Nmap

network scanning

Network Scanning

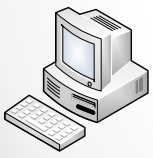
- Procedura per identificare gli host attivi su una rete:
  - Per valutare la sicurezza della rete
  - Per vedere tutte le porte aperte
  - Per attaccare un sistema
- La procedura di scansione restituisce informazioni su quali servizi questi host (indirizzi IP attivi nella rete)
  - ping sweeps
  - port scans

nmap

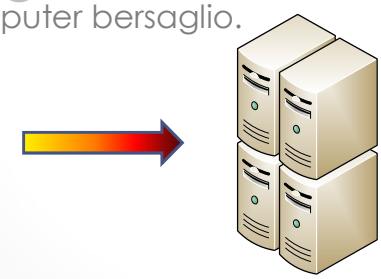
- È fra i più potenti e diffusi software open source per effettuare network scanning.
- Nmap (acronimo di "**Network Mapper**") è presente sulla rete ormai da anni
- Sono state rilasciate varie versioni, compatibili con molti sistemi operativi. http://nmap.org/download.html
- Nmap è uno degli strumenti praticamente indispensabili della "cassetta degli attrezzi" di un network administrator
- Usato per
  - analisi dei sistemi
  - test di penetrazione
  - analisi di sicurezza

nmap

- Creato per effettuare port scanning: individuare porte aperte e servizi disponibili su un computer target.
- Utilizza la tecnica del fingerprinting: è in grado di ipotizzare quale sistema operativo sia utilizzato dal computer bersaglio.



Host nmap



server targer

```
PORT     STATE    SERVICE
21/tcp   open     ftp
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
110/tcp  open     pop3
111/tcp  filtered rpcbind
112/tcp  filtered mcidas
137/tcp  filtered netbios-ns
138/tcp  filtered netbios-dgm
139/tcp  filtered netbios-ssn
161/tcp  filtered snmp
162/tcp  filtered snmptrap
343/tcp  filtered unknown
443/tcp  open     https
445/tcp  filtered microsoft-ds
587/tcp  open     submission
995/tcp  open     pop3s
1720/tcp filtered H.323/Q.931
2049/tcp filtered nfs
3306/tcp filtered mysql
6881/tcp filtered bittorent-tracker
```

nmap

- Utilizza pacchetti IP per ottenere informazioni :
  - gli host presenti/attivi su una rete
  - i servizi che tali host rendono disponibili
  - i sistemi operativi presenti sull'host target
  - la presenza di firewall – ids/ips
  - monitor/diagnostica di host
  - troubleshooting ("eliminazione del problema")
  - altro (virus, …)

nmap: lo stato delle porte

- Le porte rilevate possono essere:
  - **Open** – Una applicazione accetta attivamente connessioni TCP o UDP su questa porta.
  - **Closed** – Una porta chiusa è accessibile, ma non vi è alcuna applicazione in ascolto su di essa.
  - **Filtered** – Nmap non può determinare con esattezza se la porta sia aperta o meno in quanto un filtro ne impedisce l'accesso.
  - **Unfiltered** – la porta è accessibile, ma Nmap non può determinarne lo stato.
  - **Open|filtered** – Nmap non è in grado di determinare se una porta è aperta oppure filtrata.
  - **Closed|filtered** – Nmap non è in grado di determinare se una porta è chiusa oppure filtrata.

nmap: Target Specification

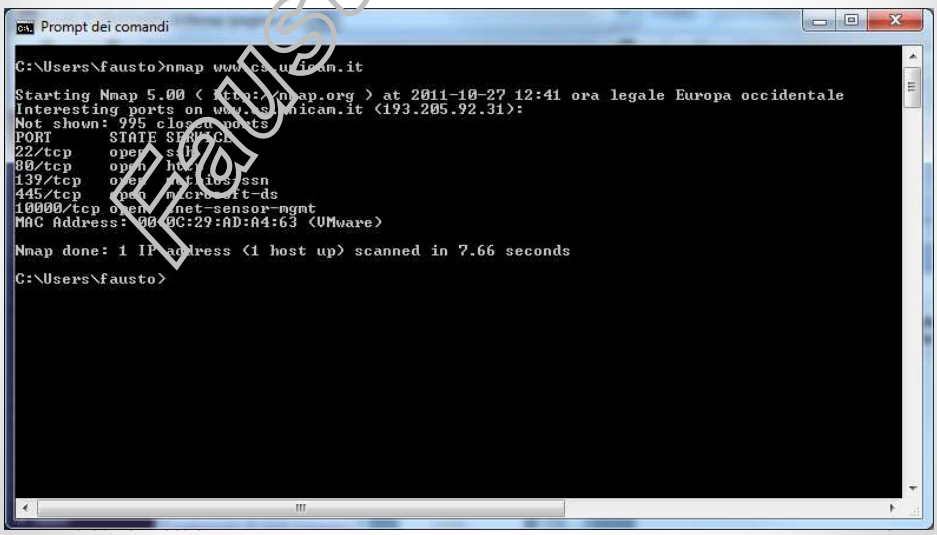
un indirizzo IP
un nome di host per la scansione
un gruppo di indirizzi IP
un'intera rete di host adiacenti notazione CIDR (192.168.3.0/24)

```
nmap 192.168.3.12

nmap www.cs.unicam.it

nmap 192.168.3.23-37

nmap 192.168.3.0/24
```

Indirizzi IPV6

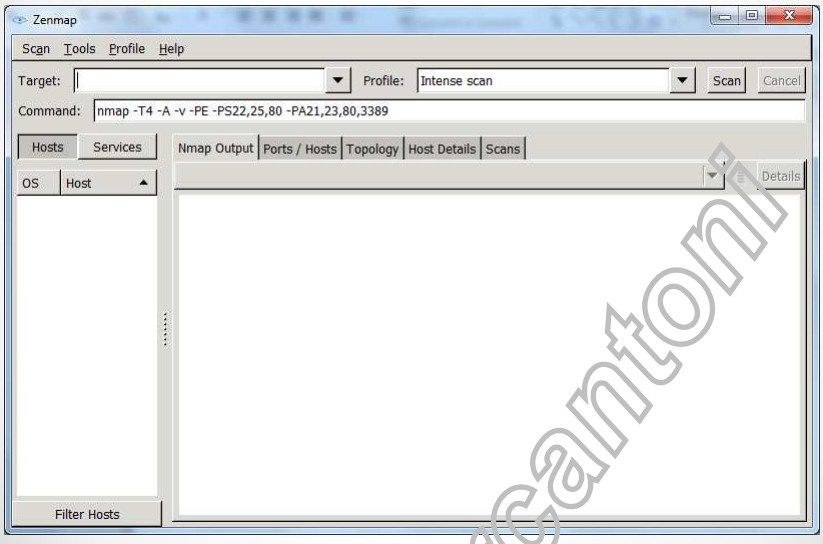
nmap [host]

```
nmap www.cs.unicam.it
```



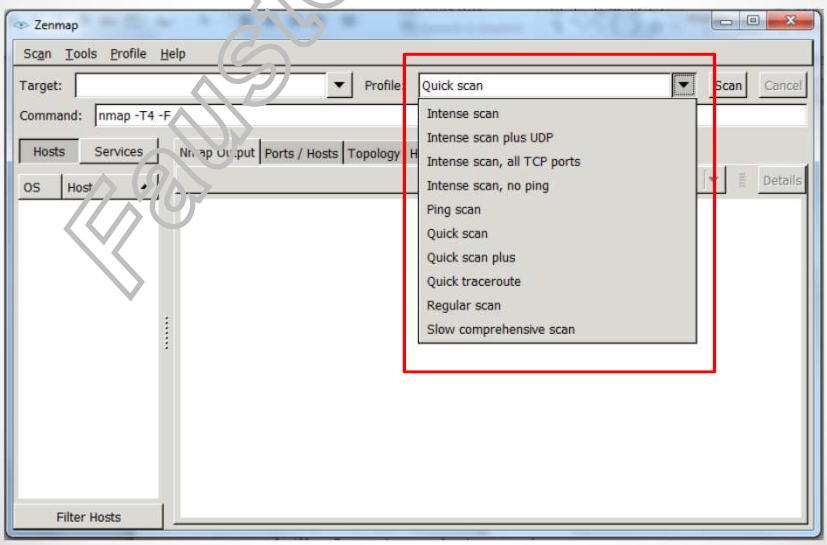
nmap: zenmap

Zenmap: Interfaccia GUI per nmap



nmap: zenmap

Zenmap: profili già definiti



Profile Zenmap

Intense scan
command = nmap -T4 -A -v
An intense, comprehensive scan. The -A option enables OS detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute). Without root privileges only version detection and script scanning are run. This is considered an intrusive scan.

Intense scan plus UDP
command = nmap -sS -sU -T4 -A -v
Does OS detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute) in addition to scanning TCP and UDP ports.

Intense scan, all TCP ports
command = nmap -p 1-65535 -T4 -A -v
Scans all TCP ports, then does OS detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute).

Intense scan, no ping
command = nmap -T4 -A -v -Pn
Does an intense scan without checking to see if targets are up first. This can be useful when a target seems to ignore the usual host discovery probes.

Ping scan
command = nmap -sn
This scan only finds which targets are up and does not port scan them.

Quick scan
command = nmap -T4 -F
This scan is faster than a normal scan because it uses the aggressive timing template and scans fewer ports.

Quick scan plus
command = nmap -sV -T4 -O -F --version-light
A quick scan plus OS and version detection.

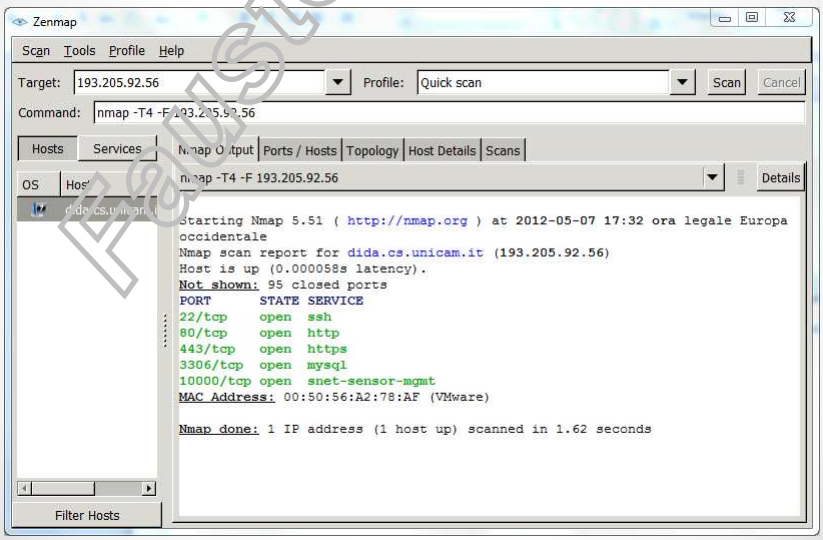
Quick traceroute
command = nmap -sn --traceroute
Traces the paths to targets without doing a full port scan on them.

Regular scan
command = nmap
A basic port scan with no extra options.

