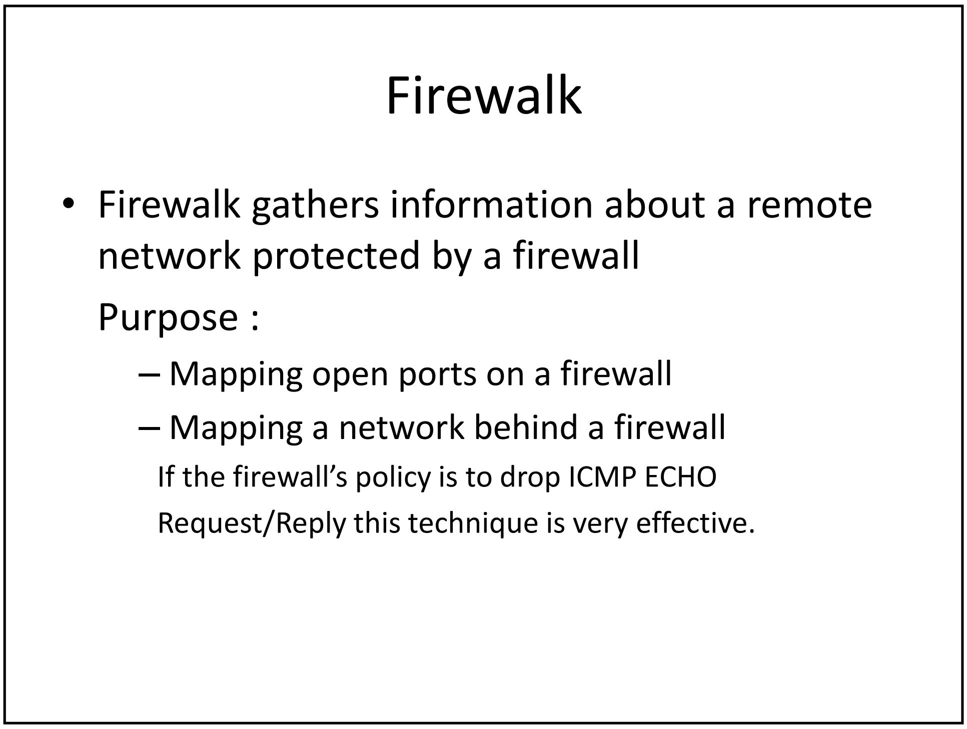


```
root@bt:~# nmap -D 192.168.1.69,192.168.1.67 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 20:26 BST
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00036s latency).
Not shown: 999 filtered ports
PORT   STATE  SERVICE
53/tcp closed domain
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap done: 1 IP address (1 host up) scanned in 4.55 seconds
```

- Firewall Logs:



```
Apr  2 20:25:41 Blackbox kernel: [378138.809349] [UFW BLOCK] IN=eth4
OUT= MAC=00:04:4b:00:0c:87:b8:70:f4:de:15:43:08:00 SRC=192.168.1.71
DST=192.168.1.64 LEN=44 TOS=0x00 PREC=0x00 TTL=56 ID=32111 PROTO=TCP
SPT=40634 DPT=993 WINDOW=1024 RES=0x00 SYN URGP=0
Apr  2 20:25:41 Blackbox kernel: [378138.809371] [UFW BLOCK] IN=eth4
OUT= MAC=00:04:4b:00:0c:87:b8:70:f4:de:15:43:08:00 SRC=192.168.1.67
DST=192.168.1.64 LEN=44 TOS=0x00 PREC=0x00 TTL=41 ID=32111 PROTO=TCP
SPT=40634 DPT=993 WINDOW=1024 RES=0x00 SYN URGP=0
Apr  2 20:25:41 Blackbox kernel: [378138.809413] [UFW BLOCK] IN=eth4
OUT= MAC=00:04:4b:00:0c:87:b8:70:f4:de:15:43:08:00 SRC=192.168.1.69
DST=192.168.1.64 LEN=44 TOS=0x00 PREC=0x00 TTL=59 ID=11003 PROTO=TCP
SPT=40634 DPT=8888 WINDOW=1024 RES=0x00 SYN URGP=0
```



