Nmap – Techniques for Avoiding Firewalls

pentestlab.blog/category/information-gathering/page/10

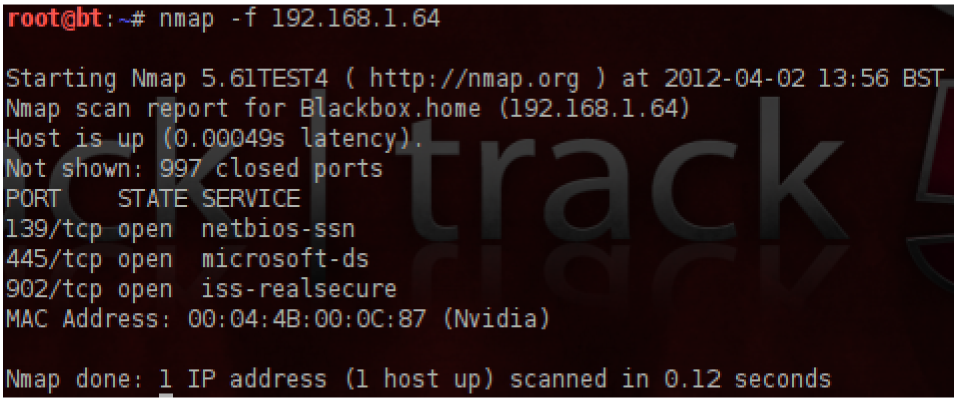
April 2, 2012

As a penetration tester you will come across with systems that are behind firewalls and they are blocking you from getting the information that you want.So you will need to know how 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 so in this article we will explore these options.

Fragment Packets

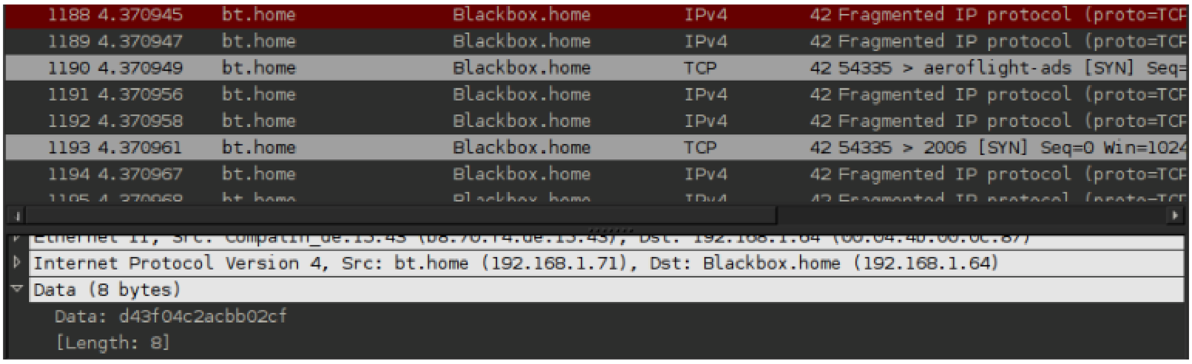
This technique was very effective especially in the old days however you can still use it if you found a firewall that is not properly configured.The Nmap offers that ability to fragment the packets while scanning with the **-f** option so it can bypass the packet inspection of firewalls.



```
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
```

Fragment Packets - Nmap

In the next image we can see that Nmap is sending packets 8-bytes size when we are doing a scan with the **-f** option.

