La collecte d’informations

Nous allons décrire le fonctionnement des outils permettant de récupérer des informations à distance. Ces utilitaires sont fréquemment utilisés par les pirates pour préparer de futures attaques. C’est pour cette raison qu’il est indispensable de les décrire dés le début. Vous apprendrez également à les utiliser pour votre propre protection.

Le Scanner

Pourquoi les scanners de ports sont-ils si importants au sein d'un réseau ? Fondamentalement parce que ce sont des outils indispensables à ceux qui souhaitent attaquer un système. Les différentes méthodes pour la préparation d'une attaque sont les suivantes :

- scanner une machine ou un réseau, observer les services en cours et les systèmes qui les exécutent et s'appuyer sur les vulnérabilités connues de ces services ou de ces systèmes.
- scanner une machine ou un réseau à la recherche d'un service ou d'un système particulier (incluant la vérification de la version) dont la vulnérabilité est connue.

Pour cette raison, un administrateur préoccupé par la sécurité, devra scanner son réseau et y chercher les points faibles avant que d'autres, aux intentions moins avouables, ne s'en chargent.

L’objectif du pirate est de repérer les serveurs offrant des services particuliers et de les identifier. Pour obtenir ces informations, le pirate va utiliser un scanner.

Qu’est ce qu’un scanner ? Lorsqu’un serveur offre un service particulier (Web, messagerie, mail), il exécute un programme assurant ce service. Ce programme est en attente de connexions. Les clients devant accéder à ce service doivent connaître l’adresse IP du serveur et le numéro de port associé au service. La plupart des services ont un numéro de port bien défini (voir tableau). Lorsqu’un service est en écoute sur un port, on dit que le numéro de port associé à ce service est ouvert.

Service	Port TCP
FTP	20 (données), 21 (contrôle
Telnet	23
SMTP (Mail)	25
DNS	53
HTTP	80
POP3	100

L’intérêt du scanner est très simple il permet de trouver dans un délai très court, tous les ports ouverts sur une machine. Il existe différents types de scanner, certains se contentent juste de donner le listing des ports ouverts, de donner le type et la version de l’OS tournant sur le serveur (ces fonctionnalités seront décrites dans la suite avec ***Nmap***). D’autres scanners comme ***Nessus*** permettent de tester différentes failles connues sur ces services

I-Exemple avec *Nmap*

Nmap est le scanner de ports le plus utilisé car il est très puissant. En effet il dispose d'un nombre impressionnant de fonctionnalités. De plus il est disponible sur un grand nombre de système d'exploitation en plus de GNU/Linux. Nmap vous sera très utile pour auditer votre réseau car c'est aussi un outil utilisé par les hackers. Dans la suite seront présentées les principales fonctionnalités. Pour comprendre les tenants de Nmap il faut se rappeler quelques notions de base de TCP.

Rappel sur le fonctionnement de TCP

TCP (qui signifie *Transmission Control Protocol*, soit en français: *Protocole de Contrôle de Transmission*) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes:

- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP ;
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau ;
- TCP permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne ;
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise ;

En résumé, TCP permet le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, en encapsulant les données, c'est-à-dire qu'on ajoute aux paquets de données un en-tête qui va permettre de synchroniser les transmissions et d'assurer leur réception.

La fiabilité du transport TCP dépend de l'établissement d'une connexion entre deux machines qui veulent dialoguer. L'établissement d'une connexion est réalisé par l'échange d'informations telles que le numéro de port, le numéro de séquence et la taille de fenêtre.

Signification des champs:

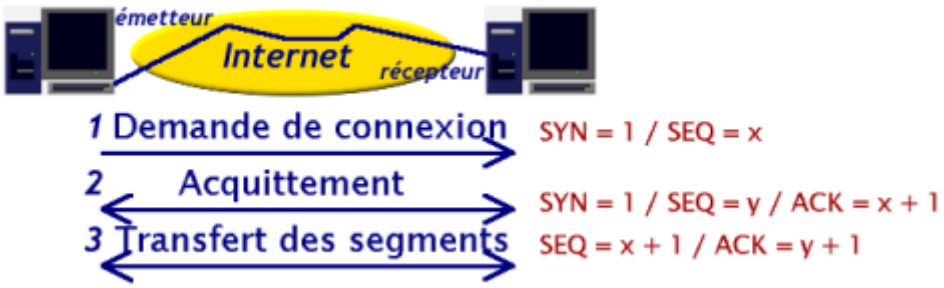
- **Port source**: Numéro du port source
- **Port destination**: Numéro du port destination
- **Numéro de sequence**: Si **SYN = 0**, le numéro de séquence est celui du premier octet de données de ce segment. Si **SYN = 1**, il s'agit du numéro de séquence initital **NSI**. le premier octet de donnée est à **NSI+1**.