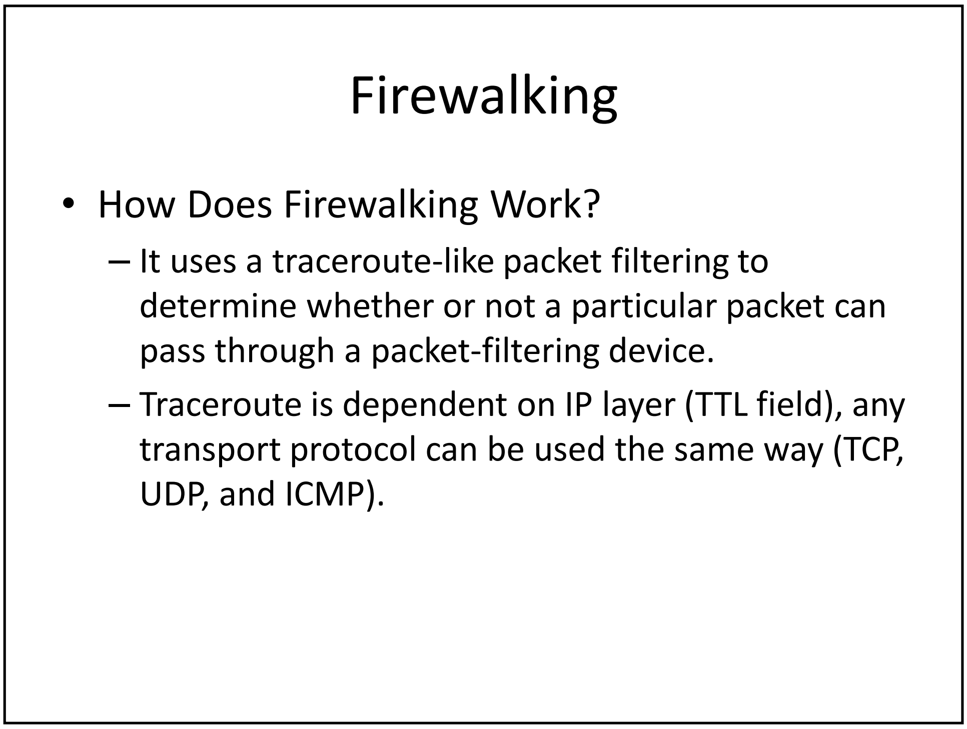
Firewalk

- Firewalk gathers information about a remote network protected by a firewall

  Purpose :

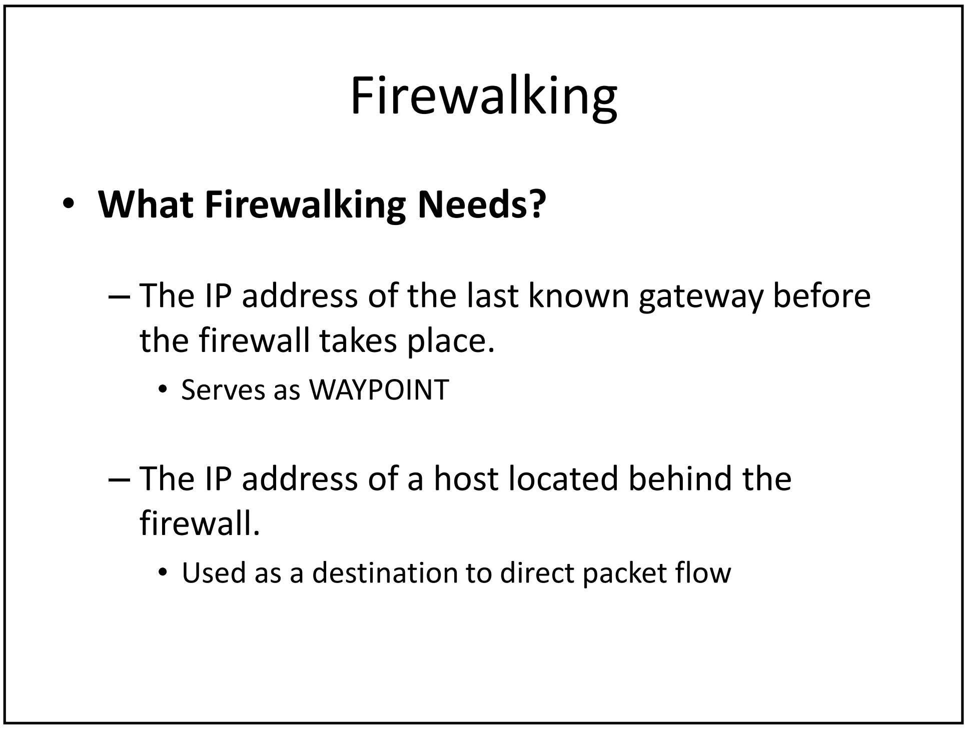
  - Mapping open ports on a firewall
  - Mapping a network behind a firewall