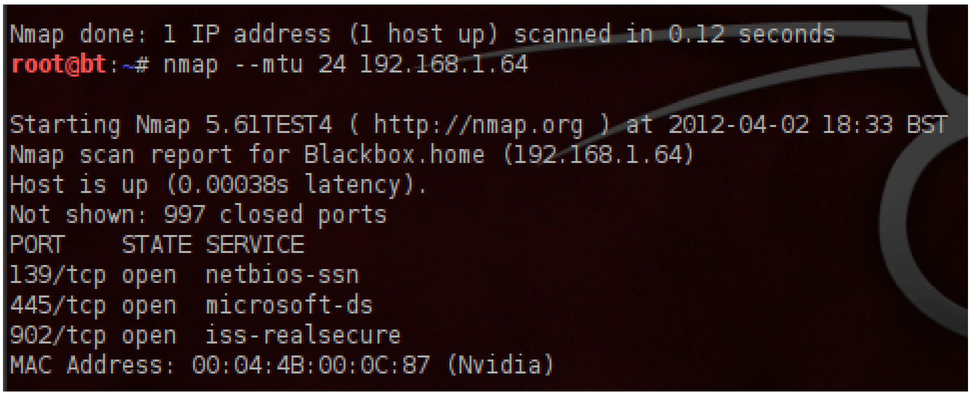


```
1188 4.370945   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP
1189 4.370947   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP
1190 4.370949   bt.home    Blackbox.home   TCP    42 54335 > aeroflight-ads [SYN] Seq=
1191 4.370956   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP
1192 4.370958   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP
1193 4.370961   bt.home    Blackbox.home   TCP    42 54335 > 2006 [SYN] Seq=0 Win=1024
1194 4.370967   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP
1195 4.370968   bt.home    Blackbox.home   IPv4   42 Fragmented IP protocol (proto=TCP

Ethernet II, Src: Compatib_de:13:43 (b8:70:f4:de:13:43), Dst: 192.168.1.64 (00:04:4b:00:0c:87)
Internet Protocol Version 4, Src: bt.home (192.168.1.71), Dst: Blackbox.home (192.168.1.64)
Data (8 bytes)
   Data: d43f04c2acbb02cf
   [Length: 8]
```

Capture a fragment packet

Specify a 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 that we have explained above.During the scan that size of the 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). You can specify the MTU of your choice with the command **–mtu number target.**



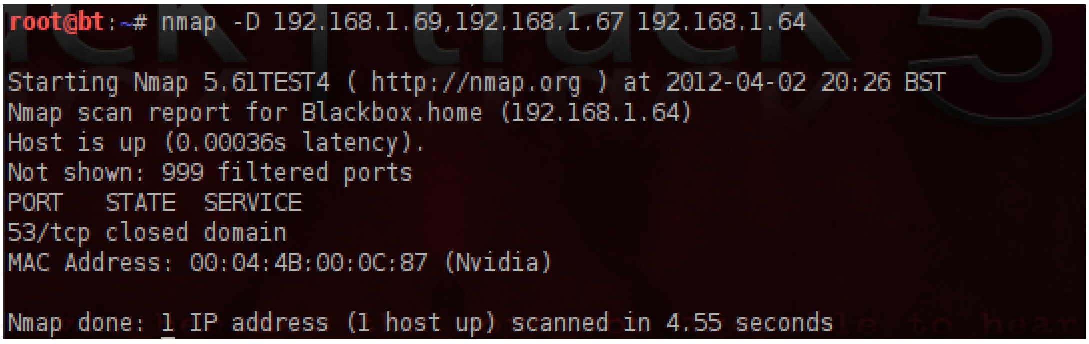
```
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
root@bt:~# nmap --mtu 24 192.168.1.64

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)
```

Specify a specific MTU to the packets

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 and 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:

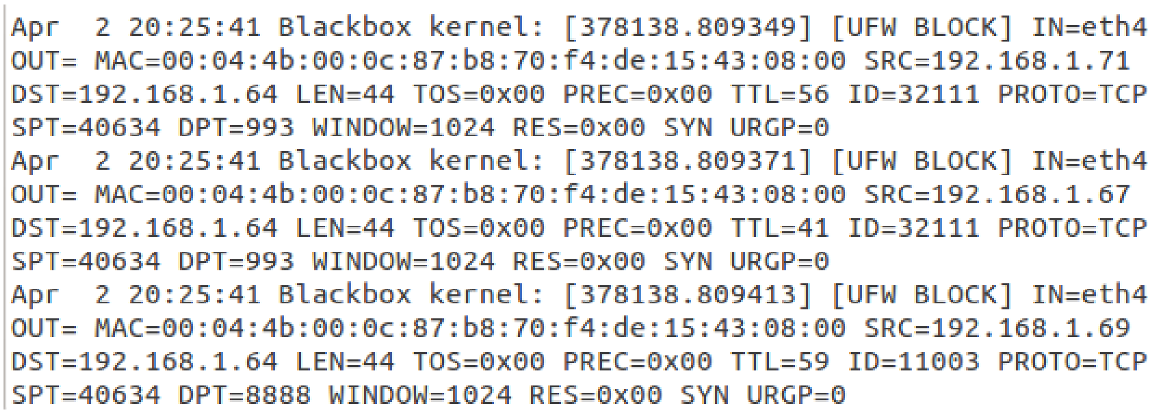
1. **nmap -D RND:10 [target]** (Generates a random number of decoys)
2. **nmap -D decoy1,decoy2,decoy3** etc. (Manually specify the IP addresses of the decoys)



```
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
```

