Nmap en Ligne de Commande

Claude DUTREILLY

claude.dutreilly@univ-nantes.fr

Laboratoire de Planétologie et Géodynamique
de Nantes

UMR 6112

Nmap

- Définition *
- Fonctionnement *
- Principes et interprétation des résultats
- Différentes options
- Conclusion
- Annexes

* source: Wikipedia (traduction du site: http://nmap.org/man/man-briefoptions.html)

Définition

- **Nmap = Network Mapper**

Nmap est un **scanner de ports** open source créé par Fyodor et distribué par Insecure.org.

Il est conçu pour *détecter les ports ouverts*, les *services hébergés* et les *informations sur le système d'exploitation* d'un ordinateur distant.

Ce logiciel est devenu une référence pour les administrateurs réseaux car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau

INSECURE.ORG

Fonctionnement

- Pour scanner les ports d'un ordinateur distant, **Nmap** utilise diverses techniques d'analyse basées sur des protocoles tels que **TCP**, **IP**, **UDP** ou **ICMP**.
- Par défaut **Nmap** scanne les port de **1 à 1024** et les ports indiqués dans le fichier *nmap-services*.
- De même, il se base sur les *réponses particulières* qu'il obtient à des *requêtes particulières* pour obtenir une **empreinte de la pile IP**, souvent *spécifique du système* qui l'utilise. C'est par cette méthode que l'outil permet de reconnaître la version d'un système d'exploitation et aussi la version des services en écoute.
- Le code source est disponible sous la licence *GNU GPL*.

Principes (1)

But de Nmap: Solliciter des réponses de la machine cible pour montrer la présence ou non d'une application ou d'un service

6 états de ports reconnus par **Nmap**

- **Open**
- **Closed**
- **Filtered**
- **Unfiltered**
- **Open | Filtered**
- **Closed | Filtered**.

Principes (2)

1) Open

Une application qui tourne sur la machine cible accepte les connexions TCP ou les paquets UDP sur ce port.

Les ports ouverts montrent également les services disponibles sur le réseau

2) Closed

Accessible (reçoit et répond aux paquets envoyés par Nmap) mais il n'y a pas d'application à l'écoute sur ce port.

Utilité: machine cible « UP ». détection de l'OS par Nmap

Remarque: il peut être utile de bloquer de tels ports avec un firewall

Principes (3)

3) Filtered

Nmap ne peut déterminer si le port est ouvert car il est intercepté avant d'atteindre le port.

Peut être causé par un firewall, des règles de routage ou bien un firewall intégré à la machine cible.

Conséquences: ralentissement du scan (nmap réitère plusieurs fois son scan)

4) Unfiltered

Le port est accessible, mais nmap est incapable de déterminer s'il est ouvert/fermé.

Il faut alors tester avec d'autres types de scan: Windows scan, ou FIN scan pour savoir si le port est ouvert.

Principes (4)

5) Open | Filtered

Nmap est incapable de déterminer si le port est ouvert ou filtré.

-> cela arrive, par ex, lorsqu'un port ouvert ne donne pas de réponse !

-> l'absence de réponse peut vouloir dire également qu'un filtrage a « droppé » le paquet généré par Nmap ou la réponse obtenue.

6) Closed | Filtered

Cet état est utilisé quand Nmap est incapable de déterminer si un port est fermé ou filtré.

Différentes options (1)

- commande de base:

nmap <machine_cible>

nmap -P0 <machine_cible> ..... passe le firewall

```
[root@localhost ~]#  nmap -P0 himalia

Starting Nmap 4.20 ( http://insecure.org ) at 2008-03-26 08:28 CET
Interesting ports on himalia:
Not shown: 1694 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
6000/tcp open  X11

Nmap finished: 1 IP address (1 host up) scanned in 0.093 seconds
```

Différentes options (2)

- Découvrir application/services en écoute sur les ports TCP:

nmap -sV <machine_cible>

```
[root@localhost ~]# nmap -sV himalia
Starting Nmap 4.20 ( http://insecure.org ) at 2008-03-26 08:31 CET
Interesting ports on himalia:
Not shown: 1694 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 3.6.1p2 (protocol 1.99)
111/tcp  open  rpcbind  2 (rpc #100000)
6000/tcp open  X11      (access denied)
Service Info: OS: Unix
Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 6.300 seconds
```

Différentes options (3)

- Découvrir le système d'exploitation:

nmap -sO <machine_cible>

```
[root@localhost ~]#  nmap -O himalia
Starting Nmap 4.20 ( http://insecure.org ) at 2008-03-26 08:32 CET
Interesting ports on himalia:
Not shown: 1694 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
6000/tcp open  X11
Device type: general purpose|WAP|storage-misc
Running: Linux 2.4.X, Linksys Linux 2.4.X, Asus Linux 2.4.X, Maxtor Linux 2.4.X
OS details: Linux 2.4.20 - 2.4.32, Linux-based embedded device (Linksys WRT54GL WAP, Buffalo
      AirStation WLA-G54 WAP, Maxtor Shared Storage Drive, or Asus Wireless Storage Router)
Uptime: 68.640 days (since Thu Jan 17 17:11:08 2008)
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 1.785 seconds
```