    If the firewall's policy is to drop ICMP ECHO Request/Reply this technique is very effective.



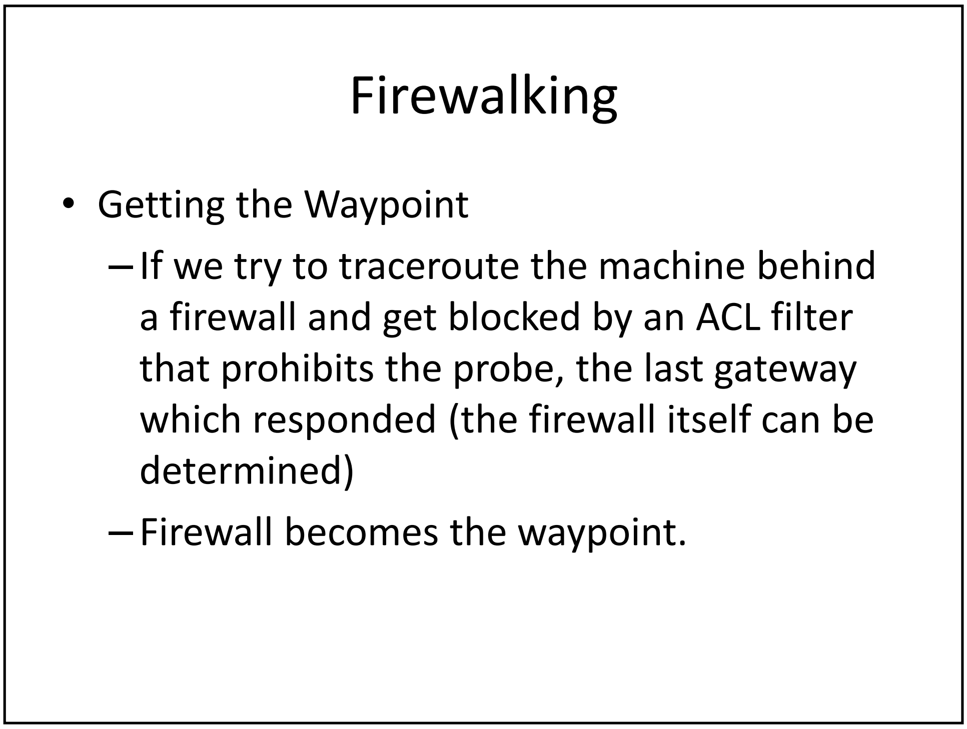
Firewalking

- How Does Firewalking Work?
  - It uses a traceroute-like packet filtering to determine whether or not a particular packet can pass through a packet-filtering device.
  - Traceroute is dependent on IP layer (TTL field), any transport protocol can be used the same way (TCP, UDP, and ICMP).



Firewalking

- **What Firewalking Needs?**
  - The IP address of the last known gateway before the firewall takes place.
    - Serves as WAYPOINT
  - The IP address of a host located behind the firewall.
    - Used as a destination to direct packet flow



Firewalking

- Getting the Waypoint
  - If we try to traceroute the machine behind a firewall and get blocked by an ACL filter that prohibits the probe, the last gateway which responded (the firewall itself can be determined)
  - Firewall becomes the waypoint.