Scanning with decoy addresses

In the next image we can see that in the firewall log files exist 3 different IP address.One is our real IP and the others are the decoys.



```
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
```

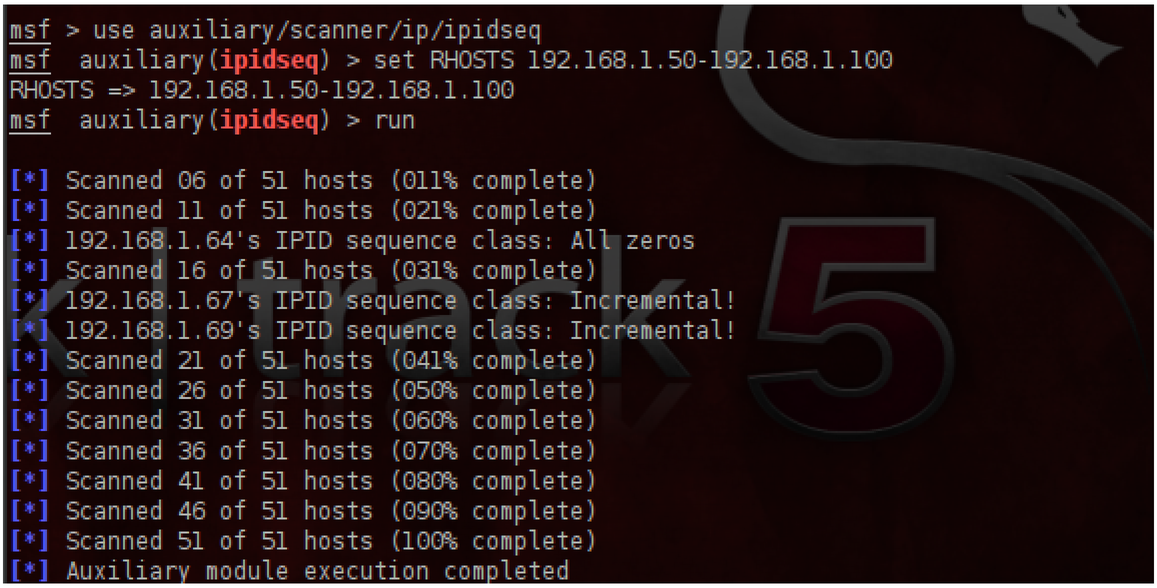
Log Files flooded with decoy addresses

You need to have in mind that the host that you will use as decoys must be online in order this technique to work.Also using many decoys can cause network congestion so you may want to avoid that especially if you are scanning the network of your client.

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 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.

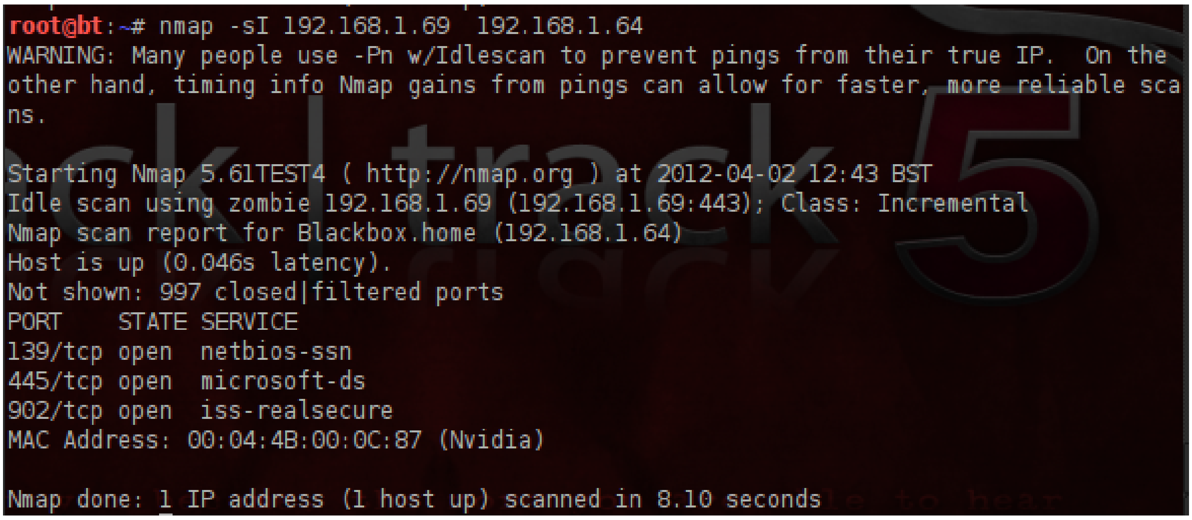
Metasploit framework has a scanner that can help us to discover hosts that are idle on the network and it can be used while implementing this type of scan.



```
msf > use auxiliary/scanner/ip/ipidseq
msf  auxiliary(ipidseq) > set RHOSTS 192.168.1.50-192.168.1.100
RHOSTS => 192.168.1.50-192.168.1.100
msf  auxiliary(ipidseq) > run

[*] Scanned 06 of 51 hosts (011% complete)
[*] Scanned 11 of 51 hosts (021% complete)
[*] 192.168.1.64's IPID sequence class: All zeros
[*] Scanned 16 of 51 hosts (031% complete)
[*] 192.168.1.67's IPID sequence class: Incremental!
[*] 192.168.1.69's IPID sequence class: Incremental!
[*] Scanned 21 of 51 hosts (041% complete)
[*] Scanned 26 of 51 hosts (050% complete)
[*] Scanned 31 of 51 hosts (060% complete)
[*] Scanned 36 of 51 hosts (070% complete)
[*] Scanned 41 of 51 hosts (080% complete)
[*] Scanned 46 of 51 hosts (090% complete)
[*] Scanned 51 of 51 hosts (100% complete)
[*] Auxiliary module execution completed
```