- **Numéro d'acquittement**: Si le bit ACK = 1, ce champ contient le numéro de séquence attendu par l'émetteur du segment.
- **Data Offset** (Taille de l'en-tete): Longueur de l'en-tête en mots de 32 bits.
- **Réservé**: Réservé pour un usage futur
- **Drapeaux**:
  - **URG**: Signale la présence de données URGentes
  - **ACK**: Signale que le paquet est un accusé de réception
  - **PSH**: Données à envoyer tout de suite (PUSH)
  - **RST**: Rupture anormale de la connexion (ReSeT)
  - **SYN**: Demande de SYNchronisation
  - **FIN**: Demande la fin de la connexion
- **Fenêtre**: Taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- **Checksum**: Somme de contrôle (CRC, cyclic redundancy check) des données de l'en-tête
- **Pointeur de données urgentes**: Ce champ est interprété uniquement si le bit de contrôle URG est à 1. Le pointeur donne le numéro de séquence de l'octet qui suit les données << urgentes >>.
- **Options**: Facultatives ,
- **Padding** (Bourrage): Caractères de bourrage (zéro), si nécessaire
- **Données**: Séquences d'octets transmis par l'application (par exemple: OK POP3 server ready, ...)

Établissement d'une connexion

Le protocole TCP assure une fiabilité de la remise des paquets grâce à des numéros de séquences qui les distingue un à un. Chaque paquet TCP possède deux numéros, le numéro de séquence et le numéro d'acquittement. Ces deux nombres sont codés sur 32 bits. Ils sont uniques, afin de ne pas confondre les paquets lors de leurs traitements.

Pour l'établissement d'une connexion TCP, le client envoie un paquet avec un numéro de séquence initial (**SEQ**). Le serveur va répondre avec un paquet d'acquittement ayant son propre numéro de séquence (**SEQ2**), mais ayant un numéro d'acquittement (**ACK**) égal au numéro de séquence initial incrémenté d'une unité (**ACK=SEQ+1**). Ensuite le client renvoie un paquet avec un numéro d'acquittement (**ACK=SEQ2+1**). Une connexion TCP s'établit donc en trois parties.



Ce principe de numéros de séquences et d'acquittement est utilisé tout le long de la transaction pour en assurer la fiabilité. La subtilité de l'attaque réside dans le fait que le serveur génère la valeur **SEQ2** suivant un cycle particulier. Il peut utiliser, par exemple, soit une fonction générant un nombre aléatoire, soit incrémenter une valeur initiale de 128 toutes les secondes et de 64 après chaque connexion. C'est sur ces bases que nous allons décrire quelques attaque.

I-1 Les différentes méthodes de scans

Dans la suite seront présentées les principales fonctionnalités de Nmap.

TCP connect() : Cette méthode utilise l'option **-sT**. C'est le type de scan le plus simple et il n'a pas besoin d'avoir les droits de l'utilisateur root. Pour tester si un port est ouvert ou non, Nmap emploi la fonction **connect()** du langage C. Elle permet d'initialiser une connexion à une socket. On a alors une connexion TCP dite complète (méthode three way handshake). Si le port est ouvert l'appel de la fonction connect() fonctionnera sinon cela veut dire que le port est fermé.

```
N-Box:/home/neeho# nmap -sT 192.168.164.80
Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-16 13:45 CEST
Interesting ports on nserver (192.168.164.80):
(The 1617 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
80/tcp     open        http
81/tcp     open        hosts2-ns
111/tcp    open        sunrpc
139/tcp    open        netbios-ssn
Nmap run completed -- 1 IP address (1 host up) scanned in 1.138 seconds
```

Ci-dessus on peut voir que Nmap nous a retourné la liste des ports ouverts sur la machine cible 192.168.164.80. Pour chaque port ouvert Nmap affiche le service qui devrait être lancer sur le port correspondant d'après le fichier nmap-services. Ici Nmap ne vérifie pas si c'est vraiment le service affiché qui tourne sur ce port. En effet, on peut très bien lancer un serveur FTP sur le port 80 qui est habituellement réservé à un serveur HTTP. Son principal inconvénient pour une personne mal-intentionnée est qu'il sera facile à détecter. L'attaque sera alors visible dans les fichiers logs. Par contre, certains IDS (voir ci-après) n'en tienne pas compte car c'est une connexion TCP classique en trois temps. En effet ceux-ci, se préoccupe plus des méthodes décrites après.

