Le détournement de session TCP

But de cet article :

Le but de cet article est de comprendre la technique utilisé par K.Mitnick lorsqu'il a hacké le système de Shimomura en utilisant la technique de détournement de session. Cette technique consiste , lorsqu'il y a par exemple deux ordinateurs connectés entre eux , à se faire passer pour l'un des d'eux et à exécuter des commandes sur l'autre.Cette technique sert à pénétrer sur un système avec l'identité d'un autre afin d'y ouvrir une porte pour que n'importe qui puisse y revenir sans avoir a passer au travers des formalités d'authentification .Cette technique est difficile à réaliser mais est toujours faisable de nos jours.

Ingrédients :

Le point clé de cette attaque est la prédiction des numéros de séquence des paquets TCP afin de ' s'introduire ' en quelque sorte dans la connexion en cours.
Il faut pour cela que le trafic entre les deux ordinateurs soit nul (vous comprendrez mieux pourquoi plus tard) c'est d'ailleurs pour cela que Mitnick a effectué cette attaque la veille de Noël au soir.

Petit rappel de l'en tête TCP :

<table>
<tr><td colspan="3">PORT src</td><td>PORT dest</td></tr>
<tr><td colspan="4">No Seq</td></tr>
<tr><td colspan="4">No ACK</td></tr>
<tr><td>O</td><td>R</td><td>F</td><td>WIN</td></tr>
<tr><td colspan="3">Cheksum</td><td>URG</td></tr>
</table>

Processus en trois temps lors de la connexion d'un hôte A vers un hôte B :

A envoie un SYN à B (demande de synchronisation) (ex : seq=12000 , ack=0)
B renvoie un SYN/ACK à A (seq=8500 , ack=12001)
A envoie un ACK à B (seq=12001 , ack=8501)

La connexion est établie. Le point clé est le numéro de séq de B (que l'on ne peux avoir que dans le retour syn/ack)

Que va faire le hacker à présent ?

1ere phase : le hacker va envoyer une dizaine de demande de connexion sur le port shell de B et récuperer via un sniffer le numéro de séq contenu dans le syn/ack de retour.Que va t'il faire avec ?
Faire quelques soustractions , exemple du retour syn/ack de B:

1 : seq = 8500 ack=12001
2 : seq= 9000 ack= 12501
3 : seq= 9500 ack= 13001
4 : seq= 10000 ack= 13501
5: seq = 10500 ack = 14001
etc, etc…

Le hacker va déterminer si il y a un rapport constant à chaque paquets,car à chaque connexion est généré un nouveau numéro de séq., il va soustraire le paquet 4 au 5 , le 3 au 4 , etc etc…et regarder si le résultat est à chaque fois le même. Si c'est le cas on dira alors que le numéro de séquence est prévisible. Dernier point , sur ces dix connexions lancées le hacker ne va pas terminer chaque connexions et enverra donc comme 3eme temps un rst et non le ack habituel. Le but n'étant pas de faire du syn flooding mais de déterminer la réaction du numéro de séq. Dans cet exemple le numéro est incrémenté de 500 à chaque connexions.

2ème phase : le hacker va flooder l'hôte A (qui est l'ordinateur dont il va usurper l'identité) par un flot de requêtes SYN mais contenant cette fois comme IP source un adresse inexistante.
Pourquoi ?
Il est un fait que chaque système peut soutenir un nombre limité de connexion en attente. Exemple , vous demandez une connexion sur le port telnet d'un ordi (vous envoyez donc le syn) et avant même que la réponse revienne votre système est déconnecté pour une raison ou une autre, bref l'hôte qui vous reverra le syn/ack l'enverra donc bien mais comme l'adresse n'existe plus le paquet ira se balader de routeurs en routeurs et le port (23 dans ce cas) sera donc en attente de cette connexion.
D'une part chaque système selon qu'il est configuré , peut attendre 30s comme 10 min avant d'éjecter ces paquets, et d'autre part ne peut accepter qu'un nombre limité de ce genre de connexions. Le but du hacker va être donc de saturer la capacité d'attente du système , et bien entendu aucune connexion ne pourra être échangée avec un hôte bloqué de cette façon.

Le hacker va donc inonder de requêtes syn l'hôte A afin qu'il ne puisse plus échanger un seul paquet avec B . Mais pourquoi donc ?

3ème phase :

Le hacker va à présent envoyer une commande à l'hôte B en se faisant passer pour A.
1) Il va donc envoyer à B un Syn de connexion avec l'IP de l'hôte A
2) B va retourner le Syn/ack non pas au hacker mais à l'hôte A (à cause de l'ip usurpée)
NB :
C'est pour cette raison qu'il à été important de prédire le numéro de séquence car celui de l'hôte B n'est visible que dans le syn/ack de retour (rappel…)
Autre point important , le fait que A ne puisse pas répondre, car s'il le pouvait , il retournerait à B un RST (A n'ayant pas demandé de connexion et recevant comme premier paquet un syn/ack) et couperait court à la tentative de

connexion du hacker.

3) le hacker va fabriquer le 3ème paquet avec le bon ack (voir le processus de connex expliqué plus haut).

NB :

Lors de l'envoi des paquets pour tester le num de séquence de B , il prendra le dernier paquet recu (par exemple , le dernier de mon exemple de la phase 1) qui est seq=10500 et comme il connaît l'incrémentation de chaque paquet (+500) il saura que le prochain paquet (celui qu'il envoie à la phase 3) aura dans la réponse syn/ack 10500+500=11000

Le 3ème temps donc le ack sera donc seq=No de A et ack = 11001

Il lui reste alors maintenant à envoyer un paquet contenant la commande adéquate (le cas de Mitnick est un cas Unix avec ses fichiers rhosts) qui permettra de donner un accès à n'importe qui sur la machine B.

La 4ème et dernière phase sera pour le hacker de vider la file d'attente qui bloque l'hôte A par quelques RST bien placés afin que tout revienne à la normale.

La difficulté reste aussi du côté du trafic entre les hôtes A et B qui doit être nul sous peine d'avoir fait bouger le numéro de séquence.

Repérer un détournement de session commence par surveiller l'activité de son réseau , détecter les prises d'empreintes et installer un Ids qui relève le traffic et particulièrement dans ce cas le champ TTL des paquets, analyser quels sont les hôtes qui restent connectés et sans surveillance.

Le firewall n'est ici d'aucune utilité car le hacker se fait passer le temps d'une commande par un hôte approuvé du reseau.

Auteur : Philippe.F

Date : 14 juin 2001