Guide de référence Nmap (Man Page, French translation)

Table of Contents

Name

nmap — Outil d'exploration réseau et scanneur de ports/sécurité

`nmap [ Types de scans ...] [ Options ] { spécifications des cibles }`

Description

Nmap (“Network Mapper”) est un outil open source d'exploration réseau et d'audit de sécurité. Il a été conçu pour rapidement scanner de grands réseaux, mais il fonctionne aussi très bien sur une cible unique. Nmap innove en utilisant des paquets IP bruts (raw packets) pour déterminer quels sont les hôtes actifs sur le réseau, quels services (y compris le nom de l'application et la version) ces hôtes offrent, quels systèmes d'exploitation (et leurs versions) ils utilisent, quels types de dispositifs de filtrage/pare-feux sont utilisés, ainsi que des

douzaines d'autres caractéristiques. Nmap est généralement utilisé pour les audits de sécurité mais de nombreux gestionnaires des systèmes et de réseau l'apprécient pour des tâches de routine comme les inventaires de réseau, la gestion des mises à jour planifiées ou la surveillance des hôtes et des services actifs.

Le rapport de sortie de Nmap est une liste des cibles scannées ainsi que des informations complémentaires en fonction des options utilisées. L'information centrale de la sortie est la “table des ports intéressants”. Cette table liste le numéro de port et le protocole, le nom du service et son état. L'état est soit `ouvert (open)`, `filtré (filtered)`, `fermé (closed)` ou `non-filtré (unfiltered)`. `Ouvert` indique que l'application de la machine cible est en écoute de paquets/connexions sur ce port. `Filtré` indique qu'un pare-feu, un dispositif de filtrage ou un autre obstacle réseau bloque ce port, empêchant ainsi Nmap de déterminer s'il s'agit d'un port `ouvert` ou `fermé`. Les ports `fermés` n'ont pas d'application en écoute, bien qu'ils puissent quand même s'ouvrir n'importe quand. Les ports sont considérés comme `non-filtrés` lorsqu'ils répondent aux paquets de tests (probes) de Nmap, mais Nmap ne peut déterminer s'ils sont ouverts ou fermés. Nmap renvoie également les combinaisons d'états `ouverts|filtré` et `fermés|filtré` lorsqu'il n'arrive pas à déterminer dans lequel des deux états possibles se trouve le port. La table des ports peut aussi comprendre des détails sur les versions des logiciels si la détection des services est demandée. Quand un scan du protocole IP est demandé (`-sO`), Nmap fournit des informations sur les protocoles IP supportés au lieu de la liste des ports en écoute.

En plus de la table des ports intéressants, Nmap peut aussi fournir de plus amples informations sur les cibles comme les noms DNS (reverse DNS), deviner les systèmes d'exploitation utilisés, obtenir le type de matériel ou les adresses MAC.

Un scan classique avec Nmap est présenté dans Example 1, “Un scan Nmap représentatif” . Les seuls arguments de Nmap utilisés dans cet exemple sont `-A` , qui permet la détection des OS et versions de logiciels utilisés, `-T4` pour une exécution plus rapide, et les noms d'hôte des cibles.

Example 1. Un scan Nmap représentatif

```
# nmap -A -T4 scanme.nmap.org playground

Starting nmap ( http://www.insecure.org/nmap/ )
Interesting ports on scanme.nmap.org (205.217.153.62):
(The 1663 ports scanned but not shown below are in state: filtered)
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 3.9p1 (protocol 1.99)
53/tcp  open   domain
70/tcp  closed gopher
80/tcp  open   http    Apache httpd 2.0.52 ((Fedora))
113/tcp closed auth
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.7 - 2.6.11, Linux 2.6.0 - 2.6.11
Uptime 33.908 days (since Thu Jul 21 03:38:03 2005)

Interesting ports on playground.nmap.org (192.168.0.40):
(The 1659 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn
```

```
389/tcp  open  ldap?
445/tcp  open  microsoft-ds  Microsoft Windows XP microsoft-ds
1002/tcp open  windows-icfw?
1025/tcp open  msrpc         Microsoft Windows RPC
1720/tcp open  H.323/Q.931   CompTek AquaGateKeeper
5800/tcp open  vnc-http      RealVNC 4.0 (Resolution 400x250; VNC TCP port:
5900)
5900/tcp open  vnc           VNC (protocol 3.8)
MAC Address: 00:A0:CC:63:85:4B (Lite-on Communications)
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Pro RC1+ through final release
Service Info: OSs: Windows, Windows XP

Nmap finished: 2 IP addresses (2 hosts up) scanned in 88.392 seconds
```

Résumé des options

Ce résumé des options est affiché quand Nmap est exécuté sans aucun argument; la plus récente version est toujours disponible sur http://www.insecure.org/nmap/data/nmap.usage.txt . Il sert d'aide-mémoire des options les plus fréquemment utilisées, mais ne remplace pas la documentation bien plus détaillée de la suite de ce manuel. Les options obscures n'y sont pas incluses.

Utilisation: nmap [Type(s) de scan] [Options] {spécifications des cibles}

SPÉCIFICATIONS DES CIBLES:
Les cibles peuvent être spécifiées par des noms d'hôtes, des adresses IP, des adresses de résea ux, etc.
Exemple: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0-255.0-255.1-254
-iL <inputfilename>: Lit la liste des hôtes/réseaux cibles à partir du fichier
-iR <num hosts>: Choisit les cibles au hasard
--exclude <host1[,host2][,host3],...>: Exclut des hôtes/réseaux du scan
--excludefile <exclude_file>: Exclut des hôtes/réseaux des cibles à partir du fichier

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
-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 -n/-
R: Ne jamais résoudre les noms DNS/Toujours résoudre [résout les cibles actives par défaut]

TECHNIQUES DE SCAN:
-sS/sT/sA/sW/sM: Scans TCP SYN/Connect()/ACK/Window/Maimon -
sN/sF/sX: Scans TCP Null, FIN et Xmas
--scanflags <flags>: Personnalise les flags des scans TCP
-sI <zombie host[:probeport]>: Idlescan (scan paresseux)
-sO: Scan des protocoles supportés par la couche IP
-b <ftp relay host>: Scan par rebond FTP

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

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-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)

DÉTECTION DE SYSTÈME D'EXPLOITATION:
-O: Active la détection d'OS
--osscan-limit: Limite la détection aux cibles prométeuses --osscan-
guess: Détecte l'OS de façon plus agressive

TEMPORISATION ET PERFORMANCE:
-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
--host-timeout <msec>: Délai d'expiration du scan d'un hôte --scan-delay/--max_scan-
delay <msec>: Ajuste le délai de retransmission entre deux paquets de tests

ÉVASION PARE-FEU/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
--ttl <val>: Spécifie le champ time-to-live IP
--spoof-mac <adresse MAC, préfixe ou nom du fabriquant>: Usurpe une adresse MAC

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)
--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

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
-h: Affiche ce résumé de l'aide

EXEMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sP 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -P0 -p 80

Spécification des cibles

Tout ce qui n'est pas une option (ou l'argument d'une option) dans la ligne de commande de Nmap est considéré comme une spécification d'hôte cible. Le cas le plus simple est de spécifier une adresse IP cible ou un nom d'hôte à scanner.

Si vous désirez scanner un réseau entier d'hôtes consécutifs, Nmap supporte l'adressage du style CIDR. Vous pouvez ajouter / *`numbits`* à une adresse IP ou à un nom d'hôte de référence et Nmap scannera toutes les adresses IP dont les *`numbits`* bits de poids fort sont les mêmes que la cible de référence. Par exemple, 192.168.10.0/24 scannerait les 256 hôtes entre 192.168.10.0 (en binaire: `11000000 10101000 00001010 00000000`) et 192.168.10.255 (en binaire:`11000000 10101000 00001010 11111111`) inclusivement. 192.168.10.40/24 ferait donc aussi la même chose. Étant donné que l'hôte scanme.nmap.org est à l'adresse IP 205.217.153.62, scanme.nmap.org/16 scannerait les 65 536 adresses IP entre 205.217.0.0 et 205.217.255.255. La plus petite valeur autorisée est /1 qui scanne la moitié d'Internet. La plus grande valeur autorisée est 32, ainsi Nmap ne scanne que la cible de référence car tous les bits de l'adresse sont fixés.

La notation CIDR est concise mais pas toujours des plus pratiques. Par exemple, vous voudriez scanner 192.168.0.0/16 mais éviter toutes les adresses se terminant par .0 ou .255 car se sont souvent des adresses de diffusion (broadcast). Nmap permet de le faire grâce à l'adressage par intervalles. Plutôt que de spécifier une adresse IP normale, vous pouvez spécifier pour chaque octet de l'IP une liste d'intervalles séparés par des virgules. Par exemple, 192.168.0-255.1-254 évitera toutes les adresses se terminant par .0 ou .255. Les intervalles ne sont pas limités aux octets finals: 0-255.0-255.13.37 exécutera un scan de toutes les adresses IP se terminant par 137.37. Ce genre de spécifications peut s'avérer utile pour des statistiques sur Internet ou pour les chercheurs.

Les adresses IPv6 ne peuvent être spécifiées que par une adresse IPv6 pleinement qualifiée ou un nom d'hôte. L'adressage CIDR ou par intervalles n'est pas géré avec IPv6 car les adresses ne sont que rarement utiles.

Nmap accepte les spécifications de plusieurs hôtes à la ligne de commande, sans qu'elles soient nécessairement de même type. La commande **nmap scanme.nmap.org 192.168.0.0/8 10.0.0,1,3-7.0-255** fait donc ce à quoi vous vous attendez.

Même si les cibles sont souvent spécifiées dans les lignes de commandes, les options suivantes sont également disponibles pour sélectionner des cibles :

`-iL <inputfilename>`(Lit la liste des hôtes/réseaux cibles depuis le fichier)

Lit les spécifications des cibles depuis le fichier *`inputfilename`*. Il est souvent maladroit de passer une longue liste d'hôtes à la ligne de commande. Par exemple, votre serveur DHCP pourrait fournir une liste de 10 000 baux que vous souhaiteriez scanner. Ou alors voudriez scanner toutes les adresses IP *sauf* celles des baux DHCP pour identifier les hôtes qui utilisent des adresses IP statiques non-autorisées. Générez simplement la liste des hôtes à scanner et passez ce fichier comme argument de l'option `-iL`. Les entrées peuvent être spécifiées dans n'importe quel des formats acceptés par la ligne de commande de Nmap (adresses IP, noms d'hôtes, CIDR, IPv6 ou par intervalles). Les entrées doivent être séparées par un ou plusieurs espaces, tabulations ou retours chariot. Vous pouvez utiliser un tiret (`-`) comme nom de fichier si vous souhaitez que Nmap lise les hôtes depuis l'entrée standard.

`-iR <num hosts>`(Choisit des cibles au hasard)

Pour des études à l'échelle d'Internet ou autres, vous pourriez désirer de choisir vos cibles au hasard. L'argument *`num hosts`* indique à Nmap combien d'IPs il doit générer. Les IPs à éviter, comme les plages d'adresses privées, multicast ou non allouées sont automatiquement évitées. On peut aussi utiliser l'argument `0` pour effectuer un scan sans fin. Rappelez-vous bien que certains administrateurs de réseau s'irritent lorsqu'on scanne leur réseau sans permission et peuvent porter plainte. Utilisez cette option à vos risques et périls! Un jour de pluie où vous ne savez pas quoi faire, essayez la commande **nmap -sS -PS80 -iR 0 -p 80** pour trouver des serveurs Web au hasard sur lesquels fureter.

`--exclude <host1[,host2][,host3],...>` (Exclut des hôtes/des réseaux des cibles)

Spécifie une liste de cibles séparées par des virgules à exclure du scan, même si elles font partie de la plage réseau que vous avez spécifiée. La liste que vous donnez en entrée utilise la syntaxe Nmap habituelle, elle peut donc inclure des noms d'hôtes, des blocs CIDR, des intervalles, etc. Ceci peut être utile quand le réseau que vous voulez scanner comprend des serveurs à haute disponibilité, des systèmes reconnus pour réagir défavorablement aux scans de ports ou des sous-réseaux administrés par d'autres personnes.

`--excludefile <exclude_file>` (Exclut des hôtes/des réseaux des cibles depuis le fichier)

Cette option offre les mêmes fonctionnalités que l'option `--exclude`, à la différence qu'ici les cibles à exclure sont spécifiées dans le fichier *`exclude_file`* au lieu de la ligne de commande. Les cibles sont séparées entre elles dans le fichier par des retours chariot, des espaces ou des tabulations.

Découverte des hôtes

Une des toutes premières étapes dans la reconnaissance d'un réseau est de réduire un ensemble (quelques fois énorme) de plages d'IP à une liste d'hôtes actifs ou intéressants. Scanner tous les ports de chacune des IP est lent et souvent inutile. Bien sûr, ce qui rend un hôte intéressant dépend grandement du but du scan. Les administrateurs de réseau peuvent être uniquement intéressés par les hôtes où un certain service est actif tandis que les auditeurs de sécurité peuvent s'intéresser à tout équipement qui dispose d'une adresse IP. Alors que l'administrateur se satisferait d'un ping ICMP pour repérer les hôtes de son réseau, l'auditeur pourrait utiliser un ensemble varié de douzaines de paquets de tests (probes) dans le but de contourner les restrictions des pare-feux.

Parce que les besoins de découverte des hôtes sont si différents, Nmap propose une grande panoplie d'options pour individualiser les techniques utilisées. La découverte d'hôte est souvent appelée « scan ping » (ping scan), mais celle-ci va bien au delà d'une simple requête echo ICMP associée à l'incontournable outil ping. Les utilisateurs peuvent entièrement éviter l'étape scan ping en listant simplement les cibles (`-sL`), en désactivant le scan ping(`-P0`) ou alors en découvrant le réseau avec des combinaisons de tests TCP SYN/ACK, UDP et ICMP. Le but de ces tests est de solliciter une réponse des cibles qui prouvera qu'une adresse IP est effectivement active (utilisée par un hôte ou un équipement réseau). Sur de nombreux réseaux, seul un petit pourcentage des adresses IP sont actives à un moment donné. Ceci est particulièrement courant avec les plages d'adresses privées (définies par la sainte RFC 1918) comme 10.0.0.0/8. Ce réseau comprend 16 millions d'IPs, mais il s'est déjà vu utilisé par des entreprises disposant de moins d'un millier de machines. La découverte des hôtes permet de trouver ces machines dans l'immensité de cet océan d'adresses IP.

Lorsqu'aucune option de découverte n'est spécifiée, Nmap envoie un paquet TCP ACK sur le port 80 ainsi qu'une requête d'echo ICMP à chaque machine cible. Une exception à cette règle est qu'un scan ARP est utilisé pour chaque cible du réseau Ethernet local. Pour les utilisateurs UNIX non-privilégiés, un paquet SYN est utilisé à la place du ACK en utilisant l'appel système `connect()`. Ces options par défaut sont équivalentes à la combinaison d'option `-PA -PE`. Cette méthode de découverte des hôtes est souvent suffisante lors de scans de réseaux locaux, mais un ensemble plus complet de tests de découverte est recommandé pour les audits de sécurité.

Les options suivantes contrôlent la découverte des hôtes.

`-sL` (Liste simplement)

> Cette forme dégénérée de découverte d'hôtes liste simplement chaque hôte du(des) réseau(x) spécifié(s), sans envoyer aucun paquet aux cibles. Par défaut, Nmap utilise toujours la résolution DNS inverse des hôtes pour connaître leurs noms. Il est souvent étonnant de constater combien ces simples informations peuvent être utiles. Par exemple, `fw.chi.playboy.com` est le pare-feu du bureau de Chicago de Playboy Enterprises. Nmap rend également compte du nombre total d'adresses IP à la fin de son rapport. Cette simple liste est un bon test pour vous assurer que vos adresses IP cibles sont les bonnes. Si jamais ces noms de domaines ne vous disent rien, il vaudrait mieux s'arrêter là afin d'éviter de scanner le réseau de la mauvaise entreprise.