SYN scan : Cette méthode consiste à ne pas ouvrir une connexion TCP complète comme précédemment. Cette technique est généralement appelée "half-open" ou scan furtif. Celle-ci utilise l'option -sS de Nmap. Le principe de fonctionnement est le suivant: un flag SYN est envoyer à la station cible et on attend la réponse. Si on reçoit un SYN/ACK cela veut dire que le port est ouvert, sinon si la réponse est un RST, c'est que le port est fermé. Si Nmap détecte un port ouvert il coupe brutalement la connexion par un RST.

```
N-Box:/home/neeho# nmap -sS 192.168.164.80
Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-16 13:52 CEST
Interesting ports on nserver (192.168.164.80):
(The 1617 ports scanned but not shown below are in state: closed)
Port       State       Service
```

```
21/tcp    open       ftp
22/tcp    open       ssh
80/tcp    open       http
81/tcp    open       hosts2-ns
111/tcp   open       sunrpc
139/tcp   open       netbios-ssn
Nmap run completed -- 1 IP address (1 host up) scanned in 0.990 seconds
```

Ce type de scan à pour avantage d'être plus rapide et il est moins détectable que la méthode précédente. En effet de nos jours, la majorité des IDS détecte ce type de scan à cause de sa grande popularité. Par contre il nécessite les droits de l'utilisateur root mais en général le méchant a normalement un accès root sur sa machine :)

FIN, XMAS et NULL scan : Ces trois méthodes fonctionnent de la même manière. Avec l'option -sF (scan FIN), comme son nom l'indique on envoie un paquet avec un flag FIN. Le scan XMAS (option -sX), quand à lui, il envoie un paquet avec les flags FIN|URG|PSH. Et le NULL scan utilisant le paramètre -sN n'envoie aucun flag. Chaque port fermé retourne un RST.

```
N-Box:/home/neeho# nmap -sX 192.168.164.80
Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-16 14:12 CEST
Interesting ports on nserver (192.168.164.80):
(The 1617 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
80/tcp     open        http
81/tcp     open        hosts2-ns
111/tcp    open        sunrpc
139/tcp    open        netbios-ssn
Nmap run completed -- 1 IP address (1 host up) scanned in 4.325 seconds
```

L'inconvénient principal de cette méthode est quelle ne fonctionne pas sous Windows. Tout simplement parce que ce système d'exploitation ne respecte pas les standards établis dans les RFC (Les RFC (*Request For Comments*) sont un ensemble de documents contenant les spécifications techniques sur divers points de TCP/IP (protocoles, services, ...)). Ceux-ci renvoient un RST lorsqu'un port est ouvert au lieu de rien du tout. Ces techniques se révèlent aussi plus lente que les deux précédentes. Il y a aussi quelques firewalls bien configurés qui rejettent ces paquets particulièrement avec l'option -sN. Cependant cela fonctionne sur la plus part des machines et peut être plus ou moins bien détecté.

ACK et Window scan: Ces deux fonctionnalités utilisent respectivement les options -sA et -sW. La principale utilité des ces méthodes est de déterminer le type de firewall de la cible, si le firewall bloque les paquets SYN entrants. L'inconvénient majeur est que cela ne fonctionne pas sur la totalité des systèmes d'exploitation. La liste de systèmes supportant ces techniques

sont disponibles dans la page man et sûrement dans les archives de la mailling-list nmap-hackers.

Le ACK scan envoie un numéro de séquence aléatoire au port voulu. Tandis que le Window scan fournie une taille de fenêtre invalide. En retour, si on obtient un RST cela indique que le port n'est pas filtrer. Si on a rien, le port est filtré.

Idle scan : Cette méthode permet de garder son anonymat en utilisant une troisième machine. Cette dernière est appelée zombie. Voici comment Nmap met en place cette méthode :