Slow comprehensive scan
command = nmap -sS -sU -T4 -A -v -PE -PS80,443 -PA3389 -PP -PU40125 -PY --source-port 53 --script all
This is a comprehensive, slow scan. Every TCP and UDP port is scanned. OS detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute) are all enabled. Many probes are sent for host discovery. This is a highly intrusive scan.

nmap: Regular Scan

Regular Scan



Elenco delle opzioni

http://www.insecure.org/nmap/data/nmap.usage.txt

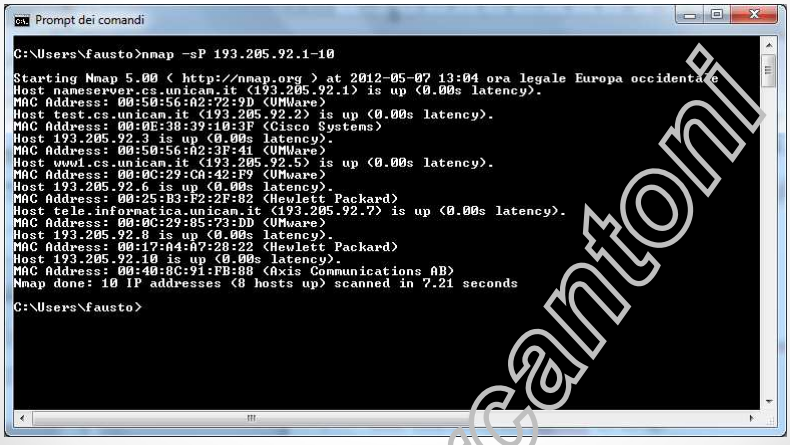
```
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0-255.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random 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
  -sP: Ping Scan - go no further than determining if host is online
  -P0: Treat all hosts as online -- skip host discovery
  -PS/PA/PU [portlist]: TCP SYN/ACK or UDP discovery probes to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes resolve]
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idlescan
  -sO: IP protocol scan
  -b <ftp relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
  -F: Fast - Scan only the ports listed in the nmap-services file)
  -r: Scan ports consecutively - don't randomize
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version_light: Limit to most likely probes for faster identification
  --version_all: Try every single probe for version detection
  --version_trace: Show detailed version scan activity (for debugging)
OS DETECTION:
  -O: Enable OS detection
```


Ping sweep

- Vengono inviati dei pacchetti **ICMP** (**Internet Control Message Protocol**), in particolare dei pacchetti Tipo 8 (**echo request**), conosciuti anche come pacchetti ping che identificano se un host è attivo.
- Ci sono anche altri pacchetti ICMP che possono risultare utili, ad esempio:
  - **Tipo 13** (**timestamp request**) – viene richiesto il tempo di sistema di un host target
  - **Tipo 15** (**information request**) – è un particolare messaggio creato per supportare alcuni sistemi auto-configuranti (ad esempio diskless workstation) che nel boot devono poter trovare il proprio indirizzo nella rete.
  - **Tipo 17** (**subnet address mask request**) – con questo messoggio riusciamo a determinare la subnet mask usata dall'host.

Ping Sweep - Nmap

- Nmap può effettuare ping sweep scans, utilizzando l'opzione –sP
- esempio:
- `nmap -sP 193.205.92.0/24 oppure 193.205.92.1-10`



Port Scanning

- Il Port Scanning è un processo di connessione a porte TCP e UDP del sistema target al fine di determinare **quali porte siano aperte** e quali servizi siano in stato di esecuzione o in stato di listening su quelle porte
- Un amministratore di sistema deve poter **identificare i servizi offerti dai propri server** ed identificare eventuali debolezze della propria rete e dei propri sistemi, col fine di evitarne lo sfruttamento da parte di utenti indesiderati.

TCP port scanning

- Esistono diversi tipi di port scanning per controllare se le porte TCP di un sistema sono aperte o chiuse, questi sono classificabili in tre classi:
  - **Standard scanning methods**
    - Vanilla TCP connect() scanning
    - Half-open SYN flag scanning
  - **Stealth TCP scanning methods**
    - Inverse TCP flag scanning
    - ACK flag probe scanning
    - TCP fragmentation scanning
  - **Third-party and spoofed TCP scanning methods**
    - FTP bounce scanning
    - Proxy bounce scanning
    - Sniffer-Based spoofed scanning
    - IP ID header scanning

Standard scanning methods

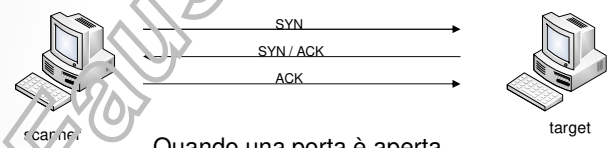
- I metodi definiti nella categoria Standard scanning methods sono delle tecniche **molto semplici** e dirette, usate per identificare le porte e i servizi TCP accessibili in modo accurato.
- Questi sono metodi certi, ma possono essere **facilmente rilevati e loggati**.

Vanilla TCP connect() method

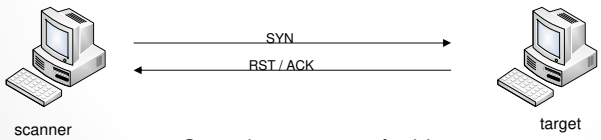
- È il metodo più semplice, conosciuto anche come **TCP connect()** o **Vanilla connect()**
- Viene stabilita **un'intera connessione TCP/IP** con una porta TCP dell'host target
- Data l'affidabilità del protocollo TCP/IP, questo metodo è **molto accurato** per determinare quali servizi sono attivi in un dato host

Vanilla TCP connect() method

pacchetti inviati da questo metodo (**three way handshake**)



Quando una porta è aperta

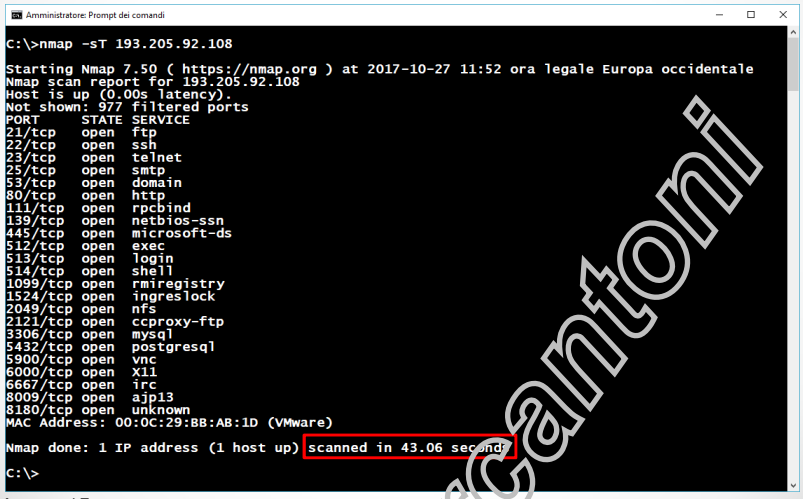


Quando una porta è chiusa

Vanilla TCP connect() method - Nmap

Nmap può effettuare un TCP connect() portscan, utilizzando l'opzione –sT

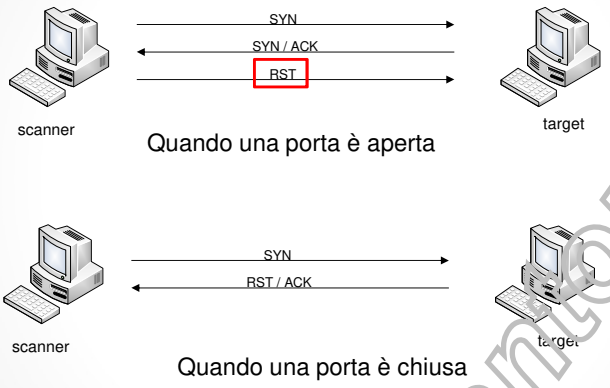
nmap -sT 193.205.92.108



Half-open SYN flag scanning

- In modo simile al metodo TCP connect(), si **simula un inizio di connessione** come se fosse un three way handshake per sincronizzazione i due host
- Nel caso dell'**half open SYN port scanning**, quando una porta viene rilevata aperta, viene inviato un pacchetto con **flag RST** che resetta la connessione TCP
- Questo fa si che non ci sia una completa connessione fra i due host, e quindi sia più **difficile rilevare e loggare lo scanning**

Half-open SYN flag scanning



Half-open SYN flag scanning - Nmap

Nmap può effettuare un SYN portscan utilizzando l'opzione –sS

```
nmap -sS 193.205.92.108
```

```
Amministratore: Prompt dei comandi
C:\>nmap -sS 193.205.92.108

Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-27 11:59 ora legale Europa occidentale
Nmap scan report for 193.205.92.108
Host is up (0.00024s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:BB:AB:1D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.33 seconds

C:\>
```

Stealth TCP scanning methods

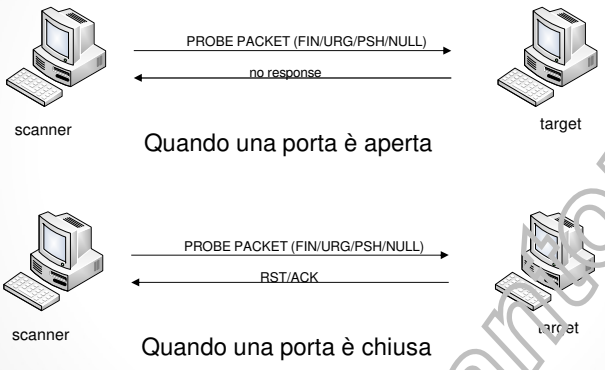
- Questi metodi sfruttano **alcune lacune del protocollo TCP/IP**, provocate inviando pacchetti con configurazioni di bit non standard.
- Queste tecniche non mappano le porte aperte in modo accurato, però **lavorano in modo nascosto** e **difficilmente possono essere rilevati e loggati**.

Inverse TCP flag scanning

- Questo tipo di scanning viene chiamato **inverse** in quanto rispondono **solo le porte chiuse** di un sistema
- Vengono inviati dei pacchetti "**sonda**" con diversi flag settati, ad esempio:
  - FIN probe, con il FIN TCP flag settato
  - L'XMAS probe con i FIN,URG,PUSH TCP flag settati
  - NULL probe con nessun flag TCP settato

Inverse TCP flag scanning

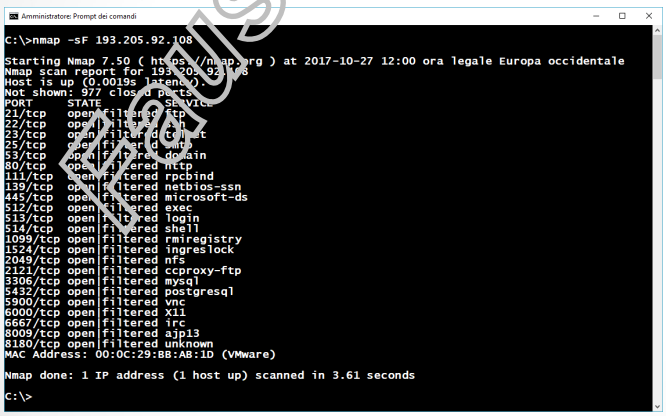
Ecco come avvengono le comunicazioni utilizzando questo metodo:



Inverse TCP flag scanning - Nmap

- Nmap può effettuare un TCP flag portscan, utilizzando le opzioni:
  -sF per il FIN probe
  -sX per l'XMAS Tree probe
  -sN per il NULL probe

```
nmap -sF 193.205.92.108
```



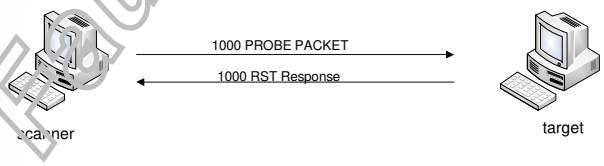
Come vediamo in questo caso **Nmap riesce anche a dirci se la porta in questione è open|filtered** in quanto il sistema, se una porta è aperta non risponde, ma lo stesso avviene se una porta è filtered e quindi inaccessibile

