I°) Le modèle OSI

Le modèle OSI (Open Systems Intercommunication, ou communication entre systèmes ouverts) fournit des règles et standards internationaux qui permettent à n'importe quel système qui obéit à ces protocoles de communiquer avec n'importe quels systèmes les utilisant aussi. Dans le modèle OSI, ces règles sont séparées en 7 couches communiquantes. Chaque couche s'occupant d'un aspect différent de la communication. Ce modèle a été établit par l'ISO (International Standards Organisation). Nous allons maintenant vous présenter brièvement chacune des couches du modèle OSI :

La couche physique (Physical Layer) est la couche la plus basse. Comme son nom l'indique, elle permet la connexion physique entre deux instances communiquantes. Autrement dit, elle maintient, créé, désactive ou transmet des flux d'octets entre deux systèmes. Par conséquent, cette couche est sollicitée dans n'importe quel type de communication.

La couche de liaison des données (Data-Link Layer) se préoccupe du transfert de données entre les deux systèmes interconnectés. Elle complète la couche physique, qui n'émettait que de simples octets sans signification et sans erreurs (la couche physique vérifie que 1 = 1 et que 0 = 0 en début et fin de connexion), en des transmissions plus complexes, pourvues de systèmes d'erreurs et de contrôle du flux. Elle segmente les données à envoyer et les gère. Elle est capable de maintenir, détruire ou créer un envoi de données entre deux systèmes.

La couche réseau (Network Layer) est une couche primordial de l'organisation des réseaux informatiques, car elle gère l'adressage et le routage sur le réseau. Cette couche, au milieu des autres, a pour but de transmettre les données entre les couches basses et les couches plus hautes.

La couche de transport (Transport Layer) est responsable de l'intégrite des paquets transmis des couches supérieures à la couche réseau. En réception, elle réassemble les segments de données pour qu'elles soient traitées par les couches supérieures. Elle permet aux couches supérieures de s'occuper d'autres choses que l'efficacité ou les moyens de transmission effectifs

La couche de session (Session Layer) est responsable de l'établissement et du maintien d'une connexion entre les applications communicantes au travers des différents systèmes.

La couche de présentation (Presentation Layer) s'occupe comme son nom l'indique de la présentation des données de manières c ompréhensible à l'application locale communiquante (par exemple, encryptage ou compression des données).

La couche d'application (Application Layer) transmet simplement les ordres et besoins de l'application aux couches inférieures.

Nous avons eu l'occasion de citer la segmentation en paquets des données lors des communications. Ces paquets sont donc des "morceaux" de données, chacuns utilisant scrupuleusement les protocoles de communication établis (un peu comme une suite de paquets passant dans un bureau de la poste, triés par destination, étiquettés, etc..).
Partant de la couche de l'application, un paquet va traverser toutes les couches, de la plus haute à la plus basse, jusqu'à la couche physique. Ce procédé est appellé communément l'encapsulation du paquet. Chaque paquet contient des en-têtes et un corps. Les en-têtes

définissent le protocole de communication suivi, le type de données présentes, leur longueur. Le corps contient les données transmises elles-mêmes. Ainsi, chaque couche apporte sa contribution au corps du paquet en ajoutant des données à celles inscrites par les couches qui lui sont supérieures.
Souvent, on métaphorise les couches réseaux à l'aide d'une entreprise d'envoi constituée de sept services différents. Chaque service est spécialisé dans son travail et n'est pas capable d'effectuer le travail des autres. Quand le service d'enpaquettage a mis les données primaires dans le paquet, il met le paquet dans un sac inter-service, n'oublie pas de remplir l'étiquette attachée au sac pour que le service suivant comprenne ce qui a été fait et l'envoi. Ainsi, chaque service fait son office et le paquet arrive au service d'envoi qui l'expédie à l'adresse indiquée.
Quand deux systèmes communicants font transiter des paquets par des sources intermédiaires (routeurs, réseau local, internet, etc..), les paquets ne se transmettent qu'aux couches physique, liaison de données et réseau, puisque arrivé ici, le réseau reroute le paquet vers sa destination réelle. Ainsi, les données ne sont pas traitées par les systèmes non concernés par l'échange.
Après ces connaissances de base sur l'architecture réseau, nous allons désormais étudier trois couches particulièrement intéressantes et dont la connaissance est nécessaire pour les sections qui suivront.

