DESCRIPTION
Nmap est conçu pour permettre aux administrateurs système
ainsi qu'aux personnes curieuses de scanner de larges réseaux pour
déterminer quels sont les hôtes connectés et quels services ils offrent.
Nmap supporte un grand nombre de techniques de scan tel que:
UDP, TCP connect(), TCP SYN (mi-ouvert), ftp proxy (bounce attack),
reverse-ident, ICMP (ping sweep), FIN, ACK sweep, Xmas tree, SYN sweep,
et NULL scan. Voir la section types de scan pour plus de détails.
Nmap possède aussi un certain nombre de fonctions avancés telles que
la détection de l'OS distant grâce à l'empreinte digitale TCP/IP, le
stealth scanning, le dynamic delay, le calcul
de retransmission, le scan en parallèle, la détection d'hôtes éteints
grâce au ping en parallèle, les scans avec leurres,la détection de ports
filtrés, le scan RPC direct (sans-portmap), le scan avec fragmentations
de paquets et une flexibilité dans l'écriture des cibles et des ports.

Des efforts ont été fournis pour faciliter l'utilisation
aux utilisateurs n'ayant pas d'accès root. Malheureusement,
de nombreuses interfaces de kernels (ex: raw sockets) nécessitent
les privilèges root.

Le résultat de l'exécution de nmap nous donne normalement
une liste de ports intéressants sur la machine scannée.
Nmap donne toujours le nom du service correspondant
(s'il est connu), le numéro, l'état, et le protocole. L'état
est soit 'open', 'filtered', 'unfiltered'. Open signifie que
la cible acceptera - 'accept()' - des connexions sur ce port.
Filtered signifie qu'il y a un firewall/filtre ou un obstacle
qui couvre ce port et qui empêche nmap de déterminer s'il
est ouvert ou non. Unfiltered signifie que le port est connu par nmap
pour être fermé et qu'aucun firewall/filtre n'apparaît interférer avec
nmap. Les résultats 'unfiltered' sont courants et sont affichés
uniquement quand la plupart des ports sont trouvés à l'état filtrés.

En fonction des options utilisées, les caractéristiques suivantes
peuvent être affichées :
Le type d'OS.
Le séquencement TCP.
Les noms des utilisateurs sous lesquels fonctionnent les programmes
ayant ouverts des ports.
Le nom DNS.
Si l'hôte est une adresse de smurf.
...

OPTIONS
Les options dont la combinaison paraît logique peuvent
généralement être utilisées ensemble dans nmap. Certaines
options sont spécifiques à certains modes de scans.
nmap essaie de prévenir quand vous spécifiez des

options mal placées ou ne fonctionnant pas.
Si vous êtes impatients vous pouvez passer directement
à la section exemples, à la fin, elle vous montre les
utilisations les plus fréquentes. Vous pouvez aussi
taper nmap -h pour avoir une référence des options.

Types de scan

-sT
TCP connect() scan:
C'est le type le plus basique des scans TCP. L'appel
système connect() est utilisé pour ouvrir un port sur l'hôte distant.
Si le port est en écoute, connect() fonctionnera, autrement,
le port ne sera pas accessible. Un gros avantage de cette technique est
qu'elle ne nécessite pas les privilèges root. N'importe quel utilisateur
sur la plupart des UNIX est libre d'utiliser cette option.

Ce type de scan est, par contre, facilement détectable.
Les logs du système montreront des connections et des
erreurs résultantes au service qui accept() les connections.
En effet, elles seront rapidement arrêtées par le serveur suite au
connect() car le client n'enverra pas d'autres demandes pour cette
connexion ouverte.

-sS
TCP SYN scan:
Cette commande fait souvent référence à la technique de
scan "mi-ouvert" (half open), car vous n'ouvrez pas
une connexion TCP complète. Vous envoyez un paquet avec
le flag SYN, comme pour faire la requête d'une connexion
normale et vous attendez la réponse.
Si vous recevez comme réponse un SYN/ACK, c'est que
le port est en écoute, donc ouvert.
Un paquet avec le flag RST signifie que le port n'est
pas en écoute. Si un SYN/ACK est reçu, alors quelques
secondes plus tard vous recevrez un RST (le kernel de
l'OS fait cela pour nous) . Cela ne sera pas le cas si
vous avez répondu au précèdent SYN/ACK.
Le premier avantage de ce type de scan est qu'il
ne sera pas la plupart du temps détecté par la cible.
Malheureusement, les privilèges root seront nécessaires.