Discover Zombies

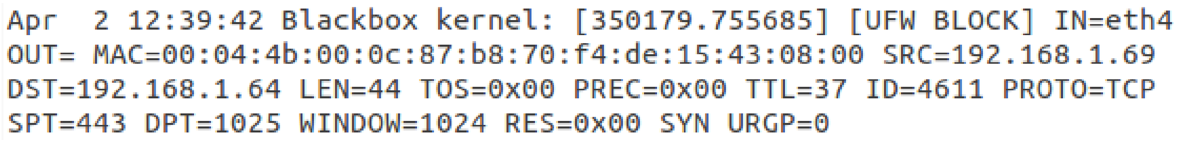
As we can see from the above image the scanner has discovered that the IP addresses 192.168.1.67 and 192.168.1.69 are idle on the network and are potential candidates for use on an Idle Zombie Scan.In order to implement an Idle Zombie scan we need to use the command **nmap -sI [Zombie IP] [Target IP]**



```
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
```

Executing an Idle Scan

We can see the effectiveness of this scan just by checking the firewall logs.As we can see the log files record the IP address of the Zombie host (SRC=192.168.1.69) and not our IP address so our scan was stealthy.

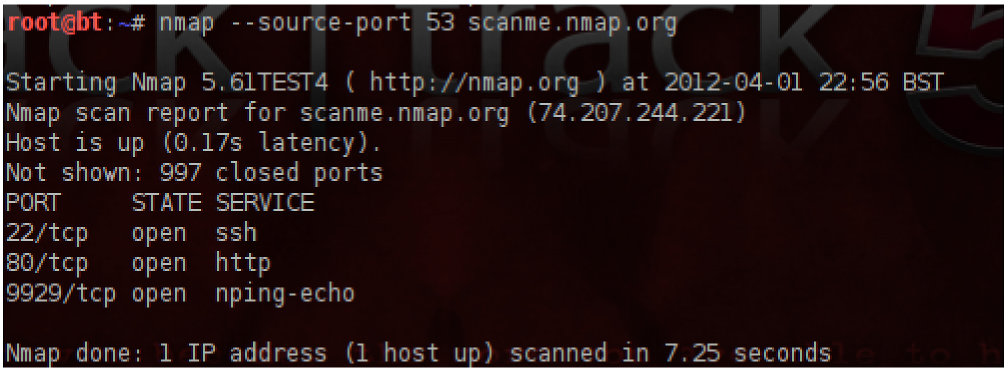


```
Apr  2 12:39:42 Blackbox kernel: [350179.755685] [UFW BLOCK] IN=eth4
OUT= MAC=00:04:4b:00:0c:87:b8:70:f4:de:15:43:08:00 SRC=192.168.1.69
DST=192.168.1.64 LEN=44 TOS=0x00 PREC=0x00 TTL=37 ID=4611 PROTO=TCP
SPT=443 DPT=1025 WINDOW=1024 RES=0x00 SYN URGP=0
```

Firewall Log Files - Idle Scan

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.

