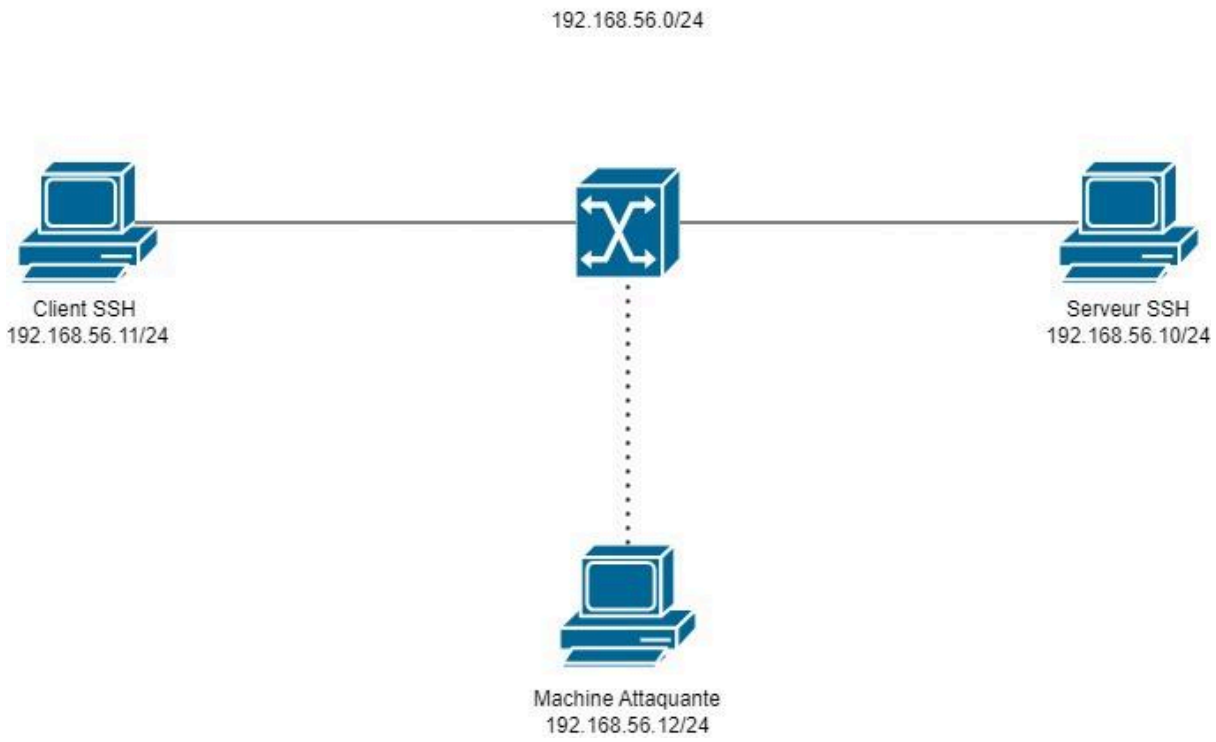
TP MITM SSH

Sommaire :

Schéma réseau



@MAC serveur ssh : 08:00:27:08:BA:20
@MAC client : 08:00:27:DA:9A:A4
@MAC attaquant : 08:00:27:6F:31:8C

Première Utilisation (Q5/Q7)

Lorsque l'on souhaite se connecter au serveur ssh, ce message apparaît car on est en train d'établir cette connexion pour la première fois. c'est un avertissement nous disant si on souhaite bien utiliser ce certificat pour se connecter au serveur

```
etusio@clissh:~$ ssh etusio@srvssh.local.sio.fr
The authenticity of host 'srvssh.local.sio.fr (192.168.56.10)
 can't be established.
ECDSA key fingerprint is SHA256:lPIxEKxsYHPP3i7iMiZZXLYUoW9vi
wSfF39MNoWlM4.
Are you sure you want to continue connecting (yes/no)?
```

Après s'être connecté au serveur, le nouveau certificat est enregistré dans le fichier known_hosts(fichier stockant les clé d'hôtes des serveurs ssh et contient une empreinte de clé publique associée) et sera donc connu

Découverte des hôtes et services présents sur un réseau local (Q8)

