Information gathering and Scanning Tools

Ping:

Ping is a utility used to send out ICMP packets to an address to see how fast the response is (check if the host exists). Ping is important when it comes to website latency as it corresponds with the delay time (in milliseconds) for how long it takes the data to travel across the internet, to its destination address, and then back to you.

#ping domain name
#ping ip address

Nslookup:

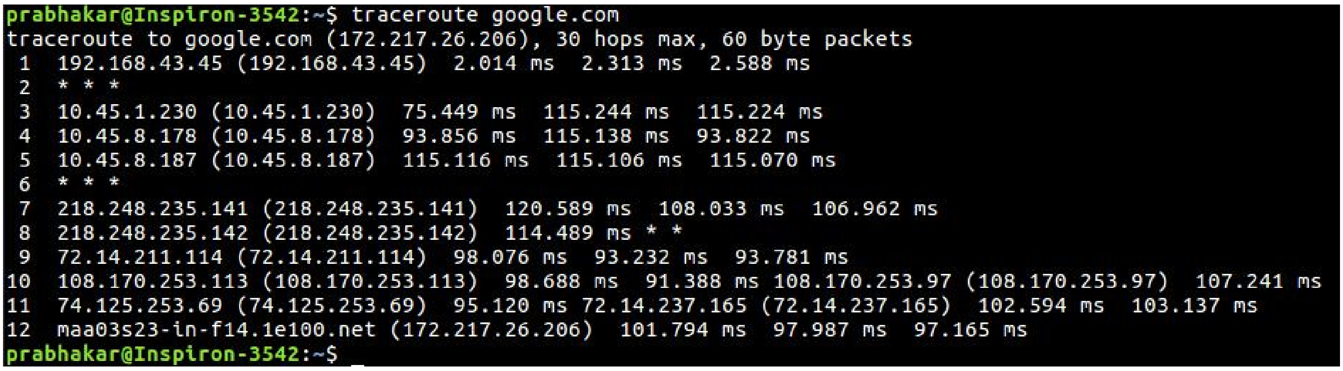
DNS lookup tool is to find the IP address of a certain domain name. The results will include the IP addresses in the DNS records received from the name servers.

#nslookup domain name

Traceroute:

A traceroute is a network tool used to show the route taken by packets across an IP network.
The Traceroute tool will show you each hop sequentially, and total hops required. For each hop, it will display the hop #, roundtrip times, best time (ms), IP address, TTL, and country.

#traceroute domain name



```
prabhakar@Inspiron-3542:~$ traceroute google.com
traceroute to google.com (172.217.26.206), 30 hops max, 60 byte packets
 1  192.168.43.45 (192.168.43.45)  2.014 ms  2.313 ms  2.588 ms
 2  * * *
 3  10.45.1.230 (10.45.1.230)  75.449 ms  115.244 ms  115.224 ms
 4  10.45.8.178 (10.45.8.178)  93.856 ms  115.138 ms  93.822 ms
 5  10.45.8.187 (10.45.8.187)  115.116 ms  115.106 ms  115.070 ms
 6  * * *
 7  218.248.235.141 (218.248.235.141)  120.589 ms  108.033 ms  106.962 ms
 8  218.248.235.142 (218.248.235.142)  114.489 ms * *
 9  72.14.211.114 (72.14.211.114)  98.076 ms  93.232 ms  93.781 ms
10  108.170.253.113 (108.170.253.113)  98.688 ms  91.388 ms 108.170.253.97 (108.170.253.97)  107.241 ms
11  74.125.253.69 (74.125.253.69)  95.120 ms 72.14.237.165 (72.14.237.165)  102.594 ms  103.137 ms
12  maa03s23-in-f14.1e100.net (172.217.26.206)  101.794 ms  97.987 ms  97.165 ms
prabhakar@Inspiron-3542:~$ _
```

Whois:

WHOIS (pronounced as the phrase "who is") is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block or an autonomous system, but is also used for a wider range of other information. The protocol stores and delivers database content in a human-readable format.

#whois domain name
#whois ip adress

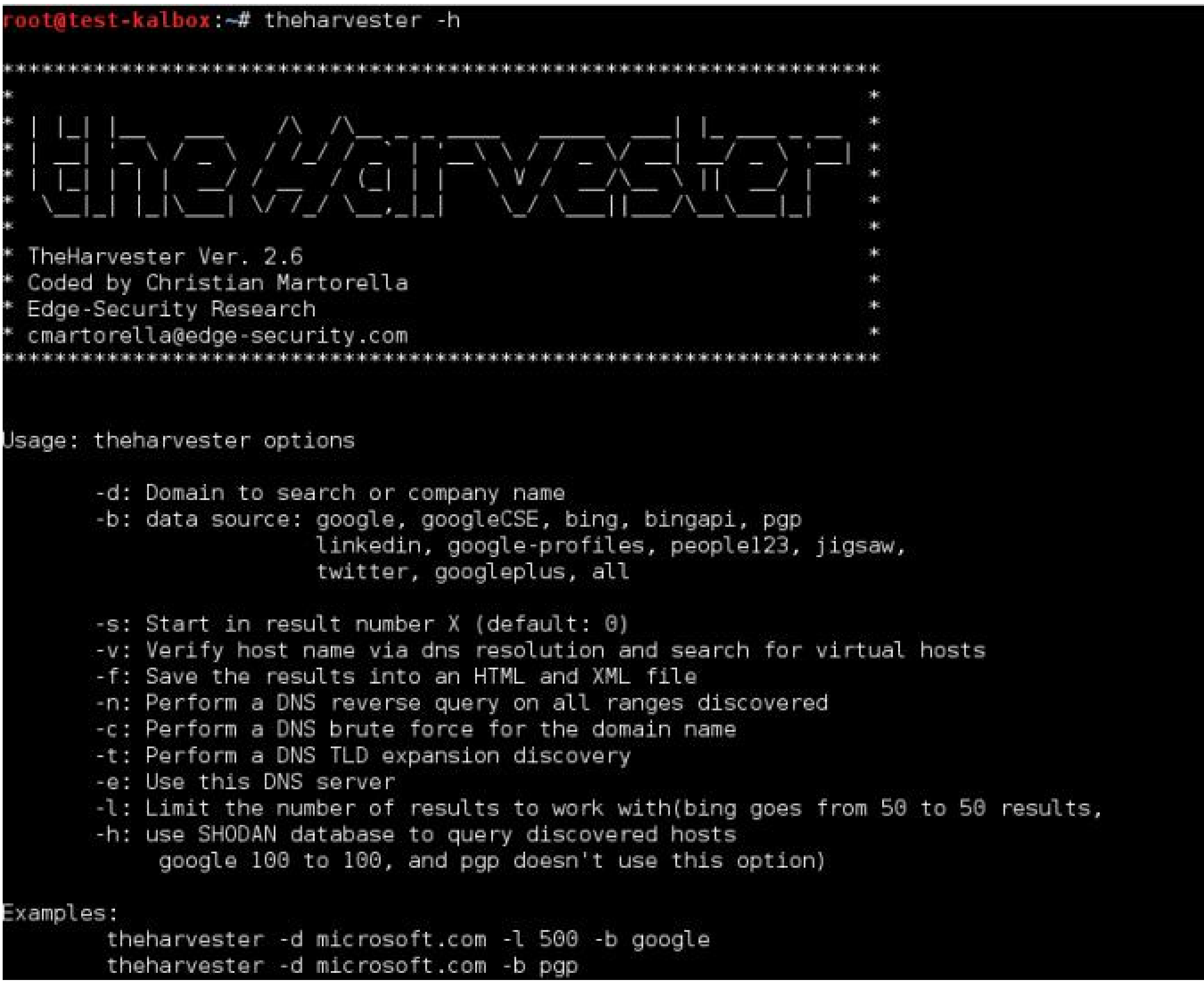
WHOIS search results

```
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited
https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited
https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited
https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited
https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited
https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form:
https://www.icann.org/wicf/
>>> Last update of whois database: 2020-08-23T17:25:42Z <<<
```