-sF -sX -sN
Stealth (scan de type "mi-ouvert") FIN, Xmas Tree,
et le mode de NULL scan : certaines fois, le scan de type SYN
n'est plus suffisamment clandestin. Certains
firewall, IDS, filtreur de paquets repèrent
ce type de scans sur des ports restreints. Des programmes
tel que Synlogger et Courtney sont connus pour cela.
D'un autre côté, ces types de scans auront de moins bon

résultats, et pourront passer à côté de ports ouverts.

L'idée est que les ports fermés doivent nous répondre par un RST, alors que ceux ouverts ne doivent pas prendre en compte le paquet en question (RFC 793 pp 64).
Le scan FIN utilise un paquet avec le fag FIN, alors que l'Xmas lui utilise la combinaison des flags FIN, URG et PUSH. Le scan NULL ne met aucun flag.
Comme d'habitude, microsoft ne respecte pas du tout les règles et fait à sa manière. A cause de cela, ces types de scans ne fonctionneront pas face à des machines de type windows95/NT.

D'une autre manière, c'est un bon moyen pour faire la distinction entre des machines windows ou non. Si le scan trouve des ports ouverts, vous pouvez alors être sûr que ce n'est pas un windows. Si un scan de type -sF, -sX ou encore -sN vous montre tous les ports fermés, et qu'avec un -sS il y en a d'ouverts, alors vous pouvez être quasiment sûr que c'est un windows. Ceci est moins utile toutefois, depuis que nmap possède un module de "remote OS identification" par le "TCP/IP fingerprinting". Windows n'est pas non plus le seul à travailler de cette manière.
Cisco, BSDI, HU/UX, MSV et IRIX renvoyent aussi des RST depuis leurs ports ouverts alors qu'ils devraient simplement jeter les paquets et ne pas en tenir compte.

-sP
Ping scanning : quelques fois, vous avez besoin de connaître uniquement si tel ou tel hôte est là ou non. Nmap fait cela en envoyant une requête 'ICMP echo' à chaque IP de la liste. Les hôtes qui répondent sont présents. Malheureusement, certains sites tel que microsoft.com ne respectent pas cette règle et bloquent l'ICMP en entrée. De cette manière nmap ne peut envoyer que des paquets avec des TCP ack en direction du port 80 (par défaut).Si nous avons un RST en réponse alors l'hôte est présent. Une autre technique est d'envoyer un paquet SYN et d'attendre un RST ou SYN/ACK. Les utilisateurs n'ayant pas le root, utiliseront la méthode connect().

Par défaut (pour les utilisateurs root) nmap utilise à la fois la technique ICMP et ACK en parallèle. Vous pouvez changer cela grâce à l'option -P décrite plus bas.

Il faut savoir que le ping est effectué par défaut, et qu'uniquement les hôtes qui répondent sont scannés. Utilisez cette méthode uniquement si vous voulez faire un ping sweep sans aucun port scan.

-sU UDP scan: Cette méthode est utilisée pour connaître tous les ports UDP (User Datagram Protocol, RFC 768) ouverts sur la

machine. La technique consiste à envoyer un paquet UDP de 0 octets sur chaque ports de la machine. Si nous recevons un message "ICMP port unreachable", alors le port est fermé. Autrement, nous considèrerons qu'il est ouvert.

Certaines personnes considèrent que le scan UDP est inutile.
J'ai pour habitude de leur rappeler un bug récent sur
Solaris concernant rpcbind.
Rpcbind peut être caché quelque part sur un port UDP
au dessus de 32770. Donc si le port 111 est
bloqué par le firewall, ce n'est pas grave; Vous pouvez chercher
sur les 30 000 ports plus hauts.
Avec un scanner UDP c'est possible. C'est aussi
le cas du cheval de troie du cDc (BackOrifice) qui est en écoute
sur un port UDP configurable. Il n'est pas nécessaire de mentionner
non plus les services très couramment vulnérables qui écoutent aussi
sur de l'UDP tel que snmp, tftp, NFS, etc...

Malheureusement, le scanning UDP est quelques fois extrêmement
lent depuis que quelques hôtes implémentent une suggestion de
la RFC 1812 (section 4.3.2. ) qui limite la vitesse de transmission
des messages d'erreur ICMP. Par exemple, le Kernel de Linux limite
les messages "destination unreachable" (dans net/ipv4/icmp.h) à 80
toutes les 4 secondes, avec 1/4 de seconde de pénalité si cela excède.
Solaris possède des limites encore plus strictes (à peu près 2 messages
par seconde) et donc cela prend encore beaucoup plus de temps
pour les scanner.
nmap détecte cette limite et ralentie en s'accordant
à celle-ci, plutôt que d'innonder le réseau avec des paquets
qui seront ignorés par la cible.

