Projet Firewall : Test d'entrée à LaPlateforme en deuxième année

Teva Clairefond

Juillet 2025

Table des matières

Partie 1 Questions

1.1 Système

1.1.1 Question 1 : Visualiser les processus SSH

Commande pour afficher les processus SSH actifs :

```
ps -ef | grep ssh
```

Pour une vue interactive :

```
sudo apt install htop
htop
```

1.1.2 Question 2 : Obtenir l'adresse IP de la machine

```
ip addr
```

1.1.3 Question 3 : Hyperviseur de type 1 vs type 2

— **Type 1 (bare metal)** : Exécute directement sur le matériel. Utilisé en production pour virtualiser des serveurs dans les datacenters.
— **Type 2 (hosted)** : Fonctionne au-dessus d'un OS. Utilisé pour le test et le développement (ex : VirtualBox, VMware Workstation).

1.2 Réseau

1.2.1 Question 1 : Protocoles, classification OSI et sécurité

Classification des protocoles

Couche OSI	Protocoles
Application (7)	HTTP, HTTPS, FTP, SFTP, DNS,
Transport (4)	TCP, UDP
Réseau (3)	IP (IPv4, IPv6)
Liaison de données (2)	Ethernet

Protocoles sensibles au niveau sécurité
— **HTTPS / SSH** : Chiffrés, sécurisés.
— **DNS** : Vulnérable au spoofing (empoisonnement).
— **FTP** : Transfert non chiffré → écoute possible.
— **TCP** : Vulnérable aux attaques SYN flood, spoofing.

1.2.2 Question 2 : Couches OSI associées

Élément	Couche OSI
Switch	Couche 2 – Liaison de données
Routeur	Couche 3 – Réseau
TCP	Couche 4 – Transport
HTTP	Couche 7 – Application

1.2.3 Question 3 : Enregistrements DNS

— **A record** : Lien nom de domaine → IPv4.
— **AAAA record** : Lien nom de domaine → IPv6.

1.2.4 Question 4 : SYN et ACK

— **SYN (Synchronize)** : Demande de connexion.
— **ACK (Acknowledge)** : Accusé de réception.
— **Handshake TCP** : SYN → SYN-ACK → ACK.

1.3 Sécurité

1.3.1 Question 1 : ISO/IEC 27001

Norme internationale pour la mise en place d'un **système de management de la sécurité de l'information (SMSI)**. Elle définit les bonnes pratiques pour protéger la confidentialité, l'intégrité et la disponibilité des données.

1.3.2 Question 2 : Institutions de cybersécurité

— **France** :
  — ANSSI : Agence nationale de la sécurité des systèmes d'information
  — CNIL : Autorité de protection des données personnelles
— **International** :
  — ENISA : Agence européenne pour la cybersécurité
  — NIST : National Institute of Standards and Technology (USA)

1.3.3 Question 3 : Moments de vulnérabilité et conséquences

Moments de vulnérabilité
— Connexion à un Wi-Fi public non sécurisé
— Téléchargement d'applications non vérifiées
— Navigation sur des sites non HTTPS
— Utilisation de mots de passe faibles ou réutilisés
— Absence de mises à jour des systèmes/appareils

Conséquences possibles
— Vol de données ou d'identifiants
— Piratage de comptes personnels ou professionnels
— Usurpation d'identité, chantage, rançongiciel

Personnes impactées
— **Famille** : Partage de réseau ou d'appareils
— **Collègues / Clients** : Risques professionnels ou fuite de données
— **Entreprise** : Perte d'image, d'argent ou interruption d'activité

Partie 2 Projet Firewall

2.1 Configuration du parefeu via iptables

2.1.1 Mise en place du serveur web et SSH

Après ouverture du terminal, passage en mode super-utilisateur :

```
sudo -s
apt install apache2
ip addr
apt install openssh-server
apt install iptables
apt install tables  # erreur car paquet inexistant
```

2.1.2 Réinitialisation des règles iptables

Suppression des règles et chaînes pour les tables filter, nat, et mangle. Réinitialisation des politiques à ACCEPT. Affichage des règles du parefeu pour vérifier la bonne application :

```
iptables -nvL
```