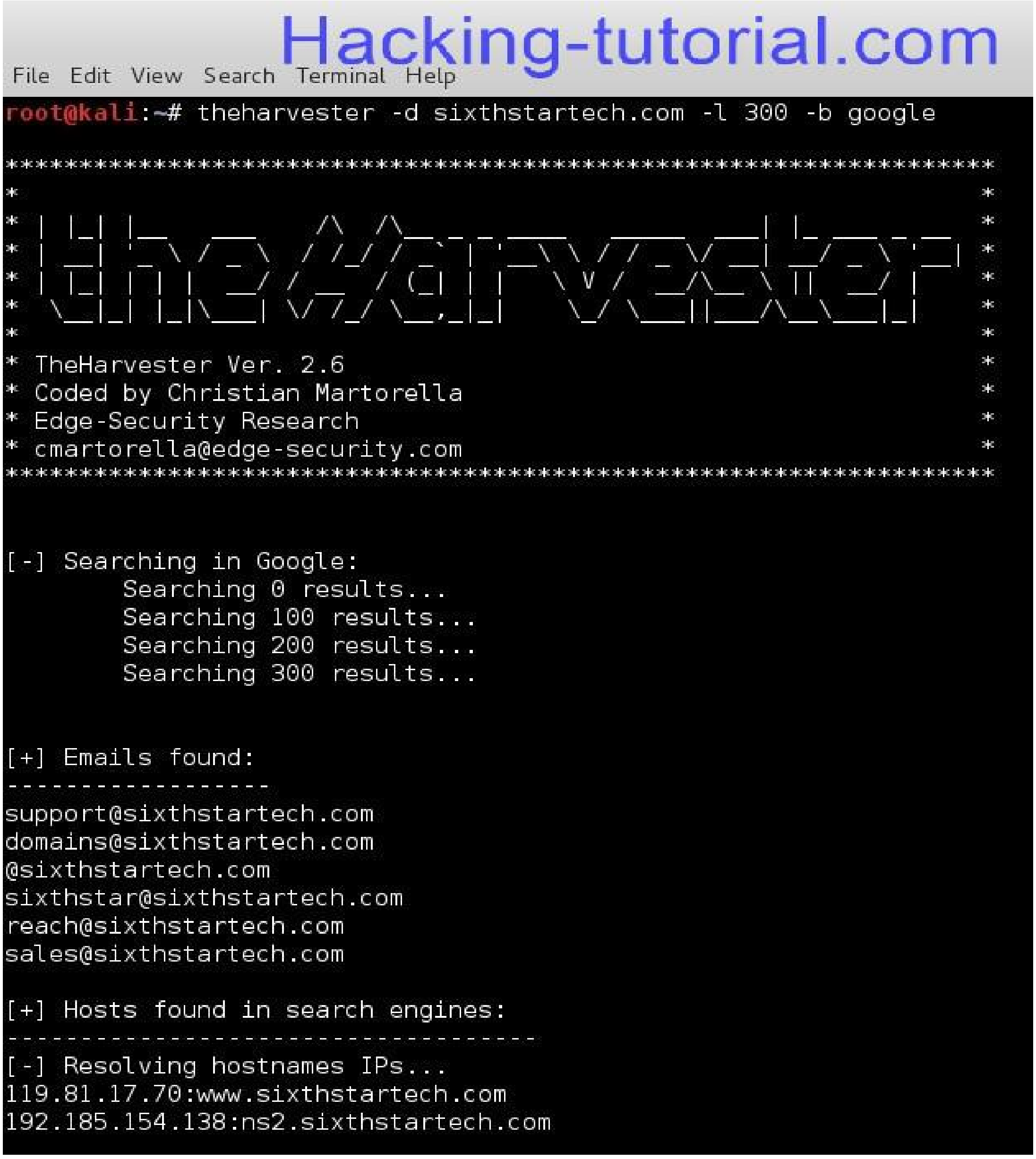
The harvester tool:

The Harvester is a tool that was developed in python. Using this you can gather information like emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers, and SHODAN computer database.