Firewalking

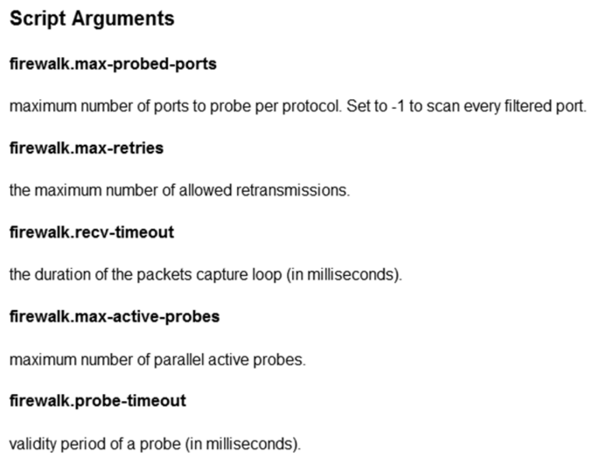
- Getting the Destination
  - Traceroute the same machine with a different traceroute-probe using a different transport protocol.
  - If we get a response
    - That particular traffic is allowed by the firewall.
    - We know a host behind the firewall.
  - If we are continuously blocked, then this kind of traffic is blocked.
  - Sending packets to every host behind the packet-filtering device can generate an accurate map of a network’s topology.

Nmap NSE Firewalk (1)

- Tries to discover firewall rules using an IP TTL expiration technique known as firewalking.

- To determine a rule on a given gateway, the scanner sends a probe to a metric located behind the gateway, with a TTL one higher than the gateway. If the probe is forwarded by the gateway, then we can expect to receive an ICMP_TIME_EXCEEDED reply from the gateway next hop router, or eventually the metric itself if it is directly connected to the gateway. Otherwise, the probe will timeout.

Nmap NSE Firewalk (2)



Script Arguments

firewalk.max-probed-ports

maximum number of ports to probe per protocol. Set to -1 to scan every filtered port.

firewalk.max-retries

the maximum number of allowed retransmissions.

firewalk.recv-timeout

the duration of the packets capture loop (in milliseconds).

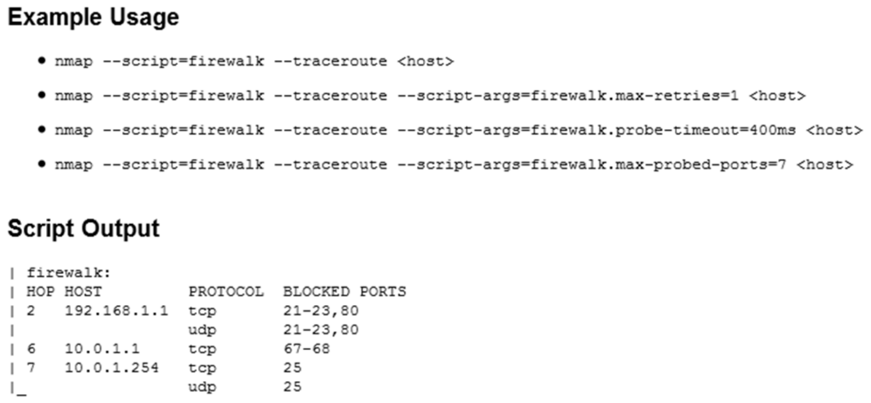
firewalk.max-active-probes

maximum number of parallel active probes.

firewalk.probe-timeout

validity period of a probe (in milliseconds).

Nmap NSE Firewalk (3)



