I°) Les Denial of Service (DoS)

Notre but n'est pas de fournir des outils tout prêts à utiliser afin d'effectuer des DoS performants. Ceci dit, il nous paraît important de se familiariser avec les techniques de détournement de TCP/IP (TCP/IP Hijack), c'est pourquoi nous allons ici vous exposer brièvement la théorie de certaines techniques de Déni de Service. Une attaque de type DoS vise tout simplement à empêcher la victime (un poste personnel, un réseau) de communiquer, c'est-à-dire l'isoler, en le faisant crasher ou en lui envoyant beaucoup plus de requêtes qu'il n'est capable de traiter afin de remplir sa bande passante (Flood). Voici donc 5 DoS très communs.

Le ping de la mort
Dans les spécifications du protocole ICMP, les paquets echo n'ont été conçus que pour contenir 216 octets de données dans le corps du paquet. A vrai dire, cela importe peu, puisque dans le cas des requêtes et réponses ICMP, seuls les en-têtes importent, aucun réel message n'est nécessaire. Un DoS très connu est simplement l'utilisation de minimum 65 537 (= 216 + 1) octets de données. Le système recevant le message paniquera à la réception du paquet et crashera. Cette attaque est très ancienne et connue comme The Ping of Death. Seuls les systèmes relativement anciens sont affectés par cette attaque car elle a bien sûr été patchée depuis. Ceci dit, elle illustre très bien comment les personnes ayant conçu ces systèmes ne se préparaient pas à l'éventualité que quelqu'un puisse sortir du protocole attendu.

Le ping-flood
Cette attaque est la plus commune et la plus connue. Cette fois, on ne veut pas crasher l'ordinateur distant mais le saturer, de façon à ce qu'il ne puisse plus communiquer. Les attaques de flood TCP/IP essaient de simplement surcharger la bande passante sur le réseau de la machine concerner et ainsi l'isoler. En pratique, cette attaque consiste juste à envoyer beaucoup de requêtes PING à la victime, avec des paquets contenant beaucoup de données. Cette attaque n'a rien de spécialement intelligente car elle se résume à un combat entre bandes passantes, et que la meilleure gagne..

Les attaques amplifiées
A ce stade, l'utilisation de TCP commence à devenir intéressante. Les attaques amplifiées sont juste une façon d'exploiter des floods, mais intelligemment. Nous avons parlé dans notre article sur l'adressage dans la couche liaison de données d'une adresse spéciale, la broadcast. Cette adresse donne la possibilité d'envoyer un paquet à toutes les machines connectées au réseau. Ainsi, si on envoie une requête PING à la broadcast, tous les systèmes connectés nous enverrons un paquet ICMP reply en retour. Comme se flooder soi-même n'est pas d'une immense utilité, il faut trouver un moyen de faire répondre tous ces systèmes à qui on l'entend, car cela représente un gros potentiel. La solution est simple : envoyer un paquet PING echo spoofé contenant l'adresse de la victime comme source et la broadcast comme destinataire. Ainsi, chaque système du réseau recevra un ping qu'ils croieront venir de la machine de la victime et y répondront. En envoyant beaucoup de paquets spoofés, la bande passante de la victime est prise d'assaut de manière exponentielle avec le nombre de systèmes connectés au réseau.
Une manière de se prémunir de ces attaques est d'interdir les pings sur la broadcast ou de ne les rerouter que vers le serveur principal, ce qui est couramment effectué dans les grands réseaux (universités, entreprises). Cette attaque s'appelle communément un smurf.

Le flood DDoS (Distributed Denial of Service)
Comme nous l'avons fait remarquer pour l'attaque de type ping-flood, le flooding revient à un combat entre bandes passantes. L'idée du Déni de Service Distribué est simple et rejoint celle du smurf, à savoir, pourquoi faire le combat tout seul et pas à plusieurs ? Effectivement, plus il y a d'ordinateurs qui floodent une seule machine ou un seul réseau, plus ça devient