ACK flag probe scanning

- Un altro metodo consiste nell'inviare dei pacchetti ACK ed analizzare il pacchetto RST ottenuto in risposta.
- Ci sono due parametri principali da osservare:
  - il time-to-live (ttl)
  - il parametro WINDOW (win)
- Questo sistema è molto difficile da rilevare, ma per funzionare sfrutta un bug nell'implementazione dello standard TCP/IP che non è più presente nei sistemi più aggiornati

ACK flag probe scanning

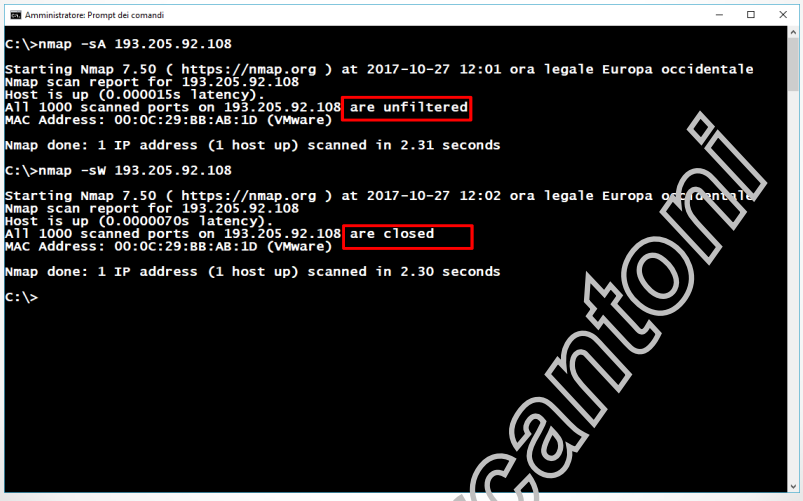
Vengono spediti moltissimi pacchetti ACK verso porte TCP diverse



Questo scan fa affidamento a un dettaglio implementativo di una minoranza di sistemi presenti in internet, segue che questo non è sempre affidabile

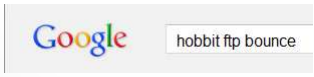
ACK flag probe scanning

Nmap può effettuare un ACK flag probe scanning, usando le opzioni:
-sA per analizzare il parametro ttl
-sW per analizzare il parametro win



FTP bounce attack

- Una "caratteristica" interessante del protocollo ftp (RFC 959) è il supporto per le connessioni ftp "proxy". In altre parole, io dovrei essere in grado di connettermi da mioftp.com al server FTP di target.com e richiedere che tale server mandi un file OVUNQUE su internet!
- Ora questo poteva andare bene nel 1985 quando la RFC fu scritta.
- Come Hobbit scrisse nel 1995, questo punto debole nel protocollo "può essere usato per postare mail e news virtualmente irrintracciabili, riempire i dischi, provare a scavalcare i firewall, e generalmente è fastidioso e difficile da rintracciare allo stesso tempo."

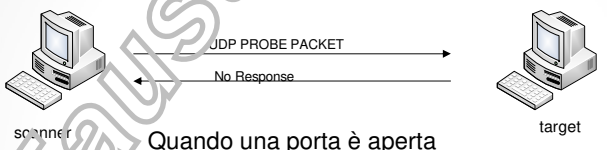


https://nmap.org/hobbit.ftpbounce.txt

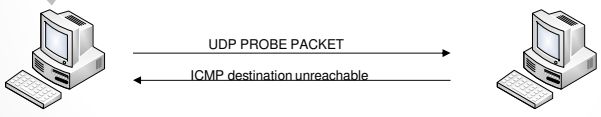
UDP port scanning

- Il protocollo **UDP** (**user datagram protocol**), consente di stabilire quali porte possono essere aperte in un host, in soli due modi:
  - Inviando **pacchetti UDP probe** a tutte le porte UDP di un host (65535), e aspettando i messaggi "ICMP destination port unreachable"
  - Oppure **usando degli specifici client per servizi UDP** e controllando se ci sono delle risposte da degli host (che in caso positivo possiedono il servizio)
- La maggior parte dei firewall però consente di **filtrare i messaggi ICMP**, da e per gli host protetti, questo rende difficile capire quali servizi UDP sono accessibili attraverso un semplice UDP port scanning

UDP port scanning



Quando una porta è aperta

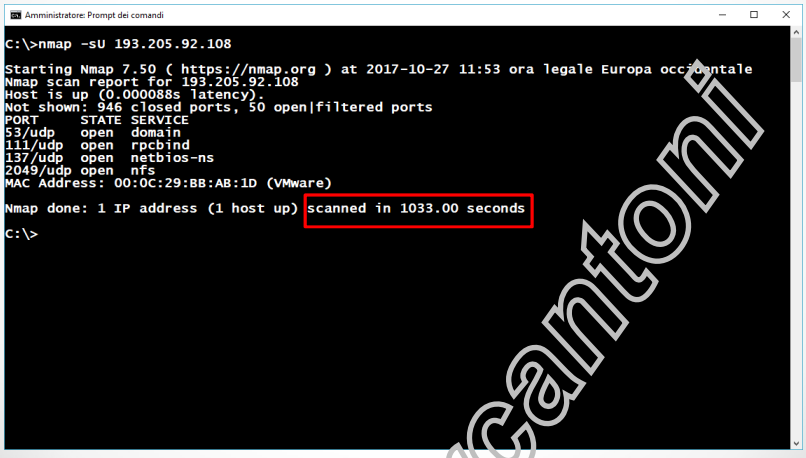


Quando una porta è chiusa

UDP port scanning - Nmap -

- Nmap può effettuare l'UDP port scanning, utilizzando l'opzione -sU

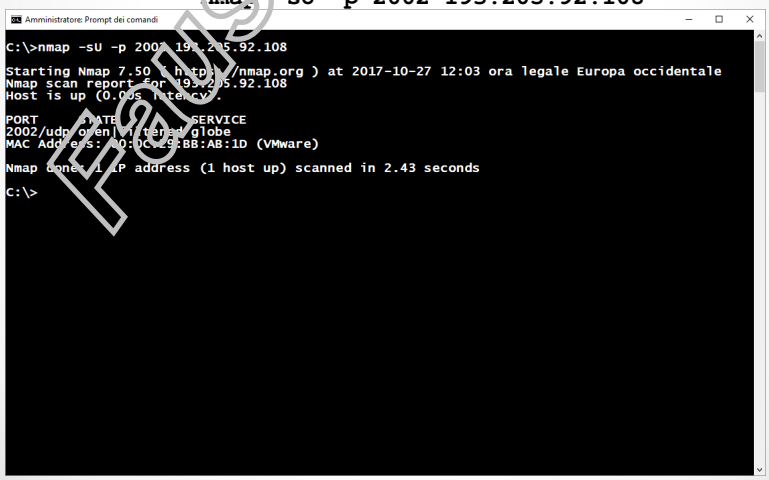
```
nmap -sU 193.205.92.108
```



UDP port scanning - Nmap -

- Nmap può effettuare l'UDP port scanning, con una porta specifica o con un gruppo di porte utilizzando l'opzione -sU

```
nmap -sU -p 2002 193.205.92.108
```



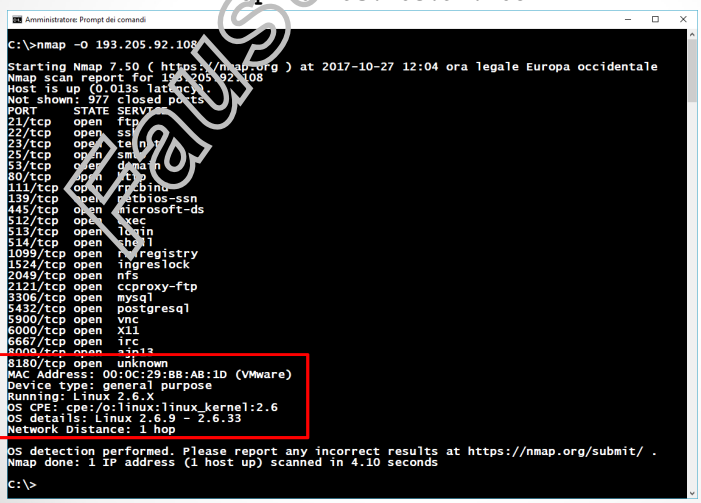
Fingerprinting

- Ogni sistema operativo possiede una propria interpretazione degli standard del protocollo IP
- Analizzando le risposte ottenute, inviando alcuni pacchetti TCP ed UDP ad un host remoto, si può determinare il sistema operativo presente
- Nmap ad esempio compara i risultati ottenuti con il suo database, contenente oltre 1500 SO, e ne visualizza i dettagli se trova riscontri

Fingerprinting - Nmap -

- Nmap effettua il fingerprinting utilizzando l'opzione –O

nmap -O 193.205.92.108



Service e Version Detection

- Si possono sfruttare i messaggi ricevuti in risposta da porte aperte in un host target, oltre che per la determinazione del SO, anche per altre funzionalità avanzate
- Nmap, ad esempio, è in grado di analizzare i messaggi ottenuti da una porta di un host e confrontarli con un suo database interno con oltre 2200 servizi

Service e Version Detection

- Questa funzionalità diventa di fondamentale importanza per un amministratore di sistema.
- Infatti conoscendo queste informazioni, e confrontandole con opportuni database, si può determinare a quali bugs è vulnerabile il server.
- Purtroppo lo stesso procedimento può essere effettuato da utenti malintenzionati che possono sfruttare tali bugs per controllare servizi e dati dell'host target.

Service e Version Detection

- nmap può utilizzare l'opzioni -sV per "Version Detection"

nmap -sV 193.205.92.108

Service e Version Detection

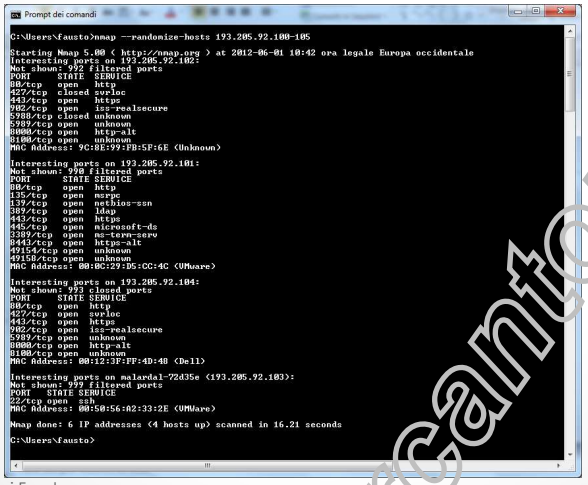
- nmap può utilizzare le opzioni -A per abilitare contemporaneamente sia "OS Detection" che "Version Detection"

nmap -A 193.205.92.108

nmap – Random Order

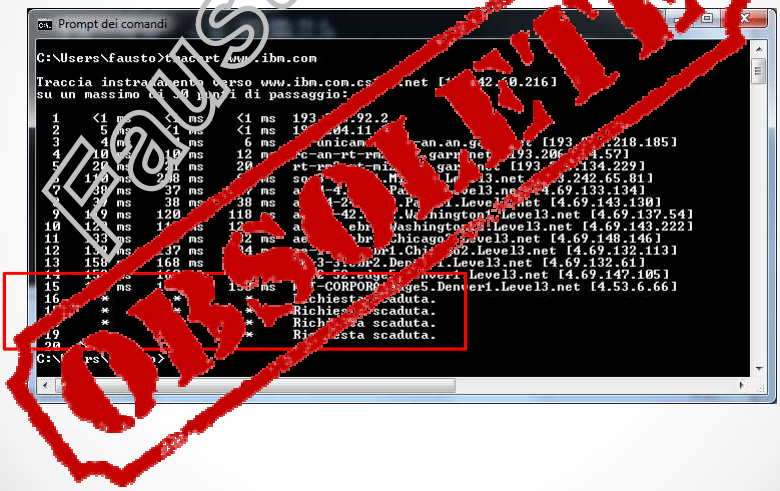
nmap può randomizzare lo scan degli host

```
nmap --randomize-hosts 193.205.92.0/24
nmap --randomize-hosts 193.205.92.100-200
```