Différentes options (4)

balayage sur des réseaux entiers:rechercher les serveurs Web present sur un LAN, par ex

nmap -sS -sV -p 80 <@réseau_classeC / 24> ou <X.Y.Z.80 - 443>

```
[root@localhost ~]# nmap -P0 -sV -p 80 172.16.13.16-32

Starting Nmap 4.20 ( http://insecure.org ) at 2008-03-26 11:13 CET
Interesting ports on 172.16.13.16:
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.3 ((CentOS))
MAC Address: 00:14:5E:57:A2:5C (IBM)

Interesting ports on venus (172.16.13.18):
PORT   STATE  SERVICE VERSION
80/tcp closed http
MAC Address: 40:13:97:9D:00:14 (Unknown)

Interesting ports on pc-sig (172.16.13.22):
PORT   STATE    SERVICE VERSION
80/tcp filtered http
MAC Address: 00:14:22:26:DF:F2 (Dell)

Interesting ports on pleiades (172.16.13.25):
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 1.3.31 ((Unix) mod_perl/1.24 ApacheJserv/1.1.2)
MAC Address: 00:03:BA:14:A4:F4 (Sun Microsystems)

Interesting ports on glpi (172.16.13.29):
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.3 ((Mandriva Linux/PREFORK-1mdv2007.0))
MAC Address: 00:02:A5:62:E5:E7 (Compaq Computer)

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/ .
Nmap finished: 17 IP addresses (13 hosts up) scanned in 6.914 seconds
```

Option « -n » permet de désactiver la résolution DNS et ainsi accroître la rapidité du balayage.

Conclusions (1)

Nmap est un outil puissant, versatile, **indispensable** à tout ASR.

En combinant ses diverses options, il offre une grande souplesse qui permet d'analyser tout réseau, tester le filtrage, les filtres IP, de découvrir les ports ouverts, de découvrir de nouvelles machines (!)

Un petit plus, il est libre et gratuit et c''est le meilleur outil pour faire cela (source HSC)

TOUTEFOIS il faut être prudent, un scan est équivalent à une tentative d'intrusion, et certaines méthodes de scan peuvent entraîner des dysfonctionnements sur une machine.

« Scanme.nmap.org est gracieusement fourni par l'auteur de Nmap afin de tests ! »

Conclusions (2)

James Messer, auteur de *Secrets of a Network Cartography*, écrit:

The bad guys are already using nmap for reconnaissance, because a single scan can tell you a lot about the open doors and windows in a computer's house.

The good guys are using nmap to make their network safer.

Comment empêcher les scans ?

Un scan précéde souvent une attaque .... utilisation d'un IDS.

ex à la Fac sciences : Detescan

Biblio

- présentation de nmap par Marie Claude Quidoz – Outils de sécurité. 15/11/2001 -
- doc officielle de Nmap sur http://www.insecure.org/nmap/man
- NMAP Open Source Security Tools for Information Technology Professional par Aron Trauring – 7/11/2005
- NMAP v1.4 par Gaël Beauquin CNRS/UREC 24/3/2006
- Article paru dans GNU Linux Magazine Avril 2005 n°71 Cyril Nocton p38 à 48
- et surtout: Secrets Of Network Cartography, A comprehensive guide to nmap. James Messer

Options disponibles (1)

Pour Nmap 4.20, voici les options que nous pouvons utiliser :

- Les cibles peuvent être spécifiées par des noms d'hôtes, des adresses IP (v4 ou v6), des adresses de réseaux...

  ex: www.cnrs.fr, 192.168.1.1, 192.168.1.0/24, microsoft.com/24, 10.0-255.0-255.1-254

- ***Découverte des hôtes :***
  - **-sL**: List Scan - Liste simplement les cibles à scanner
  - **-sP**: Ping Scan - Ne fait que déterminer si les hôtes sont en ligne -P0: Considère que tous les hôtes sont en ligne-- évite la découverte des hôtes
  - **-PN**: Considérer tous les hôtes comme étant connectés -- saute l'étape de découverte des hôtes
  - **-PS/PA/PU [portlist]**: Découverte TCP SYN/ACK ou UDP des ports en paramètre
  - **-PE/PP/PM**: Découverte de type requête ICMP echo, timestamp ou netmask
  - **-PO** [num de protocole]: Ping IP (par type)
  - **-n/-R**: Ne jamais résoudre les noms DNS/Toujours résoudre [résout les cibles actives par défaut]
  - --dns-servers <serv1[,serv2],...>: Spécifier des serveurs DNS particuliers
  - --system-dns: Utilise le resolveur DNS du système d'exploitation

Options disponibles (2)

- ***Techniques de scan :***
  - **-sS/sT**/sA/sW/sM: Scans TCP SYN/Connect()/ACK/Window/Maimon
  - -sN/sF/sX: Scans TCP Null, FIN et Xmas
  - **-sU**: Scan UDP (Cette option ne fonctionne pas en IPv6)

    --scanflags <flags>: Personnalise les flags des scans TCP
  - **-sI** <zombie host[:probeport]>: Idlescan (scan passif)
  - **-sO**: Scan des protocoles supportés par la couche IP
  - **-b <ftp relay host>**: Scan par rebond FTP

    --traceroute: Détermine une route vers chaque hôte

    --reason: Donne la raison pour laquelle tel port apparaît à tel état

Options disponibles (3)

- ***Spécifications des ports et ordre de scan :***
  - **-p <plage de ports>**: Ne scanne que les ports spécifiés

    Exemple: -p22; -p1-65535; -pU:53,111,137,T:21-25,80,139,8080
  - **-F**: Fast - Ne scanne que les ports listés dans le fichier nmap-services
  - **-r**: Scan séquentiel des ports, ne mélange pas leur ordre