Comme l'idée est de simplement afficher une liste des cibles, les options de fonctionnalités plus haut niveau comme le scan de ports, la détection du système d'exploitation ou la découverte des hôtes ne peuvent pas être combinées avec la liste simple. Si vous voulez juste désactiver la découverte des hôtes mais quand même effectuer des opérations de plus haut niveau, lisez sur l'option `-P0`.

`-sP`(Scan ping)

Cette option indique à Nmap de n'effectuer *que* le scan ping (la découverte des hôtes), puis d'afficher la liste des hôtes disponibles qui ont répondu au scan. Aucun autre test (comme le scan des ports ou la détection d'OS) n'est effectué. Ce scan est légèrement plus intrusif que la simple liste, et peut souvent être utilisé dans le même but. Il permet un survol d'un réseau cible sans trop attirer l'attention. Savoir combien d'hôtes sont actifs est plus précieux pour un attaquant que la simple liste de chaque IP avec son nom d'hôte.

Les gestionnaires des systèmes apprécient également cette option. Elle peut facilement être utilisée pour compter le nombre de machines disponibles sur un réseau ou pour contrôler la disponibilité d'un serveur. Cette option est souvent appelée « balayage ping » (ping sweep). Elle est plus fiable que sonder par ping l'adresse de diffusion (broadcast) car beaucoup d'hôtes ne répondent pas à ces requêtes.

L'option `-sP` envoie une requête d'echo ICMP et un paquet TCP sur le port par défaut (80). Lorsqu'exécutée par un utilisateur non-privilégié, un paquet SYN est envoyé (en utilisant l'appel système `connect()`) sur le port 80 de la cible. Lorsqu'un utilisateur privilégié essaie de scanner des cibles sur un réseau local Ethernet, des requêtes ARP (`-PR`) sont utilisées à moins que l'option `--send-ip`soit spécifiée. L'option `-sP` peut être combinée avec chacun des tests de découverte des hôtes (les options `-P*`, sauf `-P0`) pour une plus grand flexibilité. Dès qu'un test de ce type est utilisé avec un numéro de port, il est prépondérante sur les tests par défaut (ACK et requête echo). Quand des pare-feux restrictifs sont présents entre la machine exécutant Nmap et le réseau cible, il est recommandé d'utiliser ces techniques avancées. Sinon des hôtes peuvent être oubliés quand le pare-feu rejète les paquets ou leurs réponses.

`-P0` (Pas de scan ping)

Cette option évite complètement l'étape de découverte des hôtes de Nmap. En temps normal, Nmap utilise cette étape pour déterminer quelles sont les machines actives pour effectuer un scan approfondi. Par défaut, Nmap n'examine en profondeur, avec le scan des ports ou la détection de version, que les machines qui sont actives. Désactiver la détection des hôtes avec l'option `-P0`conduit Nmap à effectuer les scans demandés sur *toutes* les adresses IP cibles spécifiées. Ainsi, si une adresse IP de classe B (/16) est spécifiée à la ligne de commande, toutes les 65 536 adresses IP seront scannées. Le deuxième caractère dans l'option `-P0` est bien un zéro et non pas la lettre O. La découverte des hôtes est évitée comme avec la liste simple, mais au lieu de s'arrêter et d'afficher la liste des cibles, Nmap continue et effectue les fonctions demandées comme si chaque adresse IP était active.

`-PS [portlist]`(Ping TCP SYN)

Cette option envoie un paquet TCP vide avec le drapeau (flag) SYN activé. La destination par défaut de ce paquet est le port 80 (configurable à la compilation en changeant la définition DEFAULT_TCP_PROBE_PORT dans `nmap.h`), mais un autre port peut être spécifié en paramètre (ex.: `-PS22,23,25,80,113,1050,35000`), auquel cas les paquets de tests (probes) seront envoyés en parallèle sur chaque port cible.

Le drapeau SYN fait croire que vous voulez établir une connexion sur le système distant. Si le port de destination est fermé, un paquet RST (reset) est renvoyé. Si le port s'avère être ouvert, la cible va entamer la seconde étape de l'établissement de connexion TCP en 3 temps (TCP 3-way-handshake) en répondant par un paquet TCP SYN/ACK. La machine exécutant Nmap avortera alors la connexion en cours d'établissement en répondant avec un paquet RST au lieu d'un paquet ACK qui finaliserait normalement l'établissement de la connexion. Le paquet RST est envoyé par le noyau (kernel) de la machine exécutant Nmap en réponse au paquet SYN/ACK inattendu; ce n'est pas Nmap lui-même qui l'émet.

Nmap ne tient pas compte si le port est réellement ouvert ou fermé. Les paquets RST ou SYN/ACK évoqués précédemment indiquent tout deux que l'hôte est disponible et réceptif.

Sur les systèmes UNIX, seuls les utilisateurs privilégiés `root` sont généralement capables d'envoyer et de recevoir des paquets TCP bruts (raw packets). Pour les utilisateurs non-privilégiés, Nmap contourne cette restriction avec l'appel système connect() utilisé sur chaque port de la cible. Ceci revient à envoyer un paquet SYN sur l'hôte cible pour établir une connexion. Si connect() réussi ou échoue avec ECONNREFUSED, la pile TCP/IP sous-jacente doit avoir reçu soit un SYN/ACK soit un RST et l'hôte est alors considéré comme étant actif. Si la tentative de connexion est toujours en cours jusqu'à l'expiration du délai d'établissement, l'hôte est considéré comme étant inactif. Cette technique est aussi utilisée pour les connexions IPv6, du fait que les paquets bruts IPv6 ne sont pas encore supportés par Nmap.

`-PA [portlist]`(Ping TCP ACK)

Le ping TCP ACK ressemble fortement aux tests SYN précédemment évoqués. À la différence que, comme on l'imagine bien, le drapeau TCP ACK est utilisé à la place du drapeau SYN. Un tel paquet ACK acquitte normalement la réception de données dans une connexion TCP précédemment établie, or ici cette connexion n'existe pas. Ainsi, l'hôte distant devrait systématiquement répondre par un paquet RST qui trahirait son existence.

L'option `-PA` utilise le même port par défaut que le test SYN (80), mais peut aussi prendre une liste de ports de destination dans le même format. Si un utilisateur non-privilégié essaie cette option, ou si une cible IPv6 est spécifiée, la technique connect() précédemment évoquée est utilisée. Cette technique est imparfaite car connect() envoie un paquet SYN et pas un ACK.

La raison pour laquelle Nmap offre à la fois les tests SYN et ACK est de maximiser les chances de contourner les pare-feux. De nombreux administrateurs configurent leurs routeurs et leurs pare-feux pour bloquer les paquets entrants SYN sauf ceux

destinés aux services publics comme les sites Web de l'entreprise ou le serveur de messagerie. Ceci empêche les autres connexions entrantes dans l'organisation, tout en permettant un accès complet en sortie à l'Internet. Cette approche sans état de connexion est peu consommatrice des ressources des pare-feux/routeurs et est largement supportée dans les dispositifs de filtrage matériels ou logiciels. Le pare-feu logiciel Linux Netfilter/iptables par exemple propose l'option `--syn` qui implante cette approche sans état (stateless). Quand de telles règles de pare-feu sont mises en place, les paquets de tests SYN (`-PS`) seront certainement bloqués lorsqu'envoyés sur des ports fermés. Dans ces cas là, les tests ACK contournent ces règles, prenant ainsi toute leur saveur.

Un autre type courant de pare-feux utilise des règles avec état de connexion (statefull) qui jettent les paquets inattendus. Cette fonctionnalité était à la base fréquente sur les pare-feux haut-de-gamme, mais elle s'est répandue avec le temps. Le pare-feu Linux Netfilter/iptables supporte ce mécanisme grâce à l'option `--state` qui catégorise les paquets selon les états de connexion. Un test SYN marchera certainement mieux contre ces systèmes, car les paquets ACK sont généralement considérés comme inattendus ou bogués et rejetés. Une solution à ce dilemme est d'envoyer à la fois des paquets de tests SYN et ACK en utilisant conjointement les options `-PS` et `-PA`.

`-PU [portlist]`(Ping UDP)

Une autre option de découverte des hôtes est le ping UDP, qui envoie un paquet UDP vide (à moins que l'option `--data-length` ne soit utilisée) aux ports spécifiés. La liste des ports est écrite dans le même format que les options `-PS` et `-PA` précédemment évoquées. Si aucun port n'est spécifié, le port par défaut est le 31338. Cette valeur par défaut peut être modifiée à la compilation en changeant la définition DEFAULT_UDP_PROBE_PORT dans le fichier `nmap.h`. Un numéro de port très peu courant est utilisé par défaut, car envoyer des paquets sur un port ouvert n'est que peu souhaitable pour ce type de scan particulier.

Lorsqu'on atteint un port fermé sur la cible, le test UDP s'attend à recevoir un paquet ICMP « port unreachable » en retour. Ceci indique à Nmap que la machine est active et disponible. De nombreuses autres erreurs ICMP, comme « host/network unreachable » ou « TTL exceeded » indiquent un hôte inactif ou inaccessible. Une absence de réponse est également interprétée de la sorte. Si un port ouvert est atteint, la majorité des services ignorent simplement ce paquet vide et ne répondent rien. Ceci est la raison pour laquelle le port par défaut du test est le 31338, qui n'a que très peu de chances d'être utilisé. Très peu de services, comme chargen, répondront à un paquet UDP vide, dévoilant ainsi à Nmap leur présence.

L'avantage principal de ce type de scan est qu'il permet de contourner les pare-feux et dispositifs de filtrage qui n'observent que TCP. Les routeurs sans-fil Linksys BEFW11S4 par exemple sont de ce type. L'interface externe de cet équipement filtre tous les ports TCP par défaut, mais les paquets de tests UDP se voient toujours répondre par des messages ICMP « port unreachable », rendant ainsi l'équipement désuet.

`-PE; -PP; -PM`(Types de ping ICMP)

En plus des inhabituels types de découverte des hôtes TCP et UDP précédemment évoqués, Nmap peut également envoyer les paquets standard émis par l'éternel programme ping. Nmap envoie un paquet ICMP type 8 (echo request) aux adresses IP cibles, attendant un type 0 (echo reply) en provenance des hôtes disponibles. Malheureusement pour les explorateurs de réseaux, de nombreux hôtes et pare-feux bloquent désormais ces paquets, au lieu d'y répondre comme indiqué par la RFC 1122. Pour cette raison, les scans « purs ICMP » sont rarement fiables contre des cibles inconnues d'Internet. Cependant, pour les administrateurs surveillants un réseau local cette approche peut être pratique et efficace. Utilisez l'option `-PE` pour activer ce comportement de requête echo.

Même si la requête echo est le standard de la requête ICMP, Nmap ne s'arrête pas là, Le standard ICMP (RFC 792) spécifie également les requêtes « timestamp », « information » et « adress mask », dont les codes sont respectivement 13, 15 et 17. Si le but avoué de ces requêtes est d'obtenir des informations comme le masque réseau ou l'heure courante, elles peuvent facilement être utilisées pour la découverte des hôtes: un système qui y répond est actif et disponible. Nmap n'implante actuellement pas les requêtes d'informations, car elles ne sont que rarement supportées. La RFC 1122 insiste sur le fait “qu'un hôte ne DEVRAIT PAS implanter ces messages”. Les requêtes timestamp et masque d'adresse peuvent être émises avec les options `-PP` et `-PM`, respectivement. Une réponse timestamp (code ICMP 14) ou masque d'adresse (code ICMP 18) révèle que l'hôte est disponible. Ces deux requêtes peuvent être très utiles quand les administrateurs bloquent spécifiquement les requêtes echo mais oublient que les autres requêtes ICMP peuvent être utilisées dans le même but.

`-PR`(Ping ARP)

Un des usages les plus courant de Nmap est de scanner un LAN Ethernet. Sur la plupart des LANS, particulièrement ceux qui utilisent les plages d'adresses privées de la RFC 1918, la grande majorité des adresses IP sont inutilisées à un instant donné. Quand Nmap essaie d'envoyer un paquet IP brut (raw packet) comme une requête ICMP echo, le système d'exploitation doit déterminer l'adresse matérielle (ARP) correspondant à la cible IP pour correctement adresser la trame Ethernet. Ceci est souvent lent et problématique, car les systèmes d'exploitation n'ont pas été écrits pour gérer des millions de requêtes ARP contre des hôtes indisponibles en un court intervalle de temps.

Les requêtes ARP sont prises en charge par Nmap qui dispose d'algorithmes optimisés pour gérer le scan ARP. Si Nmap reçoit une réponse à ces requêtes, il n'a pas besoin de poursuivre avec les ping basés sur IP car il sait déjà que l'hôte est actif. Ceci rend le scan ARP bien plus rapide et fiable que les scans basés sur IP. Ainsi, c'est le comportement adopté par défaut par Nmap quand il remarque que les hôtes scannés sont sur le réseau local. Même si d'autres types de ping (comme `-PE` ou `-PS`) sont spécifiés, Nmap utilise ARP pour chaque cible qui sont sur le même sous-réseau que la machine exécutant Nmap. Si vous ne souhaitez vraiment pas utiliser le scan ARP, utilisez l'option `--send-ip`

`-n`(Pas de résolution DNS)

Indique à Nmap de ne *jamais* faire la résolution DNS inverse des hôtes actifs qu'il a trouvé. Comme la résolution DNS est souvent lente, ceci accélère les choses.

`-R`(Résolution DNS pour toutes les cibles)

Indique à Nmap de *toujours*faire la résolution DNS inverse des adresses IP cibles. Normalement, ceci n'est effectué que si une machine est considérée comme active.

`--system_dns`(Utilise la résolution DNS du système)

Par défaut, Nmap résout les adresses IP en envoyant directement les requêtes aux serveurs de noms configurés sur votre machine et attend leurs réponses. De nombreuses requêtes (souvent des douzaines) sont effectuées en parallèle pour améliorer la performance. Spécifiez cette option si vous souhaitez utiliser la résolution de noms de votre système (une adresse IP à la fois par le biais de l'appel getnameinfo()). Ceci est plus lent est rarement utile à moins qu'il n'y ait une procédure erronée dans le code de Nmap concernant le DNS -- nous contacter s'il vous plaît dans cette éventualité. La résolution système est toujours utilisée pour les scans IPv6.

Les bases du scan de ports

Même si le nombre de fonctionnalités de Nmap a considérablement augmenté au fil des ans, il reste un scanner de ports efficace, et cela reste sa fonction principale. La commande de base **nmap *`target`*** scanne plus de 1 660 ports TCP de l'hôte *`target`*. Alors que de nombreux autres scanners de ports ont partitionné les états des ports en ouverts ou fermés, Nmap a une granularité bien plus fine. Il divise les ports selon six états: `ouvert (open)`, `fermé (closed)`, `filtré (filtered)`, `non-filtré (unfiltered)`, `ouvert|filtré (open|filtered)`, et `fermé|filtré (closed|filtered)`.

Ces états ne font pas partie des propriétés intrinsèques des ports eux-mêmes, mais décrivent comment Nmap les perçoit. Par exemple, un scan Nmap depuis le même réseau que la cible pourrait voir le port 135/tcp comme ouvert alors qu'un scan au même instant avec les mêmes options au travers d'Internet pourrait voir ce même port comme `filtré`.