#theharvester -d [domain name] -l [no. Of searches] -b [search engine name / all][options] [parameters]
#theharvester -d microsoft.com -l 500 -b google
#theharvester -d microsoft -l 200 -b linkedin
#theharvester -d microsoft.com -l 500 -b google -h myresults.html



```
root@test-kalbox:~# theharvester -h

*******************************************************************
*                                                                 *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __| '_ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 2.6                                           *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************


Usage: theharvester options

       -d: Domain to search or company name
       -b: data source: google, googleCSE, bing, bingapi, pgp
                        linkedin, google-profiles, people123, jigsaw,
                        twitter, googleplus, all

       -s: Start in result number X (default: 0)
       -v: Verify host name via dns resolution and search for virtual hosts
       -f: Save the results into an HTML and XML file
       -n: Perform a DNS reverse query on all ranges discovered
       -c: Perform a DNS brute force for the domain name
       -t: Perform a DNS TLD expansion discovery
       -e: Use this DNS server
       -l: Limit the number of results to work with(bing goes from 50 to 50 results,
       -h: use SHODAN database to query discovered hosts
            google 100 to 100, and pgp doesn't use this option)

Examples:
        theharvester -d microsoft.com -l 500 -b google
        theharvester -d microsoft.com -b pgp
```



```
root@kali:~# theharvester -d sixthstartech.com -l 300 -b google

*******************************************************************
*                                                                 *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __| '_ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 2.6                                           *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************


[-] Searching in Google:
        Searching 0 results...
        Searching 100 results...
        Searching 200 results...
        Searching 300 results...


[+] Emails found:
------------------
support@sixthstartech.com
domains@sixthstartech.com
@sixthstartech.com
sixthstar@sixthstartech.com
reach@sixthstartech.com
sales@sixthstartech.com

[+] Hosts found in search engines:
------------------------------------
[-] Resolving hostnames IPs...
119.81.17.70:www.sixthstartech.com
192.185.154.138:ns2.sixthstartech.com
```

DMitry:

DMitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. DMitry has the ability to gather as much information as possible about a host. Base functionality is able to gather possible subdomains, email addresses, uptime information, tcp port scan, whois lookups, and more.

The following is a list of the current features:

- An Open Source Project.
- Perform an Internet Number whois lookup.

- Retrieve possible uptime data, system and server data.
- Perform a SubDomain search on a target host.
- Perform an E-Mail address search on a target host.
- Perform a TCP Portscan on the host target.

```
# dmitry -h
```

```
# dmitry -winsepo [file_name.txt] [domain_name]
```

```
# dmitry -winsepo example.txt example.com
```

hping3:

hping3 is another tool used for scan network. it is available in kali linux by default it is one of DOS attack software, ddos stand for distributed denial of service attack. you can launch and stop dos attack, whenever you want.

A simple DOS (not DDOS) attack

```
#hping3 -S --flood -V -p 80 <ip address>
```

Where:
note:need rootprivileges to run hping3.
hping3: calls hping3 program.
-S: specifies SYN packets.
–flood: shoot at discretion, replies will be ignored (that's why replies wont be shown) and packets will be sent fast as possible.
-V: Verbosity.
-p 80: port 80, you can replace this number for the service you want to attack.

The following example portrays a SYN attack against lacampora.org:

```
#hping3 lacampora.org -q -n -d 120 -S -p 80 --flood --rand-source
```

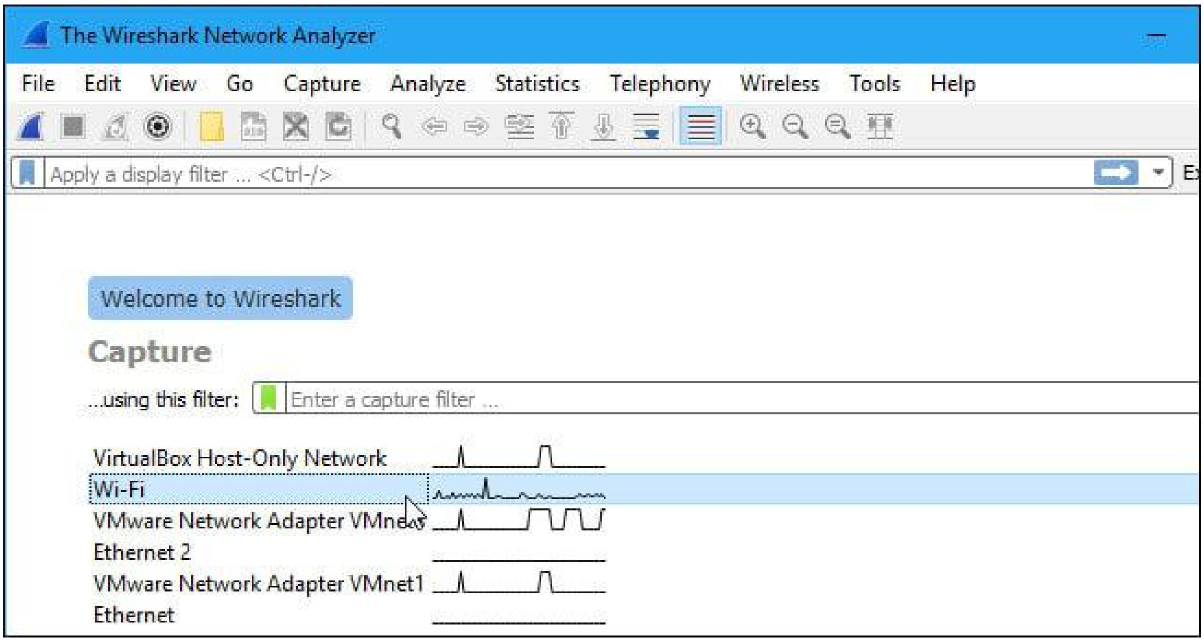
Where:
Lacampora.org: is the target
-q: brief output
-n: show target IP instead of host.
-d 120: set packet size
–rand-source: hide IP address.