    --top-ports <nombre>: Scan <nombre> de ports parmis les plus courants

    --port-ratio <ratio>: Scan <ratio> pourcent des ports les plus courants
- ***Détection de service/version :***
  - **-sV**: Teste les ports ouverts pour déterminer le service en écoute et sa version

    --version-light: Limite les tests aux plus probables pour une identification plus rapide

    --version-intensity <niveau>: De 0 (léger) à 9 (tout essayer)

    --version-all: Essaie un à un tous les tests possibles pour la détection des versions

    --version-trace: Affiche des informations détaillées du scan de versions (pour débogage)

Options disponibles (4)

- ***Script scan*** :
  - **-sC**: équivalent de --script=safe,intrusive

    --script=<lua scripts>: <lua scripts> est une liste de répertoires ou de scripts séparés par des virgules

    --script-args=<n1=v1,[n2=v2,...]>: passer des arguments aux scripts

    --script-trace: Montre toutes les données envoyées ou recues

    --script-updatedb: Met à jour la base de données des scripts. Seulement fait si -sC ou --script a été aussi donné.
- ***Détection de système d'exploitation*** :
  - **-O**: Active la détection d'OS

    --osscan-limit: Limite la détection aux cibles prométeuses

    --osscan-guess: Devine l'OS de facon plus agressive

Options disponibles (5)

- ***Temporisation et performance*** :

  Les options qui prennent un argument de temps sont en milisecondes a moins que vous ne spécifiiez 's' (secondes), 'm' (minutes), ou 'h' (heures) à la valeur (e.g. 30m).

  - **-T[0-5]**: Choisit une politique de temporisation (plus élevée, plus rapide)

  --min-hostgroup/max-hostgroup <msec>: Tailles des groupes d'hôtes à scanner en parallèle
  --min-parallelism/max-parallelism <msec>: Parallélisation des paquets de tests (probes)
  --min_rtt_timeout/max-rtt-timeout/initial-rtt-timeout <msec>: Spécifie le temps d'aller-retour des paquets de tests
  --min_rtt_timeout/max-rtt-timeout/initial-rtt-timeout <msec>: Spécifie le temps d'aller-retour des paquets detests
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Précise le round trip time des paquets de tests.
  --max-retries <tries>: Nombre de retransmissions des paquets de tests des scans de ports.
  --host-timeout <msec>: Délai d'expiration du scan d'un hôte
  --scan-delay/--max-scan-delay <time>: Ajuste le délais entre les paquets de tests.

Options disponibles (6)

- ***Evasion parefeu/ids et usurpation d'identité :***
  - **-f**; --mtu <val>: Fragmente les paquets (en spécifiant éventuellement la MTU)
  - **-D <decoy1,decoy2[,ME],...>**: Obscurci le scan avec des leurres
  - **-S <IP_Address>**: Usurpe l'adresse source
  - **-e <iface>**: Utilise l'interface réseau spécifiée
  - **-g/--source-port <portnum>**: Utilise le numéro de port comme source

    --data-length <num>: Ajoute des données au hasard aux paquets émis

    --ip-options <options>: Envoi des paquets avec les options IP spécifiées.

    --ttl <val>: Spécifie le champ time-to-live IP

    --spoof-mac <adresse MAC, préfixe ou nom du fabriquant>: Usurpe une adresse MAC

    --badsum: Envoi des paquets TCP/UDP avec une somme de contrôle erronée.

Options disponibles (7)

- ***Sortie :***
  - **-oN/-oX/-oS/-oG <file>**: Sortie dans le fichier en paramètre des résultats du scan au format normal, XML, s|<rIpt kIddi3 et Grepable, respectivement
  - **-oA <basename>**: Sortie dans les trois formats majeurs en même temps
  - **-v**: Rend Nmap plus verbeux (-vv pour plus d'effet)
  - **-d[level]**: Sélectionne ou augmente le niveau de débogage (significatif jusqu'à 9)

    --packet-trace: Affiche tous les paquets émis et reçus

    --iflist: Affiche les interfaces et les routes de l'hôte (pour débogage)

    --log-errors: Journalise les erreurs/alertes dans un fichier au format normal

    --append-output: Ajoute la sortie au fichier plutôt que de l'écraser

    --resume <filename>: Reprend un scan interrompu

    --stylesheet <path/URL>: Feuille de styles XSL pour transformer la sortie XML en HTML

    --webxml: Feuille de styles de références de Insecure.Org pour un XML plus portable

    --no_stylesheet: Nmap n'associe pas la feuille de styles XSL à la sortie XML

Options disponibles (8)

- ***Divers :***
  - **-6**: Active le scan IPv6
  - **-A**: Active la détection du système d'exploitation et des versions

    --datadir <dirname>: Spécifie un dossier pour les fichiers de données de Nmap

    --send-eth/--send-ip: Envoie des paquets en utilisant des trames Ethernet ou des paquets IP bruts

    --privileged: Suppose que l'utilisateur est entièrement privilégié -V: Affiche le numéro de version

    --unprivileged: Suppose que l'utilisateur n'a pas les privilèges d'usage des raw socket
  - **-h**: Affiche ce résumé de l'aide