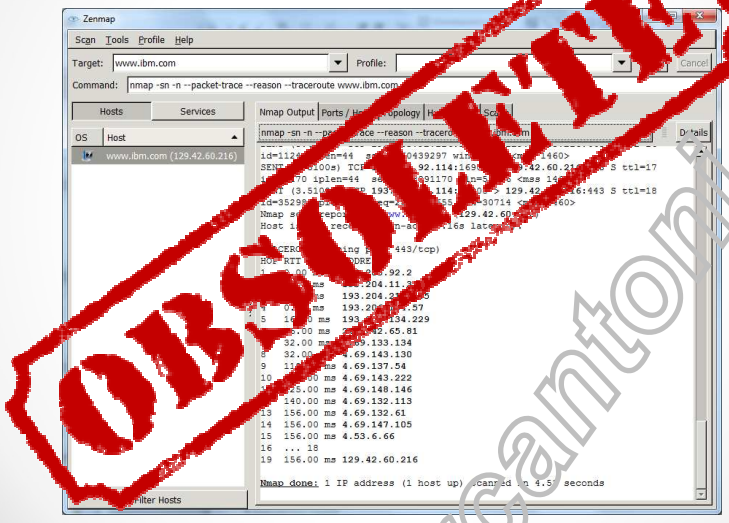
nmap - traceroute

```
tracert www.ibm.com
```



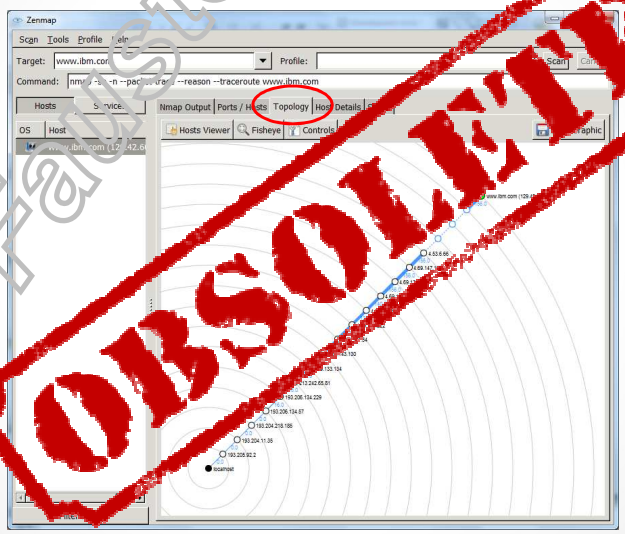
nmap - traceroute

```
nmap -sP --traceroute --reason -n --packet-trace www.ibm.com
```



nmap - traceroute

```
nmap -sP --traceroute --reason -n --packet-trace www.ibm.com
```



nmap - traceroute

nmap -sP --traceroute --reason -n --packet-trace www.ibm.com

```
root@pentest:~# nmap -sP --traceroute --reason -n --packet-trace www.ibm.com
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-19 09:40 CET
SENT (0.2003s) ICMP [193.205.92.99 > 23.1.74.26 Echo request (type=8/code=0) id=54920 seq=0] IP [ttl=59 id=54451 iplen=28 ]
SENT (0.2004s) TCP 193.205.92.99:53261 > 23.1.74.26:443 S ttl=38 id=5931 iplen=44  seq=4150009185 win=1024 <mss 1460>
SENT (0.2005s) TCP 193.205.92.99:53261 > 23.1.74.26:80 A ttl=51 id=9013 iplen=40  seq=0 win=1024
SENT (0.2006s) ICMP [193.205.92.99 > 23.1.74.26 Timestamp request (type=13/code=0) id=49965 seq=0 orig=0 recv=0 trans=0] IP [ttl=47 id=38902 iplen=40 ]
RCVD (0.2009s) TCP 23.1.74.26:80 > 193.205.92.99:53261 RA ttl=254 id=54826 iplen=40  seq=4150009185 win=1024
SENT (0.2883s) TCP 193.205.92.99:61584 > 23.1.74.26:80 A ttl=10 id=38809 iplen=40  seq=4081214763 win=21712
SENT (0.2884s) TCP 193.205.92.99:61585 > 23.1.74.26:80 A ttl=9 id=36249 iplen=40  seq=920528368 win=17104
SENT (0.2885s) TCP 193.205.92.99:61586 > 23.1.74.26:80 A ttl=8 id=17641 iplen=40  seq=669687590 win=29749
SENT (0.2886s) TCP 193.205.92.99:61587 > 23.1.74.26:80 A ttl=7 id=33161 iplen=40  seq=2641256326 win=58111
SENT (0.2887s) TCP 193.205.92.99:61588 > 23.1.74.26:80 A ttl=6 id=53994 iplen=40  seq=3121907236 win=64099
SENT (0.2888s) TCP 193.205.92.99:61589 > 23.1.74.26:80 A ttl=5 id=58910 iplen=40  seq=495142060 win=9420
SENT (0.2889s) TCP 193.205.92.99:61590 > 23.1.74.26:80 A ttl=4 id=51535 iplen=40  seq=1326898320 win=54301
SENT (0.2890s) TCP 193.205.92.99:61591 > 23.1.74.26:80 A ttl=3 id=35037 iplen=40  seq=136010506 win=64540
SENT (0.2891s) TCP 193.205.92.99:61592 > 23.1.74.26:80 A ttl=2 id=58767 iplen=40  seq=178037081 win=57859
SENT (0.2891s) TCP 193.205.92.99:61593 > 23.1.74.26:80 A ttl=1 id=24133 iplen=40  seq=2706992586 win=6059
RCVD (0.2886s) TCP 23.1.74.26:80 > 193.205.92.99:61584 RA ttl=254 id=41485 iplen=40  seq=885924960 win=21712
RCVD (0.2887s) TCP 23.1.74.26:80 > 193.205.92.99:61585 RA ttl=254 id=49398 iplen=40  seq=3445877334 win=17104
RCVD (0.2889s) TCP 23.1.74.26:80 > 193.205.92.99:61587 RA ttl=254 id=33619 iplen=40  seq=616124656 win=58111
RCVD (0.2889s) TCP 23.1.74.26:80 > 193.205.92.99:61586 RA ttl=254 id=52028 iplen=40  seq=71692901 win=29749
RCVD (0.2890s) TCP 23.1.74.26:80 > 193.205.92.99:61588 RA ttl=254 id=35730 iplen=40  seq=933830085 win=64099
RCVD (0.2891s) TCP 23.1.74.26:80 > 193.205.92.99:61589 RA ttl=254 id=57684 iplen=40  seq=3897875125 win=942[illegible]
RCVD (0.2891s) TCP 23.1.74.26:80 > 193.205.92.99:61590 RA ttl=254 id=62691 iplen=40  seq=3597012536 win=543[illegible]
RCVD (0.2892s) TCP 23.1.74.26:80 > 193.205.92.99:61591 RA ttl=254 id=35649 iplen=40  seq=3315646858 win=645[illegible]
RCVD (0.2895s) TCP 23.1.74.26:80 > 193.205.92.99:61592 RA ttl=254 id=58525 iplen=40  seq=3939971650 win=578[illegible]
RCVD (0.2895s) ICMP [193.205.92.2 > 193.205.92.99 TTL=0 during transit (type=11/code=0) ] IP [ttl=255 id=23191 iplen=[illegible]]
Nmap scan report for www.ibm.com (23.1.74.26)
Host is up, received reset ttl 254 (0.00040s latency).
Other addresses for www.ibm.com (not scanned): 2a02:26f0:4:19a::b3a 2a02:26f0:4:183::b3a

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.40 ms 193.205.92.2
2   0.48 ms 23.1.74.26

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
root@pentest:~#
```

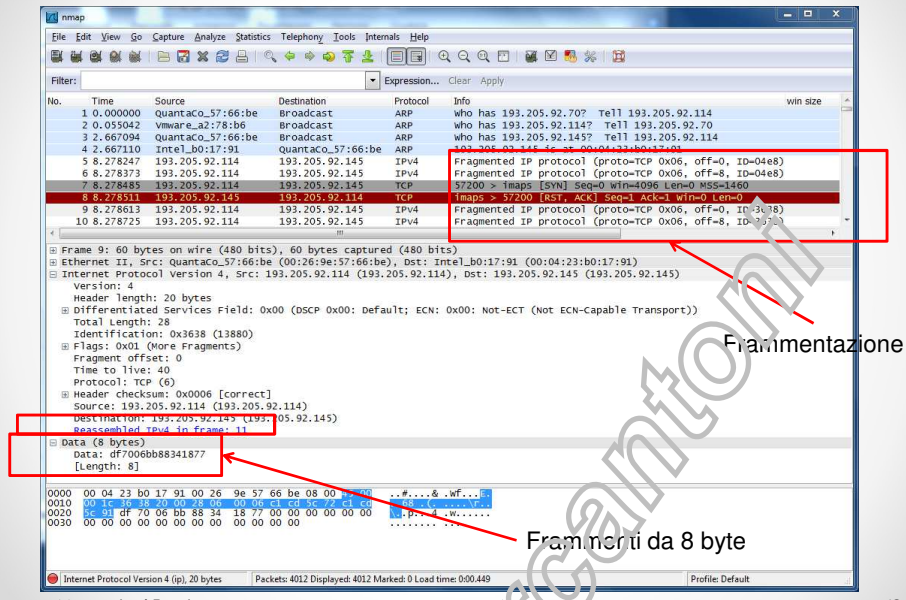
nmap - fragmentation

nmap può frammentare i pacchetti per evitare un firewall

nmap -f 193.205.92.108

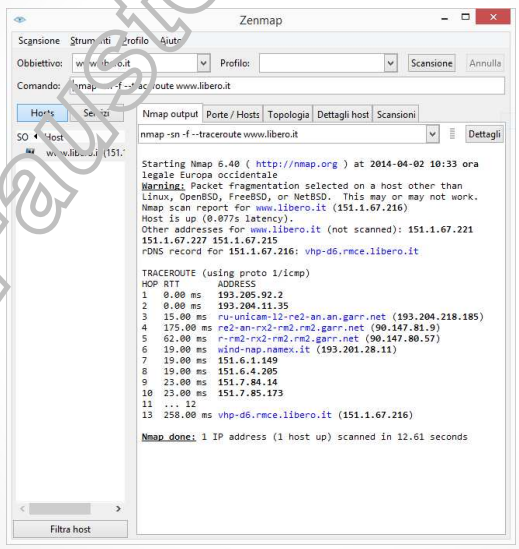
```
Amministratore: Prompt dei comandi

C:\>nmap -f 193.205.92.10[illegible]
Warning: Packet fragmentation selected on a host other than Linux, OpenBSD, FreeBSD, or NetBS
D.  This may or may not work.

Starting Nmap 7.50 ( https://nmap.org ) at 2017-10-27 12:17 ora legale Europa occidentale
Nmap scan report for 193.205.92.108
Host is up (0.00047s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:BB:AB:1D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.43 seconds

C:\>
```