II°) La couche réseau

Cette couche est responsable de l'adressage (envoyeur et destinataire) ainsi que de la méthode d'envoi utilisée pour la transmission. Par exemple, la couche réseau utilise un protocol bien connu pour la transmission sur Internet, appellé Internet Protocol, ou IP. Ici, nous parlerons exclusivement de la version 4 du protocole (IPv4), IPv6 étant toujours plutôt obsolète.
Chaque système connecté à Internet a une adresse IP. Elle consiste en une succession de 4 bytes de la forme xxx.xxx.xxx.xxx. Par exemple, l'IP de bases-hacking est 213.186.33.87, la votre est 62.166.196.0 (ou du moins celle que vous utilisez pour naviguer). Dans cette couche sont traités les paquets IP et ICMP (Internet Control Message Protocol). Les paquets IP servent à envoyer les données et les paquets ICMP fournissent des moyens de diagnostic de la connexion et de la transmission (comme le PING par exemple). Par exemple, si un paquet IP n'arrive pas à destination, un paquet ICMP est renvoyé au destinataire pour le prévenir de l'erreur intervenue.
Comme cité plus haut, ICMP sert aussi au test de la connectivité, notamment au moyen de la commande PING avec les requêtes et réponses ICMP Echo. Si un système veut tester la possibilité d'envoi d'un paquet à un client, il lui envoi un paquet ICMP Echo Request et reçoit un paquet ICMP Echo Reply si tout se pase bien. De plus, la différence entre le départ de l'arrive de ces paquets permet aussi de déterminer le temps de latence, ou lag entre les systèmes.
De plus, IP est capable de fragmenter ses propres paquets (par exemple pour répondre aux exigences d'un système interdisant les paquets longs). Ainsi, il va transformet un paquet constitué d'une en-tête et d'un corps en une multitude de paquets dont le premier sera constitué de l'en-tête avec le début du corps, le deuxième avec l'en-tête et la suite du corps, etc.. jusqu'au dernier paquet constituté de l'en-tête de la fin du message. Chaque paquet contient dans l'en-tête son numéro dans l'agencement de la segmentation des données. On comprend qu'il est primordial que chaque paquet contienne l'en-tête pour pouvoir réassembler aisément le paquet à l'arrivée.
Comme expliqué dans la page précédente, le travail de segmentation et d'assemblage dépend de la couche de transport que nous allons étudier maintenant.

III°) La couche de transport

La couche de transport est finalement le premier niveau de traitement des paquets, puisque immédiatement après la couche de routage. La couche de transport est une couche de transition où passent les paquets entre le traitement réel des données et le traitement de l'en-tête et de la gestion du flux. En fait, elle s'occupe du retour, de l'envoi, des authorisations d'envois des paquets.
Les deux protocoles majeurs vous sont sûrement familiers de nom : ce sont UDP (User Diagram Protocol) et TCP (Transport Control Protocol). TCP est le protocol le plus utilisé à travers l'internet (pour tout ce qui est HTTP, SSH, FTP, Telnet, SMTP, POP, IMAP, etc..). TCP permet une liaison bidirectionnel (envoi/réception), plutôt efficace (vérification de l'intégrité et de l'ordre des données reçues et envoyées) et transparente entre deux adresses IP, c'est ce qui fait qu'il est très utilisé. Une particularité de la vérification de l'intégrité pour TCP/IP est l'attente des paquets manquants, c'est-à-dire que si un paquet numéroté X arrive, tous les paquets suivants seront empilés dans l'attente de l'arrivée du paquet X+1 pour assurer la continuité du message transmis.
Tout ceci est rendu possible par l'inscription de marques (TCP flags) sur les paquets ainsi que le stockage de nombres particuliers appellés les nombres de séquence (sequence numbers).
Voici une description brève des 6 marques TCP :
URG, pour urgent permet d'identifier les données importantes
ACK, pour reconnaissance (Acknowledgment). Cette marque reconnait l'activité de la connexion. Elle est à on pour la majorité de la connexion.
PSH, pour pousser Push. On force le passage à la couche supérieure plutôt que de stocker le paquet en mémoire tampon
RST, pour réinitialisation (Reset). Réinitialise une connexion.
SYN, pour synchronisation (Synchronize). Synchronise les nombres de séquence pendant le début de la connexion.
FIN, pour finir (Finish). Termine une connexion de façon propre.