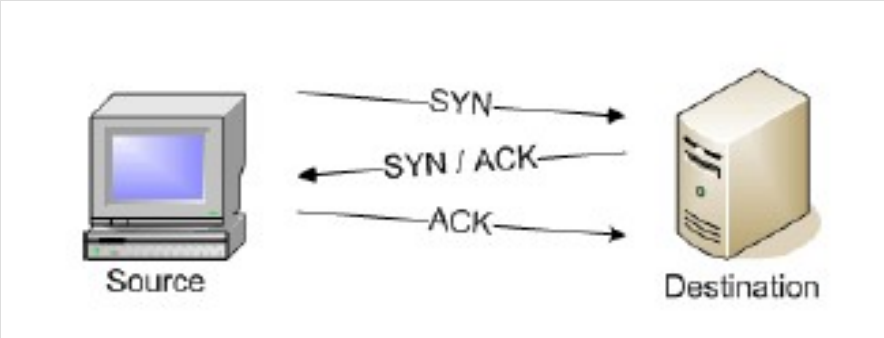
Résumé des techniques de Scan

Nmap Scan	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	`-sS`	YES	YES	NO
TCP connect() Scan	`-sT`	NO	YES	NO
FIN Scan	`-sF`	YES	YES	NO
Xmas Tree Scan	`-sX`	YES	YES	NO
Null Scan	`-sN`	YES	YES	NO
Ping Scan	`-sP`	NO	NO	NO
Version Detection	`-sV`	NO	NO	NO
UDP Scan	`-sU`	YES	NO	YES
IP Protocol Scan	`-sO`	YES	NO	NO
ACK Scan	`-sA`	YES	YES	NO
Window Scan	`-sW`	YES	YES	NO
RPC Scan	`-sR`	NO	NO	NO
List Scan	`-sL`	NO	NO	NO
Idlescan	`-sI`	YES	YES	NO
FTP Bounce Attack	`-b`	NO	YES	NO

exemple de mail envoyé par detescan

```
Analyse du fichier :    /var/stats/firewall.tmp
Taille du fichier :     42 Mb
Destinataire :          support@sciences.univ-nantes.fr,claude.dutreilly@univ-nantes.fr
Lignes :                212554
Debut de l'analyse :    26/03/2008 06:00:10
Fin de l'analyse :      26/03/2008 06:51:38
Parametres pour detection scans : nb machines minimum >= 3, nb ports minimum >= 5
Detescan v20040308 a detecte les scans suivants a partir des logs du routeur iptables
Statistiques des connexions rejetees (au moins 20 connexion(s) par port) :
          5672 connexion(s) sur le port     53/udp (domain)
                dont   1161 connexion(s) de 172.16.13.59 sur 193.52.103.2 ()
                dont   2378 connexion(s) de 172.16.40.52 sur 192.168.100.2 ()
                dont    814 connexion(s) de 172.16.11.24 sur 193.252.19.4 ()
                dont   1082 connexion(s) de 172.16.11.24 sur 193.252.19.3 ()
```

TCP Handshake

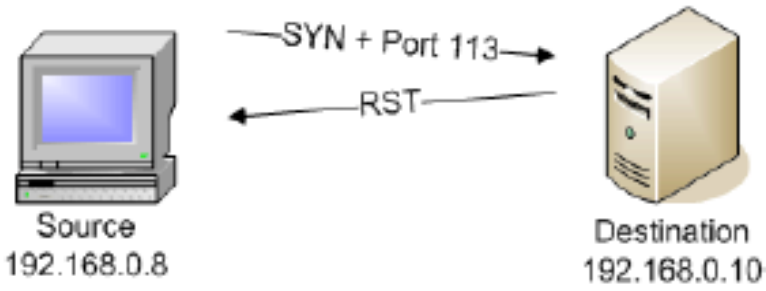


- Une « poignée de main » en trois étapes établit une connexion TCP.
- Elle assure l'échange (drapeau SYN) et l'acquittement (drapeau ACK) des numéros de séquence initiaux.

Scan TCP SYN : -sS

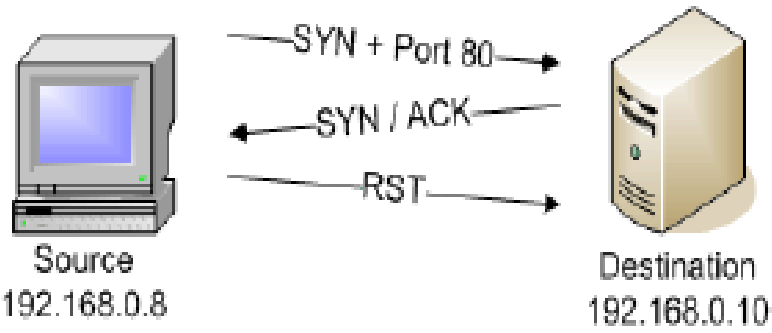
- cas d'un port fermé

TCP SYN Scan Operation
Most of the ports queried during the TCP SYN scan will probably be closed. These closed port responses to the TCP SYN frame will be met with a RST frame from the destination station.



- cas d'un port ouvert

If nmap receives an acknowledgment to a SYN request, then the port is open. Nmap then sends an RST to reset the session, and the handshake is never completed.



Scan TCP SYN : -sS (suite)

Advantages of the TCP SYN Scan
The TCP SYN scan never actually creates a TCP session, so isn't logged by the destination host's applications. This is a much "quieter" scan than the TCP connect() scan, and there's less visibility in the destination system's application logs since no sessions are ever initiated. Since an application session is never opened, the SYN scan is also less stressful to the application service.

Disadvantages of the TCP SYN Scan
The TCP SYN scan requires that nmap have privileged access to the system. Without privileged access, nmap can't create the raw packets necessary for this half-open connection process.