nmap - fragmentation



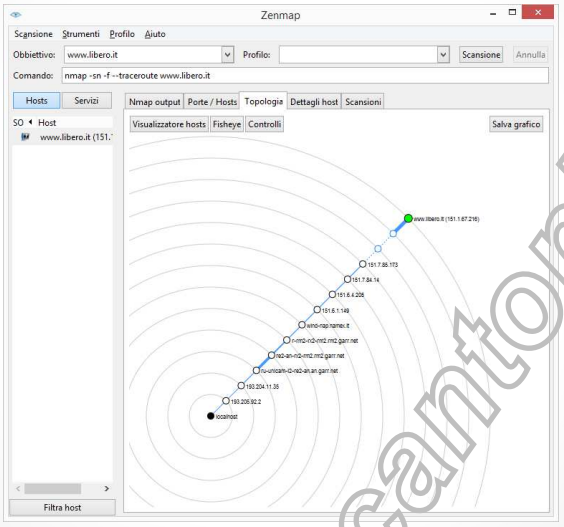
nmap – traceroute - "2" LA VENDETTA

nmap -sn -f --traceroute www.libero.it



nmap – traceroute - "2" LA VENDETTA

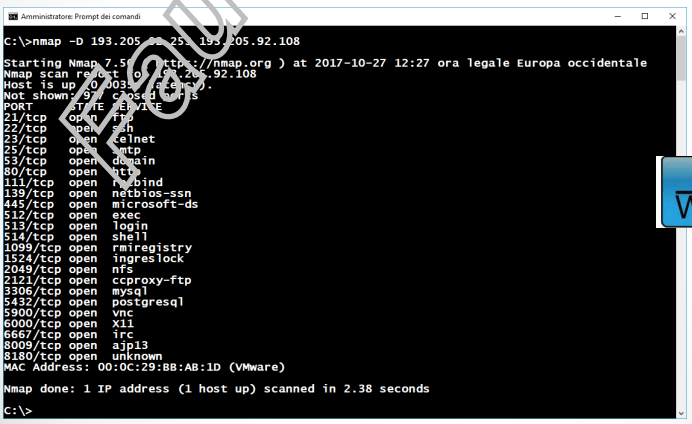
```
nmap -sn -f --traceroute www.libero.it
```



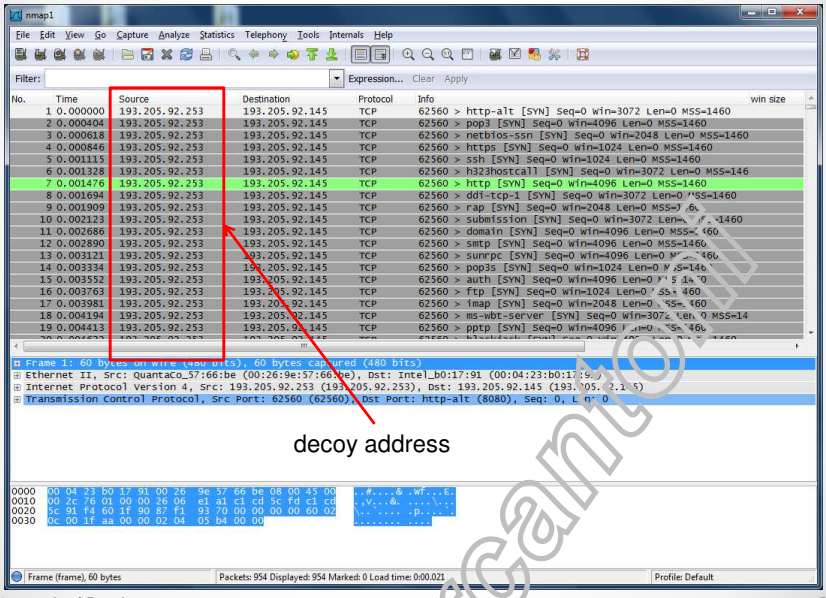
nmap – decoy address

nmap può inviare i pacchetti con un indirizzo «esca» per evitare un firewall

```
nmap -D 193.205.92.253 193.205.92.108
nmap -D [decoy] [target]
nmap -D RND:10 [target] (Generates a random number of decoys)
nmap -D decoy1,decoy2,decoy3 etc. (Manually specify the IP addresses of the decoys)
```



nmap – decoy address



nmap – decoy address

L'opzione -D è usata per mascherare uno scan di porte usando uno o più IP "esca", questo fa sembrare all'host remoto che gli hosts specificati come esche stiano facendo anche loro una scansione della rete di destinazione.

`nmap -D [decoy] [target]`

Sintassi: nmap -D [decoy1, decoy2, decoy3, etc| RND:Number] [IP destinazione]

nmap -D 192.168.1.33 192.168.1.31 192.168.1.32 **192.138.1.34**

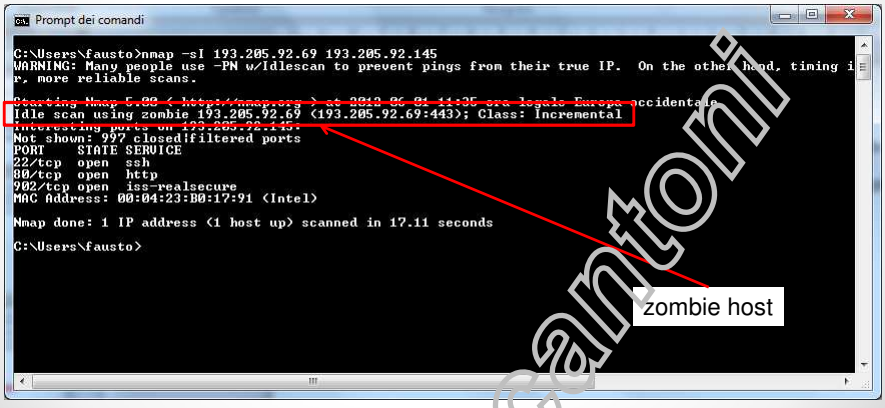
In questo esempio l'indirizzo ip in grassetto è l'indirizzo IP del bersaglio remoto

Inoltre, durante l'esecuzione di una scansione con esca, Nmap manderà ulteriori pacchetti dal numero specificato di indirizzo esca.
Questo fa sembrare effettivamente che l'obiettivo sia sottoposto a scansione da più sistemi contemporaneamente.
Utilizzando le esche l'ip sorgente si "***mescola nella folla***" cosa che lo rende più difficile da rintracciare.

nmap – **Idle Zombie Scan**

Questa tecnica consente di utilizzare un altro host della rete che è inattivo. Il principale vantaggio di questo metodo è che i file di log del firewall registra l'indirizzo IP dello Zombie e non il nostro IP.
Tuttavia per avere un risultato ottimale **bisogna trovare padroni di casa che sono inattivi sulla rete**.

```
nmap -sI 193.205.92.69 193.205.92.145
nmap -sI [zombie host] [target]
```



nmap – Idle Zombie Scan

bisogna trovare padroni di casa che sono inattivi sulla rete

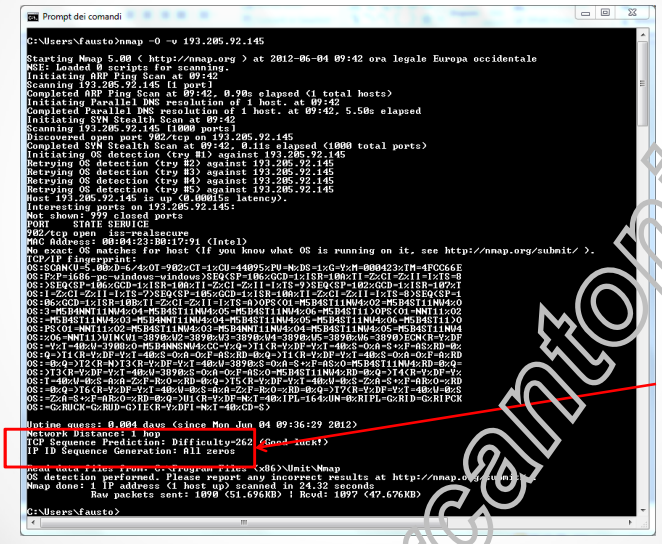
Internet si basa sull'unione di due protocolli fondamentali TCP e IP, quando host A si collega ad host B, invierà un pacchetto TCP contenente un flag SYN attivo all'interno del quale vi sarà presente un numero K compreso tra 0 e 2^32-1.
Tale numero viene chiamato Identification Sequence Number (da ora ISN).
Host B, ricevuto il pacchetto TCP da host A, risponderà a sua volta con un altro pacchetto contenente flag SYN e ACK con ISN J proprio e ISN K+1 (ovvero l'ISN di host A aumentato di 1).
Al ricevimento del pacchetto SYN+ACK, host A provvederà all'invio di un terzo pacchetto contenente esclusivamente flag ACK con i due numeri (K e J) aumentati di 1.
Dopo questi tre banali passaggi la connessione TCP/IP è stabilita; a questo punto vi domanderete cosa mai potrà aver a che fare il funzionamento del protocollo TCP/IP con l'ultima "apparizione" dell' argomento di nmap.
La risposta è semplice: Kevin Mitnick (famoso hacker americano) dimostrò che i numeri ISN potevano essere facilmente prevedibili per il fatto che venivano generati in sequenza fissa... vediamo come.
Prendiamo in considerazione host A (attaccante), host B (Zombie) ed host C (Target).
Prima di tutto inviamo un pacchetto SYN+ACK dall'host attaccante (A) all'host zombie (B): per quanto abbiamo detto prima ciò non dovrebbe essere consentito in quanto la comunicazione TCP viene inizializzata col il solo flag SYN, ciò nonostante host B risponde alla richiesta con un pacchetto contenente il flag RST che a sua volta include il numero, esemplificativo, ISN 1000.
Fatto questo siamo a conoscenza dell'ISN di B e sappiamo anche che lo stesso verrà incrementato linearmente: inviamo da A un pacchetto "alterato" a C contenente l'ip di B facendo ricadere qui la stessa responsabilità al medesimo: abbiamo in questa circostanza introdotto il concetto di spoofing.
Nel caso in cui la porta H di C è aperta questi inviarà un pacchetto a B (impersonificazione di A) contenente un pacchetto SYN+ACK ma, non avendo B legittimo richiesto alcunchè, questi chiuderà la connessione con un RST, incrementando logicamente l'ISN di 1: 1001.
Dopo aver fatto questo l'attaccante invierà un pacchetto SYN+ACK a B che, come in precendenza, chiuderà la comunicazione con un RST ed incrementerà l'ISN di 2: 1002 facendo denotare lo stato open della porta H.
Se invece la porta H di C è chiusa questi invierà un pacchetto a B (impersonificazione di A) con flag RST e, pertanto, l'ISN di B non cambierà.
Quando A invierà un nuovo pacchetto SYN+ACK a B ed in risposta troverà un RST, constaterà che l'ISN sarà aumentato solo di 1 facendo notare lo stato closed di H.
Nmap è ingrado di fare tutto ciò ed è proprio qui che entra in gioco l'opzione -I dello stesso (dove I sta, guarda caso, per idle scan).
Tramite il TCP Sequence Prediction, nmap segnala il "grado di difficoltà" al fine di "intrappolare" un host zombie.