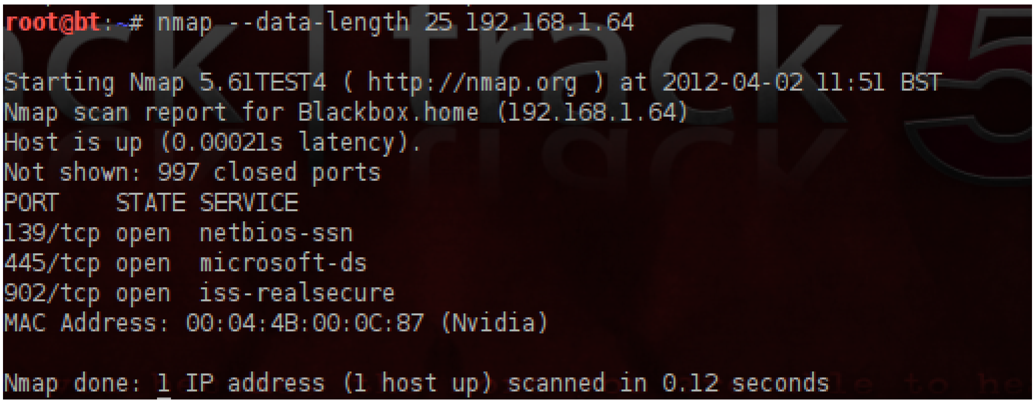


```
root@bt:~# nmap --source-port 53 scanme.nmap.org

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
```

Source port scan

Append Random Data

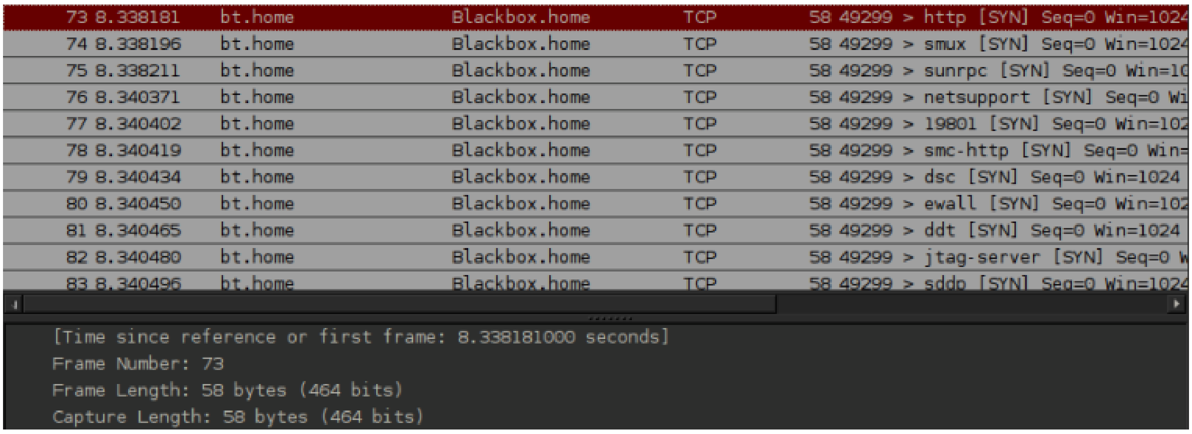
Many firewalls are inspecting packets by looking at their size in order to identify a potential port scan.This is because many scanners are sending packets that have specific size.In order to avoid that kind of detection you can use the command **–data-length** to add additional data and to send packets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



```
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
```