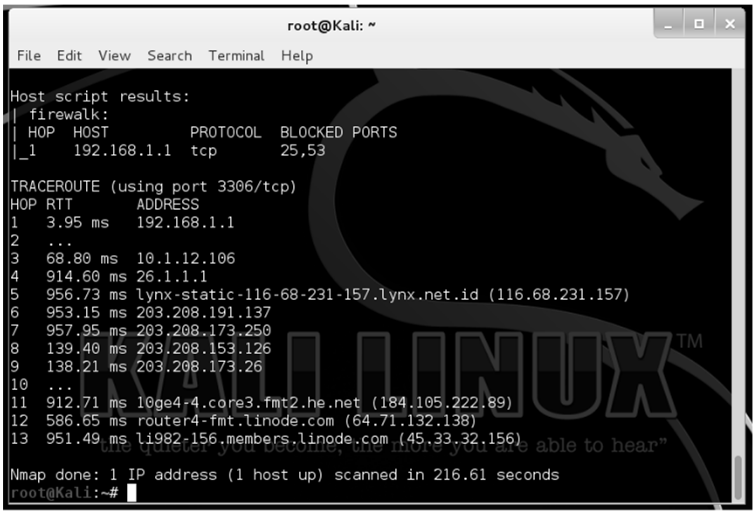
Example Usage

- nmap --script=firewalk --traceroute <host>
- nmap --script=firewalk --traceroute --script-args=firewalk.max-retries=1 <host>
- nmap --script=firewalk --traceroute --script-args=firewalk.probe-timeout=400ms <host>
- nmap --script=firewalk --traceroute --script-args=firewalk.max-probed-ports=7 <host>

Script Output

```
| firewalk:
| HOP HOST         PROTOCOL  BLOCKED PORTS
| 2   192.168.1.1  tcp       21-23,80
|                  udp       21-23,80
| 6   10.0.1.1     tcp       67-68
| 7   10.0.1.254   tcp       25
|_                 udp       25
```

Nmap NSE Firewalk (4)



```
Host script results:
| firewalk:
| HOP  HOST         PROTOCOL  BLOCKED PORTS
|_1    192.168.1.1  tcp       25,53

TRACEROUTE (using port 3306/tcp)
HOP RTT       ADDRESS
1   3.95 ms   192.168.1.1
2   ...
3   68.80 ms  10.1.12.106
4   914.60 ms 26.1.1.1
5   956.73 ms lynx-static-116-68-231-157.lynx.net.id (116.68.231.157)
6   953.15 ms 203.208.191.137
7   957.95 ms 203.208.173.250
8   139.40 ms 203.208.153.126
9   138.21 ms 203.208.173.26
10  ...
11  912.71 ms 10ge4-4.core3.fmt2.he.net (184.105.222.89)
12  586.65 ms router4-fmt.linode.com (64.71.132.138)
13  951.49 ms li982-156.members.linode.com (45.33.32.156)

Nmap done: 1 IP address (1 host up) scanned in 216.61 seconds
root@Kali:~# 
```

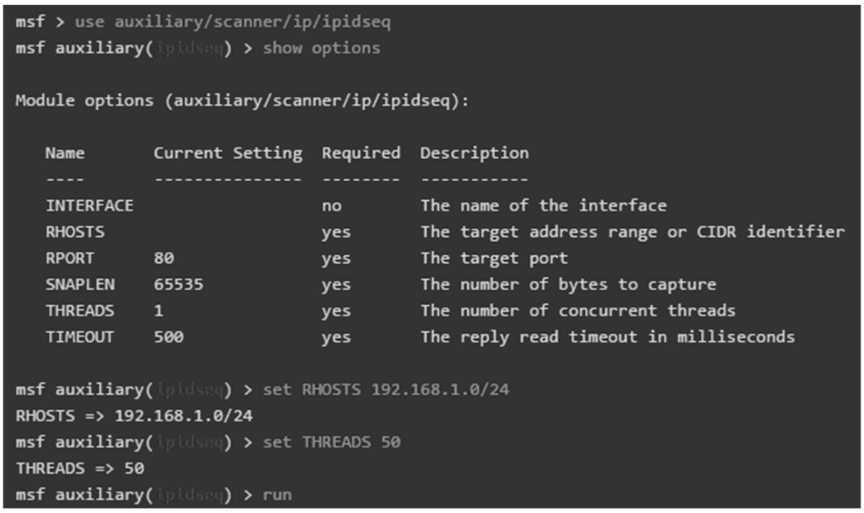
Idle Zombie Scan

- This technique allows you to use another host on the network that is idle in order to perform a port scan to another host.
- The main advantage of this method is that it is very stealthy, because the firewall log files will record the IP address of the Zombie and not our IP.
- However, in order to have proper results, we must found hosts that are idle on the network.
- Metasploit framework has a scanner that can help us to discover hosts that are idle on the network and it can be used while implementing this type of scan.

Idle Zombie Scan

- Nmap's IPID Idle scanning allows us to be a little stealthy scanning a target while spoofing the IP address of another host on the network.
- In order for this type of scan to work, we will need to locate a host that is idle on the network and uses IPID sequences of either Incremental or Broken Little-Endian Incremental.
- Metasploit contains the module 'scanner/ip/ipidseq' to scan and look for a host that fits the requirements.