http://openskill.info/infobox.php?ID=1274

nmap – Idle Zombie Scan

bisogna trovare padroni di casa che sono inattivi sulla rete

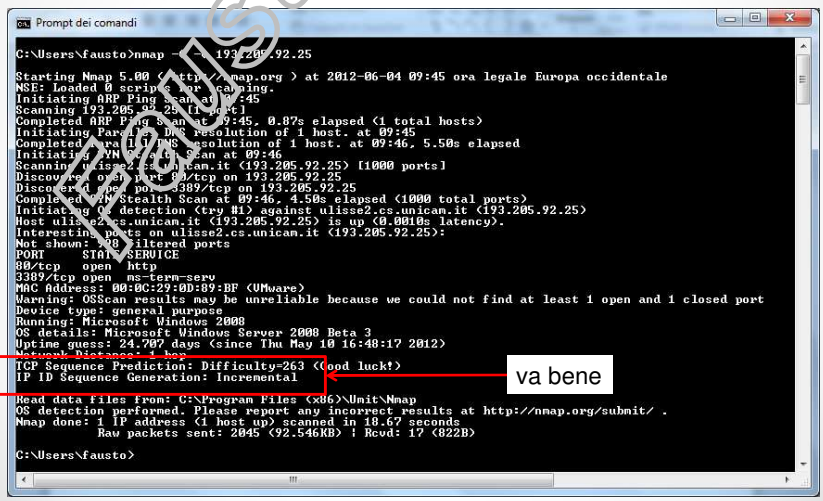
nmap –O –v 193.205.92.145



nmap – Idle Zombie Scan

bisogna trovare padroni di casa che sono inattivi sulla rete

nmap –O –v 193.205.92.25



nmap – Idle Zombie Scan

bisogna trovare padroni di casa che sono inattivi sulla rete

```
[*] Auxiliary module running as background job 1.
[*] 90.147.42.12's IPID sequence class: Randomized
[*] 90.147.42.2's IPID sequence class: Randomized
[*] 90.147.42.39's IPID sequence class: Unknown
[*] 90.147.42.30's IPID sequence class: All zeros
[*] 90.147.42.31's IPID sequence class: Unknown
[*] 90.147.42.36's IPID sequence class: All zeros
[*] 90.147.42.32's IPID sequence class: Incremental!
[*] 90.147.42.35's IPID sequence class: Unknown
[*] 90.147.42.41's IPID sequence class: All zeros
[*] 90.147.42.40's IPID sequence class: Incremental!
[*] 90.147.42.45's IPID sequence class: Unknown
[*] 90.147.42.43's IPID sequence class: Unknown
msf auxiliary(scanner/ip/ipidseq) >
```

nmap – MAC Address Spoofing

Un altro metodo per bypassare le restrizioni del firewall, è lo spoofing dell'indirizzo MAC del proprio host.

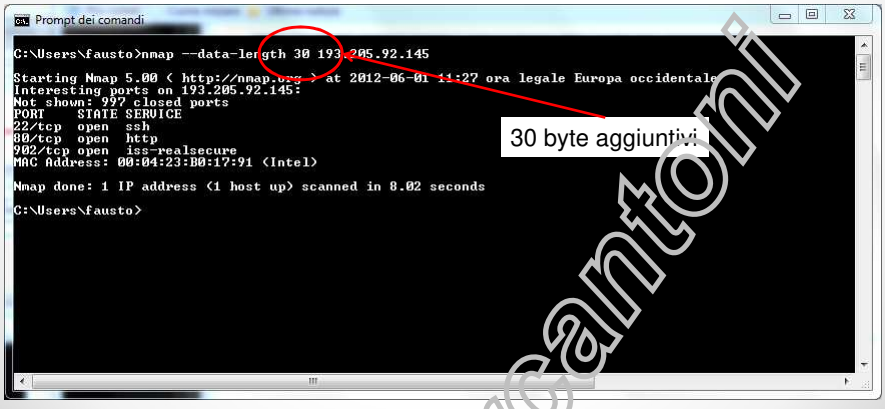
```
nmap --spoof-mac 00:01:02:25:56:AE 193.205.92.145

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
```

A ME (NON) FUNZIONA …
anzi funziona male …
e poi …

nmap – Append Random Data

Molti firewall ispezionano i pacchetti, cercando in loro dimensioni e contenuti, al fine di identificare un potenziale scan.
Per evitare questo tipo di rilevazione è possibile utilizzare il comando --data-lenght aggiungendo altri dati e di inviare pacchetti di dimensione diversa da quella default.

```
nmap --data-length 30 193.205.92.145
```

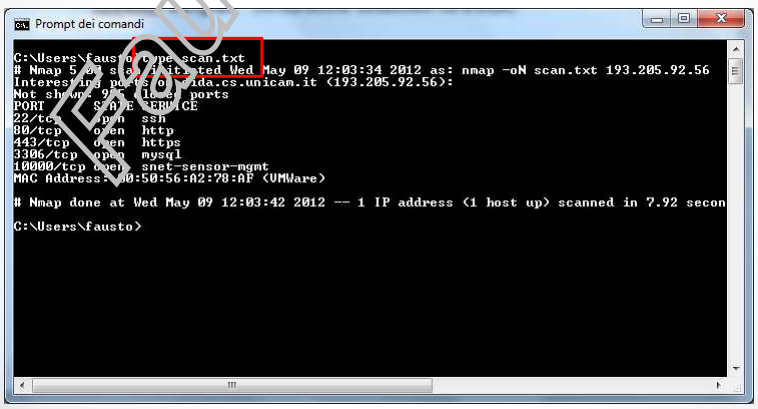


nmap - output

I Formati di Output di Nmap

```
nmap -oN scan.txt 193.205.92.108
```

```
--append-output
```
(Accoda anziché sovrascrivere i file di output)

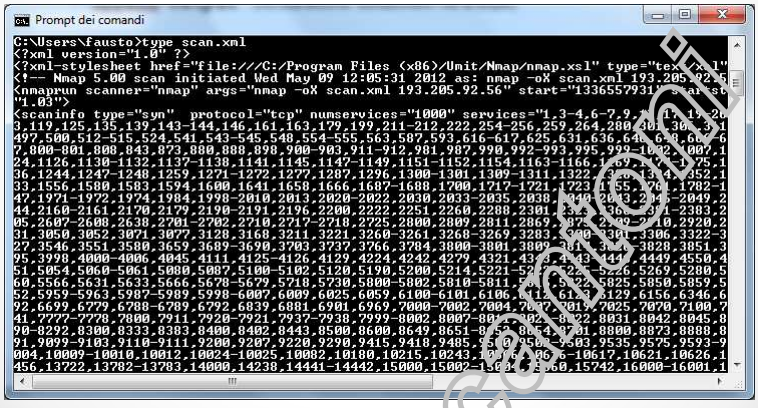


nmap - output

I Formati di Output di Nmap

`nmap -oX scan.xml 193.205.92.108`

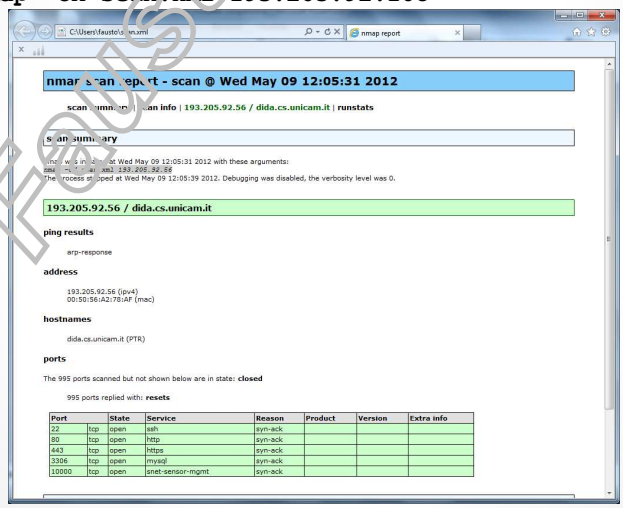
`--append-output` (Accoda anziché sovrascrivere i file di output)



nmap - output

I Formati di Output di Nmap

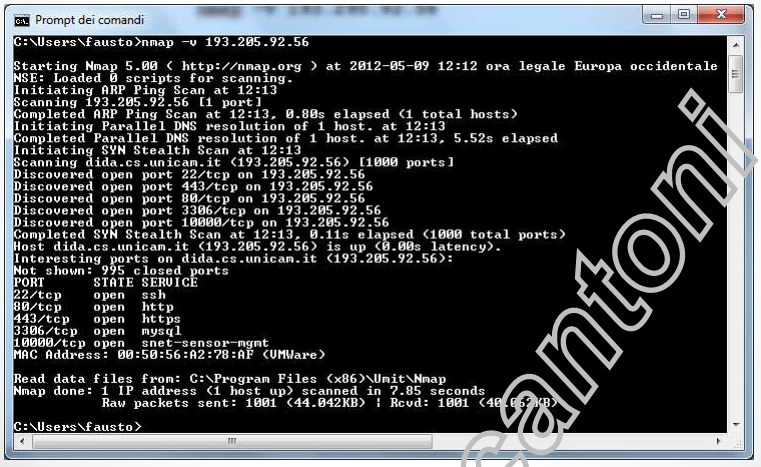
`nmap -oX scan.xml 193.205.92.108`



nmap - verbosità

Livello di verbosità

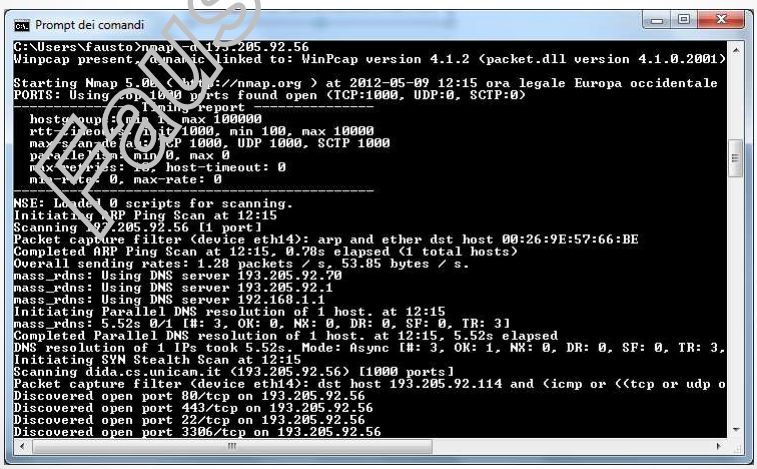
```
nmap -v 193.295.92.108
```



nmap - debug

Livello di debugging

```
nmap -d[0-9] 193.295.92.108
```



Nmap Scripting Engine

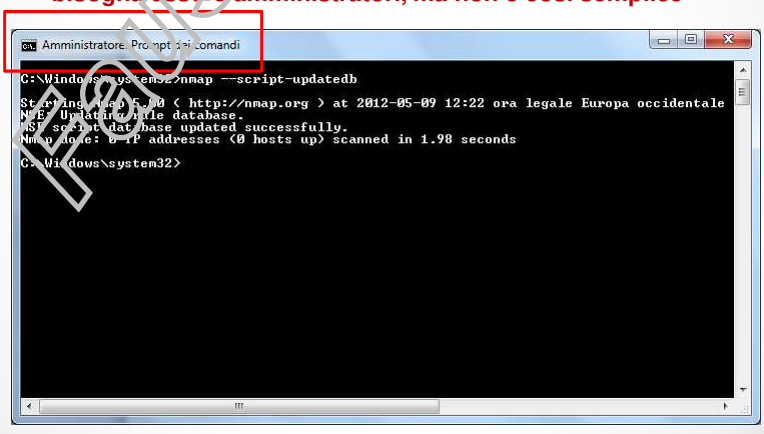
Con nmap vengono forniti anche una serie di script ufficiali elaborati all'interno del progetto.
Introdotto relativamente di recente, il **N**map **S**cript **E**ngine è un versatile motore di scripting per estendere le capacità di Nmap senza forzare gli sviluppatori ad approcciare un compito impegnativo quale la modifica dei sorgenti C del programma.
NSE è un motore per script in linguaggio **LUA** e permette agli utenti di automatizzare una serie di procedure e test verso gli host target.

http://www.lua.org/

Nmap Scripting Engine

Aggiornare gli script (ma serve ????)

