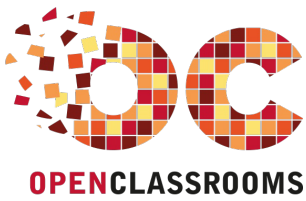
L'idle port scan

Par elalitte



www.openclassrooms.com

Dernière mise à jour le 5/09/2010

Sommaire

L'idle port scan

Par  elalitte

Mise à jour : 05/09/2010
Difficulté : Difficile

Qui n'a jamais rêvé de scanner la machine du voisin en se faisant passer pour l'autre voisin ?
L'idle scan est une technique qui vous permettra de scanner une machine en vous faisant passer pour quelqu'un d'autre.

Ce tuto nécessite d'avoir déjà de très bonnes connaissances en réseau et notamment sur le fonctionnement de la couche 3 du modèle OSI et du protocole IP.

Par ailleurs, si vous ne savez pas très précisément ce qu'est un scan de ports, je vous invite à lire le tutoriel sur le scan de ports avant de commencer la lecture de celui-ci.

L'idle port scan

Qu'est-ce que l'idle port scan ?

Comme dit précédemment, cette technique vous permettra de scanner une autre machine en vous faisant passer pour quelqu'un d'autre.
Idle en anglais veut dire, à peu de choses près, inactif. Le principe est donc de scanner quelqu'un en faisant semblant d'être inactif (et faire porter le chapeau à Mme Michu, cette vieille bique !).

Comment est-ce possible ?

Pour tous ceux qui ont déjà essayé de se faire passer pour quelqu'un d'autre sur un réseau, vous savez d'ores et déjà que ce n'est pas simple.
Le principal problème étant que si j'envoie un paquet sur le réseau en me faisant passer pour mon voisin, les réponses vont être envoyées à mon voisin, je ne les verrai pas, et cela ne me servira donc à rien...

Le principe ici sera donc d'envoyer les paquets en se faisant passer pour le voisin, d'essayer de savoir si le voisin a reçu des réponses, et si oui, de quelles réponses il s'agit.

La théorie

Comment scanner un port, sans recevoir les réponses ?

Vous qui avez lu le tutoriel sur le scan de ports, vous savez comment s'établit une connexion TCP :

- Envoi d'un segment SYN ;
- Réponse d'un segment SYN+ACK ;
- Réponse d'un ACK.

Imaginons que nous voulions nous faire passer pour notre voisin vis à vis de Mme Michu, pour scanner un de ses ports.
Etant donné que nous allons envoyer le premier SYN en nous faisant passer pour notre voisin, Mme Michu va lui répondre à lui, et nous ne saurons pas quelle est sa réponse.
Il serait nécessaire pour nous de savoir si elle a répondu, et surtout ce qu'elle a répondu !

Deux cas sont possibles :

- Soit son port est ouvert, et elle a répondu SYN+ACK.
- Soit son port est fermé, et elle a répondu RST.

Quelle différence y a-t-il entre ces deux cas ?

Une différence énorme !
Dans le premier cas, mon voisin qui jusqu'à maintenant n'a rien demandé à personne va recevoir un segment SYN+ACK venant de Mme Michu. Étant donné qu'il n'a rien demandé, il va le faire savoir en renvoyant un RST, vu qu'il ne veut pas parler à Mme Michu.
Dans le second cas, il ne va rien répondre du tout. Il reçoit une demande de réinitialisation de connexion pour une connexion qu'il n'a jamais sollicitée. Il ne va donc pas répondre à une demande de fermeture de connexion qu'il n'a jamais demandé d'ouvrir !

- Premier cas : il répond.
- Deuxième cas : il ne répond pas.

Cette différence est-elle extraordinaire ?
OUI !

La question à se poser est donc :

Qu'est-ce qui change dans ma machine quand je réponds à un paquet ?

L'IPID

~~Les lipides constituent la matière grasse des êtres vivants~~... Ah non, pas ceux-là. 