facile de saturer sa bande passante, toute immense qu'elle puisse être. Les attaques de flood les plus remarquables de ces années suivent ce principe, on peut notamment nommer l'attaque des 13 root servers (les serveurs qui maintiennent l'Internet) en Février dernier (2007). Cette attaque avait été particulièrement simple : environ 5 000 machines ont submergé les root servers de requêtes DNS et UDP. Au moins 2 serveurs sont tombés dans cette nuit du 6 au 7 Février et au moins 3 autres ont étés saturés, ce qui s'est ressenti parfois par un très léger lag sur l'Internet. Vous nous direz que rassembler 5 000 machines n'est pas à la portée du premier venu. Et bien si. En fait, ces machines ne sont autres que des machines lambda, comme vous ou nous qui ont étés affectés par un virus, reproducteur, qui est resté silencieux sur tous ces postes jusqu'à ce que leur "maître" leur demande de passer à l'action. Tous les postes infectés étaient sous Windows et plus de 65% en Corée du Sud, ce qui laisse penser que là-bas se trouvait le chef-lieu de l'attaque. De plus, utiliser autant d'ordinateurs à la fois permet naturellement de rendre quasiment impossible le traçage de la source du flood.

Le SYN flood
Cette manière d'exploiter TCP/IP est beaucoup plus maligne. Elle se sert d'une limitation obligatoire du protocole : la pile TCP/IP. Comme un ordinateur doit connaître l'état des connexions en cours pour pouvoir notamment les maintenir ou les établir, il faut stocker les informations (notamment numéros de reconnaissance et de session) quelque part, c'est ce qu'on appelle la pile TCP/IP. Or, il est bien sûr impossible d'avoir une pile TCP/IP infinie, par conséquent le nombre d'initialisations de connexion que peut surveiller un simple ordinateur est limité. Ainsi, l'attaquant floode la victime avec beaucoup de paquets SYN spoofés (d'adresses prises au hasard). La victime va donc répondre à ces paquets SYN par un paquet ACK/SYN en attente d'un paquet ACK, comme la coutume le veut. Puisque les réponses ne viendront pas, la machine gardera dans la pile les informations sur chaque host qui a demandé l'initialisation d'une connexion. Pour que ces connexions soient enlevées de la queue, il faut qu'elles timeout, ce qui prendre relativement longtemps. Par conséquent, tant que l'attaquant continue l'envoi des paquets SYN, aucun autre réelle connexion ne pourra parvenir à la cible et elle sera isolée.

De la même façon que ces attaques se servent du protocole TCP pour mener à bien leurs objectifs malveillants, il est possible de scanner un ordinateur de façon beaucoup plus savante que celle utilisée par les scanners classiques (vérification de l'établissement d'une connexion TCP/IP), c'est ce que nous nous proposons de vous expliquer maintenant.

II°) Les scans intelligents

Tout comme dans la page précédente, nous ne vous donnerons pas ici de code permettant de scanner des ordinateurs distants. Tout d'abord, des scanners de vulnérabilité tels Nmap ou Nessus contiennent ces fonctionnalités. De plus, après ces explications théoriques il devrait vous être facile de construire de tels scanners. Ces techniques sont surtout effectuées pour contourner les méthodes traditionelles de détection des scans.

Le scan SYN, dit scan "mi-ouverture"
L'idée de ce scan est tout simplement de ne pas établir une connexion complète en omettant la troisième étape de l'initialisation de la connexion. En fait, on envoit un paquet SYN sur le port concerné. Si le port est en écoute, un paquet SYN/ACK est renvoyé, le port est donc ouvert. On envoie un paquet RST pour terminer la connexion (nécessaire car si on scanne beaucoup de ports sans RST, on peut DoS la victime de la même façon qu'un Flood SYN pourrait le faire). Cette technique simplissime évite l'ouverture complète de la connexion et ainsi le log de l'IP par la majorité des services ayant accepté une connexion. Sur les systèmes un minimum sécurisé, les demi-connexions sont repérées et logguées, les trois scans suivants sont donc apparus.

Les scans FIN, X-mas ou NULL

Ces trois techniques préfèrent envoyer des paquets qui n'ont aucun sens lorsque que la connexion est fermée et s'appuyer sur les différences d'interprétation du serveur. Effectivement, si un paquet quelconque est envoyé sur un port ouvert, il sera ignoré puisque ce port attend avant toute chose un paquet SYN. Par contre, si le port est fermé, la RFC 793 prévoit le retour d'un paquet RST. En utilisant cette différence, on peut déterminer les ports ouverts et les ports fermés. Le scan FIN envoie un paquet FIN, le scan X-MAS envoie un paquet FIN/URG/PSH et le NULL, vous l'avez compris, envoit un paquet avec tous les flags TCP à off. Attention, cette technique ne marche pas sur certains OS Microsoft, car le géant de l'informatique n'aimant pas à faire les choses comme tout le monde, ne suit tout simplement pas la RFC... Bien sûr, patcher son kernel en supprimant les RST des ports fermés est très efficace contre ces scans. Dans la source de linux, on affiche le code de l'implémentation de TCP pour IPv4 (dans /usr/src/linux-source-2.6.18/net/ipv4/tcp_ipv4.c pour un kernel 2.6.18 par exemple). On cherche la fonction d'envoi des RST :

```
static void tcp_v4_send_reset(struct sk_buff *skb)
{

    o

            return;

        struct tcphdr *th = skb->h.th;
        struct tcphdr rth;
        struct ip_reply_arg arg;

        /* Never send a reset in response to a reset. */
        if (th->rst)
        [...]
}
```