SYN flood against port 80:

```
# sudo hping3 --rand-source ivan.com -S -q -p 80 --flood
```

Wireshark:

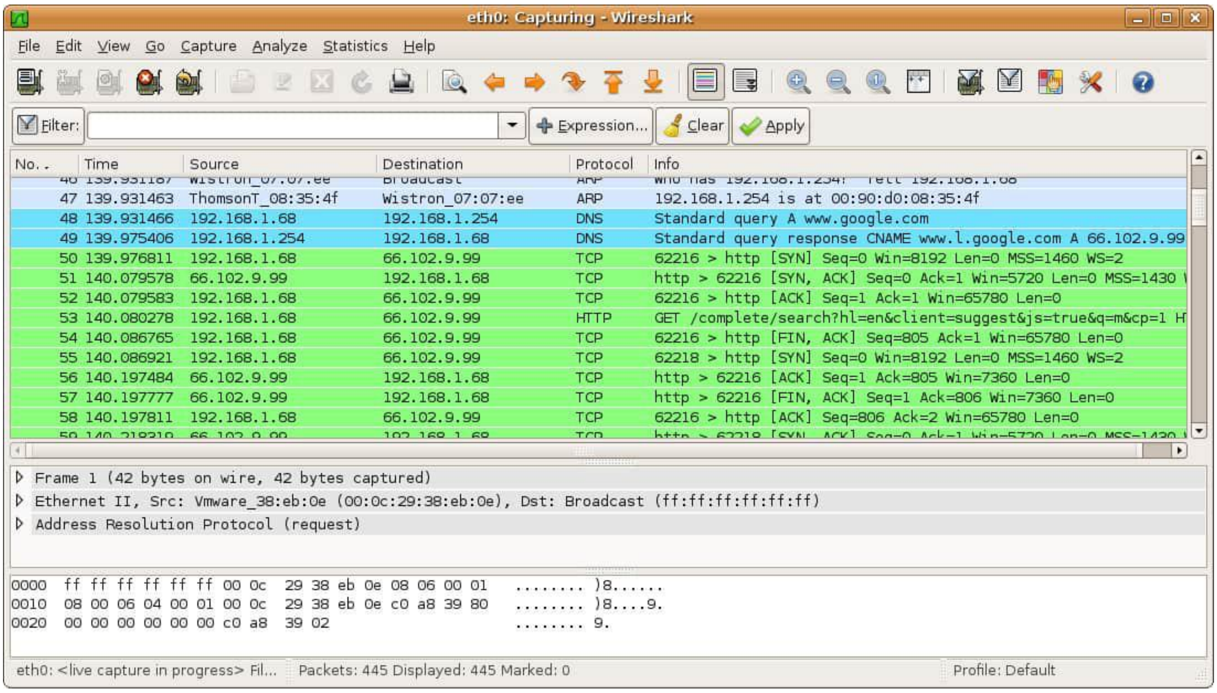
Wireshark is the world’s foremost network protocol analyzer. It lets you see what’s happening on your network at a microscopic level.



Capture

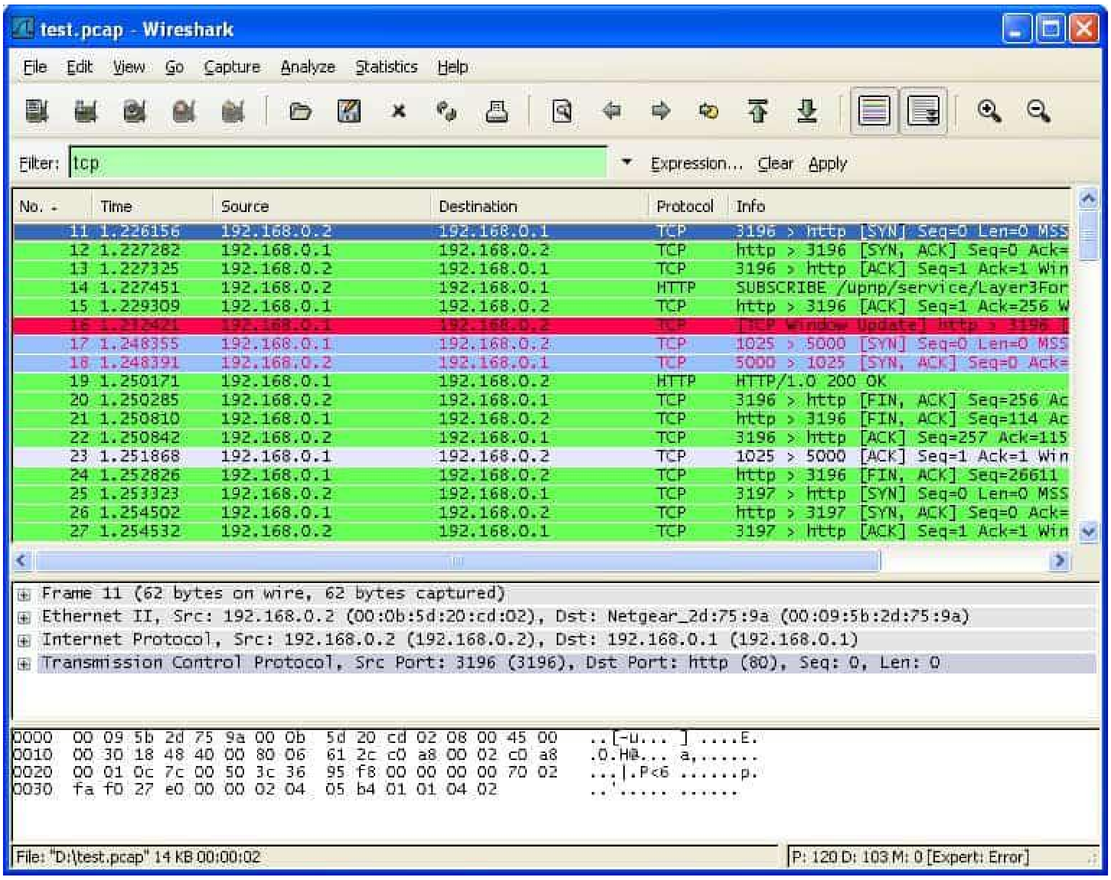
...using this filter: Enter a capture filter ...

VirtualBox Host-Only Network
Wi-Fi
VMware Network Adapter VMnet8
Ethernet 2
VMware Network Adapter VMnet1
Ethernet



```
No. .  Time        Source            Destination      Protocol  Info
 46 139.931187 Wistron_07:07:ee  Broadcast        ARP       Who has 192.168.1.254?  Tell 192.168.1.68
 47 139.931463 ThomsonT_08:35:4f Wistron_07:07:ee ARP       192.168.1.254 is at 00:90:d0:08:35:4f
 48 139.931466 192.168.1.68      192.168.1.254    DNS       Standard query A www.google.com
 49 139.975406 192.168.1.254     192.168.1.68     DNS       Standard query response CNAME www.l.google.com A 66.102.9.99
 50 139.976811 192.168.1.68      66.102.9.99      TCP       62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
 51 140.079578 66.102.9.99       192.168.1.68     TCP       http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
 52 140.079583 192.168.1.68      66.102.9.99      TCP       62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
 53 140.080278 192.168.1.68      66.102.9.99      HTTP      GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
 54 140.086765 192.168.1.68      66.102.9.99      TCP       62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
 55 140.086921 192.168.1.68      66.102.9.99      TCP       62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
 56 140.197484 66.102.9.99       192.168.1.68     TCP       http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
 57 140.197777 66.102.9.99       192.168.1.68     TCP       http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
 58 140.197811 192.168.1.68      66.102.9.99      TCP       62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0
 59 140.218319 66.102.9.99       192.168.1.68     TCP       http > 62218 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430

▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 0c  29 38 eb 0e 08 06 00 01   ........ )8......
0010  08 00 06 04 00 01 00 0c  29 38 eb 0e c0 a8 39 80   ........ )8....9.
0020  00 00 00 00 00 00 c0 a8  39 02                     ........ 9.

eth0: <live capture in progress> Fil...   Packets: 445 Displayed: 445 Marked: 0          Profile: Default
```