Comme d'habitude, Microsoft ignore la suggestion de la RFC
et donc ne fait pas de limitation, ce qui à pour avantage
de pouvoir scanner d'un trait les 64K port d'une machine
de type windows. Wahoo!!

-sR RPC scan. Cette méthode fonctionne avec la plupart des
combinaisons de scan. Il prend les divers ports TCP et UDP et
les inondent sous des commandes SunRPC NULL avec espoir
de découvrir s'ils sont des ports RPC et si oui, il essaie de
savoir quel est le programme derrière et sa version.
De cette manière vous pouvez obtenir les mêmes informations que
'rpcinfo -p' même si la cible est derrière un firewall (ou si
elle possède un wrapper). Decoy (voir ci-dessous) ne fonctionne
pas avec le scan RPC, mais, un jour viendra ou je l'ajouterai.

-b <ftp relay host>
FTP bounce attack : Une fonctionnalité intéressante du protocole

ftp (RFC 959) est le support du proxy pour les connexions ftp.
En d'autres mots, je dois être capable de me connecter depuis
evil.com sur le serveur ftp de cible.com et de demander à celui-ci
d'envoyer un fichier N'IMPORTE où sur Internet! En fait cela
devait bien fonctionner en 1985 quand la RFC a été écrite.
Mais sur l'Internet d'aujourd'hui, il n'est plus possible
que des gens 'hijack' (=détournent) des serveurs ftp pour disperser
les données à travers l'Internet. Comme *Hobbit* l'a écrit en
1995, cette faiblesse du protocole "peut permettre de poster
des mails, news totalement intraçables, remplir jusqu'à
saturation des disques, et généralement être gênant et
être plutôt difficiles à tracer". Nous allons utiliser
donc cette faille pour, "surprise, surprise...", scanner
des ports TCP depuis un serveur ftp "proxy". De cette
manière, nous pouvons scanner des ports qui sont
généralement bloqués (139 est un bon choix). Si le
serveur ftp permet la lecture, écriture dans un
répertoire (ex: /incoming), alors, il vous est possible
d'envoyer des données sur les ports trouvés ouverts
(nmap ne fait pas cela pour vous).

Les arguments donnés à l'option 'b' sont les serveurs que
vous souhaitez utiliser comme proxy, en utilisant
la notation standard d'URL. Le format est:
pseudo:pass mot@serveur:port. Tout sauf serveur
est optionnel. Pour déterminer quels sont les hôtes vulnérables,
vous pouvez lire mon article dans Phrack 51. Et une version
plus récente est présente sur le site de nmap
(http://www.insecure.org/nmap/).

OPTIONS GENERALES
Aucunes d'entre elles ne sont obligatoires, mais elles
sont parfois très utiles.

-P0 N'essaie pas de "pinger" les hôtes avant de les scanner.
Ceci permet de scanner des réseaux qui n'accèptent pas les requêtes
(ou réponses) ICMP echo à traverser leur firewall.
Microsoft est un exemple qui utilise ce type de filtrage dans
son réseau, donc il vous faudra utiliser -P0 ou encore
-PT80 si vous voulez le scanner.
Mais de toute manière Microsoft n'est qu'une petite
crotte de nez.

-PT Utilise la technique de ping TCP pour repérer
les hôtes qui sont présents.
A la place d'envoyer des requêtes ICMP echo et d'attendre
une réponse, nous envoyons des paquets TCP ACK à travers
le réseau (ou machine) cible et attendons les réponses.
Les hôtes présents doivent répondre par un RST.

Ceci nous permet de passer à travers des réseaux
n'autorisant pas le ping. Les utilisateurs n'ayant
pas d'accès root utiliserons connect(). Pour spécifier
le port destination pour faire sa requête, nous
utilisons -PT<numerodeport>. Le port par défaut
est le 80 car c'est celui qui est le moins souvent filtré.

-PS Cette option utilise les paquets SYN
(requête de connexion) à la place des paquets ACK
(pour les utilisateurs root). Les hôtes présents doivent
répondre par un RST, et (très rarement par un SYN|ACK).

-PI Cette option utilise le vrai ping (requêtes ICMP echo).
Il regarde quels sont les hôtes présents et cherche
des adresses qui autorisent les broadcasts de sous-réseaux sur
votre réseau. Ce sont des adresses IP extrêmement accessibles
et qui transmettent les paquets entrants en des broadcasts en
direction des ordinateurs du sous réseau.
Si des adresses de ce type sont trouvées, elles doivent aussitôt
être éliminées car elles permettent de nombreux types de dénie de
service (DoS), tel que les attaques Smurf.

