1. **Pourquoi l'accès aux machines virtuelles via root n'est-il pas possible directement ?** L'accès direct au super-administrateur root est désactivé pour des raisons de sécurité. Cela empêche des actions malveillantes ou accidentelles avec un accès complet et limite les attaques automatisées ciblant ce compte.
2. **À quoi sert sudo, et quels sont ses avantages sur `su -` ?**
   - `sudo` permet d'exécuter des commandes spécifiques avec des privilèges administratifs sans passer en mode root permanent. Cela limite les risques d'erreurs graves et offre un journal des actions effectuées.
   - `su -` change complètement l'utilisateur pour root, exposant potentiellement tout le système.
3. **Quelles commandes permettent de vérifier si OpenSSH est installé et démarré ?**

Vérification de l'installation :
bash
Copier le code
```
dpkg -l | grep openssh
```

   -

Vérification du statut du service :
bash
Copier le code
```
sudo systemctl status ssh
```

   -

4. **Où sont stockées les clés SSH ?**
   - Clés publiques et privées d'un utilisateur : `~/.ssh/`
   - Permissions typiques : Clé privée : `600`, clé publique : `644`.
   - Fichier de configuration du serveur SSH : `/etc/ssh/sshd_config`

V. Première utilisation : Analyse d'une première connexion SSH

5. **Signification de l'alerte lors de la connexion SSH (clé non reconnue)** L'alerte signifie que le client SSH ne possède pas encore l'empreinte de la clé publique du serveur dans son fichier `known_hosts`. Il demande confirmation avant de continuer pour éviter de se connecter à un serveur compromis.
6. **Le message réapparaîtra-t-il lors d'une future connexion ?** Non, car l'empreinte sera ajoutée au fichier `~/.ssh/known_hosts`. Si la clé change (indiquant potentiellement un MITM), une alerte sera affichée.
7. **Rôle de `~/.ssh/known_hosts` :** Ce fichier stocke les empreintes des clés des serveurs auxquels le client s'est déjà connecté. Il permet de vérifier l'identité du serveur pour prévenir les attaques MITM.

VI. Découverte des hôtes et services présents sur un réseau local

8. **Informations accessibles via Nmap pour un attaquant :**
    - Adresse IP des machines actives.
    - Ports ouverts et services associés.
    - Versions des logiciels exécutés, révélant potentiellement des vulnérabilités exploitables.

VII. Simulation d'une attaque MITM

9. Principes généraux d'une attaque MITM :

Une attaque MITM consiste à intercepter et éventuellement modifier les communications entre deux parties sans qu'elles s'en aperçoivent. L'attaquant agit comme un relais entre le client et le serveur, pouvant espionner ou manipuler les données échangées.

10. Cache ARP des machines:

Le cache ARP relie les adresses IP aux adresses MAC. Avant une attaque, les associations IP-MAC sont correctes. Après une attaque MITM, les adresses IP des cibles pointent vers l'adresse MAC de l'attaquant, prouvant la redirection des flux.

11. Routage sur la machine attaquante:

Le routage permet à la machine attaquante de transférer les paquets entre le client et le serveur, assurant que la communication continue malgré l'interception.

12. Redirection de ports :

La commande suivante redirige les flux SSH interceptés vers un port spécifique utilisé par le logiciel d'attaque :

```
bash
iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-ports 2222
```

Cela permet à l'attaquant d'intercepter les données tout en les relayant au serveur.

13. Utilité de l'attaque ARP Spoofing :

Cette attaque redirige les communications du client et du serveur vers l'attaquant en modifiant les associations IP-MAC. Elle permet de positionner l'attaquant comme intermédiaire.

14. Comparaison des caches ARP après l'attaque :

Avant l'attaque : les adresses MAC des hôtes correspondent à leurs IP.

Après l'attaque : les IP des hôtes pointent vers l'adresse MAC de l'attaquant, confirmant l'interception.

15.Fonctionnement de l'attaque ARP Spoofing :

- L'attaquant envoie de fausses réponses ARP au client et au serveur.
- Les hôtes mettent à jour leur cache ARP avec de fausses informations.
- Les paquets sont redirigés vers l'attaquant, qui peut les relayer ou les modifier.

16. Vérification du ping passant par l'attaquant :

Une capture de trames sur la machine attaquante montrera que les requêtes ICMP (ping) passent par elle avant d'atteindre leur destination.

VIII. Renforcement de la sécurité d'OpenSSH

1. Authentification par clés de chiffrement

21. Pourquoi ECDSA est préféré au RSA ?

- ECDSA offre une sécurité équivalente avec des clés plus courtes, réduisant l'espace mémoire et le temps de calcul.
- RSA est moins efficace et moins sûr contre certaines attaques modernes.

22. Importance de la phrase de chiffrement sur la clé privée :

- Elle ajoute une couche de sécurité en cas de vol de la clé privée.
- Sans la phrase, un attaquant ayant accès à la clé peut s'authentifier.

23. Affichage du répertoire et contenu de `id_ecdsa.pub` :

- La clé publique se trouve dans `~/.ssh/id_ecdsa.pub`.
- Contenu : une chaîne de caractères représentant la clé publique, partageable avec le serveur.

24. Effet des permissions incorrectes (644) sur la clé privée :

- Le client refusera d'utiliser une clé privée avec des permissions incorrectes, empêchant la connexion.
- Raison : protection contre les accès non autorisés.

26.Modification des droits de `authorized_keys` (666) sur le serveur :

- Les droits permissifs (lecture/écriture pour tous) ne respectent pas les bonnes pratiques, entraînant un refus de connexion.
- Le serveur SSH vérifie les permissions pour éviter une compromission.

27. Différence entre mot de passe et clés :

- Mot de passe : transmission d'une chaîne de caractères vérifiée par le serveur.
- Clé publique/privée : authentification basée sur un échange cryptographique sans transmettre de secret.

2. Utilisation de ssh-agent

28. Impact d'une attaque MITM après mise en place de clés :

- L'attaque échoue, car les clés cryptographiques assurent l'authenticité de la connexion.
- L'attaquant ne peut pas déchiffrer ou imiter les clés privées.

3. Vérification de l'identité du serveur avec SSHFP

29. Intérêt de SSHFP :

- Automatisation de la vérification des empreintes des clés des serveurs via DNS.
- Réduire les risques d'attaques MITM lors de la première connexion.

Durcissement de la configuration OpenSSH

30. Mise en œuvre des préconisations ANSSI :

- Vérification des permissions :

```
bash
sudo chmod 600 /etc/ssh/ssh_host_*_key
sudo chown root:root /etc/ssh/ssh_host_*_key
```

- Utilisation du protocole SSH version 2 :

Dans `/etc/ssh/sshd_config` :

```
bash
Protocol 2
```

- Changement du port SSH :

```
bash
Port 222
```

- Désactiver l'accès root :

```
bash
PermitRootLogin no
```

- Limiter les tentatives de connexion :

```
bash
MaxAuthTries 3
```

31. Définition de l'état de l'art :

L'état de l'art désigne les pratiques, outils et algorithmes les plus fiables et récents, validés par la communauté pour assurer la sécurité.

32. Principe de Kerckhoffs :

La sécurité d'un système cryptographique doit reposer uniquement sur la clé et non sur la confidentialité de l'algorithme.

33.Pertinence d'algorithmes connus :

- Les algorithmes publics sont soumis à des audits constants.
- Leur sécurité est prouvée par l'absence d'attaques efficaces, contrairement aux systèmes obscurs.