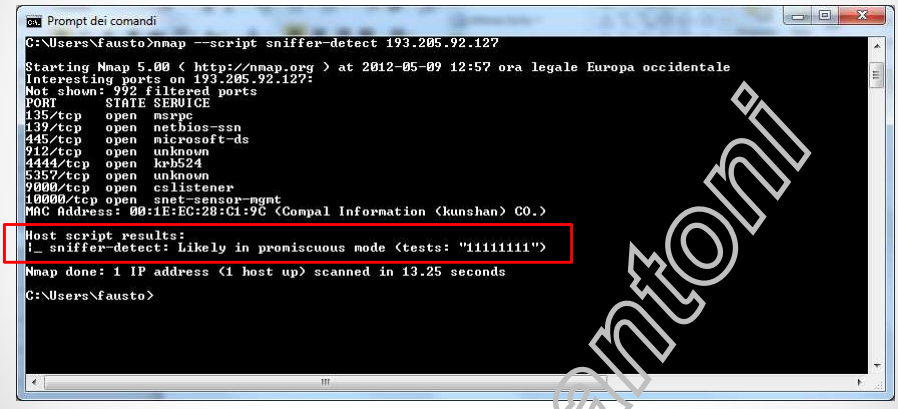
bisogna essere amministratori, ma non è così semplice



Nmap Scripting Engine

Verificare la presenza di un computer della LAN in "**promiscuous mode**"

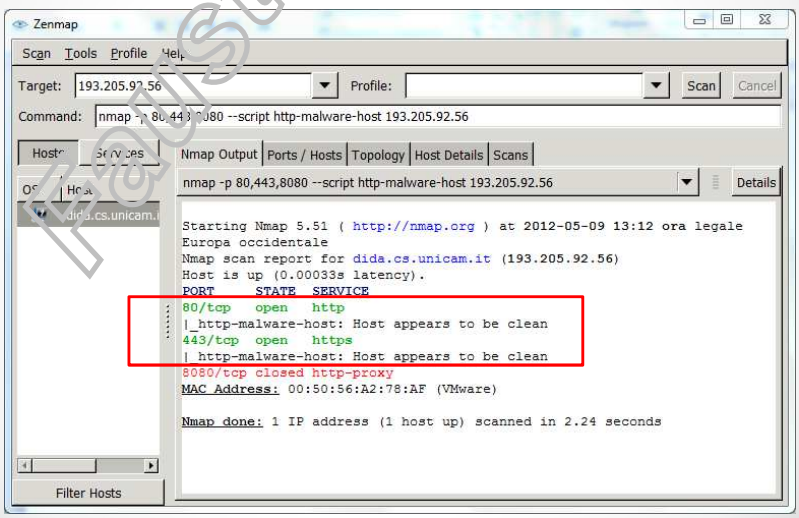
```
nmap -sV --script sniffer-detect 193.205.92.108
```



Nmap Scripting Engine

Verificare la presenza di un malware su un web serve

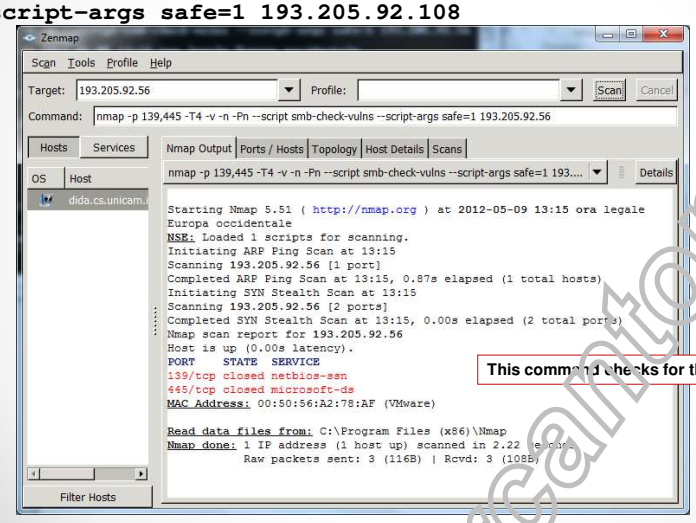
```
nmap -p80,443,8080 --script=http-malware-host 193.205.92.108
```



Nmap Scripting Engine

Verificare la presenza di Conficker tra i computer inseriti nella LAN.

```
nmap -PN -T4 -p139,445 -n -v --script=smb-check-vulns --
script-args safe=1 193.205.92.108
```



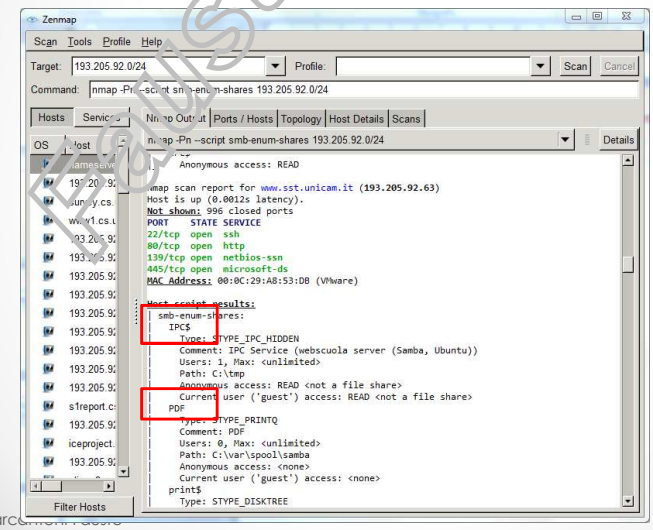
n DNS resolve
verbose

This comm checks for the MS08-067 vulnerability

Nmap Scripting Engine

Enumerare le risorse condivise in un sistema Windows o Samba file server.

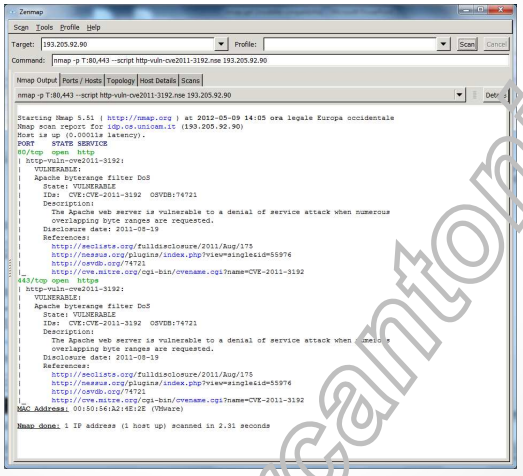
```
nmap -Pn --script=smb-enum-shares 193.205.92.0/24
```



Nmap Scripting Engine

Verificare la vulnerabilità di apache cve2011-3192

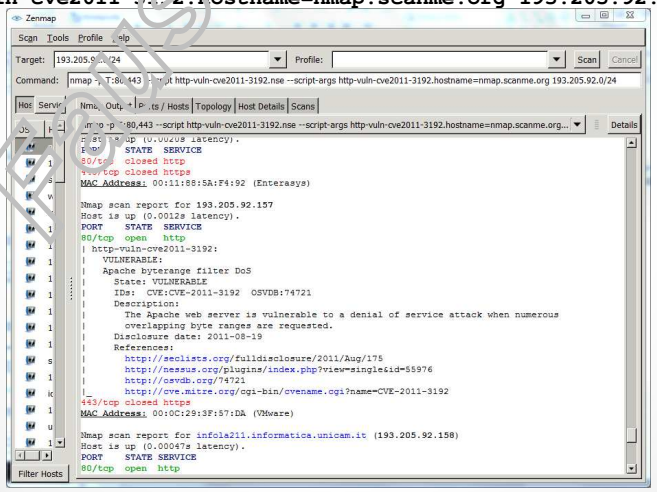
```
nmap -p T:80,443 --script http-vuln-cve2011-3192.nse 193.205.92.108
```



Nmap Scripting Engine

Verifica la vulnerabilità di apache cve2011-3192 (tutta una network)

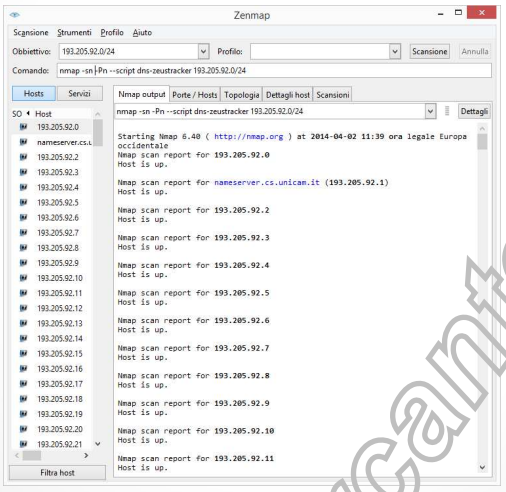
```
nmap -p T:80,443 --script http-vuln-cve2011-3192.nse --script-args
http-vuln-cve2011-3192.hostname=nmap.scanme.org 193.205.92.0/24
```



Nmap Scripting Engine

Checks if the target IP range is part of a Zeus botnet (tutta una network)

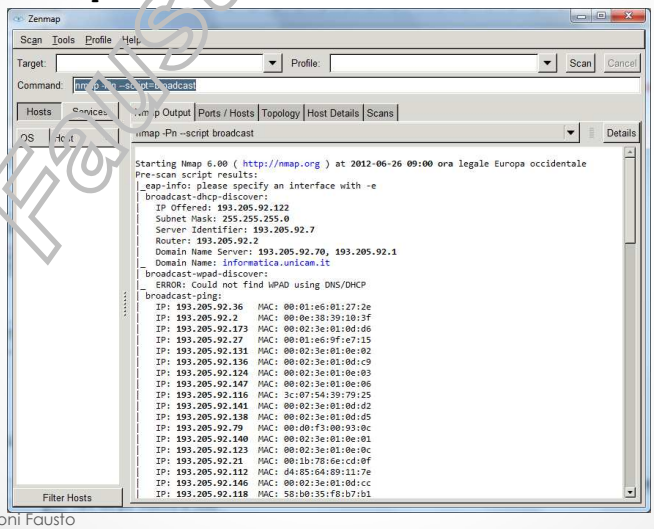
```
nmap -sn -PN --script=dns-zeustracker 193.205.92.0/24 [ -oX scan.xml ]
```



Nmap Scripting Engine

These scripts find hosts and services that advertise themselves to the network broadcast address

```
nmap -Pn --script=broadcast
```



NSE Documentation

http://nmap.org/nsedoc/index.html

NSEDoc

Index
NSE Documentation

Categories
auth
broadcast
brute
default
discovery
dos
exploit
external
fuzzer
intrusive
malware
safe
version
vuln

Scripts (show 370)

Libraries (show 96)