L'IPID, ou IP identifier, est un nombre codé sur 2 octets et contenu dans l'en-tête IP (variant donc de 0 à 65535)
Il est normalement incrémenté de 1 à chaque envoi.
C'est ce nombre qui permet de retrouver les fragments issus d'un même paquet quand il a été fragmenté en plusieurs paquets.
Donc dès que j'envoie un paquet, l'IPID envoyé dans l'en-tête IP est incrémenté de 1 par rapport au paquet précédent.

- J'envoie un paquet : IPID = 2145 ;
- J'envoie un autre paquet : IPID = 2146 ;
- J'envoie encore un paquet : IPID = 2147 ;
- J'envoie encore un autre paquet : IPID = 2148...

Vous avez saisi ?

Nous sommes bien avancés, mais dans notre problème initial, nous voulions savoir si notre voisin avait envoyé un paquet, et pour cela il faudrait connaître la valeur de l'IPID avant et après notre envoi (celui où on se faisait passer pour Mme Michu).

Peut-on connaître l'IPID de notre voisin ?

La réponse est... peut-être.
Pour cela, il faudrait pouvoir l'obliger à nous envoyer un paquet pour pouvoir lire l'IPID dans l'en-tête IP de la réponse.

Comment obliger notre voisin à nous envoyer un paquet ?

~~En le menaçant de...~~ non, il doit y avoir plus simple.
Il suffit a priori de lui envoyer un segment SYN sur un port TCP ouvert, dans ce cas il nous répondra avec un segment SYN+ACK, et nous pourrons lire dans l'en-tête de couche 3 l'IPID !
Il suffit donc que le voisin ait une application en écoute sur sa machine en TCP, ce qui est pratiquement toujours le cas... tant qu'il n'a pas de firewall d'activé.

Comment savoir si notre voisin a des ports ouverts ?

Il suffit de faire un scan de ports.
Si nous trouvons au moins un port ouvert, c'est gagné !
Nous pouvons donc maintenant connaître l'IPID de notre voisin à un instant t. Nous avons tout ce qu'il faut, c'est à nous de jouer !

Le déroulement du scan complet

- Je trouve un voisin ayant un port ouvert (ou toute autre machine sur Internet...) ;
- Je lui envoi une demande de connexion avec un SYN ;
- Il me répond avec SYN+ACK et je lis dans l'en-tête de couche 3 son IPID qui vaut, par exemple, 100 ;
- J'envoie un SYN à Mme Michu avec comme adresse IP source l'adresse de mon voisin (Mme Michu pense donc que cette demande de connexion vient de mon voisin) ;

Il y a maintenant deux cas possibles :

- 1- Le port scanné de Mme Michu est ouvert.
- 2- Le port scanné de Mme Michu est fermé.

Cas numéro 1 :
Mme Michu répond à mon voisin avec un SYN+ACK. Mon voisin s'empresse de répondre avec un RST, son IPID a donc augmenté de 1 et vaut 101.
Je renvoie alors un SYN à mon voisin sur son port ouvert. Il me répond avec SYN+ACK et je lis son IPID qui vaut maintenant... 102 ! (Puisqu'il vient d'envoyer un nouveau paquet pour me répondre !)

Cas numéro 2 :
Mme Michu répond à mon voisin avec un RST. Mon voisin ne répond pas, son IPID reste donc à 100.
Je renvoie alors un SYN à mon voisin sur son port ouvert. Il me répond avec SYN+ACK et je lis son IPID qui vaut maintenant 101 ! (Puisqu'il vient d'envoyer un nouveau paquet pour me répondre.)

La différence entre les deux ?

Dans le premier cas, où le port de Mme Michu était ouvert, je reçois un IPID de 102.
Dans le second cas, où le port de Mme Michu était fermé, je reçois un IPID de 101.

Je suis donc capable de savoir si le port de Mme Michu était ouvert sans l'avoir scanné directement !

Mais...

Pour que cette attaque fonctionne, il faut se trouver dans certaines conditions favorables.