Capture Filters and **Display Filters** are two types of distinct filters that can be used on Wireshark. Capture Filters are used to reduce the size of incoming packet capture, essentially filtering out other packets during live packet capturing. As a result, capture filters are set before you begin the live capture process.



test.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: tcp
Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Len=0 MSS
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/Layer3For
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 W
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Window Update] http > 3196 [
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Len=0 MSS
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] Seq=0 Ack=
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK
20	1.250285	192.168.0.2	192.168.0.1	TCP	3196 > http [FIN, ACK] Seq=256 Ac
21	1.250810	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] Seq=114 Ac
22	1.250842	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=257 Ack=115
23	1.251868	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [ACK] Seq=1 Ack=1 Win
24	1.252826	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] Seq=26611
25	1.253323	192.168.0.2	192.168.0.1	TCP	3197 > http [SYN] Seq=0 Len=0 MSS
26	1.254502	192.168.0.1	192.168.0.2	TCP	http > 3197 [SYN, ACK] Seq=0 Ack=
27	1.254532	192.168.0.2	192.168.0.1	TCP	3197 > http [ACK] Seq=1 Ack=1 Win

```
⊞ Frame 11 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
⊞ Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 30 18 48 40 00 80 06  61 2c c0 a8 00 02 c0 a8   .0.H@... a,......
0020  00 01 0c 7c 00 50 3c 36  95 f8 00 00 00 00 70 02   ...|.P<6 ......p.
0030  fa f0 27 e0 00 00 02 04  05 b4 01 01 04 02         ..'..... ......
```

File: "D:\test.pcap" 14 KB 00:00:02

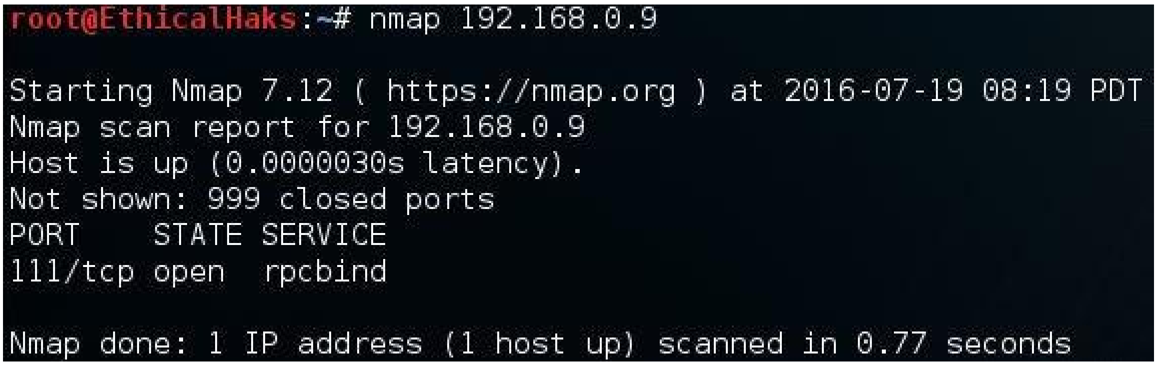
P: 120 D: 103 M: 0 [Expert: Error]

Nmap:

Nmap (Network Mapper) is the leading security scanner, written in C/C++, it is useful to discover hosts, to map and scan networks, hosts and ports and by implementing the NSE (Nmap Scripting Engine) you can also detect vulnerabilities on your target

Scan a single IP

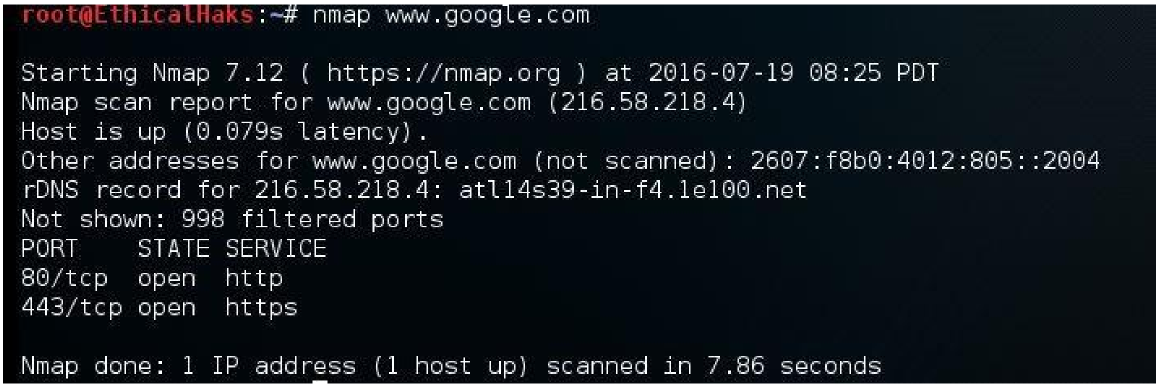
#nmap ipaddress



```
root@EthicalHaks:~# nmap 192.168.0.9

Starting Nmap 7.12 ( https://nmap.org ) at 2016-07-19 08:19 PDT
Nmap scan report for 192.168.0.9
Host is up (0.0000030s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 0.77 seconds
```

