

# Questionnaire Candidat

Bachelors 2 pour Système Réseaux et cybersécurité

## Projet Firewall

Sur un environnement Linux Debian virtualisé, vous allez configurer un firewall et le monitorer.

### Partie 1

Sur votre machine virtuelle, installez un serveur web (apache2) et le SSH.

Dans mon cas, SSH était déjà installé, mais s'il ne l'était pas, j'aurais dû l'installer à l'aide de la commande **`sudo apt install openssh-server`**.

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 111x36
pierre@debian:~$ ssh -V
OpenSSH_8.4p1 Debian-5+deb11u1, OpenSSL 1.1.1n 15 Mar 2022
pierre@debian:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-30 11:06:10 CEST; 1h 29min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 513 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 526 (sshd)
       Tasks: 1 (limit: 2337)
      Memory: 5.2M
         CPU: 222ms
    CGroup: /system.slice/ssh.service
            └─526 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Warning: some journal files were not opened due to insufficient permissions.
pierre@debian:~$
```

La commande **`sudo apt install apache2`** permet d'installer apache2 :

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 102x26
pierre@debian:~$ sudo apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
Les NOUVEAUX paquets suivants seront installés :
  apache2
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 278 ko dans les archives.
Après cette opération, 641 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://security.debian.org/debian-security bullseye-security/main amd64 apache2 amd64
2.4.56-1-deb11u1 [278 kB]
278 ko réceptionnés en 0s (986 ko/s)
Sélection du paquet apache2 précédemment désélectionné.
(Lecture de la base de données... 34323 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../apache2_2.4.56-1-deb11u1_amd64.deb ...
Dépaquetage de apache2 (2.4.56-1-deb11u1) ...
Paramétrage de apache2 (2.4.56-1-deb11u1) ...
apache-htcacheclean.service is a disabled or a static unit not running, not starting it.
pierre@debian:~$
```

## Vérification de l'activation du service apache2 :

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 102x26
pierre@debian:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-05-30 12:47:44 CEST; 2min 39s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 834 (apache2)
      Tasks: 55 (limit: 2337)
     Memory: 8.7M
        CPU: 38ms
    CGroup: /system.slice/apache2.service
            └─834 /usr/sbin/apache2 -k start
              └─836 /usr/sbin/apache2 -k start
                └─837 /usr/sbin/apache2 -k start
pierre@debian:~$
```

Après avoir vérifié que tout fonctionne bien, supprimer toutes les règles de votre firewall:

```
tables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
```

et vérifier que la commande `iptables -nvL` ne renvoi rien.

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 102x26
pierre@debian:~$ sudo iptables -V
iptables v1.8.7 (nf_tables)
pierre@debian:~$ sudo iptables -F
pierre@debian:~$ sudo iptables -X
pierre@debian:~$ sudo iptables -t nat -F
pierre@debian:~$ sudo iptables -t nat -X
pierre@debian:~$ sudo iptables -t mangle -F
pierre@debian:~$ sudo iptables -t mangle -X
pierre@debian:~$ sudo iptables -P INPUT ACCEPT
pierre@debian:~$ sudo iptables -P FORWARD ACCEPT
pierre@debian:~$ sudo iptables -P OUTPUT ACCEPT
pierre@debian:~$ sudo iptables -nvL
Chain INPUT (policy ACCEPT 50 packets, 3576 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain OUTPUT (policy ACCEPT 9 packets, 892 bytes)
 pkts bytes target    prot opt in     out     source         destination
pierre@debian:~$
```

A présent, recréez chacune des règles nécessaires pour que le serveur fonctionne (connexion ssh et serveur web).