Pour patcher cette vulnérabilité, il suffit de ne plus envoyer de paquets RST. Pour ce, on ajoute un simple **return;** après les déclarations des variables, ce qui permet de stopper directement la fonction et de ne jamais envoyer de paquets RST :

```
static void tcp_v4_send_reset(struct sk_buff *skb)
{

    o

            return;

        struct tcphdr *th = skb->h.th;
        struct tcphdr rth;
        struct ip_reply_arg arg;

        return; //On quitte directement : aucun paquet RST transmis

        /* Never send a reset in response to a reset. */
        if (th->rst)
        [...]
}
```

Et après recompilation, notre système sera immunisé contre un scan différentiant les ports envoyants des RST et les autres.

Spoofer l'host scannant

Une première technique simple pour éviter la détection du scan est de faire participer d'autres hosts du réseau au scan : par exemple, entre chaque test de port, on envoie un paquet spoofé à la cible sur un port quelconque. Ainsi, on ne lancera jamais plus d'une requête à la suite et la victime ne détectera pas un scan massif de ses ports. Vous l'aveez remarqué, les hosts spoofés doivent exister pour éviter que la cible subisse un flood SYN et à terme un DoS.

La deuxième méthode est beaucoup plus technique, elle utilise un ordinateur du réseau inactif (qui n'as pas d'autres échanges réseaux au moment X du scan). Le principe repose sur le changement prédictible de l'IP ID, incrémenté à chaque paquet que le système concerné envoie. Quand l'incrémentation est fixe (on augmente toujours du même nombre d'octets l'IP ID), on peut prédire le prochain nombre. Cette implémentation de TCP est courante, par exemple pratiquement tous les Windows incrémentent leur IP ID de 1 ou 254 (en fonction de l'agencement des bytes) selon les kernels. Notre but ici est d'observer les changements opérés sur l'host inactif afin de détecter ou non des changements dans l'IP ID pour savoir s'il a reçu des informations, explications.

Tout d'abord, l'attaquant envoie plusieurs paquets SYN/ACK au système inactif. A chaque fois, il note l'IP ID et peut ensuite facilement déterminer l'incrémentation que suit le système. Ensuite, l'attaquant fait parvenir à la victime un paquet SYN spoofé avec l'ip inactive en source. Si le port est ouvert, la victime recevant une demande de connexion, elle renvoie un paquet SYN/ACK à la machine inactive. Mais puisque l'host inactif n'a pour lui fait aucune demande de connexion, il va envoyer un paquet RST à la victime. Avec cet envoi, l'IP ID est incrémenté. Dans le cas où le port est fermé, la victime enverra un paquet RST ou rien, ce qui n'occasionnera pas de retour de paquet de la part de l'host inactif.

Ensuite, l'attaquant réenvoie un paquet ACK/SYN à l'ordinateur inactif pour recevoir l'actuel IP ID. S'il a été incrémenté d'un incrément, un seul paquet a été envoyé, donc le port est fermé (le paquet envoyé étant le RST envoyé à la suite du ACK/SYN non-sollicité de l'attaquant). S'il a augmenté de deux incréments, un autre paquet a été envoyé, a priori le paquet RST envoyé à la suite du ACK/SYN non-sollicité de la victime : le port est donc ouvert.

Cette technique est finalement puissante car l'attaquant peut ainsi scanner sans à aucun moment n'avoir besoin de réveler son IP.

Une fois les sections suivantes terminées, nous ajouterons un article sur la défense proactive, qui permet de faire croire que tous les ports sont ouverts, même si cela est faux. Nous allons étudier maintenant étudier les deux attaques réseau les plus puissantes : l'ARP Poisoning (de type MiM, *Man in the Middle*) et l'IP Spoofing, ainsi que les manières communes de les empêcher.

(Source : BasesHacking)