- Il faut trouver un voisin avec un port ouvert ;
- Il faut en plus que les IPID de ce voisin augmentent exactement de 1 à chaque envoi (ce n'est pas toujours le cas...) ;
- Il faut qu'il y ait peu de traffic vers ce voisin, sinon l'IPID peut augmenter à cause d'autres requêtes, et les résultats seront faussés.

Si ces conditions sont réunies (ce qui n'est pas bien compliqué à trouver) alors banco !

Limitations

Je peux donc scanner la maison blanche et faire porter le chapeau à mon voisin ?

Malheureusement **non**, ce serait trop beau.
Déjà, même si Mme Michu ne voit rien venir de vous, votre voisin, lui, a toutes les informations pour remonter à vous.
Si jamais la machine de votre voisin n'enregistre rien, les routeurs de son fournisseur d'accès eux le font et permettront très facilement de remonter à vous.
Donc comme pour les scans de ports simples, l'idle scan nécessite d'être effectué dans un environnement que vous maîtrisez et que vous connaissiez les conséquences associées.

La mise en pratique

La mise en pratique est on ne peut plus simple étant donné que l'outil nmap intègre déjà cette fonctionnalité.
Donc allons-y gaiement !

Etape 1, trouver un voisin

Pour cela, nous allons scanner une plage d'adresse pour voir les machines présentes dessus, en les *pingant*.

Code : Console

```
# nmap -sP 10.8.98.0/24
Starting Nmap 4.62 ( http://nmap.org ) at 2010-02-11 15:36 CET
Host 10.8.98.98 appears to be up.
MAC Address: 00:08:02:4F:08:7E (Compaq Computer)
Host 10.8.98.99 appears to be up.
MAC Address: 00:08:02:4F:09:2F (Compaq Computer)
Host 10.8.98.100 appears to be up.
MAC Address: 00:08:02:3F:EE:E3 (Compaq Computer)
Host 10.8.98.156 appears to be up.
MAC Address: 00:08:02:3F:E0:B5 (Compaq Computer)
Host 10.8.98.172 appears to be up.
MAC Address: 00:08:02:42:B1:75 (Compaq Computer)
Host 10.8.98.235 appears to be up.
Host 10.8.98.238 appears to be up.
MAC Address: 00:08:02:37:AE:EE (Compaq Computer)
Host 10.8.98.240 appears to be up.
MAC Address: 00:15:00:37:9F:A5 (Intel Corporate)
Host 10.8.98.241 appears to be up.
MAC Address: 0C:EE:E6:B5:8A:21 (Unknown)
Nmap done: 256 IP addresses (9 hosts up) scanned in 2.715 seconds
```

Il y a le choix !
Nous allons maintenant essayer de voir si l'une de ses machines a un port ouvert pour nous permettre de jouer le rôle du voisin.
Nous allons les scanner une par une, tout en essayant de connaître le système d'exploitation dessus (les derniers linux ayant une protection et incrémentant les IPIDs aléatoirement).

Code : Console

```
# nmap -sS -O 10.8.98.98
Starting Nmap 4.62 ( http://nmap.org ) at 2010-02-11 15:40 CET
Interesting ports on 10.8.98.98:
Not shown: 1706 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
111/tcp  open  rpcbind
143/tcp  open  imap
901/tcp  open  samba-swat
993/tcp  open  imaps
2049/tcp open  nfs
MAC Address: 00:08:02:4F:08:7E (Compaq Computer)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.24
Uptime: 34.971 days (since Thu Jan  7 16:21:17 2010)
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at http://nmap.org/subm
Nmap done: 1 IP address (1 host up) scanned in 1.992 seconds
```

Nous avons ici une foultitude de ports ouverts, mais malheureusement la machine semble tourner avec un noyau linux 2.6 ne permettant pas l'attaque.
Recherchons plutôt un windows.