Pour SSH, on autorise le port 22 en entrée et en sortie sur le serveur :

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 105x32
[pierre@debian:~$ sudo iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT ]
[pierre@debian:~$ sudo iptables -t filter -A OUTPUT -p tcp --dport 22 -j ACCEPT ]
[pierre@debian:~$ sudo iptables -nvl ]
Chain INPUT (policy ACCEPT 106 packets, 7552 bytes)
  pkts bytes target    prot opt in     out     source            destination
    68 4932 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:22

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 197 packets, 29950 bytes)
  pkts bytes target    prot opt in     out     source            destination
    0    0 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:22
pierre@debian:~$
```

Pour apache, on autorise le port 80 en entrée et en sortie sur le serveur :

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 105x32
[pierre@debian:~$ sudo iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT ]
[pierre@debian:~$ sudo iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT ]
[pierre@debian:~$ sudo iptables -nvl ]
Chain INPUT (policy ACCEPT 109 packets, 7937 bytes)
  pkts bytes target    prot opt in     out     source            destination
   298 21588 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:22
    0    0 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:80

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target    prot opt in     out     source            destination

Chain OUTPUT (policy ACCEPT 316 packets, 42714 bytes)
  pkts bytes target    prot opt in     out     source            destination
    0    0 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:22
    0    0 ACCEPT     tcp  --  *      *       0.0.0.0/0         0.0.0.0/0         tcp dpt:80
pierre@debian:~$
```

## Partie 2

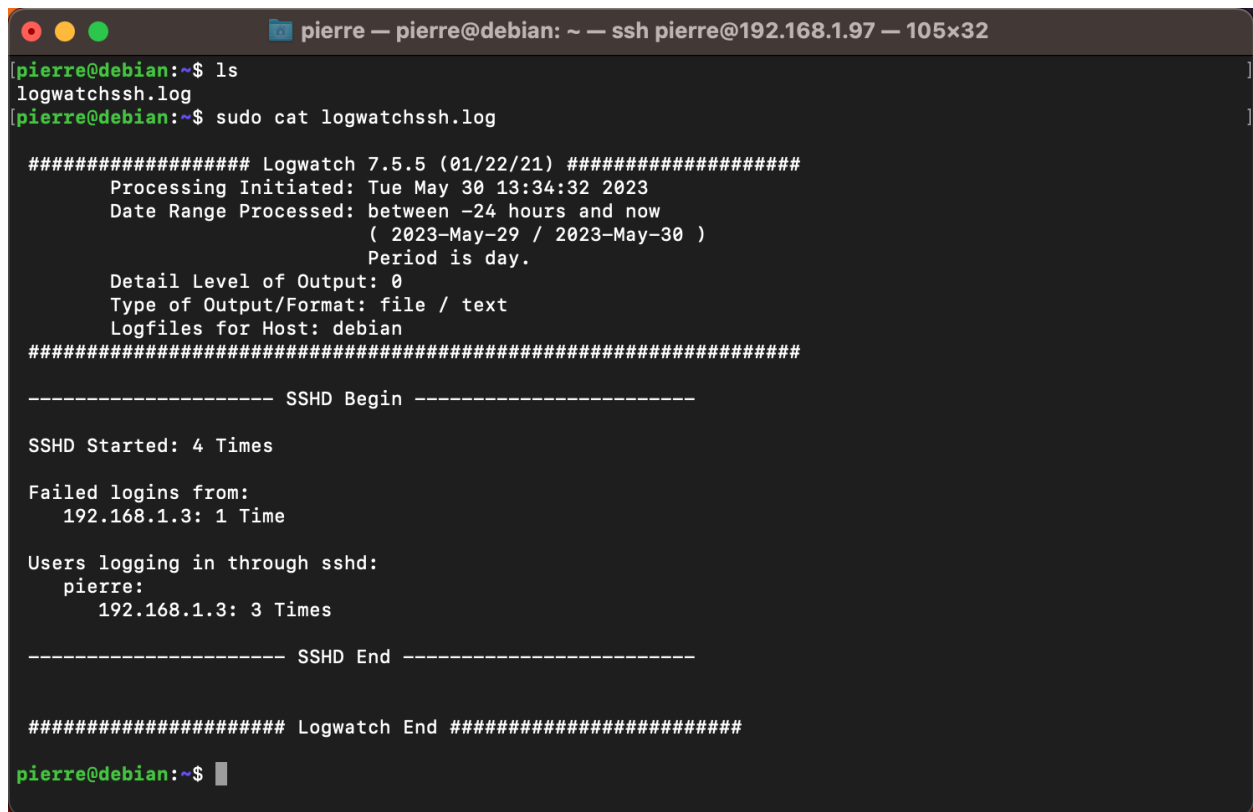
Afin d'enregistrer et de surveiller l'activité du serveur, installez le paquet **logwatch**. Configurez-le afin que soient enregistrées toutes les tentatives de connexion (par 24h) dans un fichier.

