


Corbeille



Système d'exploitation

kali@kali: ~

FichierActionsÉditerVueAide

(kali@kali)-[~]

\$

MAN IN THE MIDDLE (MITM)

Technique de cyber-attaque visant à prendre secrètement le contrôle du canal de communication entre 2 machines dans le but d'intercepter de modifier ou de remplacer le trafic.

kali@kali: ~

FichierActionsÉditerVueAide

(kali@kali)-[~]

\$

Pour réaliser cette cyber-attaque il existe plusieurs méthodes différentes en fonction de l'environnement que nous voulons contrôler et de notre objectif. Dans ce rapport je vous présenterai 2 méthodes : le **ARP Poossing** et le **Stripping SSL/TLS**

L'Usurpation ARP

Tout d'abord pour que des données confidentielles atteignent une destination il doit être encapsulé par des informations de transit servant à assurer la livraison. On appelle ce bloc de données un **paquet**. Un paquet envoyé par l'hôte émetteur se sert non seulement de l'adresse IP mais aussi de l'adresse MAC de l'hôte receveur afin d'assurer son acheminement, il récupère ses informations dans une table stockée dans son système appelé **table ARP**.

ARP est un protocole servant à mettre en lien une adresse IP avec une adresse MAC, son fonctionnement est basé sur un envoi de **requête ARP** en broadcast à chaque machine du réseau en leurs demandants :
"Peut tu m'envoyer ton adresse MAC associé à ton adresse IP ?"
Une fois la demande reçue chacune des machines répondra en lui envoyant une **réponse ARP** contenant son adresse IP et son adresse MAC. Et enfin le demandeur de la requête enregistrera en cache dans sa table ARP les réponses contenant les adresses IP et MAC de chaque machine.

Il utilisera donc par la suite cette table pour compléter les paquets avec les adresses IP et MAC des destinataires. Malheureusement ce système n'est pas **sécurisé** , pour cause la table ARP étant configurée par défaut en mode **dynamique**, dans le but de mettre à jour sa table à chaque nouvelle réponse correspondant à une adresse IP qu'elle possederait déjà. Elle peut donc être facilement modifié a la guise d'un attaquant en envoyant une réponse ARP **falsifier**, contenant son adresse MAC lié à une autre adresse IP d'hôte. Ainsi chaque paquet envoyé à partir de la machine corrompue passera d'abord par l'adresse MAC de l'attaquant pour rejoindre l'adresse IP de la destination, si l'attaquant utilise cette même méthode sur l'IP de la seconde machine, alors il servira de **relai** de communication entre les victimes. Par conséquent, l'attaquant se retrouvant au milieu de la communication peut **intercepter, modifier et remplacer** tous les paquets qui transitent entre les victimes, allant des données sans importance aux données confidentielles.

kali@kali: ~

FichierActionsÉditerVueAide

(kali@kali)-[~]

\$

Une attaque comme l'usurpation ARP peut devenir un vértiable danger de sécurité si la victime se connecte à un site **HTTP** qui transmet les messages sans chiffrement, en entrant donc ses logins et mot de passe l'attaquant pourra donc récupérer ses informations et les utiliser dans un but malveillant. C'est pour cela que le protocole **HTTPS** est apparus tres rapidement. Assurant un **chiffrement des données** et une **authentification** des serveurs afin d'éviter la possibilité des attaques MITM. Mais comme chaque systeme informatique il y à une faille a celui si.

Stripping SSL/TLS

Le protocole **HTTPS** se sert d'un autre protocole de sécurité appelé TLS ou anciennement appelé SSL lui permettant donc de rajouter une **couche de sécurité** au protocole HTTP. Les sites en HTTPS vont donc envoyer à leurs clients lors de la première connexion une **clé de certificat** assurant l'authentification du serveur où est installé le site web sur lequel il est en train de se connecter. Le système du client accuse en réception la **clé de chiffrement** du serveur et lui envoie sa propre clé permettant au serveur de l'identifier. Un attaquant se trouvant donc entre les 2 ne pourra pas **déchiffrer** les données transmises, sauf si...
Sauf si l'attaquant arrive à **rétrograder** la connexion HTTPS du client au serveur en une connexion HTTP, ce qui lui permettrait donc de revenir à une version du protocole obsolète et non sécurisé une methode appelé **Le Stripping SSL**. Une fois la connexion dégradée l'attaquant va pouvoir effectuer une attaque MITM et s'interposer lors de la prochaine connexion en HTTPS du client au serveur. Il recuperera la clé de certificat envoyé par le serveur la modifiera pour envoyer **une clé de certificat falsifier** au client, qui aura donc le choix de l'accepter ou non, dans le cas où il accepterait alors la connexion se poursuivra et l'attaquant renverra la clé du client pour établir une connexion SSL avec le serveur et en **parralele** avoir une pseudo connexion SSL avec le client. À ce stade l'attaquant déchiffre les messages du client, puis les rechiffre avant de les envoyer au serveur et vice-versa. Ainsi il servira de **relai** et pourra accéder aux informations privées de la victime les lres et les modifier comme une connexion HTTP.

Pour conclure nous venons de demontrer que la **sécurité** au sein de chaque réseau n'est pas un detail technique à prendre avec **legerter**, de telle faille de sécurité pourrait effectuer des dégats irréparables sur les machines et sur la santé des victimes. Chaque technologie possède une faille et chaque faille doivent être **réparé et contré** par les hôtes dans le but de rendre ces attaques difficilement réalisable.

Rédigé et réaliser par :
Quentin GAVILAN
n°etudiant : 20193677
L3ASR