2.1.3 Configuration des règles de sécurité

À partir d'une politique de blocage par défaut, j'ai autorisé certains ports essentiels (SSH, HTTP, HTTPS) ainsi que le trafic à destination de localhost et les connexions établies :

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

2.2 Supervision avec Logwatch et Rsyslog

2.2.1 Installation et configuration

Installation de Logwatch et de Rsyslog pour assurer la lecture des journaux, Logwatch ne supportant pas directement journalctl :

```
apt install logwatch -y
apt install rsyslog -y
```

Ajout dans /etc/rsyslog.d/20-default.conf :

```
auth,authpriv.* /var/log/auth.log
```

Création du fichier sshd.conf pour Logwatch :

```
nano /etc/logwatch/conf/logfiles/sshd.conf
```

Ajout dans /etc/logwatch/conf/logfiles/sshd.conf :

```
LogFile = auth.log
*RemoveHeaders
```

2.2.2 Test

```
logwatch --range today --service sshd --service http --detail high
    ↪ --format text
```

Logwatch repère bien les connections ssh et http.

2.2.3 Automatisation de la génération de rapports

Configuration du système de planification de tâches (cron) de manière à enregistrer les activités dans un fichier. (/etc/cron.daily/00logwatch) Définition d'une variable contenant le chemin du répertoire où les rapports seront sauvegardés :

```
OUTPUT_DIR="/var/log/logwatch"
```

Création du dossier s'il n'existait pas déjà, l'option `-p` évitant les erreurs en cas d'existence préalable :

```
mkdir -p "$OUTPUT_DIR"
```

Définition d'une variable contenant le nom du fichier de sortie, comprenant la date du jour, et situé dans le répertoire défini :

```
OUTPUT_FILE="$OUTPUT_DIR/logwatch-$(date␣+%F).log"
```

Appel de `logwatch` avec une analyse de la journée précédente (`-range yesterday`) pour les services `sshd` et `http`, avec une sortie dans le fichier défini :

```
/usr/sbin/logwatch --range yesterday --service sshd --service http
    ↪ --output file --format text --filename "$OUTPUT_FILE"
```

Transformation du fichier en fichier exécutable par `cron` grâce à la commande suivante, où `+x` donne les droits d'exécution :

```
sudo chmod +x /etc/cron.daily/00logwatch
```

2.3 Protection contre les attaques par force brute avec Fail2ban

2.3.1 Installation de fail2ban et vsftpd

```
apt install fail2ban -y
apt install vsftpd -y
```

2.3.2 Fichier jail.local

Création des jails pour SSH et FTP avec des seuils d'alerte et bannissement personnalisés :

```
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = journal
maxretry = 5
findtime = 300
bantime = 3600

[vsftpd-tbf-ip]
enabled = true
port = ftp
filter = vsftpd
logpath = journal
maxretry = 10
findtime = 300
bantime = 3600

[vsftpd-tbf-multiuser]
enabled = true
port = ftp
filter = vsftpd-multiuser
logpath = journal
maxretry = 20
findtime = 300
bantime = 3600
```

2.3.3 Filtres personnalisés

failregex analyse les logs dans journactl qui correspondent à la syntaxe indiquée. Cela permet de définir un filtre qui va trier les adresses IP qui vont être placées dans la jail vsftpd-tbf-ip :

Fichier vsftpd.conf (/etc/fail2ban/filter.d/vsftpd.conf) :

```
[Definition]
failregex = ^.*pam_unix\(vsftpd:auth\): authentication failure;.*
   ↪ rhost=(::ffff:)?<HOST>
ignoreregex =
```

Fichier vsftpd-multiuser.conf :

```
[Definition]
failregex = ^.*pam_unix\(vsftpd:auth\): authentication failure;.*
   ↪ rhost=(::ffff:)?<HOST>
ignoreregex =
```

2.3.4 Redémarrage et vérifications

```
sudo systemctl restart fail2ban
sudo systemctl enable fail2ban
```

Ajout de la règle pour autoriser le port FTP :

```
iptables -A INPUT -p tcp --dport 21 -m state --state NEW -j ACCEPT
```

Vérifications avec Putty (SSH) et FTP depuis Powershell :

```
fail2ban-client status sshd
fail2ban-client status vsftpd-tbf-ip
```