Adding random data to avoid detection

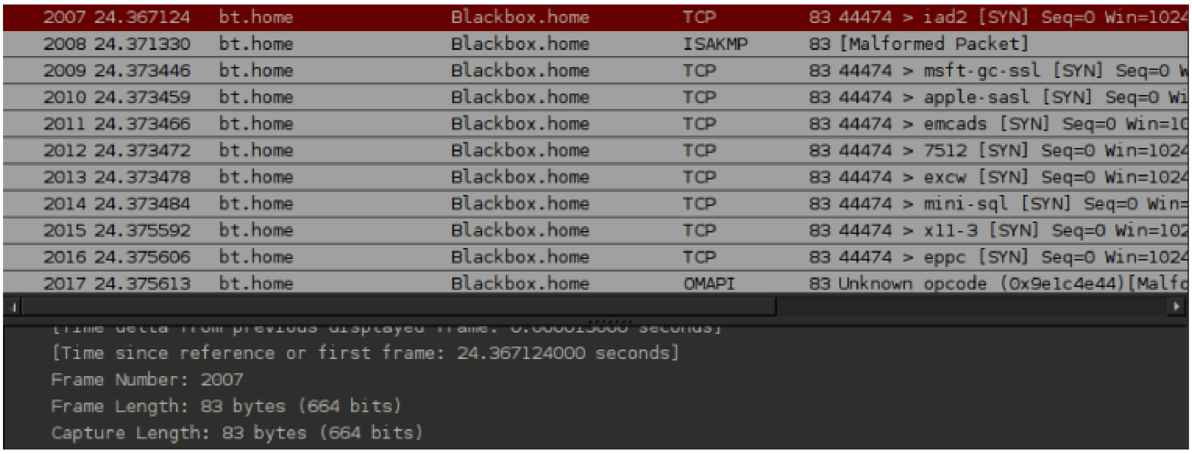
The size of a typical packet that nmap sends to the target is 58 bytes as you can see in the image below.



```
73 8.338181   bt.home   Blackbox.home   TCP   58 49299 > http [SYN] Seq=0 Win=1024
74 8.338196   bt.home   Blackbox.home   TCP   58 49299 > smux [SYN] Seq=0 Win=1024
75 8.338211   bt.home   Blackbox.home   TCP   58 49299 > sunrpc [SYN] Seq=0 Win=10
76 8.340371   bt.home   Blackbox.home   TCP   58 49299 > netsupport [SYN] Seq=0 Wi
77 8.340402   bt.home   Blackbox.home   TCP   58 49299 > 19801 [SYN] Seq=0 Win=102
78 8.340419   bt.home   Blackbox.home   TCP   58 49299 > smc-http [SYN] Seq=0 Win=
79 8.340434   bt.home   Blackbox.home   TCP   58 49299 > dsc [SYN] Seq=0 Win=1024
80 8.340450   bt.home   Blackbox.home   TCP   58 49299 > ewall [SYN] Seq=0 Win=102
81 8.340465   bt.home   Blackbox.home   TCP   58 49299 > ddt [SYN] Seq=0 Win=1024
82 8.340480   bt.home   Blackbox.home   TCP   58 49299 > jtag-server [SYN] Seq=0 W
83 8.340496   bt.home   Blackbox.home   TCP   58 49299 > sddp [SYN] Seq=0 Win=1024

    [Time since reference or first frame: 8.338181000 seconds]
    Frame Number: 73
    Frame Length: 58 bytes (464 bits)
    Capture Length: 58 bytes (464 bits)
```

Typical packet from nmap scan

With the command that we have used **–data-length 25** we changed that value to 83 in order to avoid being discovered by firewalls that will check for the default packet size that nmap generates.