Nous allons maintenant par un exemple simple expliquer l'établissement d'une connexion TCP avec les flags ACK et SYN, s'effectuant en 3 étapes.
Quand un client veut ouvrir une connexion avec un serveur, un paquet comportant la marque SYN (ie, SYN est à on) et avec la marque ACK à off. Le serveur répond avec un paquet comportant ACK et SYN à on. Enfin, le client renvoie un paquet avec la marque SYN à off et ACK à on : la connexion est reconnue. Ensuite, chaque paquet durant la connexion comportera ces deux paquets dans le même état (ACK à on et SYN à off). Par conséquent, seuls les deux premiers paquets peuvent comporter le flag SYN on, car c'est pendant ces deux premières transmissions que chaque côté synchronise les nombres de séquence. Résumons :

- Le client envoie un paquet SYN. Le paquet comporte un numéro de séquence égal à 123456 (par exemple) et un numéro de reconnaissance nul

- Le serveur renvoie un paquet SYN/ACK. Le paquet comporte un numéro de séquence égal à 654321 (par exemple) et un numéro de reconnaissance de 123457 (c'est-à-dire le numéro de séquence du client + 1)

- Enfin, le client envoie au serveur un paquet ACK avec comme numéro de séquence 123457 (le numéro de reconnaissance reçu) et comme numéro de reconnaisance 654322 (le numéro de séquence du serveur + 1)

Et la connexion est établie durablement. Les numéros de séquence sont ainsi utilisés pour assurer la fiabilité et la remise des paquets dans l'ordre, typiques de la couche de transport. De plus, ceci empechera les paquets venant d'une autre connexion d'être accidentellement mélangés car, quand une connexion est établie, chaque côté génère un nombre de séquence initial. Ce nombre est communiqué à l'autre partie au biais des deux premières étapes explicitées ci-dessus. Pendant le reste de la communication, chaque partie incrémentera son nombre de séquence du nombre de bytes de données dans le paquet envoyé. Ce numéro de séquence est inscrit dans les en-têtes du paquet. Aussi, chaque côté a, comme montré dans l'exemple précédente, un numéro de reconnaissance qui est le numéro de séquence de l'autre côté incrémenté de 1.
Par conséquent, la fiabilité des transmissions TCP semble relativement forte et c'est pourquoi il est souvent préféré dans les connexions bidirectionnelles.
UDP, lui, a beaucoup moins de fonctionnalités que TCP et finalement ressemble plus à IP : il n'y a pas de connexion qui reste ouverte et sa fiabilité est plus que faible. Plutôt que d'établir une connexion qui maintienne la véracité des données, UDP laisse à l'application le soin de s'occuper de ces problèmes d'identification des données. Ainsi, quand une connexion bidirectionnelle n'est pas requise, UDP paraît tout de suite plus adapté.