Idle Zombie Scan



```
msf > use auxiliary/scanner/ip/ipidseq
msf auxiliary(ipidseq) > show options

Module options (auxiliary/scanner/ip/ipidseq):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   RHOSTS                      yes       The target address range or CIDR identifier
   RPORT      80               yes       The target port
   SNAPLEN    65535            yes       The number of bytes to capture
   THREADS    1                yes       The number of concurrent threads
   TIMEOUT    500              yes       The reply read timeout in milliseconds

msf auxiliary(ipidseq) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf auxiliary(ipidseq) > set THREADS 50
THREADS => 50
msf auxiliary(ipidseq) > run
```

Idle Zombie Scan

- Based on the scan result we get from metasploit, we will use the hosts that have IPID Sequence class = Incremental as our Zombie hosts.

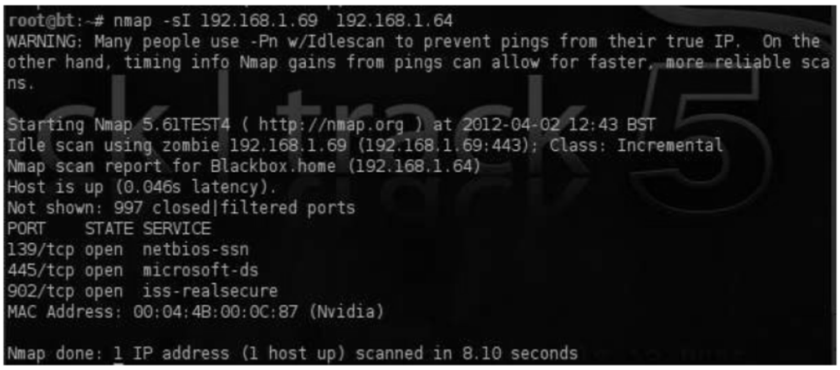
- Command:

```
Nmap –sI zombie_ip target_ip
```

or

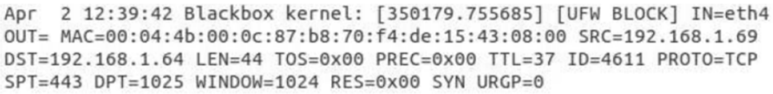
```
Nmap –Pn –sI ip_zombie –v ip_target
```

Idle Zombie Scan



```
root@bt:~# nmap -sI 192.168.1.69  192.168.1.64
WARNING: Many people use -Pn w/Idlescan to prevent pings from their true IP.  On the
other hand, timing info Nmap gains from pings can allow for faster, more reliable sca
ns.

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 12:43 BST
Idle scan using zombie 192.168.1.69 (192.168.1.69:443); Class: Incremental
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.046s latency).
Not shown: 997 closed|filtered ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap done: 1 IP address (1 host up) scanned in 8.10 seconds
```

Firewall Log:



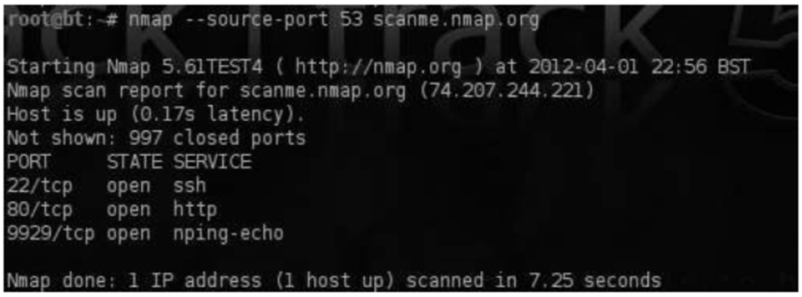
```
Apr  2 12:39:42 Blackbox kernel: [350179.755685] [UFW BLOCK] IN=eth4
OUT= MAC=00:04:4b:00:0c:87:b8:70:f4:de:15:43:08:00 SRC=192.168.1.69
DST=192.168.1.64 LEN=44 TOS=0x00 PREC=0x00 TTL=37 ID=4611 PROTO=TCP
SPT=443 DPT=1025 WINDOW=1024 RES=0x00 SYN URGP=0
```

Source Port Number Specification

- A common error that many administrators are doing when configuring firewalls is to set up a rule to allow all incoming traffic that comes from a specific port number.
- The --source-port option of Nmap can be used to exploit this misconfiguration.
- Common ports that you can use for this type of scan are: 20,53 and 67.