D'abord, dans le fichier de configuration de logwatch (`/usr/share/logwatch/default.conf/logwatch.conf`), on modifie deux paramètres :

- on indique la sortie de la commande doit se faire dans un fichier : **Output = file**
- le fichier de destination des résultats : **Filename = /home/pierre/logwatchssh.log**

Ensuite, on liste toutes les tentatives de connexion SSH par 24h avec la commande :  
**`sudo logwatch --service sshd --range 'between -24 hours and now'`**

Enfin, on retrouve dans le fichier configuré toutes les tentatives de connexion SSH :



```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 105x32
[pierre@debian:~]$ ls
logwatchssh.log
[pierre@debian:~]$ sudo cat logwatchssh.log

##### Logwatch 7.5.5 (01/22/21) #####
Processing Initiated: Tue May 30 13:34:32 2023
Date Range Processed: between -24 hours and now
                      ( 2023-May-29 / 2023-May-30 )
                      Period is day.
Detail Level of Output: 0
Type of Output/Format: file / text
Logfiles for Host: debian
#####

----- SSHD Begin -----

SSHD Started: 4 Times

Failed logins from:
  192.168.1.3: 1 Time

Users logging in through sshd:
  pierre:
    192.168.1.3: 3 Times

----- SSHD End -----

##### Logwatch End #####
pierre@debian:~$
```