Sur la machine Kali, lorsque l'on effectue dans un premier temps la commande : nmap -sP 192.168.56.0/24 (scan ping), on effectue une analyse afin que l'attaquant puisse voir quels sont les hôtes présents dans ce domaine de diffusion.

```
Nmap scan report for srvssh.local.sio.fr (192.168.56.10)
Host is up (0.00062s latency).
Nmap scan report for clissh.local.sio.fr (192.168.56.11)
Host is up (0.0041s latency).
Nmap scan report for kali.local.sio.fr (192.168.56.12)
Host is up (0.0040s latency).
```

Ensuite avec les 3 autres commandes : on peut récupérer quels sont les ports ouverts sur ces hôtes et quels services sont disponibles (Ex avec machine cliente: Port 22/tcp Status → ouvert Service → SSH Version → openssh…

```
Nmap scan report for clissh.local.sio.fr (192.168.56.11)
Host is up (0.00072s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Simulation d’une attaque MITM entre le client et le serveur SSH (Q9/Q11)

L’attaque Man in the Middle est une attaque qui a pour but d'intercepter les communications entre un client et un serveur, sans que ni l'une ni l'autre ne puisse se douter que le flux de communication entre elles a été compromis. L'attaquant se fait passer pour un utilisateur.

Dans un premier, on récupère les informations stockées dans le cache ARP à l’aide de la commande : Ip neigh show.

Client :
IP1 : 192.168.56.12 → @MAC : 31:8C → STALE
IP2 : 192.168.56.10 → @MAC : BA:20→ STALE

Serveur :

192.168.56.11 → @MAC : 9A:A4 → STALE
192.168.56.12 → @MAC : 31:8C → STALE

Les adresses correspondent entre les deux machines.
Après avoir installé et lancé le service ssh-mitm l’attaquant sera donc placé entre le serveur et le client. Pour cela, il est important d’activer la fonction de routage afin qu’il puisse récupérer le trafic entre le client et le serveur et rediriger les paquets à destination de sa machine.

Mise en place de l’ARP Spoofing (Q13/Q20)

Cette technique est utile pour l'attaquant car ce type d’attaque permet de contrôler les flux entre le client et le serveur SSH et donc, décider s' il souhaite écouter, modifier ou encore bloquer les trames.

Pour procéder à l’attaque, on se servira de la commande : “ettercap -i eth0 -T -M arp /192.168.56.10// /192.168.56.11//” ou on indique les IP des victimes (client et serveurs SSH)

Cache ARP du client et serveur après attaque :

Client :

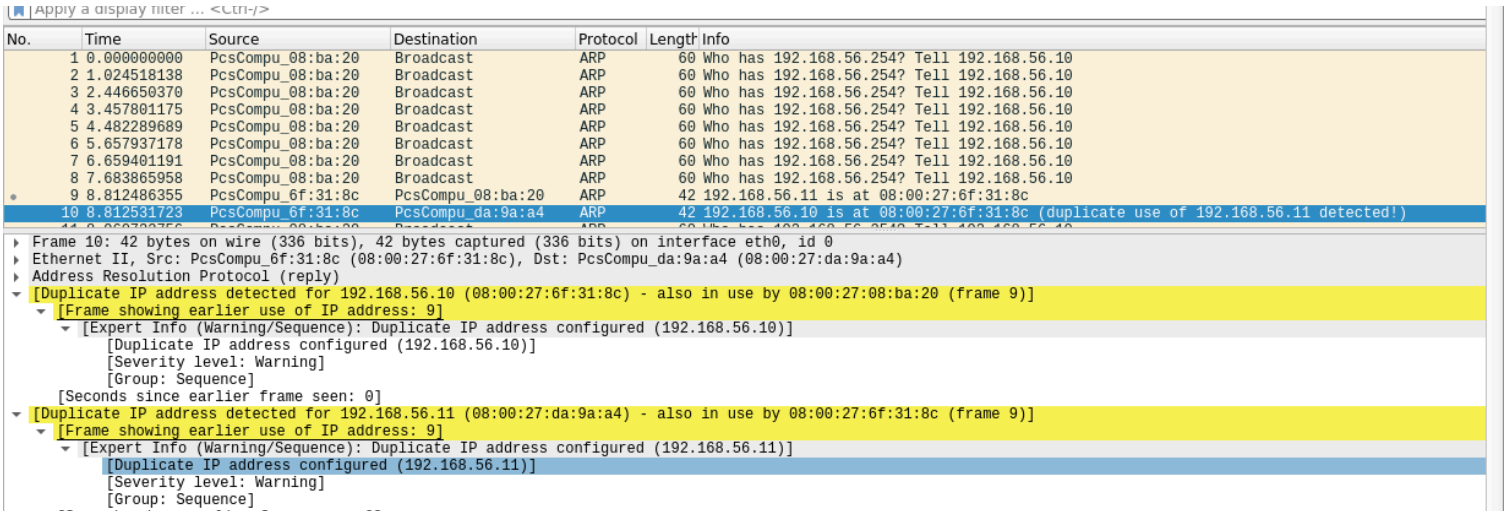
192.168.56.12 → @MAC : 31:8C → STALE
192.168.56.10 → @MAC : 31:8C → REACHABLE

Serveur :

192.168.56.11 → @MAC : 31:8C → REACHABLE
192.168.56.12 → @MAC : 31:8C →STALE

L'attaquant se fait passer pour une machine appartenant au réseau 192.168.56.0, afin d'empoisonner le cache ARP des victimes. Ce qui fait que les trames envoyées entre le client et le serveur seront redirigées sur le serveur pirate

En analysant les trames sur wireshark, on peut voir que le client envoie des trames vers le serveur sans remarquer que la machine attaquante intervient pour intercepter les communications. Sachant que le protocole ARP stocke la première liaison IP et MAC qui lui vient dans son cache, l'attaquant peut ainsi se faire passer pour une machine légale



Mise en œuvre et exploitation de l'attaque Man-in-the-Middle

Après avoir vidé le cache ARP du client et essayé de se connecter au serveur ssh, on reçoit un message d'erreur indiquant que l'empreinte du serveur a été modifiée par rapport à la première connexion suite à l'attaque.
Il est donc nécessaire de supprimer les clés d'hôtes des serveurs ssh stockées dans le fichiers known_hosts avec la commande :
SSH- keygen -f « /home/etusio/.SSH/known_hosts» -R « srvSSH.local.sio.fr »

Renforcement de la sécurité du service OpenSSH

Modifications effectués dans le fichier de configuration sshd_config du serveur