Les six états de port reconnus par Nmap

ouvert (open)

Une application accepte des connexions TCP ou des paquets UDP sur ce port. Trouver de tels ports est souvent le but principal du scan de ports. Les gens soucieux de la sécurité savent pertinemment que chaque port ouvert est un boulevard pour une attaque. Les attaquants et les pen-testers veulent exploiter ces ports ouverts, tandis que les administrateurs essaient de les fermer ou de les protéger avec des pare-feux sans gêner leurs utilisateurs légitimes. Les ports ouverts sont également intéressants pour des scans autres que ceux orientés vers la sécurité car ils indiquent les services disponibles sur le réseau.

fermé (closed)

Un port fermé est accessible (il reçoit et répond aux paquets émis par Nmap), mais il n'y a pas d'application en écoute. Ceci peut s'avérer utile pour montrer qu'un hôte est actif (découverte d'hôtes ou scan ping), ou pour la détection de l'OS. Comme un port fermé est accessible, il peut être intéressant de le scanner de nouveau plus tard au cas où il s'ouvrirait. Les administrateurs pourraient désirer bloquer de tels ports avec un pare-feu, mais ils apparaîtraient alors dans l'état filtré décrit dans la section suivante.

filtré (filtered)

Nmap ne peut pas toujours déterminer si un port est ouvert car les dispositifs de filtrage des paquets empêchent les paquets de tests (probes) d'atteindre leur port cible. Le dispositif de filtrage peut être un pare-feu dédié, des règles de routeurs filtrants ou un pare-feu logiciel. Ces ports ennuient les attaquants car ils ne fournissent que très peu d'informations. Quelques fois ils répondent avec un message d'erreur ICMP de type 3 code 13 (« destination unreachable: communication administratively prohibited »), mais les dispositifs de filtrage qui rejettent les paquets sans rien répondre sont bien plus courants. Ceci oblige Nmap à essayer plusieurs fois au cas où ces paquets de tests seraient rejetés à cause d'une surcharge du réseau et pas du filtrage. Ceci ralenti terriblement les choses.

non-filtré (unfiltered)

L'état non-filtré signifie qu'un port est accessible, mais que Nmap est incapable de déterminer s'il est ouvert ou fermé. Seul le scan ACK, qui est utilisé pour déterminer les règles des pare-feux, catégorise les ports dans cet état. Scanner des ports non-filtrés avec un autre type de scan, comme le scan Windows, SYN ou FIN peut aider à savoir si un port est ouvert ou pas.

ouvert|filtré (open|filtered)

Nmap met dans cet état les ports dont il est incapable de déterminer l'état entre ouvert et filtré. Ceci arrive pour les types de scans où les ports ouverts ne renvoient pas de réponse. L'absence de réponse peut aussi signifier qu'un dispositif de filtrage des paquets a rejeté le test ou les réponses attendues. Ainsi, Nmap ne peut s'assurer ni que le port est ouvert, ni qu'il est filtré. Les scans UDP, protocole IP, FIN, Null et Xmas catégorisent les ports ainsi.

fermé|filtré (closed|filtered)

Cet état est utilisé quand Nmap est incapable de déterminer si un port est fermé ou filtré. Cet état est seulement utilisé par le scan Idle basé sur les identifiants de paquets IP.

Techniques de scan de ports

Comme un débutant tâchant d'effectuer une réparation automobile, je peux me battre pendant des heures en essayant d'utiliser convenablement mes rudimentaires outils (marteau, clefs, etc.) pour la tâche à laquelle je me suis attablé. Une fois que j'ai lamentablement échoué et que j'ai fait remorqué ma guimbarde par un vrai mécanicien, à chaque fois il farfouille dans sa grosse caisse à outils pour y trouver le parfait bidule qui, d'un coup de cuillère à pot, répare le

truc. L'art du scan de port, c'est la même chose. Les experts connaissent des douzaines de techniques de scan et choisissent la bonne (ou une combinaison) pour une tâche donnée. D'un autre côté, les utilisateurs inexpérimentés et les script kiddies essaient de tout résoudre avec le scan SYN par défaut. Comme Nmap est gratuit, la seule barrière à franchir pour atteindre la maîtrise du scan est la connaissance. C'est bien mieux que l'automobile, où il faut une grande expérience pour déterminer que vous avez besoin d'une plieuse à tablier hydraulique, mais il faut quand même payer des centaines d'euros pour en disposer d'une.

La plupart des types de scans ne sont disponibles que pour les utilisateurs privilégiés. Ceci est dû au fait qu'ils émettent et reçoivent des paquets bruts (raw), qui nécessitent les droits root sur les systèmes UNIX. L'utilisation d'un compte administrateur est conseillé sous Windows, bien que Nmap puisse fonctionner avec des utilisateurs non-privilégiés si WinPcap est déjà chargé avec l'OS. Ce besoin des droits root était une sérieuse restriction quand Nmap a été diffusé en 1997, car beaucoup d'utilisateurs avaient seulement accès à des comptes Internet partagés. Maintenant, le monde est différent. Les ordinateurs sont moins chers, bien plus de gens disposent d'un accès 24/24 direct à Internet et les systèmes UNIX de bureau (comme Linux et Mac OS X) sont répandus. Une version Windows de Nmap est désormais disponible, permettant ainsi de le lancer sur encore plus de machines. Pour toutes ces raisons, les utilisateurs ont bien moins besoin de lancer Nmap depuis des comptes Internet limités. Ceci est heureux, car les options privilégiés rendent Nmap bien plus puissant et flexible.

Si Nmap essaie de produire des résultats précis, il faut garder à l'esprit que toute sa perspicacité est basée sur les paquets renvoyés par les machines cibles (ou les pare-feux qui les protègent). De tels hôtes ne sont pas toujours dignes de confiance et peuvent répondre dans le but de brouiller ou d'enduire Nmap d'erreurs. Les hôtes qui ne respectent pas les RFCs et ne répondent pas comme ils devraient sont encore plus courants. Les scans FIN, Null et Xmas sont les plus sensibles à ce problème. Ces points sont spécifiques à certains types de scan et sont donc abordés dans leur section propre de la documentation.

Cette section documente la douzaine de techniques de scan de ports gérées par Nmap. Les méthodes ne peuvent pas être utilisés simultanément, excepté le scan UDP (`-sU`) qui peut être combiné avec chacun des types de scan TCP. A titre d'aide mémoire, les options de type de scan sont de la forme `-sC`, où `C`est un caractère prépondérant dans le nom du scan, souvent le premier. La seule exception est le désuet scan par rebond FTP (`-b`). Par défaut, Nmap effectue un scan SYN, bien qu'il y substitue un scan connect() si l'utilisateur ne dispose pas des droits suffisants pour envoyer des paquets bruts (qui requièrent les droits root sous UNIX) ou si des cibles IPv6 sont spécifiées. Des scans listés dans cette section, les utilisateurs non-privilégiés peuvent seulement exécuter les scans connect() et le scan par rebond FTP.

`-sS`(Scan TCP SYN)

> Le scan SYN est celui par défaut et le plus populaire pour de bonnes raisons. Il peut être exécuté rapidement et scanner des milliers de ports par seconde sur un réseau rapide lorsqu'il n'est pas entravé par des pare-feux. Le scan SYN est relativement discret et furtif, vu qu'il ne termine jamais les connexions TCP. Il marche également contre toute pile respectant TCP, au lieu de dépendre des particularités environnementales spécifiques comme les scans Fin/Null/Xmas, Maimon ou Idle le sont. Il permet de plus une différentiation fiable entre les états `ouvert`, `fermé` et `filtré`.

Cette technique est souvent appelée le scan demi-ouvert (half-open scanning), car il n'établi pas pleinement la connexion TCP. Il envoie un paquet SYN et attend sa réponse, comme s'il voulait vraiment ouvrir une connexion. Une réponse SYN/ACK indique que le port est en écoute (ouvert), tandis qu'une RST (reset) indique le contraire. Si aucune réponse n'est reçue après plusieurs essais, le port est considéré comme étant filtré. Le port l'est également si un message d'erreur « unreachable ICMP (type 3, code 1,2, 3, 9, 10 ou 13) » est reçu.

`-sT`(Scan TCP connect())

Le scan TCP connect() est le type de scan par défaut quand le SYN n'est pas utilisable. Tel est le cas lorsque l'utilisateur n'a pas les privilèges pour les paquets bruts (raw packets) ou lors d'un scan de réseaux IPv6. Plutôt que d'écrire des paquets bruts comme le font la plupart des autres types de scan, Nmap demande au système d'exploitation qui l'exécute d'établir une connexion au port de la machine cible grâce à l'appel système `connect()`. C'est le même appel système haut-niveau qui est appelé par les navigateurs Web, les clients P2P et la plupart des applications réseaux qui veulent établir une connexion. Cet appel fait partie de l'interface d'application connue sous le nom de « Berkeley Sockets API ». Au lieu de lire les réponses brutes sur le support physique, Nmap utilise cette application API pour obtenir l'état de chaque tentative de connexion.

Si le scan SYN est disponible, il vaut mieux l'utiliser. Nmap a bien moins de contrôles sur l'appel système haut niveau `connect()` que sur les paquets bruts, ce qui le rend moins efficace. L'appel système complète les connexions ouvertes sur les ports cibles au lieu de les annuler lorsque la connexion est à demie ouverte, comme le fait le scan SYN. Non seulement c'est plus long et demande plus de paquets pour obtenir la même information, mais de plus la probabilité que les cibles activent la connexion est plus grande. Un IDS décent le fera, mais la plupart des machines ne disposent pas de ce système d'alarme. De nombreux services sur les systèmes UNIX standards noteront cette connexion dans le journal, accompagné d'un message d'erreur sibyllin si Nmap ouvre puis referme la connexion sans n'envoyer aucune donnée. Les services réseaux les plus piteux risquent même de tomber en panne, mais c'est assez rare. Un administrateur qui verrait un tas de tentatives de connexions dans ses journaux en provenance d'une seule machine devrait se rendre compte qu'il a été scanné.

`-sU`(Scan UDP)

Même si les services les plus connus d'Internet son basés sur le protocole TCP, les services UDP sont aussi largement utilisés. DNS, SNMP ou DHCP (ports 53, 161/162 et 67/68) sont les trois exemples les plus courants. Comme le scan UDP est généralement plus lent et plus difficile que TCP, certains auditeurs de sécurité les ignorent. C'est une erreur, car les services UDP exploitables sont courants et les attaquants eux ne les ignoreront pas. Par chance, Nmap peut aider à répertorier les ports UDP.

Le scan UDP est activé avec l'option `-sU`. Il peut être combiné avec un scan TCP, comme le scan SYN (`-sS`), pour vérifier les deux protocoles lors de la même exécution de Nmap.

Le scan UDP envoie un en-tête UDP (sans données) à chaque port visé. Si un message ICMP « port unreachable (type 3, code 3) » est renvoyé, le port est alors `fermé`. Les autres messages d'erreur « unreachable ICMP (type 3, codes 1, 2, 9, 10, or 13) » rendront le port `filtré`. À l'occasion, il arrive qu'un service répond par un paquet UDP, prouvant que le port est dans l'état `ouvert`. Si aucune réponse n'est renvoyée après plusieurs essais, le port est considéré comme étant `ouvert|filtré`. Cela signifie que le port peut être soit ouvert, soit qu'un dispositif de filtrage bloque les communications. Le scan de versions (`-sV`) peut être utilisé pour différencier les ports ouverts de ceux filtrés.

Une des grandes difficultés avec le scan UDP est de l'exécuter rapidement. Les ports ouverts et filtrés ne renvoient que rarement des réponses, laissant Nmap expirer son délai de retransmission au cas où les paquets se soient perdus. Les ports fermés posent encore un plus grand problème: ils renvoient normalement une erreur ICMP « port unreachable ». Mais à la différence des paquets RST renvoyés par les ports TCP fermés en réponse à un scan SYN ou à un connect(), de nombreux hôtes limitent par défaut la cadence d'émission de ces messages. Linux et Solaris étant particulièrement stricts à ce sujet. Par exemple, le kernel 2.4.20 limite cette cadence des destinations inaccessibles (« destination unreachable ») à un par seconde (cf.`net/ipv4/icmp.c`).

Nmap détecte cette limitation de fréquence et s'y ralenti conformément afin d'éviter de saturer le réseau avec des paquets inutiles que la machine cible rejettera. Malheureusement, une limitation à la Linux d'un paquet par seconde fera qu'un scan des 65 536 ports prendra plus de 18 heures. Les idées pour accélérer les scans UDP incluent le scan des cibles en parallèle, ne scanner que les ports les plus courants en premier, scanner derrière le pare-feu et utiliser l'option `--host-timeout`pour éviter les hôtes les plus lents.

`-sN`; `-sF`; `-sX` (Scans TCP Null, FIN et Xmas)

Ces trois types de scans (d'autres sont possibles en utilisant l'option `--scanflags` décrite dans la section suivante) exploitent une subtile faille de la RFC TCP pour différencier les ports entre `ouverts` et `fermés`. La page 65 indique que “si le port [de destination] est dans l'état fermé... un segment ne contenant pas le drapeau RST provoque l'émission d'un paquet RST comme réponse.”. La page suivante indique que pour les paquets envoyés à des ports sans aucun des drapeaux SYN, RST ou ACK activés: “il est peut vraisemblable que cela arrive, mais si cela est le cas, il faut rejeter le segment.”

Pour les systèmes respectant ce texte de la RFC, chaque paquet ne contenant ni SYN, ni RST, ni ACK se voit renvoyé un RST si le port est fermé et aucune réponse si le port est ouvert. Tant qu'aucun de ces drapeaux n'est utilisé, toute combinaison des trois autres (FIN, PSH et URG) son valides. Nmap exploite cela avec les trois types de scans:

Scan Null (`-sN`)

N'active aucun des bits (les drapeaux de l'en-tête TCP vaut 0).

Scan FIN (`-sF`)

N'active que le bit FIN.

Scan Xmas (`-sX`)

Active les drapeaux FIN, PSH et URG, illuminant le paquet comme un arbre de Noël (NDT: la fracture cognitive entre la culture anglo-saxonne et française se ressent fortement dans cette traduction...).

Ces trois types de scan ont exactement le même comportement, sauf pour les drapeaux TCP utilisés dans des paquets de tests (probes packets). Si un RST est reçu, le port est considéré comme étant `fermé`, tandis qu'une absence de réponse signifiera qu'il est dans l'état `ouvert|filtré`. Le port est marqué comme `filtré` si un message d'erreur ICMP « unreachable (type 3, code 1, 2, 3, 9, 10 ou 13) » est reçu.

L'avantage principal de ces types de scans est qu'ils peuvent furtivement traverser certains pare-feux ou routeurs filtrants sans état de connexion (non-statefull). Un autre avantage est qu'ils sont même un peu plus furtifs que le scan SYN. N'y comptez pas trop dessus cependant -- la plupart des IDS modernes sont configurés pour les détecter. L'inconvénient majeur est que tous les systèmes ne respectent pas la RFC 793 à la lettre. Plusieurs systèmes renvoient des RST aux paquets quelque soit l'état du port de destination, qu'il soit ouvert ou pas. Ceci fait que tous les ports sont considérés comme`fermé`. Les plus connus des systèmes qui ont ce comportement sont Microsoft Windows, plusieurs équipements Cisco, BSDI et IBM OS/400. Ce type de scan fonctionne cependant très bien contre la plupart des systèmes basés sur UNIX. Un autre désagrément de ce type de scan et qu'ils ne peuvent pas distinguer les ports `ouverts`de certains autres qui sont `filtrés`, vous laissant face à un laconique `ouvert|filtré`.

`-sA`(Scan TCP ACK)

Ce type de scan est différent des autres abordés jusqu'ici, dans le sens où ils ne peuvent pas déterminer si un port est `ouvert`(ni même `ouvert|filtré`). Il est utilisé pour établir les règles des pare-feux, déterminant s'ils sont avec ou sans états (statefull/stateless) et quels ports sont filtrés.

Le scan ACK n'active que le drapeau ACK des paquets (à moins que vous n'utilisiez l'option `--scanflags`). Les systèmes non-filtrés réagissent en retournant un paquet RST. Nmap considère alors le port comme `non-filtré`, signifiant qu'il est accessible avec un paquet ACK, mais sans savoir s'il est réellement `ouvert` ou `fermé`. Les ports qui ne répondent pas ou renvoient certains messages d'erreur ICMP (type 3, code 1, 2, 3, 9, 10, ou 13), sont considérés comme `filtrés`.

`-sW`(Scan de fenêtre TCP)

Le scan de fenêtre TCP est exactement le même que le scan ACK à la différence près qu'il exploite un détail de l'implémentation de certains systèmes pour identifier les ports fermés des autres, au lieu de toujours afficher `non-filtré`lorsqu'un RST est renvoyé. Sur certains systèmes, les ports ouverts utilisent une taille de fenêtre TCP positive (même pour les paquets RST), tandis que les ports fermés ont une fenêtre de taille nulle. Ainsi, au lieu de toujours afficher `non-filtré` lorsqu'un RST est reçu, le

scan de fenêtre indique que le port est `ouvert` ou `fermé` selon que la taille de fenêtre TCP de ce paquet RST est respectivement positive ou nulle.