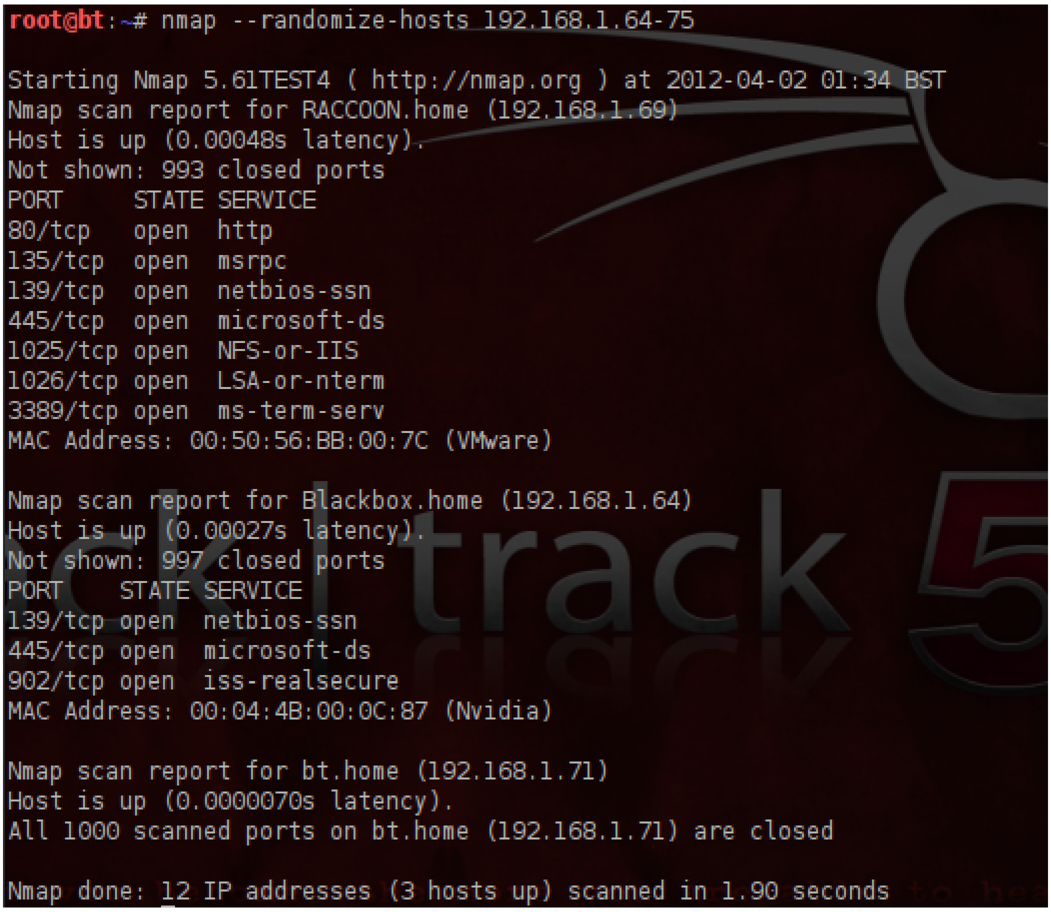


```
2007 24.367124   bt.home   Blackbox.home   TCP      83 44474 > iad2 [SYN] Seq=0 Win=1024
2008 24.371330   bt.home   Blackbox.home   ISAKMP   83 [Malformed Packet]
2009 24.373446   bt.home   Blackbox.home   TCP      83 44474 > msft-gc-ssl [SYN] Seq=0 W
2010 24.373459   bt.home   Blackbox.home   TCP      83 44474 > apple-sasl [SYN] Seq=0 Wi
2011 24.373466   bt.home   Blackbox.home   TCP      83 44474 > emcads [SYN] Seq=0 Win=10
2012 24.373472   bt.home   Blackbox.home   TCP      83 44474 > 7512 [SYN] Seq=0 Win=1024
2013 24.373478   bt.home   Blackbox.home   TCP      83 44474 > excw [SYN] Seq=0 Win=1024
2014 24.373484   bt.home   Blackbox.home   TCP      83 44474 > mini-sql [SYN] Seq=0 Win=
2015 24.375592   bt.home   Blackbox.home   TCP      83 44474 > x11-3 [SYN] Seq=0 Win=102
2016 24.375606   bt.home   Blackbox.home   TCP      83 44474 > eppc [SYN] Seq=0 Win=1024
2017 24.375613   bt.home   Blackbox.home   OMAPI    83 Unknown opcode (0x9e1c4e44)[Malfo

    [Time delta from previous displayed frame: 0.000013000 seconds]
    [Time since reference or first frame: 24.367124000 seconds]
    Frame Number: 2007
    Frame Length: 83 bytes (664 bits)
    Capture Length: 83 bytes (664 bits)
```

A sample of a packet that we have add 25 more bytes to avoid detection

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 host in random order is **–randomize-hosts**.This technique combined with slow timing options in nmap command can be very effective when you don't want to alert firewalls.



```
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
```

Scan hosts in random order

MAC Address Spoofing

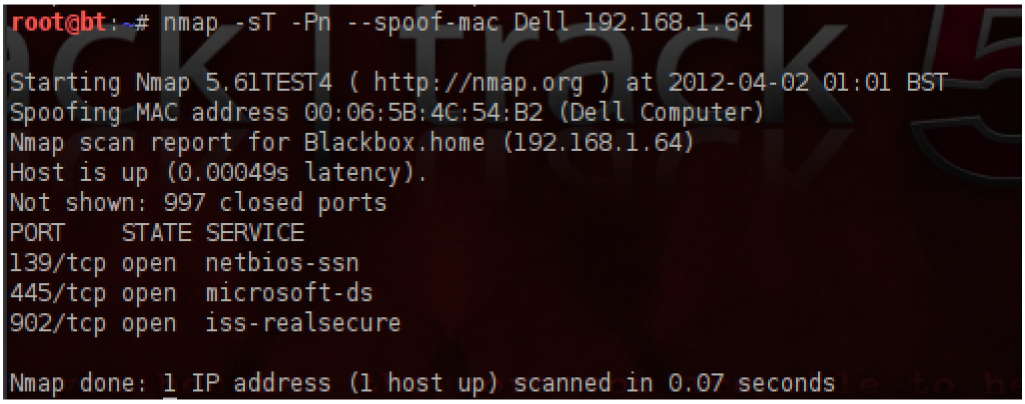
Another method for bypassing firewall restrictions while doing a port scan is by spoofing the MAC address of your host.This technique can be very effective especially if there is a MAC filtering rule to allow only traffic from certain MAC addresses so you will need to discover which MAC address you need 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 it will not appear on the firewall log files.