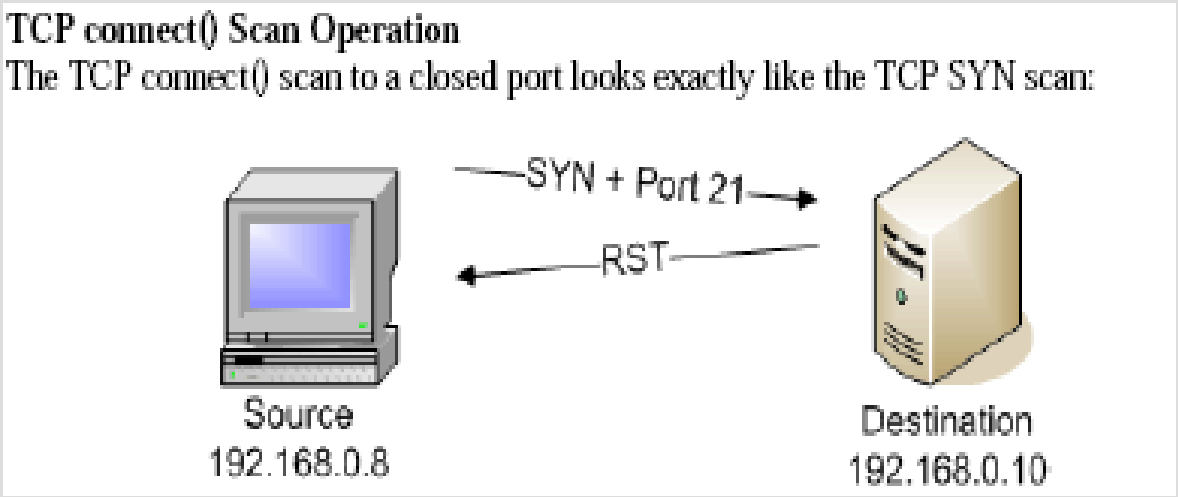
When to use the TCP SYN Scan
The SYN scan is a common scan when looking for open ports on a remote device, and its simple SYN methodology works on all operating systems. Because it only half-opens the TCP connections, it's considered a very 'clean' scan type.

The TCP SYN scan only provides open, closed, or filtered port information. To determine operating system or process version information, more intrusive scanning is required, such as the version scan (-sV) or the operating system fingerprinting (-O) option.

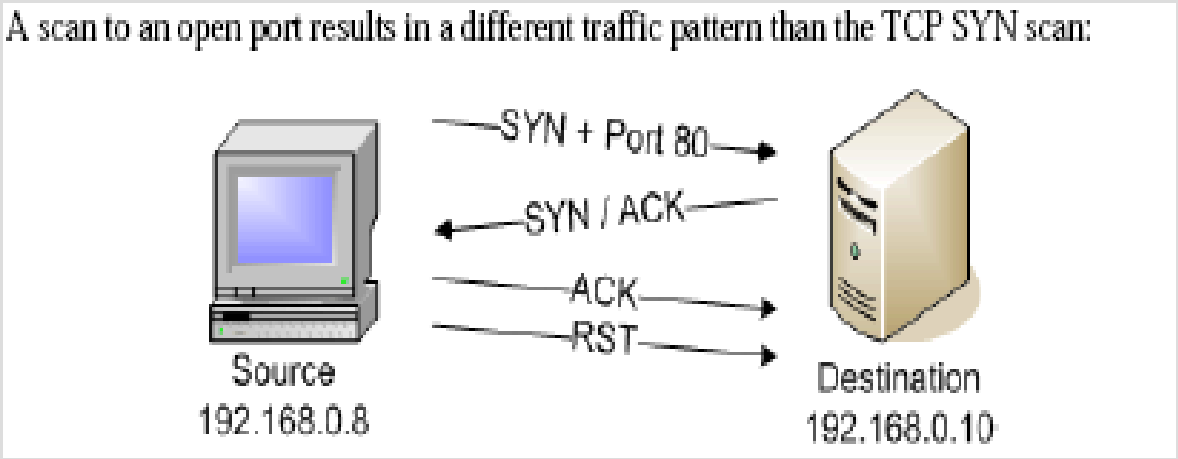
The TCP SYN scan is the most common scan to use because it works on all networks, across all operating systems, and it's invisible to applications. If the SYN scan didn't work, then TCP wouldn't work!

Scan TCP Connect : -sT

- cas d'un port fermé



- cas d'un port ouvert



Scan TCP Connect : -sT (suite)

Advantages of the TCP connect() Scan
No special privileges are required to run the TCP connect() scan. Nmap uses the operating system's normal method of connecting to remote devices via TCP before it tears down the connection with the RST packet. Because these are TCP-based methods that any user can employ, no additional rights or privileges are required.

Disadvantages of the TCP connect() Scan
The disadvantage of this scan is apparent when application connection logs are examined. Since the TCP connect() scan is completing a TCP connection, normal application processes immediately follow. These applications are immediately met with a RST packet, but the application has already provided the appropriate login screen or introductory page. By the time the RST is received, the application initiation process is already well underway and additional system resources are used.

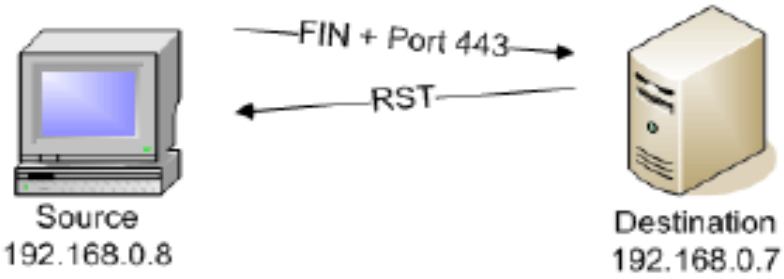
When to use the TCP connect() Scan
Because this scan is so obvious when browsing through the application event logs, it might be considered the TCP scan of last resort. If privileged access isn't available and determination of open TCP ports is absolutely necessary, however, this scan may be the only method available.