IV°) La couche de liaison de données

Cette couche permet d'adresser et d'envoyer des paquets n'importe où sur un réseau. Cette couche met en oeuvre le très célèvre Ethernet. Cette couche fournit un adressage standard pour tous les périphériques Ethernet : c'est l'adresse MAC (Media Access Control). Chaque périphérique Ethernet est pourvu d'une adresse MAC unique : c'est un moyen d'identifier de façon sûre n'importe quel périphérique établissant une connexion sur un réseau. Cette adresse est constituée de 6 bytes de la forme xx:xx:xx:xx:xx:xx, généralement communiquée en hexadécimal.
Les headers Ethernet contiennent une source (l'envoyeur) et une destination, ce qui permet a la couche réseau de router les paquets. De plus, une adresse spéciale existe sur un réseau, la broadcast, d'adresse MAC FF:FF:FF:FF:FF:FF et en général d'IP 255.255.255.255. La broadcast est un alias qui concerne tous les systèmes connectés au réseau. Autrement dit, un paquet envoyé à la broadcast sera envoyé à tous les périphériques appartenants au réseau.
L'adresse MAC ne peut a priori pas changer puisqu'elle est inscrite dans la mémoire intégrée du circuit électronique de chaque périphérique. Ceci dit, l'IP du système peut changer sur un réseau : on ne peut donc pas relier une IP à un système précis. Dans les faits, il existe un protocole qui permet de lier une adresse IP à une adresse MAC, c'est le très célèbre protocole ARP (Address Resolution Protocol)

Il y a 4 types de messages ARP : les requêtes et réponses ARP ainsi que les requêtes et réponses RARP (ARP/RARP requests ou replies). Dans l'optique de la section sur l'ARP Cache Poisoning, nous n'expliquerons ici que les paquets ARP.
Une requête ARP est un message qu'un ordinateur enverra à la broadcast qui demande "Qui a cette adresse IP ? Si c'est vous, envoyez la réponse à l'adresse MAC suivante". Le message est diffusé à tous les ordinateurs du réseau. Si cette IP existe sur le réseau, l'entité concernée va répondre en adressant une réponse ARP à l'adresse MAC indiquée dans la requête, disant "Mon adresse MAC est la suivante, et mon IP est ceci". En général, on garde en mémoire temporairement les associations MAC/IP, de façon à ne pas avoir à envoyer une requête ARP pour chaque paquet envoyé au même destinataire. Résumons ceci par l'exemple suivant, où on

prend deux ordinateurs A et B d'un réseau, d'adresses MAC et IP 12:34:56:78:9A:BC/192.168.0.101 et DE:F1:23:45:67:89/192.168.0.102 respectivement :

- L'ordinateur A forge le paquet ARP suivant : "Requête ARP - MAC Source : 12:34:56:78:9A:BC - MAC Destinataire : FF:FF:FF:FF:FF:FF - "Qui est à 192.168.0.102 ?", puis l'envoie.

- Après que la broadcast est diffusé le paquet sur tout le réseau, l'ordinateur B reçoit le paquet écrit précdemment. Reconnaissant son adresse MAC, il renvoie le paquet suivant : Réponse ARP - MAC Source : DE:F1:23:45:67:89 - MAC Destinataire : 12:34:56:78:9A:BC - "192.168.0.102 est à DE:F1:23:45:67:89".

- L'ordinateur A garde localement dans son cache la correspondance entre le MAC DE:F1:23:45:67:89 et l'IP 192.168.0.102.

Ainsi, si un programme tournant sur la machine A veut contacter 192.168.0.102 et qu'il ne trouve aucune correspondance dans le cache local, il va envoyer la requête ARP précédemment décrite. Après avoir reçu la requête et sauvegardé la correspondance, il peut désormais communiquer avec la machine B. Il est à noter que généralement, le routeur ou serveur central dans un réseau garde dans son cache local la correspondance MAC/IP de tous les ordinateurs du réseau.
Nous vous avons désormais décrites les trois couches qui seront essentielles dans ce que nous nous proposons de vous expliquer désormais, à savoir l'empoisonnement des caches ARP et le détournement du protocole TCP/IP.

(Source: "BasesHacking")