Code : Console

```
# nmap -O 10.8.98.0/24
Starting Nmap 4.62 ( http://nmap.org ) at 2010-02-11 15:43 CET
Interesting ports on 10.8.98.98:
[Bla bla]
Running: Linux 2.6.X
```

```
[Bla bla]
Running: Linux 2.6.X
[Bla bla]
Running: Linux 2.6.X
[Bla bla]
Running: Linux 2.6.X

Interesting ports on 10.8.98.240:
Not shown: 1709 closed ports
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
1026/tcp open  LSA-or-nterm
MAC Address: 00:15:00:37:9F:A5 (Intel Corporate)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS details: Microsoft Windows XP SP2, Microsoft Windows XP SP2 or Windows Server 20
Network Distance: 1 hop
Nmap done: 256 IP addresses (9 hosts up) scanned in 21.354 seconds
```

Bingo ! Une des machines semble être sous windows XP SP2.
Elle a les ports windows standards ouverts et n'a donc pas de firewall d'activé.
Nous avons trouvé notre voisin idéal ! Le gagnant est 10.8.98.240 !

Réalisation de l'idle scan

Il ne nous reste plus qu'a réaliser l'attaque, pour par exemple scanner l'adresse 10.8.98.98 dont nous avons déjà scanné les ports précédemment. Nous allons utilise le port 445 trouvé ouvert sur notre voisin idéal.

Code : Console

```
# nmap -P0 -sI 10.8.98.240:445 10.8.98.98

Starting Nmap 4.62 ( http://nmap.org ) at 2010-02-11 15:52 CET
Idle scan using zombie 10.8.98.240 (10.8.98.240:445); Class: Incremental
Interesting ports on 10.8.98.98:
Not shown: 1706 closed|filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
111/tcp  open  rpcbind
143/tcp  open  imap
901/tcp  open  samba-swat
993/tcp  open  imaps
2049/tcp open  nfs
MAC Address: 00:08:02:4F:08:7E (Compaq Computer)

Nmap done: 1 IP address (1 host up) scanned in 17.545 seconds
```

Ca a marché !
Revenons un peu sur la commande.
`nmap -P0 -sI 10.8.98.240:445 10.8.98.98`
Le -P0 est obligatoire, sinon notre machine envoie un ping à Mme Michu qui sait maintenant qui la scanne. Le -P0 oblige nmap à ne pas faire de ping avant de scanner une machine.
Nous précisons ensuite l'option -sI indiquant que nous voulons faire un idle scan.
Puis qui sera notre voisin et quel port utiliser, 10.8.98.240:445.
Et enfin la victime, Mme Michu, 10.8.98.98.
Le résultat est conforme à ce que nous avions vu auparavant.

Un autre point est important dans le résultat :

```
Idle scan using zombie 10.8.98.240 (10.8.98.240:445); Class: Incremental
```

Le *Class: Incremental* ici est très important et nous confirme que notre voisin incrémente bien les IPID de 1 et non aléatoirement.

Que se serait-il passé si nous avions essayé avec une machine linux comme voisin ?

Malheureusement il n'est pas possible d'utiliser un tel OS car il utilise des IPID aléatoires...

Code : Console

```
# nmap -P0 -sI 10.8.98.99:80 10.8.98.98

Starting Nmap 4.62 ( http://nmap.org ) at 2010-02-11 16:01 CET
Idle scan zombie 10.8.98.99 (10.8.98.99) port 80 cannot be used because IP ID seque
QUITTING!
```

Le résultat est sans appel. Impossible de réaliser l'idle scan, les IPIDs du voisin ne sont pas bons...
Nous avons donc vu comment réaliser un *idle scan*.
L'intérêt d'un tel scan est de ne pas être vu directement par la machine scannée.
Cela peut être intéressant pour être furtif, mais aussi pour voir si un filtrage différent est appliqué en fonction des adresses IP sources. Par exemple pour voir si le filtrage est le même pour ma machine que pour celle d'un administrateur réseau...

Pour tous les fans de scans, je vous invite à poursuivre ce tuto par la lecture de l'excellent site nmap qui vous en dira plus sur les différentes méthodes de scan disponibles.