Scan TCP FIN: sF

- cas d'un port fermé

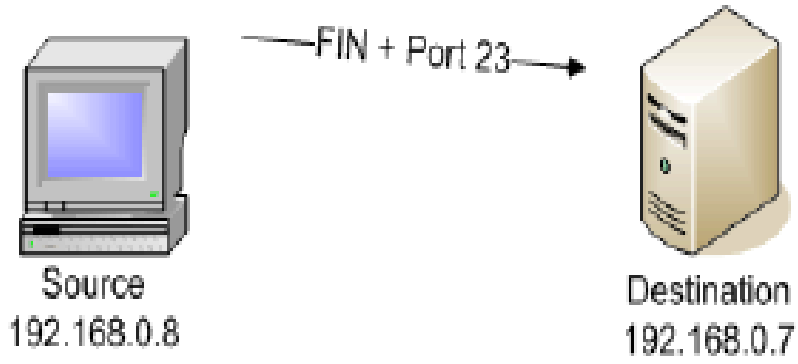
FIN Scan
The FIN scan's "stealth" frames are unusual because they are sent to a device without first going through the normal TCP handshaking. If a TCP session isn't active, the session certainly can't be formally closed!

In this FIN scan, TCP port 443 is closed so the remote station sends a RST frame response to the FIN packet:



- cas d'un port ouvert

If a port is open on a remote device, no response is received to the FIN scan:



Scan TCP FIN: -sF (suite)

The nmap output shows the open ports located with the FIN scan:

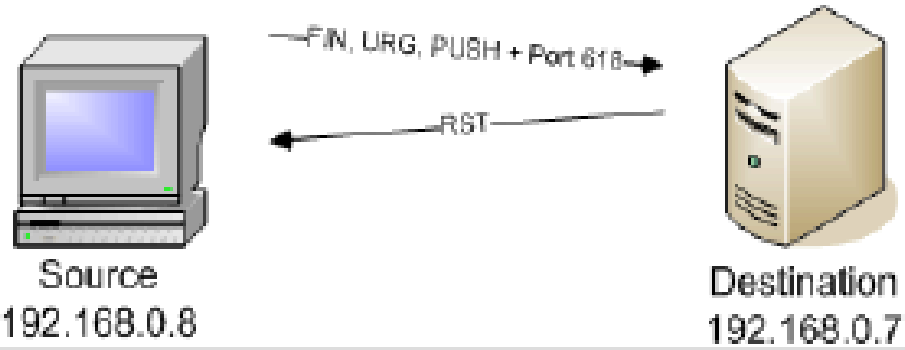
```
# nmap -sF -v 192.168.0.7

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-23
21:17 EDT
Initiating FIN Scan against 192.168.0.7 [1663 ports] at 21:17
The FIN Scan took 1.51s to scan 1663 total ports.
Host 192.168.0.7 appears to be up ... good.
Interesting ports on 192.168.0.7:
(The 1654 ports scanned but not shown below are in state: closed)
PORT     STATE          SERVICE
21/tcp   open|filtered ftp
22/tcp   open|filtered ssh
23/tcp   open|filtered telnet
79/tcp   open|filtered finger

110/tcp  open|filtered pop3
111/tcp  open|filtered rpcbind
514/tcp  open|filtered shell
886/tcp  open|filtered unknown
2049/tcp open|filtered nfs
MAC Address: 00:03:47:6D:29:D7 (Intel)

Nmap finished: 1 IP address (1 host up) scanned in 2.276 seconds
               Raw packets sent: 1674 (66.9KB) | Rcvd: 1655 (76.1KB)
#
```

Scan Xmas Tree: -sX

- cas d'un port fermé

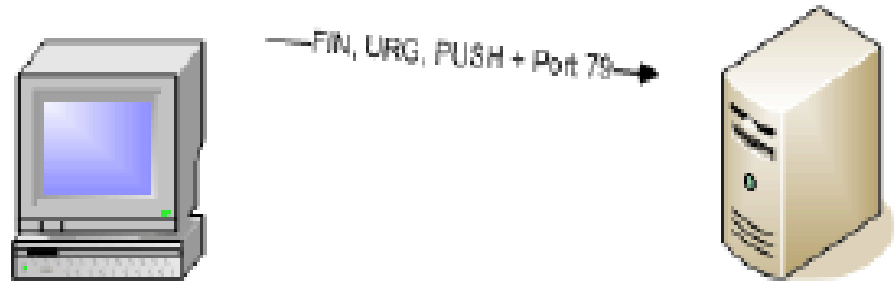
The Xmas tree scan sends a TCP frame to a remote device with the URG, PUSH, and FIN flags set. This is called a Xmas tree scan because of the alternating bits turned on and off in the flags byte (00101001), much like the lights of a Christmas tree.

