TP2 : Attaques actives

Les attaques actives sont mises en place par l'injection, la modification ou la suppression de paquets. L'attaquant peut ainsi laisser les traces des attaques qu'il lance. Ces traces peuvent être exploitées par l'administrateur réseau pour déceler l'existence d'attaques et identifier l'attaquant si possible.

1- Objectifs de ce TP:

- Implémenter quelques attaques et les tester
- Mise en place de quelques attaques en utilisant des outils d'attaques

2- Outils logiciels:

Linux, wireshark ou ethereal, utilitaire ARPflood, utilitaire dhcpstarv, logiciel "cat Karat packet builder"

3- Implémentation d'attaques

Vous trouvez avec le présent fichier, les 4 exercices ci-dessous déjà mis sous forme de programmes C. (**pingfragments.c, pingsur2frag.c ,demandeconntcp.c , my_ping.c)**

Vous pouvez compiler ces programmes sous un interpréteur de commandes shell unix en utilisant le compilateur cc ou gcc.

Exemple :
#cc -c my_ping.c pour compiler
#cc my_ping.c –o myping pour générer l'exécutable
#./myping 127.0.0.1 127.0.0.1 500 pour envoyer un paquet de taille 528 octets=500+20+8

Vous pouvez observer certains résultats en utilisant un sniffer tel que wireshark.

Exercice 1 : my_ping.c
Ce premier programme concerne un paquet IP encapsulant un paquet ICMP echo. Il s'agit d'envoyer un paquet ICMP echo d'une machine A en donnant comme adresse destination celle de B et comme adresse source celle de C. Vous devez observez en utilisant un sniffer (ethereal ou tcpdump) un paquet icmp echo et sa réponse transmise de la machine B vers la machine C. Avec ce premier exercice vous serez donc capable de générer un paquet IP et donc de maîtriser parfaitement les différents champs de IP.

Exercise 2 : pingsur2frag.c

Le deuxième exercice consiste à concevoir deux fragments de paquet IP contenant à eux deux un paquet ICMP echo avec les mêmes types d'adresses que l'exercice précédent. Cet exercice vous permet de bien maîtriser la conception de fragments IP, ce qui est nécessaire pour l'attaque Teardrop. Vous remarquerez dans cette exercice que pour l'offset on fait un décalage de trois bits ….Dans le cas on vous ne comprenez ce calcul. Il vous est demander de faire un ping avec une taille de 2000 octets sur le lien ethernet et avec le sniffer vous analyserez les différents champs des fragments IP ainsi générés.

Exercice 3 : pingfragments.c

Exercice 4 : demandeconntcp.c

4- Test de quelques outils d'attaques

Dans cette partie, nous nous intéressons à la mise en place des attaques suivantes:

1) DHCP starvation : L'attaquant inonde le serveur DHCP avec des messages DHCPREQUEST afin de réserver toutes les adresses IP disponibles sur le serveur DHCP. L'attaquant doit utiliser une nouvelle adresse MAC pour chaque requête
2) MitM basé sur l'ARP spoofing : L'attaquant empoisonne les tables ARP des victimes pour s'insérer entre eux.
3) Usurpation d'identité : l'attaquant utiliser l'adresse IP d'une autre machine comme adresse source.
4) ARP cache poisoning : Cette attaque corrompt le cache de la machine victime MV. Le pirate envoie des paquets ARP réponse à MV indiquant une fausse adresse adresse MAC correspondant à l'adresse IP d'une autre machine destinataire MD.
5) Innondation de la table de commutation par ARP flooding : L'attaquant inonde le réseau avec des trames ARP « who is » en changeant à chaque fois son adresse MAC. Les commutateurs ajoutent dans leurs tables de commutation les adresses MAC observé en correspondance avec leurs ports d'entrée.

a. Attaque DHCP starvation

Pour la réalisation de cette attaque, nous utilisons l'outil **dhcpstarv** qui lance des requêtes d'obtention de bail DHCP, sauvegarde les réponses et exécute les mises à jour des baux d'une façon normale. Pour obtenir plusieurs baux (plusieurs adresses IP), dhcpstarv utilise, pour chaque requête, une nouvelle adresse MAC.

Manipulation :

Installer, sous linux, l'outil dhcpstarv en suivant les étapes suivantes :

#tar –xvf dhcpstarv-0.2.1.tar.gz

#cd dhcpstarv-0.2.1

#./configure

#make

#make install

Sur une autre machine, configurer le serveur DHCP (/etc/dhcpd.conf) puis le lancer, (#dhcpd –lf /etc/dhcpd.leases)

Sur la première machine, lancer l'attaque dhcp starvation : #dhcpstarv –i eth0

Editer le fichier /etc/dhcpd.leases. vous remarquerez que toutes les adresse étaient assignées

b. Attaque "Man In The Middle" basée sur l'attaque "ARP spoofing"

Pour réaliser cette attaque, nous avons besoin de trois nœuds connectés à un switch (cas réel ou sur GNS3) ou à un point d'accès sans fil. Le nœud attaquant sera la machine sur laquelle est installé le système d'exploitation « **Kali Linux** »

Etape 1 : activer le routage dans le nœud attaquant (Kali Linux)

Tester si le routage est activé en utilisant la commande **sysctl** ou en cherchant la valeur de **ip_forward** dans **/proc/sys/ipv4**

```
Root# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

ou bien