Specify MAC address from a Vendor —-> **–spoof-mac** Dell/Apple/3Com

Generate a random MAC address —-> —**spoof-mac** 0

Specify your own MAC address —-> —**spoof-mac** 00:01:02:25:56:AE



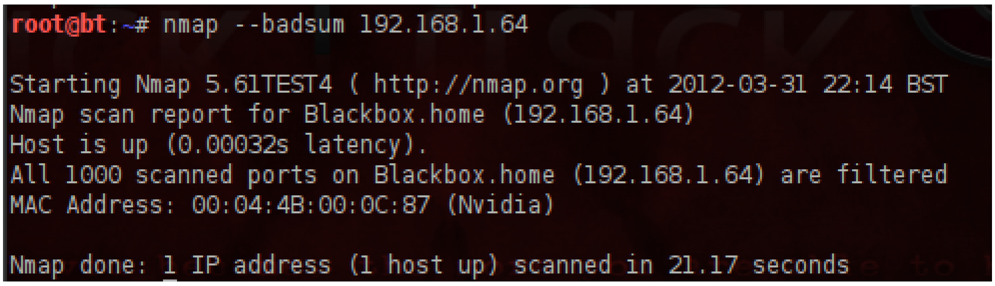
```
root@bt:~# nmap -sT -Pn --spoof-mac Dell 192.168.1.64

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
```

MAC address Spoofing

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 to discover information from systems that is not properly configured or when you are trying to avoid a firewall.

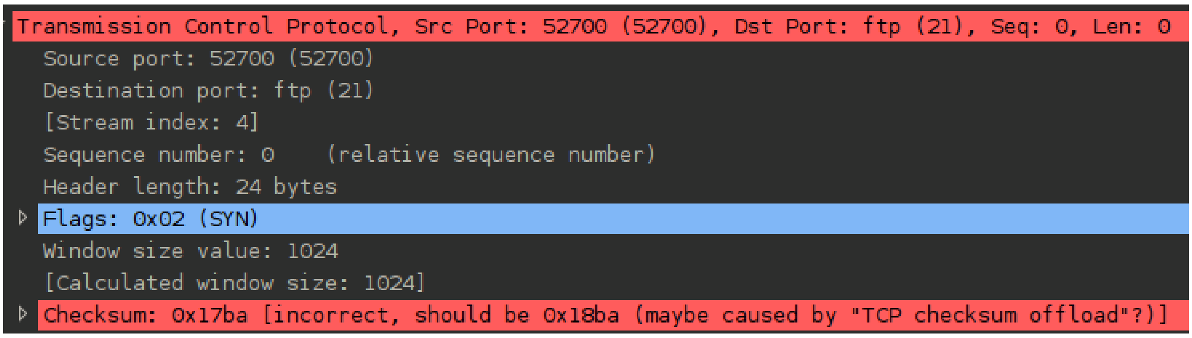
You can use the command **nmap –badsum IP** in order to send packets with bad checksums to your targets.In the image below we didn't get any results.This means that the system is suitable configured.



```
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
```

Sending packets with bad checksum

You can see below a sample of a packet with bad checksum that we have sent:



```
Transmission Control Protocol, Src Port: 52700 (52700), Dst Port: ftp (21), Seq: 0, Len: 0
    Source port: 52700 (52700)
    Destination port: ftp (21)
    [Stream index: 4]
    Sequence number: 0    (relative sequence number)
    Header length: 24 bytes
  ▷ Flags: 0x02 (SYN)
    Window size value: 1024
    [Calculated window size: 1024]
  ▷ Checksum: 0x17ba [incorrect, should be 0x18ba (maybe caused by "TCP checksum offload"?)]
```

A packet with bad checksum

Conclusion

We have seen that Nmap offers a variety of methods that it can be used to avoid a firewall that exists on the network that we are scanning and to get proper results from the target host.The problem in many of the cases that we have seen is the bad configuration of Firewalls that allowed us to get results from the target.So in a network that have IDS and firewalls properly configured many of the techniques may not work.Every situation is different so you need to decide which one will work for you.