-PB Ceci est le ping utilisé par défaut dans nmap. Il utilise
à la fois le paquet ACK (-PT) et ICMP (-PI)
en parallèle. De cette manière vous pouvez atteindre à la fois
des réseaux qui en filtrent un (mais pas les deux).

-O Cette option active l'authentification par l'empreinte
TCP/IP. En d'autres mots, il utilise quelques techniques pour
repérer des subtilités de la couche réseau de l'ordinateur
scanné. Il utilise les informations recueillies "empreinte
digitale" pour les comparer à une base de donnée d'empreintes
connues (le fichier nmap-os-fingerprints) et peut ainsi
voir quel type de système nous sommes en train de scanner.

Si vous trouvez une machine mal diagnostiquée et qui a au
minimum un port ouvert, il serait très utile que vous
m'envoyez les détails (du type: OS blabla version xxx
was detected as OS bloblo version ZZZ). Si vous trouvez
une machine avec au moins un port d'ouvert, et sur laquelle
nmap répond "unknown operating system", alors, il serait
utile que vous envoyez l'adresse IP avec l'OS et la version
de celui-ci. Si vous ne pouvez envoyer l'adresse IP, la
dernière bonne chose que vous puissiez faire est d'envoyer
le résultat de l'execution de nmap sur l'hôte donné
avec l'option -d (ceci doit vous donner trois fichiers)
ainsi que le système d'exploitation et sa version. En faisant
cela, vous contribuez à augmenter la base de données

d'empreintes connues par nmap ce qui sera mieux
pour tout le monde.

-I Met en action la fonction 'TCP reverse ident scanning'.
Comme l'a écrit en 1996 Dave Goldsmith dans un post sur Bugtraq,
le protocole ident (rfc 1413) permet de détecter le nom de
l'utilisateur qui fait fonctionner n'importe quel processus
connecté en utilisant TCP, même si ce processus n'a pas
initialisé la connexion. Donc vous pouvez, de cette manière
vous connecter sur le port 80 et utiliser identd pour trouver
si ce serveur est lancé par le root ou non. Ceci ne peut
être fait uniquement que par une connexion complète à
l'hôte distant (c'est à dire: l'option de scan -sT).
Quand -I est utilisé, le serveur identd de l'hôte distant
questionne chaque ports ouvert. Naturellement, ceci ne
fonctionne pas si l'hôte ne fait pas fonctionner identd.

-f Cette option force les scans par SYN, FIN, XMAS,
ou encore NULL à utiliser de tous petits fragments de
paquets IP. L'idée est de fragmenter le header TCP sur
plusieurs paquets pour pouvoir rendre plus compliqué
aux filtreurs de paquets ainsi qu'au détecteurs d'intrusions
(IDS), de détecter ce que vous êtes en train de faire.
Attention avec cette option! Certains programmes ont du
mal à gérer les paquets fragmentés. Mon sniffer préféré
fait une erreur de segmentation après réception des
36 premiers octets de fragments, après réception d'un
paquet de 24 octets ! Attention, ces paquets ne sont pas
repérés par les sniffers de paquets ainsi que par
les firewalls qui n'utilisent pas l'option du kernel
linux CONFIG_IP_ALWAYS_DEFRAG, et certains réseaux ne
peuvent pas supporter la nombre de hits que
cause ce type de scan.
Il faut noter que l'option ne fonctionne pas encore
correctement sur certains systèmes. Elle fonctionne
bien sur Linux, FreeBSD, et OpenBSD, certaines personnes
ont aussi reporté un succès sur certains *NIX.

-v Mode détaillé. Cette option est fortement
conseillée, car elle donne des informations sur ce qui
est en train de se passer. Vous pouvez l'utiliser
en double pour un plus grand effet. Attention, ne tapez pas
-d -d -d..... sinon les informations détaillées apparaîtront
autant de fois qu'il y a de 'd' sur la ligne de commande.

-h Option pratique pour afficher une rapide
référence des options de nmap. Comme vous pouvez le
remarquer ce fichier man n'est pas une 'référence rapide' 

-oN <fichierdelog>

Ceci crée un log du résultat et l'inscrit dans un
fichier lisible humainement.

-oM <fichierdelog>
Ceci crée un log du résultat dans un fichier, spécifié en
argument, au format utilisable par un ordinateur.
Vous pouvez mettre l'argument '-' (sans les guillemets) pour
envoyer le résultat vers stdout (pour une utilisation avec
des pipes sous shell, etc). Dans ce cas l'affichage normal
sera supprimé. Faites tout de même attention aux messages
d'erreurs si vous utilisez cela (ils seront toujours envoyés
sur stderr).