```
PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
```

Sur le client, on génère une clé privé et publié qui serviront comme moyen authentification avec la commande : ssh-keygen -b 256 -t ecdsa (la phrase de chiffrement est “TPMITMcerta”

```
etusio@clissh:~$ sudo ssh-keygen -b 256 -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ecdsa.
Your public key has been saved in /root/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:arSafiMfFVi+CxgkA2OP2WLWykK3ZCEL3xq54VnHDdQ root@clissh
The key's randomart image is:
```

Cette phase de chiffrement est très importante car cela empêchera l’attaquant de se saisir de la clé publique du client

Contenu du répertoire $HOME

```
etusio@clissh:/etc/ssh$ cd $HOME/.ssh
etusio@clissh:~/.ssh$ ls
id_ecdsa  id_ecdsa.pub  known_hosts  known_hosts.old
```

On effectue ensuite une copie de la clé publique vers le serveur ssh via la commande :
etusio@clissh:~$ ssh-copy-id -i ~/.ssh/id_ecdsa.pub etusio@srvssh.local.sio.fr

Tentative de connexion au serveur après avoir modifié les droits du fichier id_ecdsa contenant la clé privé

```
etusio@clissh:~/.ssh$ chmod 644 id_ecdsa
etusio@clissh:~/.ssh$ ssh etusio@srvssh.local.sio.fr
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/home/etusio/.ssh/id_ecdsa' are too open.
It is required that your private key files are NOT accessible by oth
ers.
This private key will be ignored.
Load key "/home/etusio/.ssh/id_ecdsa": bad permissions
etusio@srvssh.local.sio.fr: Permission denied (publickey).
```

La connexion échoue car les permissions de la clé privée (id_ecdsa) sont peu sécurisées. SSH exige que les clés privées soient accessibles uniquement par leur propriétaire.