Ce scan repose sur un détail d'implémentation d'une minorité de systèmes Internet, vous ne pouvez donc pas toujours vous y fier. Les systèmes qui ne le supportent pas vont certainement se voir considérés leurs ports comme `fermés`. Bien sûr, il se peut que la machine n'ait effectivement aucun port ouvert. Si la plupart des ports scannés sont `fermés` mais que quelques-uns courants, comme le 22, 25 ou le 53, sont `filtrés`, le système est vraisemblablement prédisposé à ce type de scan. Quelquefois, les systèmes ont le comportement exactement inverse. Si votre scan indique que 1 000 ports sont ouverts et que 3 seulement sont fermés ou filtrés, ces trois derniers sont certainement ceux qui sont ouverts.

`-sM`(Scan TCP Maimon)

Le scan Maimon est nommé ainsi d'après celui qui l'a découvert, Uriel Maimon. Il a décrit cette technique dans le numéro 49 de Phrack Magazine (Novembre 1996). Nmap, qui inclut cette technique, a été publié deux numéros plus tard. Cette technique est la même que les scans NUll, FIN et Xmas, à la différence près que le paquet de test est ici un FIN/ACK. Conformément à la RFC 793 (TCP), un paquet RST devrait être renvoyé comme réponse à un tel paquet, et ce, que le port soit ouvert ou non. Uriel a cependant remarqué que de nombreux systèmes basés sur BSD rejettent tout bonnement le paquet si le port est ouvert.

`--scanflags`(Scan TCP personnalisé)

Les utilisateurs réellement experts de Nmap ne veulent pas se limiter aux seuls types de scans proposés. L'option `--scanflags`vous permet de créer votre propre type de scan en spécifiant vos propres combinaisons de drapeaux TCP. Laisser courir votre imagination, tout en contournant les systèmes de détection d'intrusion dont les vendeurs n'ont fait qu'ajouter des règles spécifiques d'après la documentation Nmap!

L'argument de l'option `--scanflags` peut être soit un nombre comme 9 (PSH et FIN), mais l'utilisation des noms symboliques est plus facile. Mélanger simplement les drapeaux `URG`, `ACK`, `PSH`, `RST`, `SYN` et `FIN`. Par exemple, `--scanflags URGACKPSHRSTSYNFIN` les activent tous, bien que cela ne soit pas très utile pour effectuer un scan. L'ordre dans lequel les drapeaux sont spécifiés n'a pas d'importance.

En sus de la spécification des drapeaux désirés, vous pouvez spécifier également un type de scan TCP (comme `-sA` ou `-sF`). Ce type de scan de base indique à Nmap comment interpréter les réponses. Par exemple, un scan SYN considère que l'absence de réponse indique qu'un port est `filtré`, tandis qu'un scan FIN considèrera la même absence comme un port `ouvert|filtré`. Nmap se comportera de la même façon que le type de scan de base, à la différence près qu'il utilisera les drapeaux TCP que vous avez spécifié à la place. Si vous n'en spécifiez pas, le type de scan SYN par défaut sera utilisé.

`-sI <zombie host[:probeport]>`(Scan paresseux -- idlescan)

Cette méthode de scan avancé permet de faire un véritable scan de port TCP en aveugle, (dans le sens où aucun paquet n'est envoyé directement à la cible depuis votre vraie adresse IP). En effet, la technique employée consiste à récolter des informations sur les ports ouverts de la cible en utilisant un exploit basé sur la prédictibilité de la génération des identifiants de fragmentation IP de l'hôte relais (le zombie). Les systèmes IDS considéreront que le scan provient de la machine zombie que vous avez spécifié (qui doit remplir certains critères). Le mécanisme de cette incroyable technique est trop complexe pour être expliqué en détail dans ce guide; un papier informel a été posté pour rendre compte de tous ces détails:http://www.insecure.org/nmap/idlescan.html.

En plus de son incroyable furtivité (en raison du caractère aveugle de la technique), ce type de scan permet de déterminer les relations de confiance entre les machines. La liste des ports ouverts est établie *du point de vue de l'hôte zombie.* Ainsi, vous pouvez essayer de scanner une cible en utilisant différents zombies pour lesquels vous pensez qu'il existe une relation de confiance entre eux et la cible (d'après les règles des dispositifs de filtrage).

Vous pouvez ajouter les deux points (:) suivis d'un numéro de port de l'hôte zombie si vous souhaitez tester les changements d'identifiants IP sur un port particulier du zombie. Par défaut, Nmap utilisera le port utilisé pour les pings tcp (le port 80).

`-sO`(Scan du protocole IP)

Le scan du protocole IP permet de déterminer quels protocoles IP (TCP, ICMP, IGMP, etc.) sont supportés par les cibles. Ce n'est donc pas techniquement un scan de ports, car Nmap essaie les différents numéros de protocoles IP à la place des numéros de ports TCP ou UDP. Ce scan permet néanmoins d'utiliser l'option `-p` pour sélectionner les numéros de protocoles à scanner -- le rapport de Nmap étant toujours dans le style habituel des tables de ports -- et utilise le même moteur de scan utilisé pour le scan de ports. Ainsi, cette technique est suffisamment proche du scan de port pour être présenté ici.

Au delà de son intérêt propre, le scan de protocoles illustre la puissance des logiciels en libre accès. L'idée de base est assez simple: je n'avais même pas particulièrement pensé à l'ajouter ni reçu de requête me demandant une telle fonctionnalité. En fait, à l'été 2000, Gerhard Rieger a eu cette idée et a écrit un excellent programme de correction pour l'implanter; il l'a ensuite envoyé à la liste de distribution nmap-hackers. Je l'ai par la suite ajouté à l'arbre de développement de Nmap et j'ai publié la nouvelle version le lendemain même. Très peu de logiciels commerciaux peuvent se targuer d'avoir des utilisateurs si enthousiastes concevant et proposant leur propres améliorations!

Le scan de protocole fonctionne d'une façon similaire du scan UDP. Au lieu de parcourir les champs de numéro de port des paquets UDP, il envoie des paquets d'en-têtes IP et parcours les 8 bits du champ protocole IP. Les en-têtes son généralement vides, ne contenant pas de données ni même l'en-tête du protocole sollicité. Les trois seules exceptions étant TCP, UDP et ICMP. Un en-tête exact de ces protocoles est inclus, sinon certains systèmes refusent de les émettre et Nmap dispose déjà des fonctions permettant de construire ces en-têtes. Au lieu de scruter les messages ICMP

« port unreachable », comme pour le scan UDP, le scan de protocole attend de recevoir les messages ICMP «*protocol*unreachable ». Dès que Nmap reçoit une réponse d'un protocole en provenance de la cible, Nmap considère ce protocole comme `ouvert`. Une erreur ICMP « protocol unreachable » (type 3, code 2) fait en sorte que le port est considéré comme étant `fermé`. Les autres messages d'erreur ICMP « unreachable (type 3, code 1, 3, 9, 10, or 13) » font en sorte que le port est considéré comme étant `filtré` (tout en prouvant que le protocole ICMP est quant à lui `ouvert`). Si aucune réponse n'est reçue après plusieurs transmissions, le protocole est considéré comme étant `ouvert|filtré`.

`-b <ftp relay host>`(Scan par rebond FTP)

Une caractéristique intéressante du protocole FTP (RFC 959) est qu'il supporte les connexions par proxy ftp (proxy ftp connections, ainsi nommées dans la RFC). Ceci permet à un utilisateur de se connecter à un serveur FTP, puis de demander qu'un fichier soit envoyé à un tiers serveur FTP. Une telle fonctionnalité est propre à être détournée à tous les niveaux, c'est pourquoi la plupart des serveurs ont cessé de la supporter. Un des détournements possible de cette caractéristique conduit le serveur FTP à scanner les ports d'autres hôtes. Demandez simplement au serveur FTP d'envoyer un fichier à chaque port intéressant de votre cible, et il se chargera d'effectuer le scan. Le message d'erreur permettra de savoir si le port est ouvert ou non. C'est un très bon moyen de contourner les pare-feux car les serveurs FTP des organisations sont souvent situés de telle façon à avoir plus d'accès aux hôtes du réseau internes que toute autre machine Internet. Nmap supporte le scan par rebond FTP avec l'option `-b`. Cette option prend un argument du type *`username`*:*`password`*`@`*`server`*:*`port`*. *`Server`* est le nom ou l'adresse IP d'un serveur FTP vulnérable. Comme pour une adresse URL traditionnelle, vous pouvez omettre *`username`*:*`password`*, (user: `anonymous`, password: `-wwwuser@`) pour accéder de manière anonyme. Le numéro de port (et les deux points) peuvent être également omis si le port FTP par défaut (21) est utilisé par le serveur *`server`*.

Cette vulnérabilité était très répandue en 1997 quand Nmap a été publié mais a largement été corrigée depuis. Il existe encore quelques serveurs vulnérables qui traînent, autant les essayer si rien d'autre ne marche (!!!). Si votre but est de contourner un pare-feu, scannez le réseau cible pour trouver un port 21 ouvert (ou un serveur FTP sur tout autre port en activant la détection de version), essayez ensuite pour chacun d'entre eux le scan par rebond FTP. Nmap vous indiquera si chaque hôte y est vulnérable ou pas. Si vous voulez juste essayer de masquer vos attaques, vous n'avez pas besoin (et même en fait, vous ne devriez pas) vous limiter aux hôtes du réseau cible. Avant de vous lancer dans un scan sur des adresses Internet au hasard, à la recherche de serveurs FTP vulnérables, pensez bien que les gestionnaires des systèmes n'apprécieront pas trop que vous détourniez leurs serveurs à cet effet.

Spécifications des ports et ordre du scan

En plus de toutes les méthodes de scan abordées précédemment, Nmap propose des options permettant la spécification des ports à scanner ainsi que l'ordre (au hasard ou séquentiel) dans lequel le scan doit se faire. Par défaut, Nmap scanne tous les ports jusqu'au 1 024

inclusivement ainsi que les ports supérieurs listés dans le fichier `nmap-services`pour le ou les protocoles demandés).

`-p <port ranges>`(Ne scanne que les ports spécifiés)

> Cette option spécifie quels ports vous voulez scanner et remplace le comportement par défaut. Les ports peuvent être spécifiés un à un ou par plages (séparés par des tirets, notamment 1-1023). Les valeurs de début ou de fin des plages peuvent être omises, de sorte que Nmap utilisera les ports 1 et 65 535, respectivement. Ainsi, vous pouvez spécifier `-p-` pour scanner tous les ports de 1 à 65 535. Le scan du port 0 est autorisé si spécifié explicitement. Pour ce qui est du scan du protocole IP (`-sO`), cette option spécifie les numéros de protocoles que vous souhaitez scanner (0-255).
>
> Lorsque vous scannez à la fois des ports TCP et UDP, vous pouvez spécifier un protocole particulier en préfixant les numéros de ports par `T:` (pour TCP) ou `U:` (pour UDP). Le qualificateur reste actif à moins que vous n'en indiquiez un autre. Par exemple, l'argument `-p U:53,111,137,T:21-25,80,139,8080`scannerait les ports UDP numéros 53 111 et 137 et les ports TCP de 21 à 25 inclusivement, 80, 139 et 8080. Notez que si vous voulez à la fois scanner TCP et UDP, vous devez spécifier `-sU` et au moins un type de scan TCP (comme `-sS`, `-sF` ou `-sT`). Si aucun qualificateur de protocole n'est spécifié, les numéros de ports sont alors valables pour tous les protocoles.

`-F`(Scan rapide (limite aux ports connus)

> Cette option indique que vous souhaitez seulement scanner les ports listés dans le fichier `nmap-services` fourni avec Nmap (ou le fichier des protocoles avec l'option `-sO`). Ceci est bien plus rapide que de scanner les 65 535 ports d'un hôte. Comme cette liste contient beaucoup de ports TCP (plus de 1 200), la différence de vitesse avec le comportement par défaut (environ 1 650 ports) est relativement négligeable. Par contre, la différence peut être énorme si vous spécifiez votre propre mini-fichier `nmap-services` en utilisant l'option `--datadir`.

`-r`(Ne mélange pas les ports)

> Par défaut, Nmap mélange au hasard l'ordre des ports (sauf que certains ports couramment accessibles sont placés vers le début de la liste pour des raisons d'efficacité). Ce mélange est normalement souhaitable, mais vous pouvez spécifier l'option `-r` pour effectuer un scan de port séquentiel.

Détection de services et de versions

Supposons que Nmap vous ai signalé que les ports 25/tcp, 80/tcp et 53/udp d'une machine distante sont ouverts. En utilisant sa base de données `nmap-services`d'environ 2 200 services bien connus, Nmap indique que ces ports correspondent probablement à un serveur de messagerie (SMTP), un serveur Web (HTTP) et un serveur de noms (DNS), respectivement. Cette consultation est souvent pertinente -- une vaste majorité des démons écoutant sur le port 25, étant bien des serveurs de messagerie. Cependant, en sécurité, il ne faudrait pas trop parier

là-dessus ! Les gens peuvent lancer des services sur des ports bizarres et ils le font effectivement.

Même si Nmap a raison, et que les serveurs hypothétiques du dessus sont bien des serveurs SMTP, HTTP et DNS, ce n'est pas très utile. Lors d'audit de sécurité (ou bien lors de simples inventaires de réseau) de votre entreprise ou de clients, vous voulez réellement savoir de quels serveurs de messagerie et de noms il s'agit, ainsi que leurs versions. Connaître avec précision le numéro de version aide considérablement à déterminer à quels exploits un serveur est vulnérable. La détection de version vous permet d'obtenir une telle information.

Après avoir découvert les ports TCP ou UDP par une des méthodes de scan, la détection de version interroge ces ports pour savoir quelle version tourne actuellement. La base de données `nmap-service-probes` contient les tests à effectuer selon les services, ainsi que les chaînes de caractères auxquelles comparer les réponses. Nmap essaie de déterminer le protocole (p. ex.: ftp, ssh, telnet, http), le nom de l'application (p. ex.: ISC Bind, Appache httpd, Solaris telnetd), le numéro de version, le nom d'hôte, le type d'équipement (p. ex.: imprimante, routeur), la famille d'OS (p. ex.: Windows, Linux) et quelquefois des détails divers (p. ex.: si un serveur X accepte ou non des connexions, la version du protocole SSH, le nom d'utilisateur KaZaA). Bien sûr, la plupart des services ne fournissent pas autant d'informations. Si Nmap a été compilé avec le support de OpenSSL, il se connectera aux serveurs SSL pour déduire le service écoutant derrière la couche de cryptage. Quand des services RPC sont découverts, la moulinette RPC de Nmap (`-sR`) est automatiquement utilisée pour déterminer le programme RPC et sa version. Des ports peuvent rester dans l'état `ouvert|filtré` lorsqu'un scan de ports UDP a été incapable de déterminer si le port était ouvert ou fermé. La détection de version tentera d'obtenir une réponse de ces ports (comme s'ils étaient ouverts), et changera l'état à ouvert si elle y parvient. Les ports TCP `ouverts|filtré` sont traités de la même façon. Notez que l'option `-A` de Nmap active notamment la détection de version. Un papier documentant le fonctionnement, l'utilisation et la personnalisation de la détection de version est disponible à http://www.insecure.org/nmap/vscan/.

Lorsque Nmap reçoit une réponse d'un service mais ne parvient pas à le faire correspondre à un service de sa base de données, il affiche une empreinte et une adresse URL où vous pouvez l'envoyer si vous êtes sûr de ce qui tourne sur ce port. Prendre quelques minutes pour faire cette soumission permettra à tout le monde de bénéficier de votre découverte. Grâce à ces soumissions, Nmap dispose d'environ 3 000 empreintes de référence liées à plus de 350 protocoles, comme smtp, ftp et http.

La détection de version est activée et contrôlée grâce aux options suivantes:

`-sV`(Détection de version)

> Active la détection de version, tel que discuté ci-dessus. Autrement, vous pouvez utiliser l'option `-A` pour activer à la fois la détection de version et celle du système d'exploitation.