-oS <fichierdelog>
C3c1 l0G l3 R3suLt4 2oUs uN f0Rm4t à L4 scr1p|
kiDd|3 d4n2 uN fiCh1e5 9ue v0u2 spEc1f1e2 eN
4r9umEn|! v0U2 P0uVeZ util12er l'4rgum2n| '-'
(Z4nZ leZ gu1lleme7s) pouR tou| eNV0y2r sUr stDouT!@!!

-iL <nomdufichierentrant>
Récupère la spécification des cibles dans un fichier plutôt
que dans la ligne de commande. Ce fichier doit contenir
une liste d'expressions d'hôtes ou réseaux séparés par
des espaces, tabulations, ou encore des retours chariots.
Utilisez le signe '-' comme 'nomdufichierentrant' si
vous souhaitez que nmap lise les spécifications
sur stdin (comme à la fin d'un 'pipe'). Voir la section
Expression de cible pour plus
d'informations sur les expressions à utiliser dans ce
fichier.

-iR
Cette option dit à nmap de générer son propre fichier
d'hôtes à scanner, par simple utilisation de nombres
aléatoires  . Cela n'arrètera jamais. Cette option peut
être utile pour faire des statistiques sur divers sujets.
Si vous vous ennuyez réellement essayez d'utiliser:
nmap -sS -iR -p 80
pour trouver des serveur web.

-p <port champ>
Cette option spécifie quels ports doivent être scannés.
Par exemple '-p 23' n'essayera que le port 23 sur la
machine cible. Le défaut est de scanner tout les ports
entre 1 et 1024 et tous ceux présents dans le fichier
'services' livré avec nmap.

-F Méthode de scan rapide.
Spécifie que vous ne voulez scanner que les ports
listés dans le fichier 'services' fournis avec nmap. C'est de

toute manière bien plus rapide que de scanner les 65535 ports cibles.

-D <decoy1 [,decoy2][,ME],...>
Effectue un scan avec leurres (decoy), ce qui permet de faire croire à l'hôte cible que les IP que vous spécifiez en tant que 'decoy' sont aussi en train de le scanner. De cette manière, les IDS détecteront quelques fois 5-10 ports scan provenant de la même source, et seront incapables de dire qui à réellement effectué le scan et quel sont les leurres envoyés. Tandis que cette technique ne fonctionne pas avec des routeurs traçant la route, ou encore des mécanismes dits 'actifs', cette technique est très puissante pour cacher son IP.

Séparez chaque leurre (decoy) avec des virgules, vous pouvez optionnellement utiliser 'ME' en tant que leurre pour spécifier à quelle position vous souhaitez que votre IP soit utilisée. Si vous placez le 'ME' en sixième position ou plus tard, de nombreux détecteurs de port-scan (tels que l'excellent Solar Designer's scanlogd) seront incapables de montrer votre IP. Si vous n'utilisez pas le 'ME', nmap le positionnera aléatoirement.

Il faut noter que les hôtes que vous utilisez en tant que leurres doivent être présents si vous ne voulez pas créer une inondation SYN sur votre cible. De plus, il sera plus aisé de repérer qui est en train de scanner s'il ne se trouve qu'un seul hôte présent sur le réseau. Vous pouvez utiliser des IP plutôt que des noms (si vous ne souhaitez que votre IP n'apparaisse dans le fichier de log du serveur DNS de vos leurres).

Il faut aussi noter que certains "détecteurs de port scan" (stupides) refusent le routage avec des hôtes qui essayent de les scanner. De cette manière vous pouvez causer sur cette machine le refus de dialoguer avec les hôtes que vous spécifiez en tant que leurres. Cela peut poser des problèmes majeurs si vous spécifiez une machine du réseau leur passerelle internet, ou encore le "localhost".
De cette manière il faut faire attention avec cette option. La réelle morale de cette histoire est que les détecteurs de scan ne devraient pas prendre d'action lorsqu'ils détectent quelque chose car cela pourrait très bien être un leurre!

Le scan avec leurres peut être utilisé à la fois avec le ping scan (utilisant ICMP, SYN, ACK, ou n'importe quoi) et aussi durant la phase de port scan. Les leurres peuvent aussi être utilisés durant l'authentification d'OS (-O).

Il est inutile et même c'est une peine perdue d'utiliser trop de
leurres cela ne sert à rien et pourra rendre les réponses
moins justes. A l'heure actuelle uniquement quelques FAIs filtrent les
paquets spooffés.