- Nmap envoie un paquet contenant un SYN|ACK au zombie
- Le zombie retourne un RST à l'attaquant ce qui permet de récupérer son IP ID
- L'attaquant envoie alors à la cible un SYN avec comme adresse source l'IP du zombie
- Si le port de destination est ouvert, le zombie va recevoir un SYN|ACK de la cible
- Le zombie renverra alors un RST à la cible car il n'as pas établie de connexion vers la cible (l'IP ID est alors incrémenté)
- L'attaquant envoie donc un paquet SYN|ACK au zombie pour récupérer la valeur de l'IP ID
- Si le port de la cible est fermé, un RST est envoyé au zombie ce qui ne provoque pas la modification de l'IP ID
- Nmap peut alors déterminé si le port est ouvert ou non en fonction de la valeur de l'IP iD du zombie

```
N-Box:/home/neeho# nmap -P0 -sI 192.168.164.2:139 192.168.164.80
Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-08-16 15:24 CEST
Idlescan using zombie 192.168.164.2 (192.168.164.2:139); Class: Incremental
Interesting ports on nserver (192.168.164.80):
(The 1617 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
80/tcp     open        http
81/tcp     open        hosts2-ns
111/tcp    open        sunrpc
139/tcp    open        netbios-ssn
Nmap run completed -- 1 IP address (1 host up) scanned in 39.578 seconds
```

Dans l'exemple, l'option -P0 signifie que l'on ne ping pas la cible avant le scan. On utilise ici la machine 192.168.164.2 comme zombie sur le port 139. La cible étant la station 192.168.164.80. L'avantage majeur de cette technique est que la cible n'a quasiment aucun moyen de connaître le vrai attaquant. Celle-ci pense que l'attaquant est la machine zombie. Grâce à cette technique on peut alors tester certaines règles d'un firewall vu du zombie et non de l'attaquant. On peut citer comme inconvénient que c'est un peu lent et qu'il faut trouver un "bon" zombie. Sinon la méthode possède les mêmes atouts et inconvénients que le SYN scan car c'est la même méthode de scan qui est employée.

I-2 Guarder son anonimat

La première chose à laquelle va penser une personne mal-intentionnée est qu'on ne puisse pas la retrouver lorsqu'elle fait quelque chose d'illégale. Il faut donc qu'un scan ne soit pas détecter par un IDS qui est une chose de plus en plus difficile... Pour ceux qui ne sont toujours pas convaincu de la puissance de Nmap voici ce que l'on peut encore faire :

Pour éviter de se faire détecter par certains IDS il suffit de réduire le temps entre deux scans. Nmap permet cela en utilisant l'option -T

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>.

En mode Paranoid, Nmap envoie un paquet toutes les 5 minutes et ne fait pas de scans parallèles. Sneaky est similaire au paramètre Paranoid sauf que l'on a un délai de 15 secondes entre chaque paquets. Polite est censé soulager la charge réseau et réduire les chances de planter une machine. Le temps d'attente est alors de 0.4 secondes entre chaque paquets réseau. Le comportement par défaut de Nmap est bien entendu le mode Normal. Le paramètre Agressive est particulièrement intéressant pour le SYN scan qui permet d'aller plus vite. Selon la page man ce mode est conseillé aux gens impatients qui possèdent une connexion rapide. Le dernier paramètre, Insane, est pratique pour scanner rapidement une plage d'IP du moment que perdre quelques informations ne vous gène pas. A moins que vous possédiez une connexion réseau très rapide.
Une autre solution consiste à simuler un scan de ports par plusieurs machines en utilisant de fausses adresses IP source. Autrement dit, on fais croire à la cible qu'elle est scannée par plusieurs machines en même temps. Il nous suffit donc de mêlé notre IP avec celles forgées. Pour cela l'option -D est utilisée. Par exemple dans le cas suivant la cible va détecter 3 attaques simultanée : -D 192.168.164.5,ME,192.168.164.2.
Si vous êtes sur un réseau local, vous pouvez spoofer votre adresse IP et récupérer le résultat en utilisant un sniffer tel que TCPdump. Spoofer son adresse IP consiste généralement à modifier l'adresse IP source du paquet réseau. l'inconvénient est que la réponse fournie par la cible sera envoyée à l'adresse spoofée. Il est donc plus facile de le faire sur un LAN. Il existe tout un tas de méthodes pour récupérer ces paquets mais nous ne les détaillerons pas ici. l'utilisation de cette méthode étant assez simple : -S 192.168.164.85. Ici la station cible renverra la réponse à la machine ayant pour IP 192.168.164.85.