`--allports(tous les ports)`(N'exclut aucun port de la détection de version)

> Par défaut, la détection de version de Nmap évite le port TCP 9100 car certaines imprimantes impriment tout bonnement tout ce qui est envoyé sur ce port, ce qui conduit à l'impression de douzaines de pages de requêtes HTTP, des requêtes de

sessions SSL en binaire, etc. (ce qui est particulièrement furtif). Ce comportement peut être changé en modifiant ou en supprimant la directive `Exclude` du fichier `nmap-service-probes`, ou en spécifiant l'option `--allports` pour scanner tous les ports sans tenir compte d'aucune directive `Exclude`.

`--version-intensity <intensity>`(Sélectionne l'intensité du scan de version)

Lors d'un scan de version (`-sV`), Nmap envoie une série de paquets de tests, à chacun duquel est associé une valeur de rareté allant de 1 à 9. Les tests aux basses valeurs sont efficaces pour une grande variété de services courants, tandis que les hautes valeurs indiquent ceux qui ne sont que rarement utiles. Le niveau d'intensité spécifie quels tests doivent être effectués. Plus la valeur est haute, plus le service a de chances d'être correctement identifié. Cependant, ces scans-ci sont plus longs. La valeur d'intensité doit être comprise entre 0 et 9, la valeur par défaut étant le 7. Quand un test est inscrit sur le port cible par le biais de la directive `nmap-service-probes ports`, ce test est tenté quelque soit le niveau d'intensité. Cela permet de s'assurer que les tests DNS seront toujours tentés sur chaque port 53 ouvert, les tests SSL sur chaque 443, etc.

`--version-light`(Active le mode léger)

Il s'agit d'un raccourci pour `--version-intensity 2`. Ce mode léger rend le scan de version bien plus rapide, mais il est un peu moins susceptible d'identifier les services.

`--version-all`(Essaie chaque test possible)

Il s'agit d'un raccourci pour`--version-intensity 9`forçant chaque test unitaire à être testé contre chaque port.

`--version-trace`(Trace l'activité du scan de version)

Ceci force Nmap à afficher un nombre considérable d'informations de débogage à propos de ce que fait le scan de version. Il s'agit d'un sous-ensemble de ce que vous obtenez avec l'option `--packet-trace`.

`-sR`(Scan RPC)

Cette méthode fonctionne conjointement avec les différentes méthodes de scan de Nmap. Il prend tous les ports TCP/UDP ouverts et les submerge avec les commandes NULL du programme SunRPC dans le but de déterminer s'il s'agit de ports RPC, et le cas échéant, de quel programme et quel numéro de version il s'agit. Vous pouvez aussi obtenir les mêmes informations avec **rpcinfo -p**, et ce, même si le mapper de port (portmapper) de la cible se trouve derrière un pare-feu (ou protégé par des wrappers TCP). Les leurres ne fonctionnent pas avec le scan RPC. Cette option est automatiquement activée par le scan de version (`-sV`). Comme la détection de version inclus le scan RPC, et est bien plus complète, on a rarement besoin de l'option `-sR`.

Détection de systèmes d'exploitation

L'une des fonctions les plus connues de Nmap est la détection de systèmes d'exploitation utilisant la prise d'empreinte de la pile TCP/IP. Nmap envoie une série de paquets TCP et UDP à l'hôte distant puis examine presque chaque bit des réponses. Après quelques douzaines de tests comme l'échantillonnage des séquences TCP (ISN sampling), l'ordonnancement et le support d'options TCP, l'échantillonnage IPID et la vérification de la taille initiale de fenêtre, Nmap compare les résultats avec sa base de données d'empreintes,`nmap-os-fingerprints` contenant plus de 1 500 empreintes de systèmes d'exploitation connus afin d'afficher finalement ses conclusions s'il trouve correspondance. Chaque empreinte contient une description en texte libre du système d'exploitation ainsi qu'une classification qui fournit le nom du développeur (p. ex. : Sun), le système proprement dit (p. ex. : Solaris), sa génération (p. ex. : 10) et le type d'interface (généraliste, routeur, switch, console de jeu, etc.).

Si Nmap n'est pas capable de déterminer le système d'exploitation d'une machine et si les conditions sont acceptables (p. ex. : au moins un port ouvert et un port fermé ont été trouvés), Nmap fournira une adresse URL que vous pouvez utiliser afin de transmettre cette empreinte si vous connaissez avec certitude le système d'exploitation qui tourne sur cette machine. Ce faisant, vous contribuez grandement au panel de systèmes d'exploitation reconnus par Nmap et lui permettez d'être plus performant pour chacun.

La détection du système d'exploitation met en oeuvre un certain nombre de tests annexes utilisés de toute façon durant le processus global. L'un d'eux est la mesure de l'uptime, qui utilise l'option TCP timestamp (RFC 1323) afin de déterminer si une machine a été redémarrée récemment. Il n'en est fait mention que dans le cas où cette machine fournie cette information. Un autre test consiste en la Classification de la Prédictabilité de Séquence TCP (TCP Sequence Predictability Classification) qui mesure approximativement la difficulté d'établir une connexion en TCP forgée préalablement en direction de l'hôte distant. Ce test est utile dans le cas d'exploitation de la relation de confiance basée sur l'IP (cas du rlogin, des filtres de pare-feux, etc.) ou dans le but de cacher la source d'une attaque. Ce genre de mystification (spoofing) est rarement effectué quoi qu'il en soit, mais beaucoup de machines y sont toujours sensibles. La valeur indiquant la difficulté est basée sur un échantillonnage statistique et peut ainsi varier. Il est préférable d'employer une classification anglo-saxonne telle que “worthy challenge” , vraiment difficile, ou “trivial joke”, qui est très facile. Ceci n'est indiqué dans une sortie de type normale qu'en mode verbeux (verbose mode, `-v`). Lorsque le mode verbeux est utilisé avec l'option `-O`, la génération de séquence IPID est aussi précisée. La plupart des machines sont de classe incrémentielle, “incremental”, ce qui signifie qu'elles incrémentent le champ ID de l'en-tête IP pour chaque paquet qu'elles envoient. Ceci les rend vulnérables à différentes techniques avancées de collecte d'informations et de mystification (usurpation d'identité).

Un article décrivant l'usage, le réglage et le fonctionnement de la détection de version est disponible dans plus d'une douzaine de langues sur le site http://www.insecure.org/nmap/nmap-fingerprinting-article.html.

La détection de systèmes d'exploitation est activée et contrôlée avec les options suivantes :

`-O` (Active la détection du système d'exploitation)

> Active la détection du système d'exploitation, tel que discuté ci-dessus. Vous pouvez aussi utiliser l'option `-A` pour activer à la fois la détection du système d'exploitation et la détection de la version.

`--osscan-limit` (Limite la détection du système d'exploitation aux cibles potentielles)

> La détection du système d'exploitation est bien plus efficace si au moins un port ouvert et un port fermé sont trouvés. Utilisez cette option et Nmap ne tentera pas d'effectuer la détection contre des hôtes qui ne remplissent pas cette condition. Ceci peut faire gagner pas mal de temps, particulièrement dans le cas de scans `-P0` contre de nombreux hôtes. C'est évidemment seulement important lorsque la détection du système d'exploitation est demandée au moyen de l'option `-O` ou `-A`.

`--osscan-guess`; `--fuzzy` (Prédire un résultat de détection du système d'exploitation)

> Lorsque Nmap est incapable de trouver une correspondance parfaite pour le système d'exploitation, il propose souvent les correspondances les plus proches. La correspondance doit être très proche afin que Nmap puisse effectuer cette procédure par défaut. L'une ou l'autre de ces options (équivalentes) force Nmap à être plus agressif dans sa prédiction.

Timing et Performances

L'une des priorités les plus importantes dans le développement de Nmap a toujours été la performance. Un scan par défaut (**nmap *`hostname`***) d'un hôte sur mon réseau local prend un cinquième de seconde. Il s'agit donc de très peu de temps mais les minutes s'accumulent lorsque vous scannez des dizaines ou des centaines de milliers d'hôtes. De plus, certains scans tels que le scan UDP et la détection de version peuvent accroître le temps global du scan de façon significative. De plus, certains pare-feux limitent le taux de réponses dans leur configuration. Bien que Nmap utilise un fonctionnement en parallèle et beaucoup d'autres algorithmes avancés afin d'accélérer ces scans, l'utilisateur garde le contrôle total sur le fonctionnement de Nmap. Les utilisateurs confirmés choisissent avec une grande attention leurs commandes afin d'obtenir seulement les informations dont ils ont besoin en un minimum de temps.

Les techniques permettant d'affiner les temps de scan sont entre autres d'éviter les tests non essentiels et d'avoir les versions les plus récentes de Nmap (les augmentations de performance sont fréquentes). Optimiser ses paramètres de temps en temps peut ainsi faire toute la différence. Ces options sont décrites ci-dessous.

`--min-hostgroup <millisecondes>`; `--max-hostgroup <millisecondes>` (Ajuste la quantité du groupe de scans en parallèle)

> Nmap peut scanner des ports ou faire un scan de version sur de multiples hôtes en parallèle. Pour ce faire, Nmap divise la plage des adresses IP des cibles en groupe puis scanne ces groupes un à la fois. En général, scanner un grand nombre de groupes améliore l'efficacité de la procédure. En contrepartie, les résultats ne peuvent être fournis que lorsque tout le groupe d'hôtes a été scanné. Par conséquent, si Nmap a commencé avec un groupe de 50, l'utilisateur ne recevra aucun résultat tant que les premiers 50 hôtes ne seront pas terminés (exception faite des informations données en mode verbeux).

Par défaut, Nmap adopte un compromis dans son approche de ce conflit. Il commence avec une quantité aussi petite que 5 groupes de façon à obtenir rapidement les premiers résultats et augmente ensuite la quantité de groupes jusqu'à un maximum de1 024. Les valeurs exactes par défaut dépendent des options configurées. Par soucis d'efficacité, Nmap utilise une quantité de groupes plus grande lorsqu'il s'agit de scans UDP ou sur peu de ports en TCP.

Lorsqu'un maximum est spécifié en quantité de groupes avec l'option `--max-hostgroup`, Nmap ne va jamais dépasser cette valeur. Spécifiez une quantité minimale avec l'option `--min-hostgroup` et Nmap tentera de garder la quantité de groupes au-dessus de cette valeur. Nmap devra peut-être utiliser des groupes plus petits que ceux que vous demandez s'il n'y a plus assez d'hôtes cibles sur une interface donnée par rapport au minimum que vous avez spécifié Les deux valeurs doivent être déterminés pour de conserver la quantité de groupes dans une plage spécifique, quoique ceci ne soit que rarement souhaité.

Le premier usage de ces options est de spécifier un minimum assez grand pour que le scan entier se fasse plus vite. Un choix fréquent est 256 pour scanner un réseau de Classe C. S'il s'agit d'un scan incluanrt beaucoup de ports, dépasser cette valeur n'aidera pas à grand chose. S'il s'agit de scans sur peu de ports, une quantité de groupes de 2 048 ou plus peut faciliter la procédure.

`--min-parallelism <millisecondes>`; `--max-parallelism' <millisecondes>` (Ajuste la mise en parallèle des paquets de test, probes)

Ces options permettent de contrôler le nombre total de probes idéal pour un groupe d'hôtes. Elles permettent de scanner des ports et de découvrir des hôtes (host discovery). Par défaut, Nmap calcule un parallélisme idéal et variable basé sur les performances du réseau. Si des paquets sont rejetés, Nmap ralentit sa cadence en permettant moins de probes simultanés. Le nombre idéal de probes augmente graduellement en même temps que le réseau démontre ses performances. Ces options fixent les limites maximales et minimales selon cette variable. Par défaut, le parallélisme idéal peut chuter à 1 si le réseau s'avère trop faible et monter à plusieurs centaines dans des conditions parfaites.

L'usage habituel consiste à régler l'option `--min-parallelism` à une valeur supérieure à 1 pour accélérer les scans sur des réseaux de faible performance. Il est risqué de trop modifier cette option puisqu'établir une valeur trop élevée peut affecter la précision des résultats. Modifier cette option réduit aussi la capacité de Nmap à contrôler le parallélisme de façon dynamique selon les conditions du réseau. Une valeur de 10 peut être raisonnable bien que je n'ajuste personnellement celle-ci qu'en dernier ressort.

L'option `--max-parallelism` est parfois réglée à 1 afin d'éviter d'envoyer plus d'un probe en même temps vers les hôtes. Ceci peut être intéressant en combinaison avec l'option `--scan-delay` (on verra plus tard), bien que cette option serve déjà elle-même à cet effet.

`--min_rtt_timeout <millisecondes>`, `--max-rtt-timeout <millisecondes>`, `--initial-rtt-timeout <millisecondes>` (Ajuste la durée de vie des paquets de test, probe timeouts)

Nmap conserve une valeur de durée de vie qui détermine combien de temps il devra attendre avant d'envoyer une réponse à un probe avant de l'abandonner ou de le renvoyer. Cette valeur est calculée en fonction du temps de réponse des probes précédents. Si le temps de latence du réseau est significatif et variable, ce délai d'inactivité ou cette durée de vie, peut augmenter jusqu'à plusieurs secondes. Elle est également de niveau élevé et peut rester ainsi pendant un bon moment lorsque Nmap scanne des hôtes sans réponse.

Ces options acceptent des valeurs en millisecondes. Spécifier un `--max-rtt-timeout` et un `--initial-rtt-timeout` plus bas que ceux par défaut peuvent raccourcir le temps de scan de façon significative. C'est particulièrement vrai pour les scans sans ping préalable (`-P0`) et ceux contre des réseaux très filtrés. Toutefois, ne soyez pas trop agressif. Le scan peut se finir en un temps plus significatif si, au contraire, vous spécifiez des valeurs tellement basses que les durées de vie des probes sont terminées et ceux-ci renvoyés alors que leurs réponses sont en fait encore en transit.

Si tous les hôtes sont sur un réseau local, 100 millisecondes est une valeur de `--max-rtt-timeout` seront suffisantes. Si vous etes face a un routage, mesurez d'abord le temps de réponse d'un hôte sur le réseau \ l'aide du ping ICMP de Nmap ou d'un autre outil, comme hping2 qui est plus à même de passer un pare-feu si le paquet est spécialement forgé. Regardez les durées de transit sur 10 paquets ou plus. Vous pouvez doubler cette valeur pour `--initial-rtt-timeout` et tripler ou quadrupler le `--max-rtt-timeout`. Généralement, je ne règle pas le rtt maximum à moins de 100ms, et ce, quelles que soient les mesures de ping. De plus, je n'excède pas 1 000ms.

`--min_rtt_timeout` est une option rarement utilisée qui peut s'avérer utile lorsqu'un réseau est si lent que même les réglages par défaut de Nmap sont trop agressifs. Comme Nmap ne réduit le délai d'inactivité au minimum que lorsque le réseau semble suffisamment rapide, ce genre de besoin est inhabituel et devrait être rapporté en tant que procédure erronée à la liste de développement de nmap-dev.

`--max-retries <nombreessais>` (Spécifie le nombre maximum de retransmisison des paquets de test (probes))

Quand Nmap ne reçoit pas de réponse à un paquet de test sur un port, cela peut signifier que le port est filtré. Ou simplement que la réponse s'est perdue sur le réseau. Il est également possible que l'hôte cible ait limité son taux d'émission ce qui a temporairement bloqué la réponse. Pour ces raisons, Nmap recommence l'émission du paquet de test. Si Nmap détecte que le réseau est peu fiable, il peut essayer de re-émettre le paquet plus de fois encore avant de s'arrêter. Si cette technique améliore la fiabilité, elle ralonge la durée du scan. Quand la performance est un facteur critique, les scans peuvent être accélérés en limitant le nombre de retransmissions autorisé. Vous pouvez même spécifier `--max-retries 0` pour éviter toute retransmission, bien que cela ne soit pas trop recommandé.