-S <AdressesIP>
Dans certain cas, nmap sera incapable de trouver
votre adresse IP source (il vous le dira si c'est le cas).
Si cela vous arrive, utilisez -S avec votre adresse IP
(l'interface sur laquelle vous voulez envoyer les paquets).

Une autre utilité de cette option est de faire un spoofed
scan pour faire croire à la cible que quelqu'un d'autre est en train
de la scanner. Imaginez une entreprise se faisant souvant
scanner par un concurant! Je ne l'ai pas créé dans ce but,
mais plutôt pour montrer le résultat et les problèmes que
cela pouvait causer. -e sera généralement nécessaire
dans avec ce type d'usage.

-e <interface>
Spécifie à nmap l'interface sur laquelle les paquets
doivent êtres envoyés. Nmap devrait être capable de
détecter l'interface à utiliser, mais au cas ou il n'y arriverait pas,
il vous le dira.

-g <portsource>
Fixe le port source utilisé dans les scan.
De nombreuses installations de firewall et filtreurs de
paquets feront une exception dans leurs règles aux paquets
DNS (53) ou encore FTP-DATA (20) qui entreront pour
effectuer une connexion. Naturellement pour un scan UDP
il faudra mieux utiliser le port 53 et pour un scan TCP,
le port 20 avant le 53. Il faut noter que ceci n'est qu'une
requête et que nmap ne l'utilisera uniquement que si cela
est possible.
Par exemple, vous ne pouvez pas faire d'échantillonage TCP ISN
d'un hôte:port vers un hôte:port, car nmap change le port source
même si vous utilisez l'option -g.

Il faut aussi savoir qu'il y a des inconvénients à utiliser
cette option sur certains scan car nmap place des informations
utiles dans les ports sources.

-r Demande à nmap de ne PAS placer les ports scannés
dans un ordre aléatoire.

--randomize_hosts
Demande à nmap de mélanger l'ordre de 2048 hôtes

avant de les scanner. Ceci peut rendre le scan plus discret
pour des moniteurs réseau, spécialement si vous utilisez
des techniques de minuterie lente (voir ci-dessous).

-M <max sockets>
Spécifie le nombre maximum de sockets qui seront utilisées
en parallèle pour un scan avec le TCP connect() (celui
par défaut). Ceci est utile pour ralentir un petit peu
le scan et pour ne pas crasher la cible. Une autre solution
est d'utiliser le -sS, car il est généralement plus facile
à gérer pour les machines.

OPTIONS DE MINUTERIE
En général nmap se débrouille pas mal du tout pour
s'ajuster aux caractéristiques du réseau en le scannant
pour pouvoir aller le plus vite possible tout en minimisant
les chances de ne pas détecter des hôtes/ports. Seulement,
certaines fois, la politique de chronométrage de nmap, ne
représente pas les objectifs voulu. Les options suivantes
permettent de contrôler totalement ce temp:

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>
Le mode Paranoid scan très lentement dans
l'espoir d'éviter les IDS et détecteurs de scan.
Il fait les scans en série (et non pas en parallèle) et
attend au moins 5 minutes entre chaque paquet. Sneaky est
similaire, il attend seulement 15 secondes entre chaque.
Polite est conçu pour ne pas surcharger le réseau
et crasher les machines. Il fait les scans en série et
attend au moins 0.4 secondes entre chaque
paquets. Normal est l'attitude normale de nmap,
en ce sens qu'il essaie de fonctionner le plus vite
possible sans pour autant perdre en précision dans
les résultats. Les mode Agressive ajoute
5 minutes de timeout par hôte et n'attend pas plus
de 1.25 seconde pour les réponses des requêtes.
Insane n'est convenable que pour des réseaux très
rapides ou s'il ne vous importe pas de perdre
des informations. Il met un timout pour les hôtes
de 75 secondes et attend seulement 0.3 secondes pour
les requêtes individuelles. Il permet de scanner
des réseaux très rapidement  . Vous pouvez aussi faire
vos spécifications avec des nombres (0-5). Par exemple,
'-T 0' vous donne le mode paranoïd et '-T 5' Insane.

Ces types de scan ne doivent pas être utilisés en
combinaison avec les contrôles ayant un niveau inférieur
situés ci-dessous.

--host_timeout <millisecondes>

Spécifie le total de temps que doit passer Nmap à scanner
un hôte avant de laisser tomber avec celui-ci. Le mode
de minuterie par défaut n'utilise pas cette option.

--max_rtt_timeout <millisecondes>
Spécifie le temps maximum que Nmap est autorisé pour
attendre les réponses aux requêtes ou retransmettre
cette requêtes particulière. Le mode par défaut est
9000.