Scan a host

#nmap domain_name

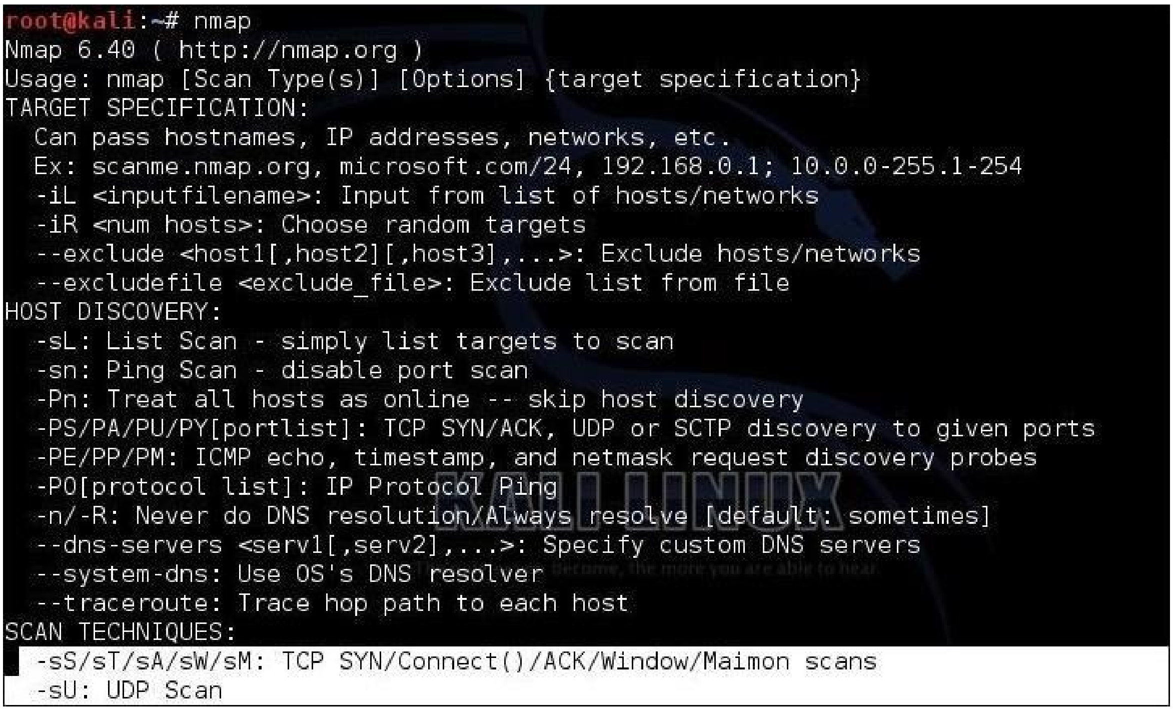


```
root@EthicalHaks:~# nmap www.google.com

Starting Nmap 7.12 ( https://nmap.org ) at 2016-07-19 08:25 PDT
Nmap scan report for www.google.com (216.58.218.4)
Host is up (0.079s latency).
Other addresses for www.google.com (not scanned): 2607:f8b0:4012:805::2004
rDNS record for 216.58.218.4: atl14s39-in-f4.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 7.86 seconds
```

Ping Scan

```
# nmap -sp 192.100.1.1/24
```

Scans the subnet for active hosts and displays the Physical address



```
root@kali:~# nmap
Nmap 6.40 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
```