Le paramétrage par défaut (sans politique `-T` spécifiée) est d'autoriser jusqu'à dic retransmissions. Si le réseau a l'air fiable et que les hôtes cibles ne limitent pas leur taux d'émission, Nmap ne fait généralement qu'une seule retransmission. Ainsi, réduire `--max-retries` à une valeur basse comme trois n'affecte pas la plupart des scans. Une telle valeur peut accélérer significativement les scans pour des hôtes lents (qui limitent leurs émissions). Généralement, vous perdez des informations si Nmap cesse de scanner un port trop tôt, mais cela peut être préférable à laisser `--host-timeout` expirer et perdre alors toutes les informations concernant la cible.

`--host-timeout <millisecondes>` (Abandon des hôtes cibles trop lents)

Certains hôtes prennent du temps *long* à scanner, tout simplement. Ceci peut être dû à du matériel ou à des logiciels réseau peu performants ou inefficaces, à un taux de paquets limité ou à un pare-feu restrictif. Le faible pourcentage de hôtes lents scannés peut ralentir le temps de scan tout entier. Il est donc parfois préférable d'écarter temporairement ces hôtes du scan initial. Ceci peut être fait en spécifiant `--host-timeout` avec le nombre de millisecondes maximales que vous êtes prêt à attendre. Je choisis souvent 1 800 000 secondes pour m'assurer que Nmap ne perde pas plus d'une demi-heure sur un seul hôte. Notez que Nmap peut être en train de scanner d'autres hôtes en même temps durant cette demi-heure, ce n'est donc pas une perte complète. Un hôte qui dépasse cette valeur est abandonné. Pas de listage des ports, de détection d'OS ou de détection de version dans les résultats pour celui-ci.

`--scan-delay <millisecondes>`; `--max_scan-delay <millisecondes>` (Ajuste le délai entre les paquets de test)

Cette option force Nmap à attendre d'obtenir au moins la valeur donnée en millisecondes entre chaque probe qu'il envoie sur un hôte donné. C'est particulièrement utile en cas de limitation de nombre de paquets (taux limite). Les machines Solaris (parmi beaucoup d'autres) vont habituellement répondre à des paquets de test d'un scan UDP par seulement un message ICMP par seconde. Tout ce qui est envoyé au-delà par Nmap serait inutile. Un `--scan-delay` de 1 000 gardera Nmap à ce taux suffisamment lent. Nmap essaie de détecter le taux limite et d'ajuster le délai en conséquence, mais il ne fait pas de mal de le préciser si vous savez déjà quelle valeur est la meilleure.

Une autre utilisation de `--scan-delay` est d'éviter les détections éventuelles des systèmes de détection et de prévention d'intrusion (IDS/IPS) basées sur ce genre de règle.

`-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>` (Régler un profil de comportement au niveau du délai)

Bien que les contrôles avancés et précis du délai dont il est fait mention dans les sections précédentes soient précis et efficaces, certains peuvent les trouver compliqués. Qui plus est, choisir les valeurs appropriées peut parfois prendre plus de temps que le scan que vous essayez d'optimiser. De ce fait, Nmap offre une approche plus simple, avec six profils de timing. Vous pouvez les spécifier grâce à l'option `-T` et aux numéros (0 à 5) ou aux noms correspondants. Les noms des profils sont paranoid (0), sneaky (1), polite (2), normal (3), agressive (4), et insane (5). Les deux premiers

sont pour éviter les IDS. Le profile « Polite » ralentit le scan afin d'utiliser moins de bande passante et moins de ressources sur la machine cible. Le profil « Normal » est celui par défaut et donc `-T3` ne fait rien. Le profil « Agressive » accélère les scans, partant du principe que vous travaillez sur un réseau suffisamment rapide et efficace. Enfin, le profil « Insane » suppose que vous êtes sur un réseau extraordinairement rapide ou que vous êtes prêt à sacrifier un peu de précision pour plus de vitesse.

Ces profils permettent à l'utilisateur de spécifier à quel point il souhaite être agressif tout en laissant Nmap choisir les valeur adéquates. Les profils effectuent aussi quelques ajustements que les options avancées ne permettent pas encore. Par exemple, `-T4` empêche la variation dynamique du délai de dépasser 10ms pour les ports TCP et `-T5` met cette valeur à 5 millisecondes. Les profils peuvent être utilisés en combinaison avec les options avancées en autant que le profil est précisé en premier. Dans le cas contraire, les valeurs normalisées pour le profil risquent d'écraser celles que vous spécifiez. Je vous recommande d'utiliser `-T4` lorsque vous scannez des réseaux plus ou moins rapides, efficaces et modernes. Utilisez cette option (en début de ligne de commande) même si vous ajoutez des options avancées afin de bénéficier des petites améliorations liée à cette option.

Si vous travaillez sur une connexion large bande ou Ethernet, je vous recommande toujours d'utiliser `-T4`. Certains aiment utiliser `-T5` quoique ce soit, à mon avis, trop agressif. Les gens utilisent parfois `-T2` parce qu'ils pensent que le rsique que les hôtes tombent en panne soit moins grand ou parce qu'ils se considèrent comme respectueux d'une façon générale. Souvent ils ne réalisent pas à quel point l'option `-T Polite` est lente en réalité. Leur scan peut prendre dix fois plus de temps qu'un scan par défaut. Les machines qui tombent en panne et les problèmes liés à la bande passante sont rares avec les options de scan par défaut (`-T3`). C'est pourquoi je les recommande habituellement pour les scanneurs précautionneux. Le fait de ne pas faire de détection de version est bien plus efficace pour limiter ces problèmes que de jouer sur les valeurs de timing.

Bien que les options `-T0` et `-T1` puissent être utiles pour éviter les alertes des IDS, elles prendront un temps énorme pour scanner des milliers de machines ou de ports. Lorsqu'il s'agit de tels scans, vous devriez régler les valeurs exactes de timing dont vous avez besoin plutôt que de vous appuyer sur les options `-T0` et `-T1` et les valeurs qui y sont associées.

Les effets principaux de `T0` sont de mettre les scans en série de façon à ce que seul un port ne soit scanné à la fois, puis d'attendre 5 minutes entre chaque envoi de probe. `T1` et `T2` sont semblables mais n'attendent que 15 secondes et 0,4 secondes, Respectivement, entre chaque probe. `T3` est le profil par défaut de Nmap et comporte la mise en parallèle. `T4` est l'équivalent de `--max-rtt-timeout 1250 --initial-rtt-timeout 500 --max-retries 6` et met le délai maximum de scan TCP à 10 millisecondes. `T5` fait la même chose que `--max-rtt-timeout 300 --min_rtt_timeout 50 --initial-rtt-timeout 250 --max-retries 2 --host-timeout 900000` tout en mettant le délai maximum de scan TCP à 5 millisecondes.

Évitement de pare-feux/IDS et mystification

Beaucoup de pionniers d'Internet envisageaient un réseau global ouvert avec un espace d'adressage IP universel permettant des connexions virtuelles entre n'importe quel noeuds. Ceci permet aux hôtes d'agir en véritables relais, recevant et renvoyant l'information les uns aux autres. Les gens pourraient accéder à l'ensemble de leur système domestique du bureau, en changeant les réglages de climatisation ou en déverrouillant leur porte pour les premiers invités. Cette vision d'une connectivité universelle a été étouffée par la réduction de l'espace d'adressage et les considérations de sécurité. Au début des années 90, les organisations commencèrent à déployer des pare-feux dans le but explicite de réduire la connectivité. De gigantesques réseaux furent cernés et coupés (NdT : le texte original dit “barrés par un cordon de police”) d'Internet non filtré par des proxies applicatifs, la conversion des adresses réseau (network address translation) et les filtrages de paquets. Le flux d'information libre céda la place à une régulation stricte de canaux de communication approuvés et du contenu qui y transitait.

Les outils d'obstruction du réseau comme les pare-feux peuvent rendre la cartographie d'un réseau beaucoup trop difficile. Ce fait ne va pas aller en s'arrangeant puisque l'étouffement de toute possibilité de reconnaissance est souvent un point clé de l'implémentation des interfaces. Nonobstant, Nmap offre un certain nombre de fonctionnalités afin d'aider à comprendre ces réseaux complexes ainsi que de s'assurer que les filtres agissent comme ils sont censés le faire. Il supporte même des mécanismes pour contourner les défenses établies de façon trop faibles. Une des meilleures méthodes pour mieux comprendre votre réseau et la sécurité qui y est déployée est de tenter de la contourner. Mettez-vous à la place de l'attaquant et déployez les techniques de cette section contre vos réseaux. Lancez un scan « FTP bounce », un « Idle scan », une attaque par fragmentation, ou tentez d'établir un tunnel à travers un de vos propres proxies.

Outre le fait de restreindre l'activité du réseau, les compagnies surveillent de plus en plus le trafic à l'aide de systèmes de détection d'intrusion (IDS). Tous les principaux IDSs sont prévus pour détecter les scans de Nmap parce que les scans sont parfois précurseurs d'attaques. Beaucoup de ces produits ont récemment migré vers des systèmes de *prévention* et d'intrusion (IPS) qui bloquent de façon active un trafic supposé malveillant. Malheureusement pour les administrateurs de réseau et les distributeurs d'IDS, la fiabilité de détection de mauvaises intentions par analyse des données de paquets demeure un problème. Les attaquants, avec de la patience, un certain niveau d'expertise et certaines quelques fonctions de Nmap, peuvent traverser un IDS sans être détectés. Dans le même temps, les administrateurs doivent composer avec un grand nombre de fausses alertes (false positive) qui bloquent et signalent une activité innocente.

De temps en temps, les gens suggèrent que Nmap ne devrait pas offrir de possibilités de contourner les règles des pare-feux ou de tromper les IDSs. Ils font valoir que ces fonctionnalités sont utilisées par les attaquants de la même façon que les administrateurs les utilisent pour renforcer leur sécurité. Le problème avec cette logique est que ces méthodes seront toujours utilisées par les attaquants, qui ne feront que trouver d'autres outils ou corriger ces fonctions sur Nmap. Dans le même temps, les administrateurs trouveront plus de difficultés à faire leur travail. Déployer seulement des serveurs FTP modernes et corrigés est une défense bien plus efficace que d'empêcher la distribution d'outils permettant les attaques « FTP Bounce ».

Il n'y a pas de méthode miracle (ni d'option dans Nmap) pour détecter et tromper les pare-feux et les systèmes IDS. Cela demande un niveau de connaissances et de l'expérience. Un tutoriel

est prévu pour ce guide de référence qui ne fait que lister les options relatives à ces sujets et ce qu'elles font.

`-f` (fragmentation de paquets); `--mtu` (utiliser le MTU spécifié)

L'option `-f` force le scan demandé (y compris les scans de type ping) à utiliser des paquets IP fragmentés en petits paquets. L'idée est de partager l'en-tête TCP en plusieurs paquets pour rendre plus difficile la détection de ce que vous faites par les dispositifs de filtrage de paquets, les systèmes de détection et d'intrusion et autres systèmes ennuyeux. Il faudra cependant faire attention ! Certains programmes ont du mal à gérer ces petits paquets. Les anciens sniffers comme Sniffit souffraient d'erreurs de segmentation immédiatement après avoir reçu le premier fragment. Spécifiez cette option une fois, et Nmap partage les paquets en 8 bytes ou moins après l'en-tête IP. Par exemple, un en-tête de 20 bytes sera fragmenté en 3 paquets. Deux avec 8 bytes d'en-tête TCP et un avec les 4 derniers. Bien entendu, chaque paquet a son en-tête IP. Spécifiez encore `-f` pour utiliser 16 bytes par fragment (ceci réduit le nombre de fragments). Vous pouvez aussi spécifier votre propre taille d'offset avec l'option `--mtu`. Par contre, ne spécifiez pas `-f` si vous utilisez `--mtu`. L'offset doit être un multiple de 8. Bien que les paquets fragmentés ne tromperont pas les filtrages de paquets et les pare-feux, tenant compte de tous les fragments IP, comme l'option CONFIG_IP_ALWAYS_DEFRAG dans le noyau Linux, certains réseaux ne peuvent supporter la perte de performance que cela entraîne et de ce fait laisse ceci désactivé. D'autres ne peuvent pas l'activer parce que les fragments peuvent prendre différentes routes au sein de leur réseau. Certains systèmes source défragmentent les paquets sortant dans le noyau. Linux, avec le module de connection « tracking iptables » est un très bon exemple. Faites donc ce genre de scan avec un sniffer comme Ethereal tournant en même temps afin de vous assurer que les paquets envoyés sont bien fragmentés. Si votre système d'exploitation causait des problèmes, essayez l'option `--send-eth` pour contourner la couche IP et envoyer des trames en raw Ethernet.

`-D <decoy1 [,decoy2][,ME],...>` (Dissimuler un scan avec des leurres)

Engendrez un scan avec des leurres, ce qui fait croire à l'hôte distant que les hôtes que vous avez spécifié exécutent eux aussi un scan contre lui. Un IDS fera état d'un scan de 5 à 10 ports depuis des adresses IP différentes, dont la vôtre, sans pouvoir faire la différence entre les leurres et la véritable origine. Bien que ceci puisse être repéré par la tracabilité des routeurs, le renvoi de réponses (response-dropping), et d'autres mécanismes actifs, ceci reste une technique généralement efficace pour cacher votre adresse IP.

Séparez chaque leure par une virgule et vous pourrez utiliser de façon facultative `ME` en tant que l'un des leurres pour représenter la position de votre véritable adresse IP. Si vous mettez `ME` en sixième position ou après, certains systèmes de détection de scans de ports (comme l'excellent scanlogd de Solar Designer) sont incapables de voir votre adresse IP. Si vous n'utilisez pas `ME`, Nmap vous placera à une position aléatoire.

Notez que les hôtes que vous utilisez comme leurres devraient être réellement actifs; sinon, vous risquez d'inonder votre cible par des SYN. Sans compter qu'il serait très facile de déterminer quel hôte est en train de scanner si en fait un seul est actif sur le

réseau. Vous pourriez utiliser des adresses IP plutôt que des noms afin de ne pas apparaître dans les logs des serveurs de nom du réseau.

Les leurres sont utilisés autant dans la phase initiale de scan ping (utilisant les ICMP, SYN, ACK, ou quoi que ce soit) que dans la phase proprement dite de scan de ports. Les leurres sont aussi utilisés pendant la détection d'OS distant (`-O`). Les leurres ne fonctionnent pas avec la détection de version ou un scan de type TCP connect().

Il est inutile d'utiliser trop de leurres car cela pourrait ralentir votre scan et potentiellement le rendre moins précis. Enfin, certains FAI peuvent filtrer vos paquets usurpés (spoofés) toutefois beaucoup ne le font pas du tout.

`-S <IP_Address>` (Usurper votre adresse source)

Dans certaines circonstances, Nmap n'est pas capable de déterminer votre adresse source (Nmap vous avisera le cas échéant). Dans cette situation, utilisez `-S` avec l'adresse IP de l'interface avec laquelle vous souhaitez envoyer les paquets.

Un autre usage possible de ce drapeau est d'usurper (spoofer) le scan afin de faire croire à la cible que *quelqu'un d'autre* est en train de les scanner. Imaginez une compagnie constamment scannée pas un concurrent ! L'option `-e` est généralement requise pour ce genre d'usage et `-P0` est à conseiller quoi qu'il en soit.

`-e <interface>` (Utiliser l'interface précisée)

Avise Nmap sur quelle interface envoyer et recevoir les paquets. Nmap devrait pouvoir la détecter automatiquement mais il vous le dira si ce n'est pas le cas.

`--source-port <portnumber>; -g <portnumber>` (Usurper le numéro du port source)

L'une des erreurs de configuration les plus surprenantes est de faire confiance au trafic sur la base du port d'où il provient. Il est facile de comprendre pourquoi une telle situation se produit. Un administrateur va régler un tout nouveau pare-feu et être noyé sous les plaintes des utilisateurs dont les applications ne fonctionnent plus. En particulier, les DNS peuvent être cassés parce que les réponses UDP DNS depuis les serveurs externes ne peuvent plus entrer sur le réseau. Le FTP est un autre exemple. Dans les transferts actifs en FTP, le serveur distant essaie d'établir une connexion en retour vers le client afin de transférer le fichier demandé.