Scripts

acarsd-info	Retrieves information from a listening acarsd daemon. Acarsd decodes ACARS (Aircraft Communication Addressing and Reporting System) data in real time. The information retrieved by this script includes the daemon version, API version, administrator e-mail address and listening frequency.
address-info	Shows extra information about IPv6 addresses, such as embedded MAC or IPv4 addresses when available.
afp-brute	Performs password guessing against Apple Filing Protocol (AFP).
afp-ls	Attempts to get useful information about files from AFP volumes. The output is intended to resemble the output of `ls`.
afp-path-vuln	Detects the Mac OS X AFP directory traversal vulnerability, CVE-2010-0533.
afp-serverinfo	Shows AFP server information. This information includes the server's hostname, IPv4 and IPv6 addresses, and hardware type (for example `Macmini` or `MacBookPro`).
afp-showmount	Shows AFP shares and ACLs.
ajp-auth	Retrieves the authentication scheme and realm of an AJP service that requires authentication.
ajp-headers	Performs a HEAD or GET request against either the root directory or any optional directory and returns the server response headers.
ajp-methods	Finds out what options are supported by the AJP server by sending an OPTIONS request and lists potentially risky methods.
amqp-info	Gathers information (a list of all server properties) from an AMQP (advanced message queuing protocol) server.
asn-query	Maps IP addresses to autonomous system (AS) numbers.
auth-owners	Attempts to find the owner of an open TCP port by querying an auth daemon which must also be open on the target system. The auth service, also known as identd, normally runs on port 113.
auth-spoof	Checks for an identd (auth) server which is spoofing its replies.
backorifice-brute	Performs brute force password auditing against the BackOrifice service. The `backorifice-brute.ports` script argument is mandatory (it specifies ports to run the script against).
backorifice-info	Connects to a BackOrifice service and gathers information about the host and the BackOrifice service itself.
banner	A simple banner grabber which connects to an open TCP port and prints out anything sent by the listening service within five seconds.
bitcoin-getaddr	Queries a Bitcoin server for a list of known Bitcoin nodes
bitcoin-info	Extracts version and node information from a Bitcoin server

ncat

Ncat is a general-purpose command-line tool for
- ➢Reading
- ➢Writing
- ➢Redirecting
- ➢Encrypting

data across a network

Swiss Army knife

http://nmap.org/ncat/guide/index.html

ncat

connect to a network service

```
ncat dida.cs.unicam.it 80
ncat -C dida.cs.unicam.it 80
```

The -C option turns on CRLF replacement

GET / HTTP/1.0

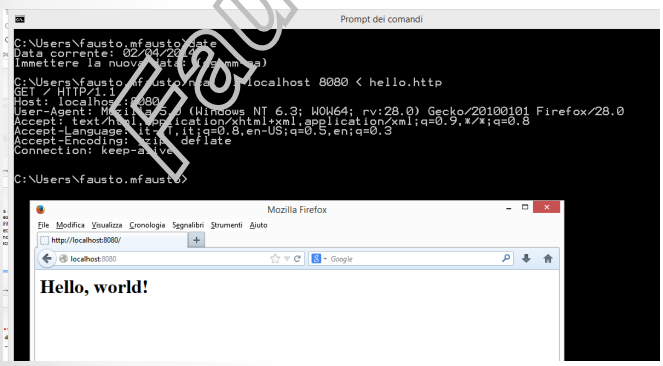
```
C:\Program Files (x86)\Nmap>ncat dida.cs.unicam.it 80
GET / HTTP/1.0

HTTP/1.1 403 Forbidden
Date: Thu, 03 Nov 2011 09:16:40 GMT
Server: Apache/2.2.3 (CentOS)
Accept-Ranges: bytes
Content-Length: 5043
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
        <head>
                <title>Apache HTTP Server Test Page powered by CentOS</title>
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
                <style type="text/css">
```


ncat

using Ncat as a web browser

```
ncat -l localhost 8080 < hello.http
```

"c:\Program Files (x86)\Nmap"\ncat -l localhost 8080 < c:\hello.http



Creare un file: hello.http

```
HTTP/1.0 200 OK

<html>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

A ME (NON) FUNZIONA …
anzi funziona male
Anzi funziona

ncat

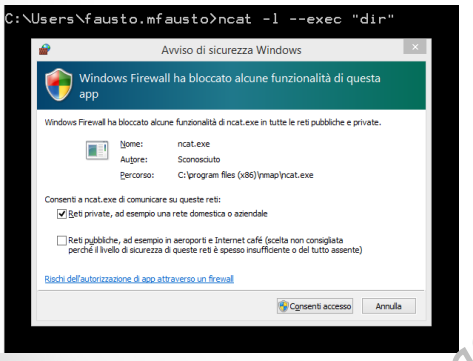
execute an external command

ncat -l --exec

ncat -l --sh-exec

Default port: 31337

The --sh-exec option (-c) works the same as --exec, except that it executes the command by passing it to /bin/sh -c on Unix or cmd.exe /C on Windows.



ncat

execute an external command

ncat -l --exec "/bin/echo HELLO"

Default port: 31337

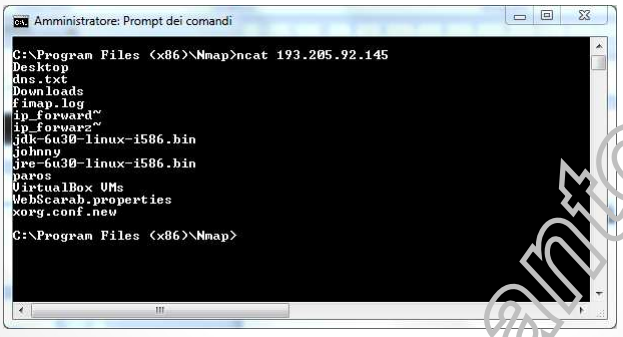
```
C:\Program Files (x86)\Nmap>ncat 193.205.92.139
HELLO

C:\Program Files (x86)\Nmap>
```

ncat

execute an external command

```
ncat -l --exec "/bin/ls"
```

Default port: 31337



ncat

execute an external command

```
ncat -l --exec "cmd.exe"
```

```
C:\Program Files (x86)\Nmap>ncat -l --exec "cmd.exe"
```

```
studente@studente-virtual-machine:~$ ncat 193.205.92.109
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Program Files (x86)\Nmap>
```

ncat

execute an external command

ncat -l --exec "cmd.exe"

```
C:\Program Files (x86)\Nmap>dir/w
dir/w
 Il volume nell'unit◆ C non ha etichetta.
 Numero di serie del volume: 28E5-29FD

 Directory di C:\Program Files (x86)\Nmap

[.]                     [..]                    3rd-party-licenses.txt
ca-bundle.crt           CHANGELOG               COPYING
COPYING_HIGWIDGETS      hello.http              icon1.ico
libeay32.dll            [licenses]              ncat.exe
ndiff.exe               NDIFF_README            nmap-mac-prefixes
nmap-os-db              nmap-payloads           nmap-protocols
nmap-rpc                nmap-service-probes     nmap-services
nmap.exe                nmap.xsl                nmap_performance.reg
nping.exe               [nselib]                nse_main.lua
[py2exe]                python26.dll            python27.dll
README-WIN32            [scripts]               [share]
ssleay32.dll            tutto_cs.txt            Uninstall.exe
zenmap.exe              ZENMAP_README
              31 File     14.310.980 byte
               7 Directory  262.013.968.384 byte disponibili

C:\Program Files (x86)\Nmap>
```


ncat

execute an external command

ncat -l --sh-exec "/bin/sh"

```
studente@studente-virtual-machine:~$ ncat -l --sh-exec "/bin/sh"
```

comando →

risultato →

```
C:\Program Files (x86)\Nmap>ncat 193.205.92.139
ls
Documenti
examples.desktop
Immagini
Modelli
Musica
Pubblici
Scaricati
Scrivania
Video
```

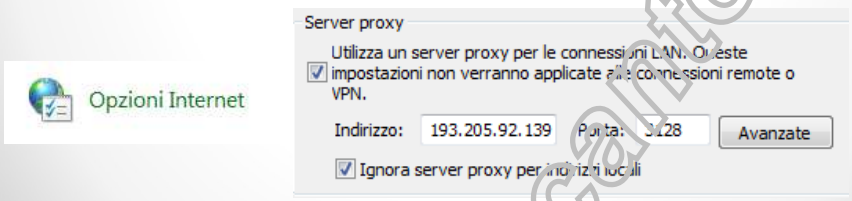
ncat

proxying

ncat -l 3128 --proxy-type http
ncat -vvv -l 3128 --proxy-type http

Ncat can act as a proxy server itself in listen mode.
The only proxy type supported is http.

```
studente@studente-virtual-machine:~$ ncat -l 3128 --proxy-type http -v
Ncat: Version 5.21 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:3128
```



ncat

Ncat simply moves bits from one place to another

ncat -l **ncat [host]**

```
studente@studente-virtual-machine:~$ ncat -l
ciao
saluti
```

```
C:\Program Files (x86)\Nmap>ncat 193.205.92.139
ciao
saluti
```

ncat

two users can communicate with each other

ncat -l --chat

```
studente@studente-virtual-machine:~$ ncat -l --chat -v
Ncat: Version 5.21 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:31337
```

ncat [host]

```
C:\Program Files (x86)\Nmap>ncat 193.205.92.139
<announce> 193.205.92.109 is connected as <user5>.
<announce> already connected: 193.205.92.109 as <user4>.
```

ncat [host]

```
C:\Program Files (x86)\Nmap>ncat 193.205.92.139
<announce> 193.205.92.109 is connected as <user4>.
<announce> already connected: nobody.
<announce> 193.205.92.109 is connected as <user5>.
<announce> already connected: 193.205.92.109 as <user4>.
```

```
studente@studente-virtual-machine:~$ ncat -l --chat -v
Ncat: Version 5.21 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:31337
Ncat: Connection from 193.205.92.109 on file descriptor 4.
Ncat: Connection from 193.205.92.109 on file descriptor 5.
```


ncat

two users can transfer files between them

receiver listens

```
ncat -l > outputfile
ncat --send-only host2 < inputfile
```

sender listens

```
ncat -l --send-only < inputfile
ncat host1 > outputfile
```

nping

Nping is an open-source tool for

- network packet generation,
- response analysis
- response time measurement

Nping allows users

- to generate network packets of a wide range of protocols

http://nmap.org/book/nping-man.html

nping

execute an external command

```
nping -c 1 --tcp -p 80 www.google.it
```

```
C:\Program Files (x86)\Nmap>nping -c 1 --tcp -p 80 www.google.it

Starting Nping 0.5.51TEST1 ( http://nmap.org/nping ) at 2011-11-03 11:52 ora solare Europa occidentale
SENT (0.3820s) TCP 193.205.92.109:29422 > 209.85.148.99:80 S ttl=64 id=52676 iplen=40  seq=3584724084 win=1480
RCVD (0.6570s) TCP 209.85.148.99:80 > 193.205.92.109:29422 SA ttl=53 id=8156 iplen=44  seq=2326870128 win=5720 <mss 1
RCVD (0.9670s) TCP 209.85.148.99:80 > 193.205.92.109:29422 SA ttl=53 id=8157 iplen=44  seq=2326870128 win=5720 <mss 1
RCVD (1.5650s) TCP 209.85.148.99:80 > 193.205.92.109:29422 SA ttl=53 id=8158 iplen=44  seq=2326870128 win=5720 <mss 1

Max rtt: 930.000ms | Min rtt: 22.000ms | Avg rtt: 428.000ms
Raw packets sent: 3 (114B) | Rcvd: 3 (138B) | Lost: 0 (0.00%)
Tx time: 0.25300s | Tx bytes/s: 213.44 | Tx pkts/s: 3.95
Rx time: 1.25300s | Rx bytes/s: 110.14 | Rx pkts/s: 2.39
Nping done: 1 IP address pinged in 1.64 seconds
```

riferimenti

- www.insecure.org
- www.nmap-tutorial.com
- nmap.org/man/it/index.html
- nmap.org/book/nse.html
- nmap.org/nsedoc/index.html

- Manuali on line:
- http://resources.infosecinstitute.com/nmap-cheat-sheet/
- http://blog.hackersonlineclub.com/2014/01/nmap-network-mapping-cheat-sheet.html