```
root# cat /proc/sys/net/ipv4/ip_forward
0
```

Activer le routage :

```
root# sysctl –w net.ipv4.ip_forward=1
```

ou

```
root#echo 1 > /proc/sys/net/ipv4/ip_forward
```

Etape 2 : empoisonner les tables ARP des nœuds victimes

Lancer une communication entre les deux nœuds légtimes (exemple : ping) puis afficher le contenu de leur table arp (arp -a). Exécuter, ensuite, l'attaque arpspoof.

root#arpspoof -i wlan0 -t @IPnoeud1 @Ipnoeud2 ==> pour empoisonner la table arp du noeud1
root#arpspoof -i wlan0 -t @IPnoeud2 @Ipnoeud1 ==> pour empoisonner la table arp du noeud2

Afficher le contenu des tables arp des nœuds légitimes. Que remarquez-vous ?

Conséquences : utiliser Wireshark pour afficher le trafic capturé par l'attaquant (penser à faire un ping entre les nœuds légitimes).

root#wireshark

Utiliser driftnet pour afficher les images se trouvant dans les sites visités par la victime

root# driftnet -i wlan0

Utiliser urlsnarf pour afficher les urls des sites visités par la victime

Root# urlsnarf -i wlan0

Répéter la même attaque en utilisant ettercap (Applications > Internet > Ettercap ou root#ettercap -G) : lancer le sniffer (Sniff > Uniffied sniffing) sur la machine attaquant, afficher les hotes du réseau local poir spécifier les victimes (Hosts > Scan for hosts, puis Hosts > Hosts list pour les voir) et enfin, lancer l'attaquer, allez dans Mitm > ARP Spoofing > Sniff remote connections

c. Attaque "usurpation d'identité"

Pour la réalisation de cette attaque, nous utilisons l'outil « **cat Karat packet builder »** qui est un générateur de paquets.

Nous travaillons sur un réseau constitué de deux postes lié par un câble réseau. Sur le premier poste, nous utilisons la machine Windows avec l'adresse IP 10.0.0.1 et la machine linux (sous vmware) avec

l'adresse 10.0.0.2. Sur le second poste, nous utilisons le logiciel **cat Karat packet builder** installé sous la machine Windows d'adresse IP 10.0.0.3.

Configurer les adresses IP des trois machines et vérifier la connectivité.

Lancer le sniffer wireshark sur la machine linux et commencer la capture sur l'interface ethernet (10.0.0.2)

Manipulation 1 : Utiliser le logiciel **cat Karat packet builder** (sur la machine 10.0.0.3) pour construire des paquets **Echo request** ayant 10.0.0.2 comme adresse IP source et 10.0.0.1 comme adresse IP destination. Spécifier aussi les adresses MAC correspondants aux deux adresses IP spécifiées. Que remarquez-vous ?

Manipulation 2 : Utiliser le logiciel **cat Karat packet builder** (sur la machine 10.0.0.3) pour construire des paquets **Echo reply** ayant 10.0.0.2 comme adresse IP source et 10.0.0.1 comme adresse IP destination. Que remarquez-vous ?

d. Attaque "ARP cache poisoning"

Pour la réalisation de cette attaque, nous utilisons l'outil « **cat Karat packet builder »**

Nous travaillons sur un réseau constitué de deux postes lié par un câble réseau. Sur le premier poste, nous utilisons la machine Windows avec l'adresse IP 10.0.0.1 et la machine linux (sous vmware) avec l'adresse 10.0.0.2. Sur le second poste, nous utilisons le logiciel **cat Karat packet builder** installé sous la machine Windows d'adresse IP 10.0.0.3.

Configurer les adresses IP des trois machines et vérifier la connectivité (ping).

Consulter les tables ARP des trois machines (commande **arp –a**) puis les vider (commande **arp –d ***)

Lancer le sniffer **ethereal** sur la machine linux et commencer la capture sur l'interface ethernet (10.0.0.2)

Utiliser le logiciel **cat Karat packet builder** pour construire des paquets **ARP reply** et les envoyer à la machine linux. Ces paquets auront comme adresse IP source 10.0.0.1 (dans l'entête ARP) et comme adresse MAC source celle de la machine 10.0.0.3.

Vérifier que la machine linux a bien reçu les paquets **ARP reply (**voir la capture ethereal).

Si la machine linux traite les paquets ARP reply sans avoir envoyé auparavant les messages ARP request correspondant, vous allez remarquer qu'une fausse entrée s'ajoute dans la table ARP (commande **arp –a**)

e. Attaque « Inondation de la table de commutation »

Pour la réalisation de cette attaque, nous utilisons l'utilitaire **ArpFlood** qui permet l'envoi massif de demande ou de réponse ARP. Nous pouvons utiliser aussi l'outil « **cat Karat packet builder ».**

Nous utilisons au moins trois machines connecté à un switch. Sur la première machine, nous lançons un ping vers la deuxième. Sur la troisième machine, nous lançons un sniffer passif. Nous allons remarquez que ce dernier n'arrive pas à sniffer les trames (du ping). Nous exécutons l'utilitaire arpflood et nous remarquons que le sniffer commence à récupérer les trames échangé entre les deux premières machines.

Lancer le serveur vsftpd sur la première machine et se connecter sur ce dernier à partir de la deuxième machines. Retrouver sur la troisième machines les trames correspondantes indiquant le login et le mot de passe.