La solution sécurisée pour ce problème existe, souvent sous la forme de proxies applicatifs ou de modules de filtrage de protocoles au niveau du pare-feu. Malheureusement, il existe aussi des solutions faciles non sécurisées. En remarquant que les réponses DNS viennent du port 53 et le FTP actif du port 20, beaucoup d'administrateurs sont tombés dans le piège de seulement permettre le trafic entrant depuis ces ports. Ils imaginent souvent qu'aucun attaquant n'aura noté et pensé exploiter de telles failles de pare-feux. Dans d'autres cas, l'administrateur va considérer que c'est une solution à court terme jusqu'à ce qu'il implémente une solution plus sécurisée. Ils oublient par la suite d'effectuer la mise à jour de sécurité.

Les administrateurs de réseau surchargés de travail ne sont pas les seuls à tomber dans ce piège. Beaucoup de produits sont pensés avec ce genre de règle mal sécurisée. Même Microsoft en a été coupable. Les filtres IPsec, fournis avec Windows 2000 et Windows XP, contiennent une règle implicite qui autorise tout trafic depuis le port 88 (Kerberos) en TCP ou UDP. Dans un autre cas bien connu, les versions du pare-feu Zone Alarm personal firewall jusqu'à 2.1.25 permettaient tout paquet UDP provenant du port 53 (DNS) ou 67 (DHCP).

Nmap propose les options `-g` et `--source-port` qui sont équivalentes pour exploiter ces faiblesses. Fournissez simplement un numéro de port et Nmap enverra les paquets depuis ce port si possible. Nmap doit utiliser certains numéros de port afin que certains tests de détection d'OS fonctionnent correctement. De plus, les requêtes DNS ignorent le drapeau `--source-port` parce que Nmap se fonde sur un système de bibliothèques pour les traiter. La plupart des scans TCP, y compris le SYN scan, supportent entièrement l'option comme le fait aussi le scan UDP.

`--data-length <number>` (Ajoute des données aléatoires aux paquets envoyés)

Normalement, Nmap envoie des paquets minimalistes contenant seulement un en-tête. Donc ces paquets TCP ne font généralement que 40 bytes et les ICMP echo request seulement 28 bytes. Cette option indique à Nmap d'ajouter le nombre donné de bytes aléatoires à la plupart des paquets qu'il envoie. Les paquets de la détection d'OS (`-O`) ne sont pas affectés, contrairement à la plupart des paquets de ping et de scan de port. Cette procédure ralentit bien entendu les choses mais permet toutefois de faire passer un scan pour un peu moins suspect.

`--ttl <value>` (Règle la valeur du champ IP de durée de vie (time-to-live))

Règle le champ IPv4 du time-to-live dans les paquets envoyés à la valeur donnée.

`--randomize-hosts` (Met les hôtes dans un ordre aléatoire)

Indique à Nmap de mélanger tous les groupes contenant jusqu'à 8 096 hôtes avant de les scanner. Ceci peut rendre les scans moins évidents pour de nombreux systèmes de surveillance réseau, spécialement si vous le combinez à des options de délai lentes. Si vous souhaitez mélanger des groupes de taille plus importante, augmentez la valeur PING_GROUP_SZ dans `nmap.h` et recompilez. Une autre solution serait de générer la liste des IP cibles avec un scan de listage (list scan, `-sL -n -oN` *`filename`*), le mélanger à l'aide d'un script Perl, puis fournir la liste complète à Nmap avec `-iL`.

`--spoof-mac <mac address, prefix, or vendor name>` (Usurpation d'adresses MAC)

Demande à Nmap d'utiliser l'adresse MAC spécifiée pour l'ensemble des trames en raw Ethernet qu'il envoie. Cette option implique `--send-eth` pour s'assurer que Nmap envoie vraiment des paquets au niveau Ethernet. Le MAC donné peut prendre plusieurs formes. S'il s'agit seulement de la chaîne “0”, Nmap choisit une adresse MAC totalement aléatoire pour la session. Si la chaîne est un nombre hexadécimal (avec les paires de nombres éventuellement séparées par les deux points), Nmap utilisera ceci comme adresse MAC. Si moins de 12 chiffres sont spécifiés, Nmap remplit le reste avec des valeurs aléatoires. Si l'argument n'est ni 0 ni une chaîne

hexadécimale, Nmap recherche dans sa base de données `nmap-mac-prefixes` un nom de fournisseur contenant la chaîne en question (non sensible à la casse). Si une correspondance est trouvée, Nmap utilise le numéro OUI du distributeur (un préfixe de 3 bytes) et utilise les 3 bytes restants de façon aléatoire. Des exemples de valeurs `--spoof-mac` valides sont `Apple`, `0`, `01:02:03:04:05:06`, `deadbeefcafe`, `0020F2` et `Cisco`.

Comptes rendus

Tout outil de sécurité n'est vraiment utile qu'en fonction des comptes rendus qu'il génère. Des tests aussi complexes soient-ils et des algorithmes n'ont finalement qu'une faible valeur s'ils ne sont pas présentés et organisés de façon compréhensible. Étant donné que les utilisateurs emploient Nmap et d'autres Logiciels de diverses façons, il n'y a pas un format qui puisse convenir à tout le monde. Nmap propose donc plusieurs formats, y compris le mode interactif permettant d'être directement intelligible et le XML pour une meilleure portabilité entre logiciels (parsing).

Outre le fait de proposer différents formats de sortie, Nmap comporte des options permettant aussi bien de contrôler la verbosité des comptes rendus que le déboggage. Les différents types de sorties peuvent être envoyés à des comptes rendus normalisés ou à des fichiers spécifiques, dont le contenu peut s'agrémenter des scans successifs ou remplacer un contenu précédent. Ces fichiers de sortie peuvent aussi être utilisés pour reprendre un scan temporairement suspendu.

Nmap rend les résultats disponibles en 5 formats différents. Le format par défaut est appelé `interactive output`. Il est envoyé en sortie standard (stdout). On trouve aussi le `normal output`, qui est semblable à `interactive` à ceci près qu'il affiche moins d'informations de fonctionnement et d'alertes étant donné qu'il est plutôt destiné à être analysé à la fin des scans au lieu de façcon interactive.

La sortie au format XML est l'une des plus importante qui peut être converti en HTML. Elle est facilement traitée par des programmes tiers comme les interfaces graphiques pour Nmap, ou importée au sein de bases de données.

Les deux autres formats restants sont le simple `grepable output`, qui inclus la plupart des informations concernant une cible dans une seule ligne, et le `sCRiPt KiDDi3 0utPUt` pour les utilisateurs qui se prennent au sérieux |<-r4d.

Alors que le format interactif représente la sortie par défaut et ne nécessite pas d'option de ligne de commande particulière, les quatre autres options de format utilisent la même syntaxe. Ils prennent un argument qui représente le nom du fichier dans lequel les résultats devraient être inscrits. Des formats multiples peuvent être spécifiés mais chaque format ne devrait être spécifié qu'une seule fois. Par exemple, vous pourriez souhaiter sauvegarder une sortie de type normal (normal output) pour votre propre usage tout en sauvegardant un XML du même scan pour une analyse par un programme. Vous pouvez le faire à l'aide des options `-oX myscan.xml -oN myscan.nmap`. Bien que ce chapitre utilise des noms de fichier simples, notamment `myscan.xml`, àdes fins pratiques, des noms plus explicites sont en général recommandés. Le choix des noms relève des préférences personnelles, toutefois pour ma part, j'en utilise de longs contenant la date du scan ainsi qu'un mot ou deux décrivant le scan. Je les

enregistre ensuite dans un répertoire nommé selon la compagnie pour laquelle je suis en train d'effectuer le scan.

Même si ces options sauvegardent les résultats dans des fichiers, Nmap continue à fournir la sortie interactive en stdout comme d'habitude. Par exemple, la commande **nmap -oX myscan.xml target** génère un fichier XML intitulé `myscan.xml` tout en donnant la sortie standard avec le même résultat interactif qu'il aurait donné si l'option `-oX` n'avait pas été spécifiée du tout. Vous pouvez changer cette procédure en entrant un tiret en argument sur l'un des types de format. Ceci force Nmap à désactiver la sortie interactive et d'inscrire à la place les résultats dans le format que vous avez spécifié pour le flux de sortie standard. Par conséquent, la commande **nmap -oX - target** enverra seulement une sortie XML en stdout. Les erreurs sérieuses sont susceptibles d'être inscrites dans le flux normal d'erreur, le stderr.

Contrairement à certains arguments de Nmap, l'espace entre le drapeau de l'option fichier (comme `-oX`) et le nom de fichier ou le tiret est obligatoire. Si vous l'omettez et entrez des arguments tels que `-oG-` ou `-oXscan.xml`, une fonction de compatibilité d'arrière-plan de Nmap forcera la création de formats de type *normal format* comme fichiers de sortie nommés `G-` et `Xscan.xml`, respectivement.

Nmap offre en outre l'option de contrôler la verbosité du scan et d'ajouter les résultats les uns à la suite des autres dans un même fichier plutôt que d'écraser les résultats précédants. Toutes ces options sont décrites ci-dessous.

Formats de Sortie sur Nmap

`-oN <filespec>` (sortie Normale)

> Demande que le format `normal output` soit appliqué au fichier donné. Tel que décrit ci-dessus, cette procédure diffère légèrement d'une sortie de type `interactive output`.

`-oX <filespec>` (sortie XML)

> Demande que le format `XML output` soit donné au fichier spécifié. Nmap contient une définition de type de document (DTD) qui permet le traitement XML des résultats de Nmap. Bien que ce soit d'abord pensé aux fins d'utilisation de programmation, cette procédure peut aussi aider à interpréter la sortie XML de Nmap. Le DTD définit les éléments légaux du format et énumère souvent les attributs et les valeurs qu'ils peuvent prendre. La dernière version est toujours disponible sur http://www.insecure.org/nmap/data/nmap.dtd.
>
> Le XML offre un format stable facilement traitable au moyen d'un logiciel. Des outils de traitement XML sont offerts gratuitement dans tous les grands langages de programmation, y compris C/C++, Perl, Python et Java. Des gens ont même écrit des outils spécifiques dans ces langages destinés au support de traitement des sorties de Nmap. Notons comme exemples le Nmap::Scanner et le Nmap::Parser en Perl CPAN. Dans la plupart des cas où une application tierce doit interagir avec Nmap, le XML est le format privilégié.

Les sorties XML font référence à une feuille de style XSL qui peut être utilisée dans le but de formater les résultats au format HTML. La façon la plus simple d'utiliser ceci est de charger la sortie XML dans un navigateur Web, comme Firefox ou IE. Par défaut, cette démarche ne pourra être appliquée qu'à partir dela machine sur laquelle vous utilisez Nmap (ou une machine configurée de façon semblable) en raison du chemin système vers `nmap.xsl` codé en dur. Utilisez l'option `--webxml` ou `--stylesheet` pour une façon de générer un fichier XML portable qui rendra un format HTML sur toute machine connectée au Web.

`-oS <filespec>` (s0r713 ScRipT KIdd|3)

Le format de sortie Script kiddie est similaire à la sortie interactive, sauf qu'il est post-traité de façon à mieux coller au style l33t HaXXorZ qui s'intéresse à Nmap soit les lettres majuscules et le contenu unique de sa prononciation. Les gens dénués d'humour devraient réaliser que cette option est surtout une moquerie envers les script kiddies avant de me descendre en flammes en m'accusant de “les aider”.

`-oG <filespec>` (sortie Grepable)

Ce format de sortie vit ses derniers instants de support parce qu'il devient désuet. Le format XML est bien plus puissant et presque aussi pratique pour les utilisateurs expérimentés. Le XML est un standard pour lequel des douzaines d'excellents outils de traitement sont disponibles alors que le format de sortie grepable est mon propre bidouillage. Le XML est évolutif afin de supporter les fonctions ultérieures de Nmap au rythme où elles sont disponibles alors que j'omets souvent ces fonctions pour les sorties grepables par manque de place.

Toutefois, le format de sortie grepable reste toujours populaire. C'est un format simple qui liste chaque hôte sur une seule ligne et peut être facilement traité à l'aide d'outils uniformisés sous UNIX, notamment grep, awk, cut, sed, diff et Perl. Je l'utilise même souvent pour certains tests en ligne de commande. Trouver tous les hôtes ayant le port ssh ouvert ou tournant sous Solaris ne prend qu'un simple grep pour identifier l'hôte, envoyé sur un awk ou traité pour afficher le champ désiré.

Le format Grepable consiste en une suite de commentaires (des lignes commençant par un dièze (#) et des lignes cibles. Une ligne cible inclus une combinaison de 6 champs étiquetés, séparés par des tabulations et suivis d'un séparatif. Les champs sont `Host`, `Ports`, `Protocols`, `Ignored State`, `OS`, `Seq Index`, `IPID` et `Status`.

Le plus important de ces champs est généralement `Ports` qui donne les détails sur chaque port considéré. C'est une liste d'entrées séparées par une virgule. Chaque entrée de port représente un port considéré et prend la forme de 7 sous-champs séparés d'une barre oblique (/) . Ces sous-champs sont les suivants : `Port number`, `State`, `Protocol`, `Owner`, `Service`, `SunRPC info` et `Version info`.

Comme pour le format XML, cette page-manuel ne permet pas de documenter de façon exhaustive l'ensemble de ce format. Une vision plus détaillée est disponible sur http://www.unspecific.com/nmap-oG-output.

`-oA <basename>` (sortie en tous formats)

À votre convenance, vous pouvez spécifier `-oA` *`basename`* pour stocker les résultats de scans en format normal, XML et grepable, et ce, en une seule fois. Ils sont stockés dans *`basename`*.nmap, *`basename`*.xml et *`basename`*.gnmap, respectivement. Comme pour la plupart des programmes, vous pouvez ajouter en préfixe au nom de fichier un chemin d'accès, comme `~/nmaplogs/foocorp/` sous UNIX ou `c:\hacking\sco` sous Windows.

options de verbosité et déboggage

`-v` (Augmenter le niveau de verbosité)

Augmente le niveau de verbosité, forçant Nmap à afficher plus d'informations sur le scan qu'il effectue. Les ports ouverts sont indiqués au fur et à mesure où ils sont trouvés ainsi qu'une évaluation du temps qui reste à scanner si Nmap pense que cela prendra quelques minutes. Utilisez cette option deux fois pour encore plus de verbosité. L'utiliser plus de deux fois n'a aucun effet.

La plupart des changements modifient seulement la sortie interactive et certains touchent aussi les sorties normales et les script kiddies. Les autres sorties sont conçues de façon à traiter par une machine, c'est pourquoi Nmap peut donner des détails importants par défaut dans ces formats sans pour autant fatiguer un utilisateur humain. Toutefois, il y a quelques modifications dans les autres modes pour lesquels les tailles de sorties peuvent être réduites substantiellement par omission de quelques détails. Par exemple, une ligne commentée dans le format grepable qui fournit une liste de tous les ports scannés n'est affichée que dans le mode verbeux parce que cela peut s'avérer très long.

`-d [level]` (Augmenter ou régler le niveau de déboggage)

Quand même le mode verbeux ne donne pas assez d'informations pour vous, le déboggage est là pour vous noyer sous encore plus de données! Comme avec l'option de verbosité (`-v`), le déboggage est mis en place avec un drapeau de ligne de commande (`-d`) et le niveau de déboggage peut être augmenté en le spécifiant plusieurs fois. Autrement, vous pouvez définir un niveau de déboggage en donnant un argument à `-d`. Par exemple, `-d9` définit le niveau 9. C'est le plus haut niveau et fournira des milliers de lignes à moins que vous ne lanciez un scan très simple avec très peu de ports et de cibles.

La sortie de déeboggage est utile lorsqu'une procédure erronée est soupçonnée dans Nmap ou si vous désirez simplement savoir ce que fait Nmap et pourquoi. Comme cette fonctionnalité est surtout faite pour les développeurs, les lignes de déboggage ne sont pas toujours tres explicites. Vous pouvez obtenir quelque chose comme : `Timeout vals: srtt: -1 rttvar: -1 to: 1000000 delta 14987 ==> srtt: 14987 rttvar: 14987 to: 100000`. Si vous ne comprenez pas une ligne, vos seuls recours sont de l'ignorer, la chercher dans le code source ou obtenir de l'aide sur la liste de développement (nmap-dev). Certaines sont quand même assez explicites, mais les messages deviennent de plus en plus obscures au fur et à mesure où le niveau de déboggage est élevé.