#nmap <scantype> IP address

Scan for Specific Port or Port Range

#nmap <IP address> -p25-150

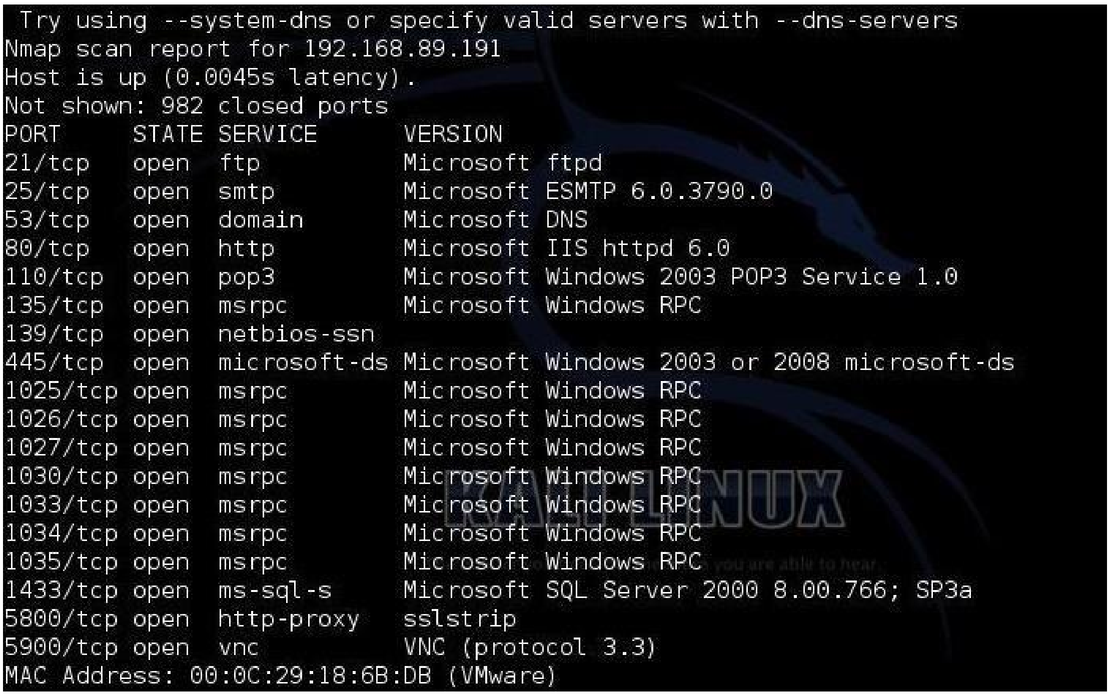
#nmap 192.168.89.0/24 -p80

Evading Firewalls

#nmap -sS -P0 <IP address>

Gathering Version Info

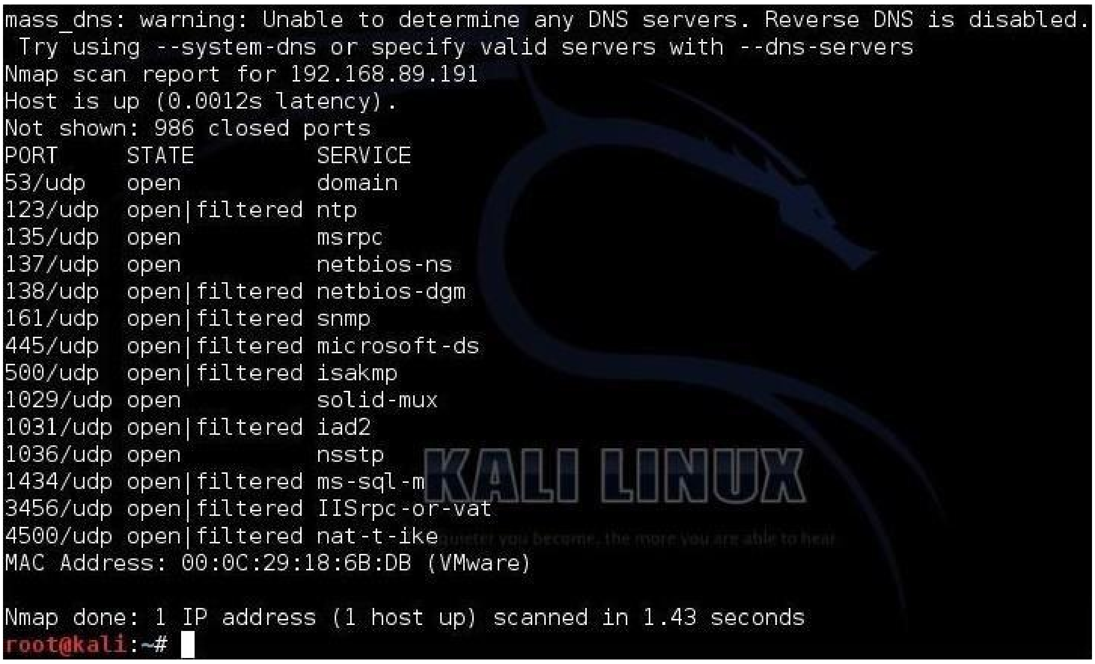
#nmap -V <IP address>



```
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.89.191
Host is up (0.0045s latency).
Not shown: 982 closed ports
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          Microsoft ftpd
25/tcp   open  smtp         Microsoft ESMTP 6.0.3790.0
53/tcp   open  domain       Microsoft DNS
80/tcp   open  http         Microsoft IIS httpd 6.0
110/tcp  open  pop3         Microsoft Windows 2003 POP3 Service 1.0
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 2003 or 2008 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
1027/tcp open  msrpc        Microsoft Windows RPC
1030/tcp open  msrpc        Microsoft Windows RPC
1033/tcp open  msrpc        Microsoft Windows RPC
1034/tcp open  msrpc        Microsoft Windows RPC
1035/tcp open  msrpc        Microsoft Windows RPC
1433/tcp open  ms-sql-s     Microsoft SQL Server 2000 8.00.766; SP3a
5800/tcp open  http-proxy   sslstrip
5900/tcp open  vnc          VNC (protocol 3.3)
MAC Address: 00:0C:29:18:6B:DB (VMware)
```

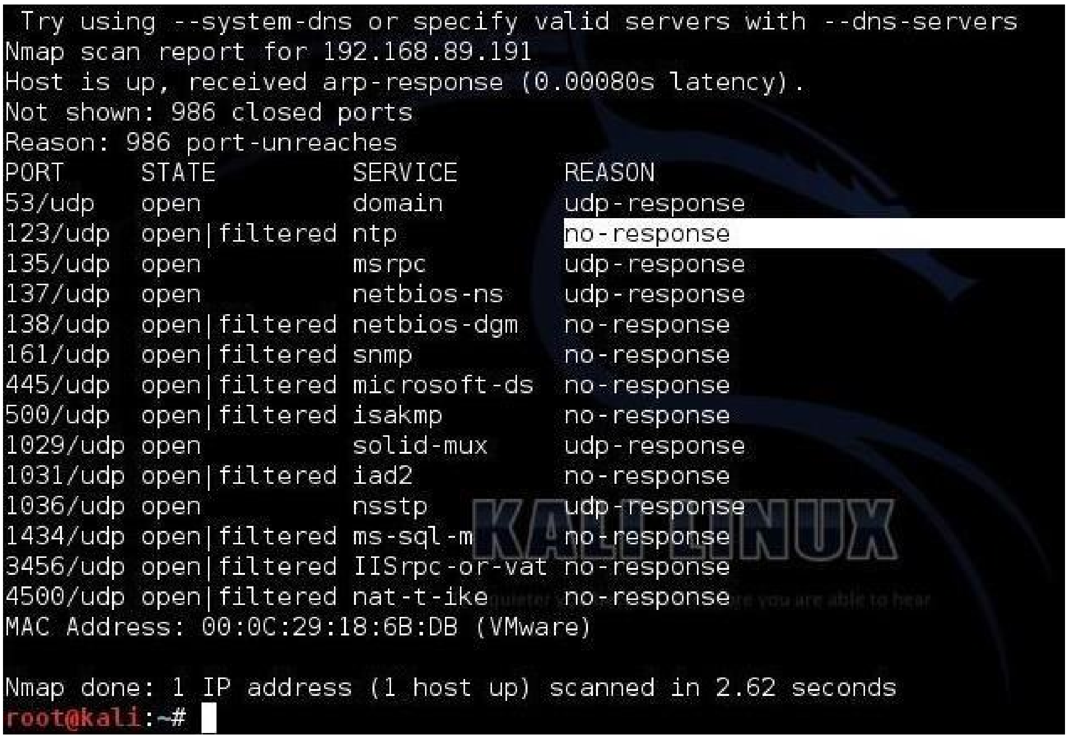
UDP Scan

nmap -sU <IP address>



```
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.89.191
Host is up (0.0012s latency).
Not shown: 986 closed ports
PORT     STATE         SERVICE
53/udp   open          domain
123/udp  open|filtered ntp
135/udp  open          msrpc
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
161/udp  open|filtered snmp
445/udp  open|filtered microsoft-ds
500/udp  open|filtered isakmp
1029/udp open          solid-mux
1031/udp open|filtered iad2
1036/udp open          nsstp
1434/udp open|filtered ms-sql-m
3456/udp open|filtered IISrpc-or-vat
4500/udp open|filtered nat-t-ike
MAC Address: 00:0C:29:18:6B:DB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.43 seconds
root@kali:~#
```