Options utiles

- -O : Détection du système d'exploitation utilisé sur la cible en fonction de l'empreinte TCP/IP capturées
- --osscan_guess : Retourne les systèmes d'exploitation les plus probables si l'option -O ne trouve rien
- -p <range> : Plage de ports à scanner, par exemple : -p 80,21,135-139
- -f : Fragmente les paquets, permet de contrer certains NIDS
- -F : Scan dit rapide, Nmap ne scan que les ports listés dans le fichier nmap-services
- -n : Ne pas faire la conversion adresse IP en nom de domaine
- -oN <file> : Sauvegarde le résultat du scan dans le fichier file
- -P0 : Ne lance pas de ping avant scanner une cible, utile car de plus en plus de serveurs ignore les paquets ICMP
- -sP : Ne scanne pas les ports, Nmap fait seulement un ping
- -PT [ports] : Lance un ping TCP sur un ou plusieurs ports (port 80 par défaut) pour tester si la cible est accéssible ou non, utile pour les machines refusant les pings ICMP

- -g : Port source utilisé lors d'un scan, utile pour contrer certains firewall qui accepte tous les paquets venant du port source 53 par exemple
- -I : "Reverse ident scanning", cela permet de connaitre l'utilisateur qui fait tourner le programme qui écoute sur un port
- --scan_delay : Temps d'attente entre chaque test
- --host_timeout : Délai d'attente d'un ping
- --packet_trace : Affiche les paquets envoyés au format TCPdump
- --max_parallelism : Nombre maximum de paquets envoyés en parallèle
- --randomize_hosts : Scan dans un ordre aléatoire une plage d'IP

Quel est l'intérêt d'utiliser Nmap ?

Nmap permet de pouvoir prévoir les futures attaques, et aussi de pouvoir connaître quels services tournent sur une machine. Une installation faite un peu trop vite peut laisser des services en écoute (donc des ports ouverts sans que cela soit nécessaire) et donc vulnérables à une attaque. N'hésitez pas à utiliser Nmap contre vos serveurs pour savoir quels ports sont en écoute. Nmap est un logiciel très complet et très évolutif, et il est une référence dans le domaine du scanning.

Comment s'en protéger ?

Configurer votre firewall pour empêcher les scans ou utiliser les IDS (voir section ci-dessous)

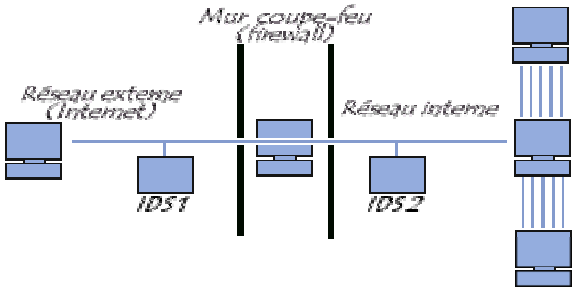
Conclusion : Si vous avez lu tout le document, vous pouvez maintenant vous en servir pour une utilisation plus avancée que vous ne l'auriez fait avant. Si vous souhaitez encore plus d'informations sur Nmap je vous recommande vivement d'aller jeter un coup d'oeil à la page man. Voir même dans son code source ou sur les mailling-lists et bien sur en visitant le site officiel ainsi qu'un petit saut chez notre ami Google ne peut pas faire de mal.

II- Systèmes de détection d'intrusions

On appelle **IDS** (*Intrusion Detection System*) un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant ainsi d'avoir une action de prévention sur les risques d'intrusion.
Il existe deux grandes familles distinctes d'IDS :

- Les **N-IDS** (*Network Based Intrusion Detection System*), ils assurent la sécurité au niveau du réseau.
- Les **H-IDS** (*Host Based Intrusion Detection System*), ils assurent la sécurité au niveau des hôtes.

Un N-IDS nécessite un matériel dédié et constitue un système capable de contrôler les paquets circulant sur un ou plusieurs lien(s) réseau dans le but de découvrir si un acte malveillant ou anormal a lieu. Le N-IDS place une ou plusieurs cartes d'interface réseau du système dédié en mode promiscuité (*promiscuous mode*), elles sont alors en mode « furtif » afin qu'elles n'aient pas d'adresse IP. Il est fréquent de trouver plusieurs IDS sur les différentes parties du réseau et en particulier de placer une sonde à l'extérieur du réseau afin d'étudier les tentatives d'attaques ainsi qu'une sonde en interne pour analyser les requêtes ayant traversé le pare-feu ou bien menée depuis l'intérieur.



Le H-IDS réside sur un hôte particulier et couvre donc une grande partie des systèmes d'exploitation tels que Windows, Solaris, Linux, HP-UX, Aix, etc .. Le H-IDS se comporte comme un démon ou un service standard sur un système hôte. Traditionnellement, le H-IDS analyse des informations particulières dans les journaux de logs (syslogs, messages, lastlog, wtmp…) et aussi capture les paquets réseaux entrant/sortant de l'hôte pour y déceler des signaux d'intrusions (Déni de Services, Backdoors, chevaux de troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par débordement de buffeurs…).