`--packet-trace` (Trace les paquets et les données envoyés et reçus)

Force Nmap à afficher un résumé de chaque paquet envoyé ou reçu. C'est souvent utilisé pour le déboggage mais c'est aussi une bonne façon pour les nouveaux utilisateurs de mieux comprendre ce que Nmap fait en arrière-lan. Afin d'éviter d'afficher des milliers de lignes, vous pouvez spécifier un nombre limité de ports à scanner, notamment `-p20-30`. Si vous ne vous préoccupez que de ce que fait le sous-système de détection de version, utilisez plutôt `--version-trace` à la place.

`--iflist` (Dresse la liste des interfaces et des routes)

Affiche la liste des interfaces et des routes système telles que détectées par Nmap. C'est utile pour le déboggage lié aux problèmes de cheminement ou de détermination des interfaces (comme lorsque Nmap traite une connexion PPP en tant qu'Ethernet).

Options de sortie diverses

`--append-output` (Ajouter au fichier plutôt que de l'écraser)

Lorsque vous spécifiez un fichier pour un format de sortie comme `-oX` ou `-oN`, ce fichier est écrasé par défaut. Si vous préférez garder le contenu existant du fichier et rajouter les nouveaux résultats, spécifiez l'option `--append-output`. Tout fichier de sortie spécifié dans cette configuration de session de Nmap se verra agrémenté des nouveaux résultats plutôt qu'écrasé. Cela ne fonctionne pas très bien pour les données de scan au format XML (`-oX`) dont le fichier résultant ne sera pas vraiment correct et devra être rectifié à la main.

`--resume <filename>` (Reprendre un scan abandonné)

Certaines sessions importantes de Nmap peuvent prendre beaucoup de temps -- de l'ordre de plusieurs jours. De tels scans ne sont pas toujours menés à terme. Des restrictions peuvent empêcher Nmap d'être utilisé pendant les heures de travail, soit parce que le réseau peut s'écrouler, la machine sur laquelle Nmap tourne peut subir une réinitialisation voulue ou non ou Nmap lui même peut tomber en panne. L'administrateur qui utilise Nmap peut l'annuler pour toute autre raison de toutes façons, en appuyant sur **ctrl-C**. Recommencer tout le scan à partir du début peut être indésirable. Heureusement, si le format normal (`-oN`) ou grepable (`-oG`) a été conservé, l'utilisateur peut demander à Nmap de reprendre le scan sur la cible qu'il traitait au moment d'être arrêté. Spécifiez simplement l'option `--resume` avec le nom du fichier de sortie normal/grepable en argument. Aucun autre argument n'est autorisé puisque Nmap va chercher dans le fichier de sortie en question sa configuration précédente. Appelez donc simplement Nmap de cette façon : **nmap --resume *logfilename***. Nmap ajoutera les nouveaux résultats aux données déjà présentes dans le fichier en question lors de la précédente exécution. Le redémarrage n'est pas possible à partir d'un format XML parce que combiner les deux sessions dans un même fichier XML serait difficile.

`--stylesheet <path or URL>` (Défini la feuille de style XSL pour transformer la sortie XML)

Nmap dispose d'une feuille de style XSL nommée `nmap.xsl` afin de visionner ou transcrire la sortie XML en HTML. La sortie XML comprend une directive `xml-`

`stylesheet` qui pointe sur `nmap.xml` où il a été initialement installé par Nmap (où dans le répertoire courant sous Windows). Chargez simplement la sortie XML de Nmap dans un navigateur à jour et il devrait retrouver `nmap.xsl` depuis le système de fichiers puis utilisez-le pour obtenir le compte rendu des résultats. Si vous préférez utiliser une feuille de style différente, spécifiez là en argument à `--stylesheet`. Vous devez donner le chemin ou l'adresse URL complète. `--stylesheet http://www.insecure.org/nmap/data/nmap.xsl` est une utilisation classique qui indique au navigateur de charger la dernière version de la feuille de style de Insecure.Org. Cette procédure rend plus facile le visionnage des résultats sur une machine qui ne dispose pas de Nmap (et donc de `nmap.xsl`) . Par conséquent, l'adresse URL est souvent plus utile toutefois le nmap.xsl local est utilisé par défaut pour des raisons de confidentialité.

`--no_stylesheet` (Ne pas déclarer de feuille de style XSL pour le XML)

Spécifiez cette option pour empêcher Nmap d'associer toute feuille de style XSL avec les sorties XML. La directive `xml-stylesheet` est omise.

Options diverses

Cette section décrit quelques options plus ou moins importantes qui ne trouvent pas vraiment leur place ailleurs.

`-6` (Activer le scan en IPv6)

Depuis 2002, Nmap a proposé le support IPv6 pour ses fonctionnalités les plus populaires. En particulier les ping scan (TCP seulement), connect() scan et détection de version qui supportent l'IPv6. La synthaxe de la commande est la même qu'habituellement, sauf que vous précisez aussi l'option `-6` . Bien sûr, vous devez utiliser la synthaxe IPv6 si vous spécifiez une adresse plutôt qu'un nom d'hôte. Une adresse doit ressembler à `3ffe:7501:4819:2000:210:f3ff:fe03:14d0`, c'est pourquoi les noms d'hôtes sont recommandés. Les résultats de sortie ressemblent à ceux obtenus habituellement avec la notation IPv6 sur la ligne “interesting ports” .

Bien qu'on ne puisse pas dire que l'IPv6 ait bouleversé le monde, son utilisation reste notable dans certains pays (particulièrement en Asie). De plus, la plupart des systèmes d'exploitation modernes le supportent. Pour utiliser Nmap avec des IPv6, la source et la cible du scan doivent être configurées pour l'IPv6. Si votre fournisseur d'accès Internet (comme dans la plupart des cas) ne vous a pas alloué d'adresse IPv6, des tunnels libres sont disponibles et fonctionnent très bien avec Nmap. L'un des meilleurs est entretenu par BT Exact sur https://tb.ipv6.btexact.com/. J'en ai aussi utilisé un que Hurricane Electric fournit sur http://ipv6tb.he.net/. Les tunnels 6to4 sont aussi une autre approche libre et populaire.

`-A` (option de scan agressif)

Cette option active des options agressives supplémentaires avancées. Je n'ai pas vraiment déterminé ce que cela signifie jusqu'à présent. Pour le moment, ceci active la détection d'OS (`-O`) et le scan de version (`-sV`). Davantage de fonctions peuvent être

ajoutées dans le futur. L'idée est d'activer un panel complet d'options de scan sans que les gens aient à se rappeler d'un grand nombre de drapeaux. Cette option ne fait qu'activer des options sans aucun réglage d'options de délai (comme `-T4`) ou de verbosité (`-v`) que vous pourriez par ailleurs souhaiter.

`--datadir <directoryname>` (Indique l'emplacement personnalisé des fichiers de données pour Nmap)

Nmap obtient certaines informations pendant son fonctionnement depuis les fichiers `nmap-service-probes`, `nmap-services`, `nmap-protocols`, `nmap-rpc`, `nmap-mac-prefixes` et `nmap-os-fingerprints`. Nmap, dans un premier temps, recherche ces fichiers dans un répertoire indiqué avec l'option `--datadir` (si elle existe). Tout fichier non trouvé à cet emplacement sera cherché dans l'emplacement spécifié par la variable d'environnement NMAPDIR . Puis vient `~/.nmap` pour les UIDs véritables et proprement dits (systèmes POSIX seulement) ou l'emplacement de l'exécutable Nmap (Win32 seulement), et enfin un emplacement comme `/usr/local/share/nmap` ou `/usr/share/nmap` . En dernier ressort, Nmap va chercher dans le répertoire courant.

`--send-eth` (Utiliser l'envoi par raw Ethernet)

Demande à Nmap d'envoyer les paquets à la couche raw Ethernet (liaison données) plutôt que sur la couche plus élevée IP (réseau). Par défaut, Nmap choisit celui qui convient le mieux à la plateforme sur laquelle il tourne. Les raw sockets (couche IP) sont en général plus efficaces sur les machines UNIX, alors que les trames Ethernet frames sont obligatoires pour Windows depuis que Microsoft a désactivé le support des raw sockets. Nmap utilise toujours des paquets en raw IP sous UNIX en dépit de cette option quand il n'y a pas d'autre choix (par exemple, une connexion non Ethernet).

`--send-ip` (Envoyer au niveau raw IP)

Demande à Nmap d'envoyer les paquets par le biais des sockets raw IP plutôt que d'envoyer des trames de niveau inférieur en Ethernet. C'est le complément de l'option `--send-eth` discuté précédement.

`--privileged` (Suppose que l'utilisateur a des privilèges)

Dit à Nmap de supposer simplement qu'il a les privilèges suffisants pour effectuer des envois en raw socket, intercepter des paquets et des opérations similaires qui, habituellement, nécessitent des privilèges root sur les systèmes UNIX. Par défaut, Nmap quitte si de telles opérations sont tentées mais que le geteuid() n'équivaut pas à zéro. `--privileged` est utile avec les capacités des noyaux Linux et des systèmes similaires pouvant être configurés pour permettre à des utilisateurs non privilégiés d'accomplir des scans avec des raw-packets. Assurez-vous de bien fournir cette option avant tout autre pour les options qui nécessitent des privilèges (SYN scan, détection de système d'exploitation, etc.). La variable NMAP_PRIVILEGED peut être utilisée comme équivalent alternatif à `--privileged`.

`--interactive` (Démarrer en mode interactif)

Démarre Nmap en mode interactif, qui offre un prompt interactif avec Nmap permettant le lancement facile de plusieurs scans (que ce soit en synchronisation ou en arrière-plan). Cette procédure est utile pour les gens qui scannent à partir de systèmes multi-utilisateurs puisqu'ils souhaitent souvent tester leur sécurité sans que d'autre utilisateur sur le système ne sache précisément quels systèmes ils sont en train de scanner. Utilisez `--interactive` pour activer ce mode puis entrez **h** pour obtenir l'aide sur les commandes. Cette option est rarement utilisée parce que de vrais shells sont en général plus familiers et complets. Cette option inclus un opérateur dit « bang » (!) pour l'exécution des commandes de shell, qui est une des raisons de ne pas installer Nmap en tant que setuid root.

`-V`; `--version` (Affiche le numéro de version)

Donne le numéro de version de Nmap et quitte.

`-h`; `--help` (Affiche le sommaire d'aide)

Affiche un petit écran d'aide avec les options les plus courantes . Lancer Nmap sans aucun argument fait la même chose.

Exemples

Voici quelques exemples d'utilisation de Nmap, du plus simple au un peu plus complexe et ésotérique. De véritables adresses IP et noms de domaine sont utilisés pour rendre les choses plus concrètes. Vous devez les substituer avec celles de *votre propre réseau.*. Bien que je ne crois pas que scanner les prots d'autres réseaux soit ou devrait être illégal, certains administrateurs de réseau n'apprécient pas les scans non sollicités de leur réseau et peuvent s'en plaindre. La meilleure approche est donc d'obtenir d'abord leur autorisation.

Pour des raisons de tests, vous avez l'autorisation de scanner l'hôte `scanme.nmap.org`. Cette permission inclus seulement les scans avec Nmap et non pas l'essai d'exploits ou d'attaques de Denis de Service. Afin de préserver la bande passante, veuillez ne lancer qu'une douzaine de scans sur cet hôte au maximum par jour. En cas d'abus de ce libre service de cible de scan, il serait fermé et Nmap afficherait le message suivant : `Failed to resolve given hostname/IP: scanme.nmap.org`. Ces permissions s'appliquent aussi à l'hôte `scanme2.nmap.org`, à `scanme3.nmap.org`, et ainsi de suite, même si ces hôtes n'existent présentement pas.

```
nmap -v scanme.nmap.org
```

Cette option scanne tous les ports réservés TCP sur la machine `scanme.nmap.org` . L'option `-v` active le mode verbeux.

```
nmap -sS -O scanme.nmap.org/24
```

Lance un scan furtif (stealth SYN scan) contre chaque machine active parmi les 255 machines du réseau de “classe C” sur lequel Scanme réside. Il essaie aussi de déterminer le système d'exploitation sur chaque hôte actif. Cette démarche nécessite les privilèges de root puisqu'on utilise un SYN scan et une détection d'OS.

```
nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127
```

Lance une recherche des hôtes et un scan TCP dans la première moitié de chacun des 255 sous-réseaux à 8 bits dans l'espace d'adressage de classe B 198.116 Cela permet de déterminer si les systèmes font tourner sshd, DNS, pop3d, imapd ou le port 4564. Pour chacun de ces ports qui sont ouverts, la détection de version est utilisée pour déterminer quelle application est actuellement lancée.

```
nmap -v -iR 100000 -P0 -p 80
```

Demande à Nmap de choisir 100 000 hôtes de façon aléatoire et de les scanner dans le but de trouver les serveurs Web (port 80). L'énumération des hôtes est désactivée avec `-P0` puisque envoyer en premier lieu quelques probes pour déterminer si un hôte est actif est inutile lorsque vous ne cherchez à tester qu'un port sur chaque hôte.

```
nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap
216.163.128.20/20
```

Cette procédure scanne 4 096 adresses IP à la recherche de serveurs Web (sans les pinguer au préalable) et sauvegarde la sortie en format grepable et XML.

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

Effectue un transfert de zone DNS afin de trouver les hôtes au sein de company.com et ensuite fournir les adresses IP à Nmap. Les commandes ci-dessus concerne mon GNU/Linux -- les autres systèmes ont d'autres commandes pour effectuer les transferts de zone.

Bogues

Comme son auteur, Nmap n'est pas parfait. Mais vous pouvez aider à l'améliorer en envoyant les rapports de bogues ou même en écrivant des programmes de correction. Si Nmap ne satisfait pas à vos attentes, mettez-le d'abord à jour en utilisant la dernière version disponible sur http://www.insecure.org/nmap/. Si le problème persiste, faites quelques recherches afin de déterminer s'il a déjà été remarqué et signalé. Essayez pour cela de mettre l'erreur en argument sur Google ou parcourez les archives de Nmap-dev sur http://seclists.org/. Lisez ce manuel en entier quoiqu'il en soit. Si rien ne semble fonctionner, envoyez un rapport de bogue à <nmap-dev@insecure.org>. Veillez à inclure tout ce que vous avez appris au sujet de ce bogue ainsi que la version de Nmap concernée et le système d'exploitation que vous utilisez. Les rapports de problèmes et les questions sur l'utilisation de Nmap envoyés à nmap-dev@insecure.org ont plus de chance de trouver une réponse que ceux envoyés à Fyodor directement.

Les codes de programmes de correction destinés à régler des bogues sont encore meilleurs que les rapports de bogues. Les instructions de base pour créer des fichiers de programmes de correction avec vos modifications sont disponibles sur http://www.insecure.org/nmap/data/HACKING. Les programmes de correction peuvent être envoyés à nmap-dev (recommandé) ou à Fyodor directement.

Auteur

Fyodor <fyodor@insecure.org> (http://www.insecure.org)

Traduction française :

Romuald THION <romuald.thion@insa-lyon.fr> 4N9e Gutek <4n9e@futurezone.biz>
Relecture et corrections : Ghislaine Landry <g-landry@rogers.com>

Bien qu'un soin particulier ait été apporté à cette traduction, il est possible que certaines erreurs s'y soient glissées. Le cas échéant, n'hésitez pas à communiquer avec les traducteurs. La traduction ne remplace pas au texte original (version anglaise), tout particulièrement en ce qui concerne les dispositions légales. Une erreur d'interprétation dans cette traduction ne peut, en aucun cas, se substituer à ces dispositions. Insecure.Com LLC n'assume aucune responsabilité en ce qui a trait aux erreurs éventuelles de traduction ou d'interprétation.

Des centaines de personnes ont apporté de précieuses contributions à Nmap au cours des années. Celles-ci sont détaillées dans le fichier `CHANGELOG` qui est distribué avec Nmap mais aussi disponible sur http://www.insecure.org/nmap/changelog.html.