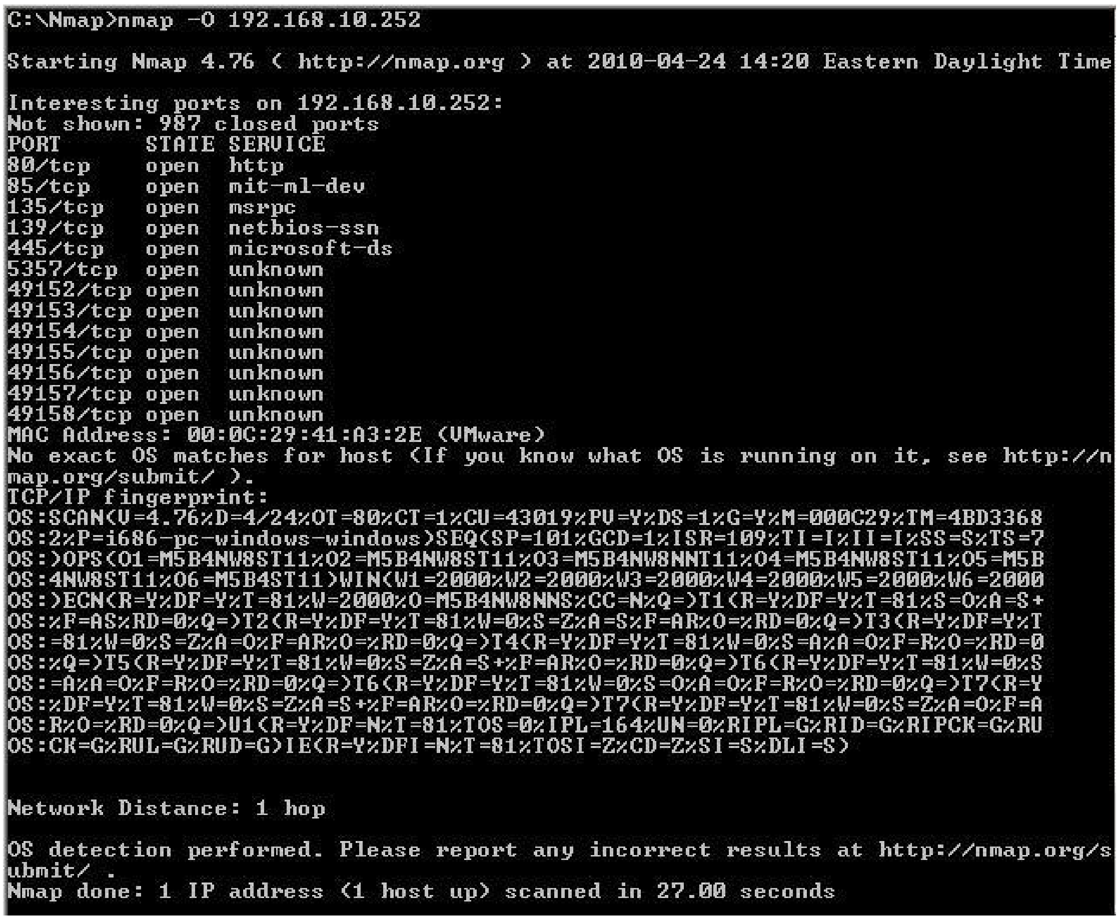
#nmap -sU --reason <IP address>



```
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.89.191
Host is up, received arp-response (0.00080s latency).
Not shown: 986 closed ports
Reason: 986 port-unreaches
PORT     STATE         SERVICE       REASON
53/udp   open          domain        udp-response
123/udp  open|filtered ntp           no-response
135/udp  open          msrpc         udp-response
137/udp  open          netbios-ns    udp-response
138/udp  open|filtered netbios-dgm   no-response
161/udp  open|filtered snmp          no-response
445/udp  open|filtered microsoft-ds  no-response
500/udp  open|filtered isakmp        no-response
1029/udp open          solid-mux     udp-response
1031/udp open|filtered iad2          no-response
1036/udp open          nsstp         udp-response
1434/udp open|filtered ms-sql-m      no-response
3456/udp open|filtered IISrpc-or-vat no-response
4500/udp open|filtered nat-t-ike     no-response
MAC Address: 00:0C:29:18:6B:DB (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.62 seconds
root@kali:~#
```

OS detection(-O)

OS detection with verbosity (-O -v)



```
C:\Nmap>nmap -O 192.168.10.252

Starting Nmap 4.76 ( http://nmap.org ) at 2010-04-24 14:20 Eastern Daylight Time

Interesting ports on 192.168.10.252:
Not shown: 987 closed ports
PORT      STATE SERVICE
80/tcp    open  http
85/tcp    open  mit-ml-dev
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  unknown
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
MAC Address: 00:0C:29:41:A3:2E (VMware)
No exact OS matches for host (If you know what OS is running on it, see http://n
map.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=4.76%D=4/24%OT=80%CT=1%CU=43019%PV=Y%DS=1%G=Y%M=000C29%TM=4BD3368
OS:2%P=i686-pc-windows-windows)SEQ(SP=101%GCD=1%ISR=109%TI=I%II=I%SS=S%TS=7
OS:)OPS(O1=M5B4NW8ST11%O2=M5B4NW8ST11%O3=M5B4NW8NNT11%O4=M5B4NW8ST11%O5=M5B
OS:4NW8ST11%O6=M5B4ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=2000
OS:)ECN(R=Y%DF=Y%T=81%W=2000%O=M5B4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=81%S=O%A=S+
OS:%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=81%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T
OS:=81%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=81%W=0%S=A%A=O%F=R%O=%RD=0
OS:%Q=)T5(R=Y%DF=Y%T=81%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=81%W=0%S
OS:=A%A=O%F=R%O=%RD=0%Q=)T6(R=Y%DF=Y%T=81%W=0%S=O%A=O%F=R%O=%RD=0%Q=)T7(R=Y
OS:%DF=Y%T=81%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T7(R=Y%DF=Y%T=81%W=0%S=Z%A=O%F=A
OS:R%O=%RD=0%Q=)U1(R=Y%DF=N%T=81%TOS=0%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RU
OS:CK=G%RUL=G%RUD=G)IE(R=Y%DFI=N%T=81%TOSI=Z%CD=Z%SI=S%DLI=S)


Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.00 seconds
```