A closed port responds to a Xmas tree scan with a RST:



- cas d'un port ouvert

Similar to the FIN scan, an open port on a remote station is conspicuous by its silence:



Scan Xmas Tree: -sX (suite)

The Xmas tree scan output shows similar results to the FIN scan:

```
# nmap -sX -v 192.168.0.7

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-23
21:18 EDT
Initiating XMAS Scan against 192.168.0.7 [1663 ports] at 21:18
The XMAS Scan took 1.55s to scan 1663 total ports.
Host 192.168.0.7 appears to be up ... good.
Interesting ports on 192.168.0.7:
(The 1654 ports scanned but not shown below are in state: closed)
PORT      STATE          SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
79/tcp    open|filtered finger
110/tcp   open|filtered pop3
111/tcp   open|filtered rpcbind
514/tcp   open|filtered shell
886/tcp   open|filtered unknown
2049/tcp  open|filtered nfs
MAC Address: 00:03:47:6D:29:D7 (Intel)

Nmap finished: 1 IP address (1 host up) scanned in 2.432 seconds
                Raw packets sent: 1674 (66.9KB) | Rcvd: 1655 (76.1KB)
#
```

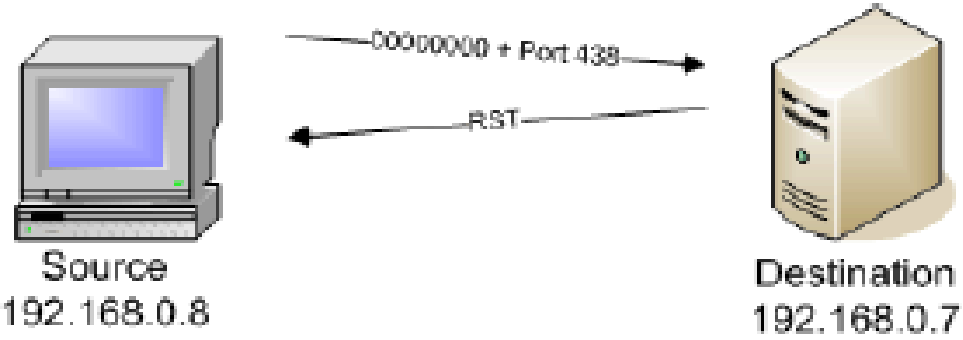
Scan NULL: -sN

- cas d'un port fermé

The Null Scan (-sN)

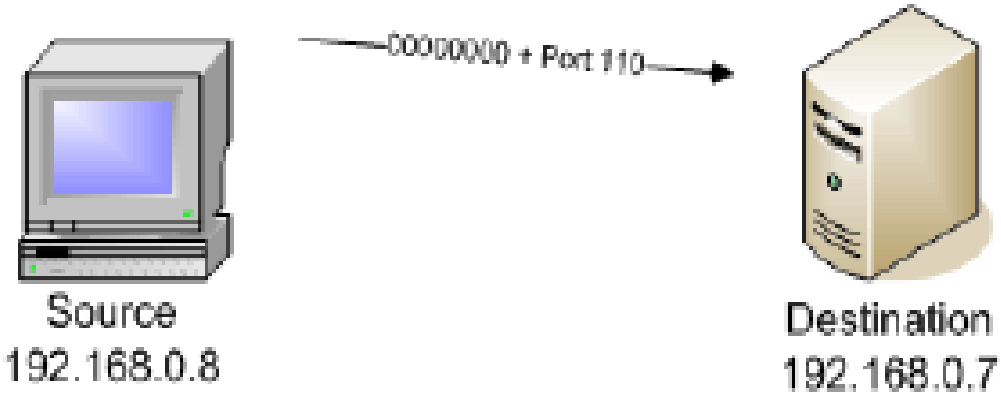
The null scan turns off all flags, creating a lack of TCP flags that should never occur in the real world.

If the port is closed, a RST frame should be returned:



- cas d'un port ouvert

As expected, the response of a null scan to an open port results in no response:



Scan NULL: -sN (suite)

Advantages of the FIN, Xmas Tree, and Null Scan
Since no TCP sessions are created for any of these scans, they are remarkably quiet from the perspective of the remote device's applications. Therefore, none of these scans should appear in any of the application logs.

These scans are also some of the most minimal port-level scans that nmap can execute. For a closed port, only two packets are transferred. A single frame is all that's necessary to find an open port!

Disadvantages of the FIN, Xmas Tree, and Null Scan
Unfortunately, Microsoft's implementation of the TCP/IP stack renders these particular scans less that useful. On a Windows-based computer, all ports will appear to be closed regardless of their actual state. This provides a backhanded advantage, since any device showing open ports must not be a Windows-based device!

These scan types are using packets that do not follow the rules of TCP. To create these specialized packets, the raw sockets capability of the operating system builds the packets from scratch. This avoids the operating system requirements that are usually forced on IP communication, but it also requires that the user running these nmap scans have privileged access to the system.