Source Port Number Specification

- Command: nmap --source-port 53 ip_target



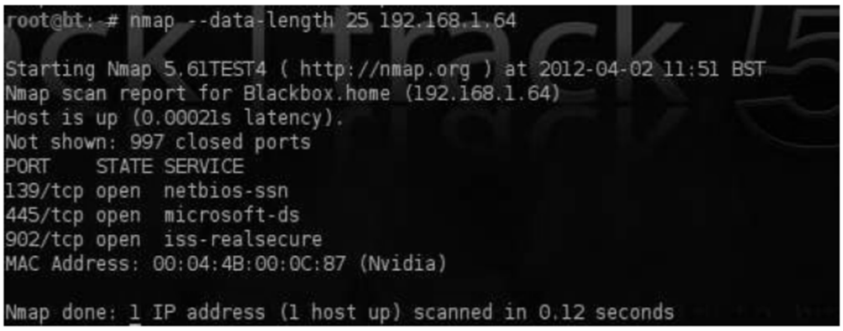
```
root@bt:~# nmap --source-port 53 scanme.nmap.org

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-01 22:56 BST
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.17s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
9929/tcp open  nping-echo

Nmap done: 1 IP address (1 host up) scanned in 7.25 seconds
```

Append Random Data

- Many firewalls are inspecting packets by looking at their size in order to identify a potential port scan.
- This is because many scanners are sending packets that have specific size.
- In order to avoid that kind of detection you can use the command --data-length to add additional data and to send packets with different size than the default.
- In the next slide we have changed the packet size by adding 25 more bytes.
- The size of a typical packet that nmap sends to the target is 58 bytes.

Append Random Data

- Command: nmap --data-length length_number ip_target

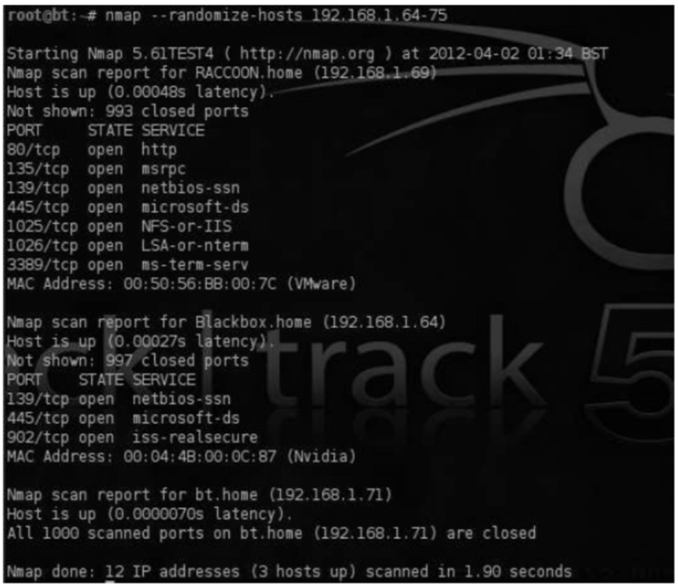


```
root@bt:~# nmap --data-length 25 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 11:51 BST
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00021s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Scan with Random Order

- In this technique you can scan a number of hosts in random order and not sequential.
- The command that you use to instruct Nmap to scan for hosts in random order is --randomize-hosts.
- This technique combined with slow timing (-T) options in nmap command can be very effective when you don’t want to alert firewalls.
- Command: nmap –randomize-hosts ip_target

Scan with Random Order



```
root@bt:~# nmap --randomize-hosts 192.168.1.64-75

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 01:34 BST
Nmap scan report for RACCOON.home (192.168.1.69)
Host is up (0.00048s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
1026/tcp open  LSA-or-nterm
3389/tcp open  ms-term-serv
MAC Address: 00:50:56:BB:00:7C (VMware)

Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00027s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap scan report for bt.home (192.168.1.71)
Host is up (0.0000070s latency).
All 1000 scanned ports on bt.home (192.168.1.71) are closed

Nmap done: 12 IP addresses (3 hosts up) scanned in 1.90 seconds
```

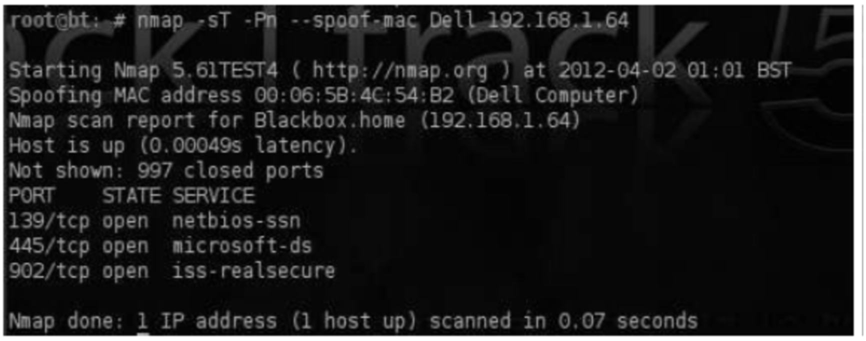
MAC Address Spoofing

- Another method for bypassing firewall restrictions while doing a port scan is by spoofing the MAC address of your host.
- This technique can be very effective especially if there is a MAC filtering rule to allow only traffic from certain MAC addresses, so you will need to discover which MAC address you need to set in order to obtain results.
- Specifically, the --spoof-mac option gives you the ability to choose a MAC address from a specific vendor, to choose a random MAC address or to set a specific MAC address of your choice.
- Another advantage of MAC address spoofing is that you make your scan more stealthier because your real MAC address will not appear on the firewall log files.

MAC Address Spoofing

- Command:
  - Specify MAC address from a Vendor: nmap --spoof-mac Dell/Apple/3Com ip_target
  - Generate a random MAC address: nmap --spoof-mac 0 ip_target
  - Specify your own MAC address: nmap --spoof-mac 00:01:02:25:56:AE ip_target

MAC Address Spoofing



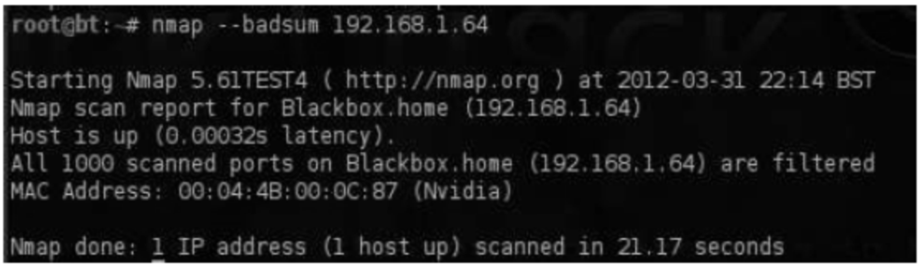
```
root@bt:~# nmap -sT -Pn --spoof-mac Dell 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-04-02 01:01 BST
Spoofing MAC address 00:06:5B:4C:54:B2 (Dell Computer)
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00049s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
902/tcp open  iss-realsecure

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Send Bad Checksums

- Checksums are used by the TCP/IP protocol to ensure the data integrity.
- However sending packets with incorrect checksums can help you discover information from systems that is not properly configured or when you are trying to avoid a firewall.
- You can use the command nmap --badsum IP_target in order to send packets with bad checksums to your targets.
- In the next image we didn’t get any results, this means that the system is correctly configured.

Send Bad Checksums

- Command: nmap –badsum ip_target



```
root@bt:~# nmap --badsum 192.168.1.64

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-03-31 22:14 BST
Nmap scan report for Blackbox.home (192.168.1.64)
Host is up (0.00032s latency).
All 1000 scanned ports on Blackbox.home (192.168.1.64) are filtered
MAC Address: 00:04:4B:00:0C:87 (Nvidia)

Nmap done: 1 IP address (1 host up) scanned in 21.17 seconds
```

End of Session

Any Questions?