--min_rtt_timeout <millisecondes>
Quand l'hôte cible commence à envoyer quelques réponses
au requêtes précédentes très rapidement, Nmap ajustera
le temps total pour chaque requête. Ceci accélère
le scan, mais peut laisser de côté des réponses un peu
trop lentes à arriver. Avec ce paramètre vous pouvez être
garantit que nmap attendra au moins le minimum de temps
possible pour une requête.

--initial_rtt_timeout <millisecondes>
Spécifie le time-out pour la requête initiale. Ceci est
généralement utile quand l'on scanne des réseaux
protégés par un firewall en utilisant l'option '-P0'.
Normalement Nmap peut obtenir une bonne estimation de RTT
grâce au ping et les première requêtes. Le mode par défaut
utilise 6000.

--max_parallelism <nombre>
Spécifie quel est le nombre maximum de scan que Nmap est
autorisé à effectuer en parallèle. En mettant ceci à 1 nmap ne
scannera pas plus d'un port à la fois. Il affecte aussi
d'autre types de scan en parallèles tel que le ping sweep,
RCP scan, etc...

--scan_delay <milliseconds>
Spécifie le temps minimum à nmap pour attendre
chaque requête. Ceci est utile pour limiter la
charge du réseau ou pour ralentir le scan en ayant
pour but d'être moins facilement repéré.

SPECIFICATION DE CIBLE
Tout ce qui n'est pas une option (ou un argument d'option)
dans nmap est traité comme une spécification de cible.
Le cas le plus simple est d'être à l'écoute de simples noms
d'hôtes ou adresses IP sur la ligne de commande. Si vous
souhaitez scanner un réseau, vous devriez utiliser
'/mask' en l'ajoutant au nom d'hôte ou à l'adresse IP.
Mask doit être entre 0 (scan tout l'Internet)
et 32 (scan un simple ordinateur). Utilisez /24 pour scanner
un réseau de class 'C' et /16 pour un de class 'B'.

Nmap a aussi une technique de notation puissante qui vous permet de spécifier les adresses IP en utilisant pour chaque élément des listes/champs. De cette manière vous pouvez décrire un réseau de class 'B' 128.210.0.0 en spécifiant '128.210.*.*' ou '128.210.0-255.0-255' ou encore '128.210.1-50,51-255.1,2,3,4,5-255'. Et bien sûr, vous pouvez utiliser la notation suivante avec mask: '128.210.0.0/16'. Ils sont tous équivalent. Si vous utilisez des astérisques ('*'), il faut juste se souvenir que de nombreux shells ont besoins de guillemets et de séquences d'échapements. Une autre chose intéressante est de diviser l'internet d'une manière différente: Au lieu de scanner les hôtes d'une classe 'B', une spécification du type '*.*.5.6-7' effectuera un scan sur toute les IP se terminant par .5.6 ou .5.7, choisissez vos propres numéros. Pour plus d'informations, sur les spécifications à utiliser pour scanner des hôtes, il faudra se reporter à la section des exemples.

EXAMPLES
Vous trouverez ici quelques exemples d'utilisation de nmap, en partant des plus simples en passant par les normaux et en terminant par les plus complexes et ésotériques. Il faut noter que certains numéros et domaines ont été choisis pour rendre cela plus concret. Dans ces parties vous devez remplacer les noms/adresses par VOS PROPRES PARAMETRES RESEAU. Je ne pense pas que faire des scannage de ports d'autres réseaux est illégal et qu'ils doivent être forcément interprétés comme des attaque. J'ai scanné des centaines de milliers de machines et reçu uniquement une seule plainte. Mais je ne suis pas juge et certaines personnes (trou du cul) peuvent être ennuyées par les requêtes de nmap. Demandez premièrement l'autorisation ou utilisez-le à vos propres risques.

nmap -v example.cible.com

Cette option scan tous les ports TCP sur la machine example.cible.com. Le -v signifie mode 'verbose', c'est à dire: bavard, détaillé.

nmap -sS -O example.cible.com/24

Lance un SYN scan sur le réseau de classe 'C' ou exemple.cible.com réside. Il essaie aussi de récupérer l'identification du système sur chacuns des hôtes qu'il trouve grâce au TCP fingerprinting. Ceci nécessite les droits root du fait du SYN scan et de l'identification de l'OS.