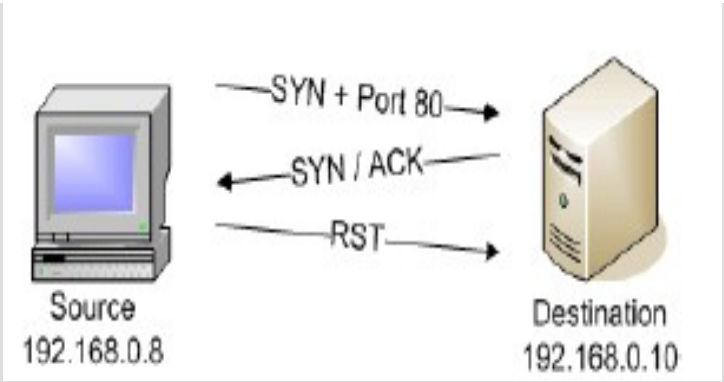
Détection de services / versions: -sV

Version Detection Operation

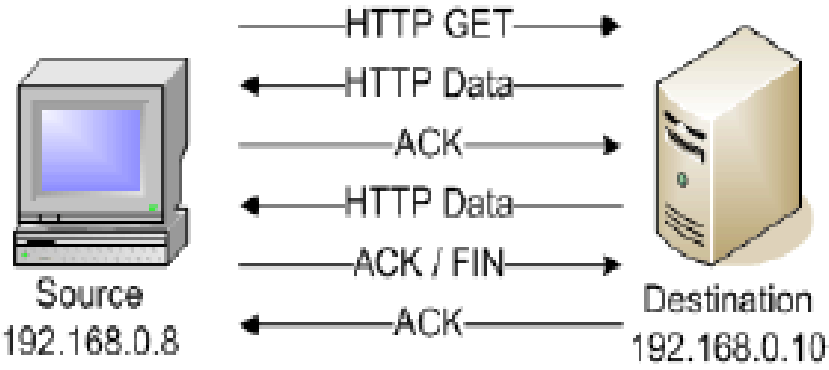
The version detection scan runs in conjunction with another scan type that will identify open ports. If another scan type is not specified on the command line, nmap will run a TCP SYN scan (for a privileged user) or a TCP connect() scan (for non-privileged users) prior to running the version detection scan.

If open ports are found, the version detection scan will begin the probing process with the remote device. The version detection scan communicates directly with the remote application to uncover as much information as possible.

In this example, the default TCP SYN scan runs prior to the version detection scan to identify the open port:



After the TCP SYN scan identifies port 80 as open, the version detection process begins. The process shown in this example is for this specific example. Other ports and application will operate differently.



Advantages of Using Version Detection

Version information is valuable information! Version scanning for service information provides easier management of patches and updates. A network security manager can scan every host in an organization to verify that software is at the correct versions. Stations showing older software revisions are identified and further action can be taken.

The version information scan can also assist in locating software that is not complaint with organizational standards. This is also an easy method of verifying the licenses of application services. Nmap can find all of the devices running a specific version of server software to determine if the quantity meets the organization's licensing agreements.

Disadvantages of Using Version Detection

The version scan is very invasive because of the probing that must occur to prompt the service for information. The fingerprint comparison must have information from the application to compare to the fingerprint in the `nmap-service-probes` file, and this process transmits a number of packets between the source and destination.

The version scan also opens sessions with the remote applications, which will often display in an application's log file. These sessions are almost always necessary, and can't be avoided if the version scan is going to decisively determine the application type and version.

Version detection will only work with TCP or UDP port scans. The ping scan (`-sP`), the list scan (`-sL`) and the IP protocol scan (`-sO`) will not run on the same command line with version detection.

Détection de services / versions: -sV (re-suite)

The version scan isn't foolproof. The version detection is only as good as the `nmap-service-probe` file, but this support file is constantly under revision. If nmap is unable to match an appropriate fingerprint, check to see if an URL and fingerprint is provided for uploading into the nmap database. Your services can assist in making the next version of nmap even better!

When to use Version Detection

The name and version of a service can provide the security team with information that it can use to keep the network applications patched and up-to-date. The server team can use version scans to confirm that a series of upgrades have been completed successfully. If unknown stations are found during a ping scan, the version scan can help determine what applications these 'rogue' stations are providing!

The version detection also shows what other scans might provide when polling network devices for version information. If the security team understands what other people can see, they can revise their security strategies to create a safer computing environment.

nmap scan report : -oX

nmap scan report - scan @ Wed Mar 26 14:10:43 2008

scan summary | **scan info** | **172.16.13.28 / monitor** | **runstats**

scan summary

nmap was initiated at Wed Mar 26 14:10:43 2008 with these arguments:
`nmap -PO -sV -oX /tmp/test1.xml cups`
The process stopped at Wed Mar 26 14:10:50 2008. Debuging was disabled, the verbosity level was 0.

172.16.13.28 / monitor

address

- 172.16.13.28 (ipv4)
- 00:50:DA:36:36:74 (mac)

hostnames

- monitor (PTR)

ports

The 1693 ports scanned but not shown below are in state: **closed**

Port		State	Service	Product	Version	Extra info
21	tcp	open	ftp	ProFTPD		
22	tcp	open	ssh	OpenSSH	4.5	protocol 2.0
111	tcp	open	rpc			
631	tcp	open	ipp	CUPS	1.2	

runstats

- 7 sec. scanned
- 1 host(s) scanned
- 1 host(s) online
- 0 host(s) offline

- nmap version: 4.20
- xml output version: 1.01
- nmap.xsl version: 0.9b

Timing Policies

Category	initial_rtt_timeout	min_rtt_timeout	max_rtt_timeout	max_parallelism	scan_delay	max_scan_delay
T0 / Paranoid	5 min	Default (100 ms)	Default (10 sec)	Serial	5 min	Default (1 sec)
T1 / Sneaky	15 sec	Default (100 ms)	Default (10 sec)	Serial	15 sec	Default (1 sec)
T2 / Polite	Default (1 sec)	Default (100 ms)	Default (10 sec)	Serial	400 ms	Default (1 sec)
T3 / Normal	Default (1 sec)	Default (100 ms)	Default (10 sec)	Parallel	Default (0 sec)	Default (1 sec)
T4 / Aggressive	500ms	100ms	1,250ms	Parallel	Default (0 sec)	10ms
T5 / Insane	250ms	50ms	300ms	Parallel	Default (0 sec)	5ms