nmap -sX -p 22,53,110,143,4564 128.210.*.1-127

Envoie un scan de type Xmas sur la première moitié des
255 possibilités dans le réseau de class 'B' 128.210.
Nous testons si le système possède sshd, DNS, pop3, imapd,
ou le port 4564. Il est bon de savoir que Xmas ne fonctionne
pas contre les machines de type windows du fait e d'une
malformation dans leur pile TCP. Même chose pour CISCO,
IRIX, HP/UX, et BSDI.

nmap -v --randomize_hosts -p 80 '*.*.2.3-5'
Au lieu de se focaliser sur une class IP, il est
parfois intéressant de découper l'internet en plusieurs
partie que l'on scanne en tranches. Cette commande trouve
tous les serveurs webs sur les machines dont l'IP se
termine par .2.3, .2.4 ou .2.5. Si vous êtes root, vous
devrez ajouter -sS. De même vous trouverez des machines
plus intéressantes en commençant par 127.

Donc vous pourriez utiliser '127-222' à la place des astérisques
car cette section possède un plus grand nombre de
machines intéressantes.

host -l company.com | cut '-d ' -f 4 | ./nmap -v -iL -

Effectue un transfert de zone pour trouver les hôtes de
'company.com', envoie le résultat à nmap. Les commandes
ci-dessus sont pour mon GNU/Linux. Vous pouvez avoir différentes
commandes sur votre OS.

BUGS
Des Bugs? Quels bugs? Envoyez moi ce que vous trouvez.
Les patches sont sympas aussi  N'oubliez pas d'envoyer
les nouveaux fingerprints pour grossir la base de donnée.
Nmap vous donnera une URL de soumission dans le cas d'un
fingerprint approprié trouvé.

AUTEUR
Fyodor <fyodor@dhp.com>

TRADUCTEUR
Hertz <hertz@webmails.com>
Texte traduit le 18 Avril 2000.
Merci à <blured@linuxmail.org>
et <Guilux> pour leur aide à la correction.

NOTE DU TRADUCTEUR
Cette traduction, n'est en aucun cas la meilleure
traduction qui puisse être de ce texte. Si vous

trouvez des erreurs, fautes... veuillez me contacter
pour que je puisse mettre à jour le document et
que tout le monde puisse profiter de ces correctifs.
En ce qui concerne le chapitre 'DISTRIBUTION', je
dois en aucun cas être tenu responsable d'une mauvaise
interprétation du texte. Si vous êtes intéressé par
celui-ci alors veuillez consulter le document en Anglais.

DISTRIBUTION
Toute nouvelle version de Nmap peut être trouvée sur
http://www.insecure.org/nmap/

Les sources sont fournies avec ce logiciel car nous
sommes persuadés que l'utilisateur à le droit de savoir
ce que va faire son programme avant même son exécution.
Cela vous permet de la même manière de faire un audit
sur le logiciel, à la recherche de failles de sécurité
(aucune n'a encore à ce jour été trouvée).

Le code source vous permet aussi de porter nmap sur
de nouvelles plateformes, de corriger les bugs, ou
encore d'ajouter de nouvelles fonctionnalités. Vous
êtes fortement encouragés à envoyer vos modification
à Fyodor, pour pouvoir les incorporer à la distribution
de Nmap. Par l'envoi de ces changements à Fyodor, ou
nmap-hackers, il faut assumer que vous lui offrez le code
et qu'il pourra être sans exclusivité, réutilisé, modifié,
et licensé. Si vous voulez spécifier un type de license
particulier, vous pouvez l'indiquer tout en haut de votre
code source.

Ce programme est distribué dans l'espoir qu'il sera utile,
mais il est fournit SANS AUCUNE GARANTIE;
sans même la garantie implicite d'être COMMERCIALISABLE ou
même de pouvoir CONVENIR A UNE
APPLICATION PARTICULIERE. Consulter la
'GNU Public License' pour plus de détails (elle se trouve dans
le fichier COPYING de la distribution de Nmap.

Il est aussi nécessaire de savoir que Nmap à été connu
pour crasher certaine applications mal programmées,
piles TCP/IP, et même des OS. Nmap ne devrait jamais
être lancé contre des serveurs à haut risque sauf si
vous êtes prêt à ce qu'il souffre d'indisponibilité.
Nous approuvons le fait que nmap peut crasher des systèmes
ou réseaux et nous déclinons toute responsabilité sur les
dommages ou problèmes que Nmap peut causer.

Toutes les versions de Nmap supérieures ou égales à 2.0
sont connues pour être compatibles AN 2000 (Y2K). Cela
à été dit, mais Nmap n'est fournit avec aucune garantie.
Il n'y a pas de raison que les versions précédentes à la
2.0 soient sujettes à des problèmes, mais nous n'avons
jamais testé.