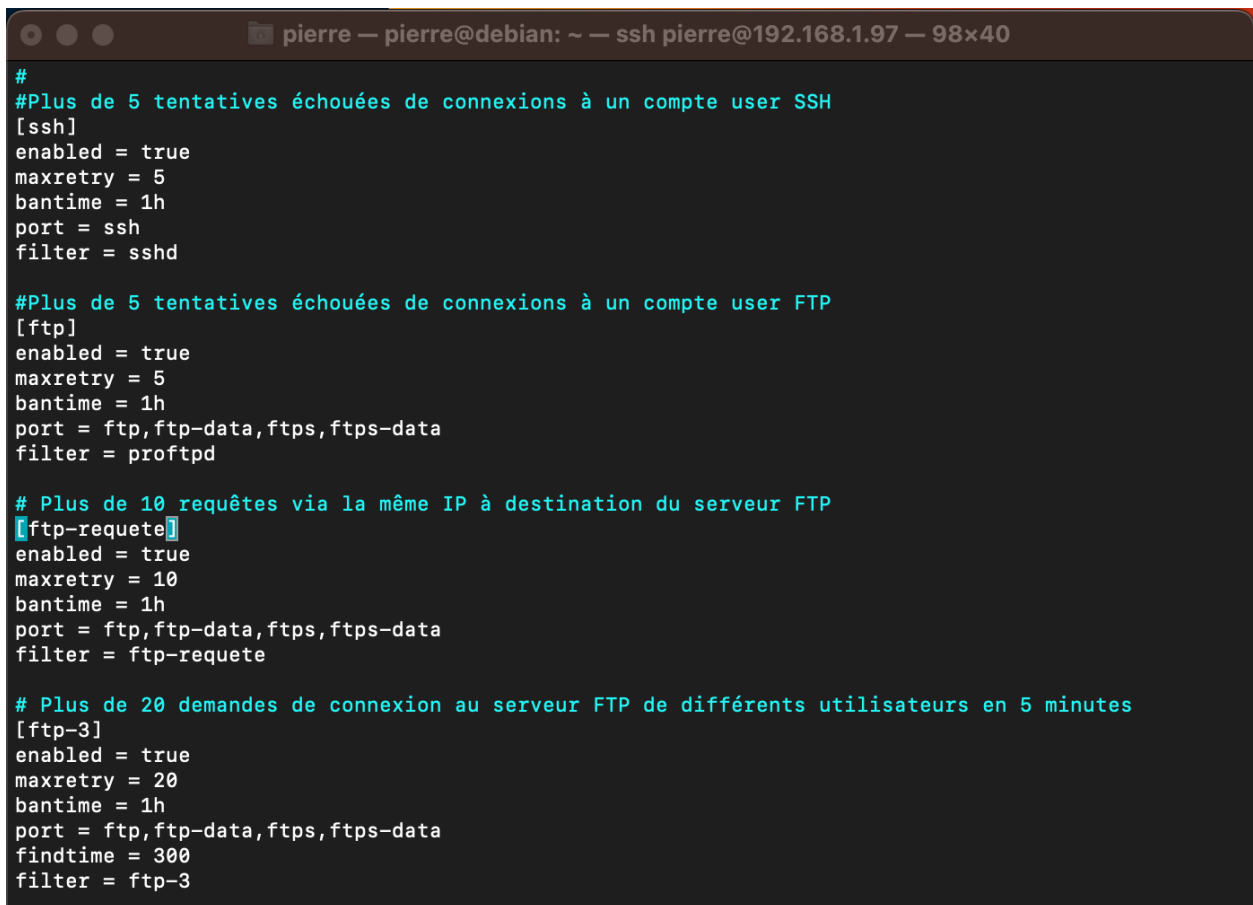
## Partie 3

A l'aide de Fail2ban, traiter les logs enregistrés afin que:

- Plus de 5 tentatives échouées de connexions à un compte user (ftp ou ssh)
- Plus de 10 requêtes via la même IP à destination du serveur FTP
- Plus de 20 demandes de connexion au serveur FTP même d'utilisateurs différents en 5 minutes

soit interprété comme un signe de TBF (Tentative de Brute Force).

On ajoute ces règles dans le fichier de configuration de Fail2ban (/etc/fail2ban/jail.conf).



```
#
#Plus de 5 tentatives échouées de connexions à un compte user SSH
[ssh]
enabled = true
maxretry = 5
bantime = 1h
port = ssh
filter = sshd

#Plus de 5 tentatives échouées de connexions à un compte user FTP
[ftp]
enabled = true
maxretry = 5
bantime = 1h
port = ftp,ftp-data,ftps,ftps-data
filter = proftpd

# Plus de 10 requêtes via la même IP à destination du serveur FTP
[ftp-requete]
enabled = true
maxretry = 10
bantime = 1h
port = ftp,ftp-data,ftps,ftps-data
filter = ftp-requete

# Plus de 20 demandes de connexion au serveur FTP de différents utilisateurs en 5 minutes
[ftp-3]
enabled = true
maxretry = 20
bantime = 1h
port = ftp,ftp-data,ftps,ftps-data
findtime = 300
filter = ftp-3
```

On utilise des filtres déjà existants pour les deux premières règles : sshd et proftpd (choix par défaut parmi les filtres ftp présents dans le répertoire filter.d)

Précisons que 'maxretry' correspond au nombre de tentatives et que 'findtime' correspond à l'intervalle de temps de 5 minutes.

Dans le répertoire filter.d on vient créer deux filtres personnalisés pour les deux dernières règles : ftp-requete et ftp-3.

```
pierre — pierre@debian: ~ — ssh pierre@192.168.1.97 — 98x40
[pierre@debian:~]$ sudo cat /etc/fail2ban/filter.d/ftp-requete.conf
[INCLUDES]

before = common.conf

[Definition]

_daemon = proftpd
failregex = .* \[.+\] \(\d+\) FTP session opened\.
            .* \[.+\] \(\d+\) FTP session closed\.
ignoreregex =
[pierre@debian:~]$ sudo cat /etc/fail2ban/filter.d/ftp-3.conf
[INCLUDES]

before = common.conf

[Definition]

_daemon = proftpd

failregex = \(\S+\[<HOST>\]\)[: -]+ USER \S+: no such user found from \S+ \[[0-9.]+\] to \S+:\S+\s
*$
            \(\S+\[<HOST>\]\)[: -]+ USER \S+ \(\Login failed\):.*\s+$
            \(\S+\[<HOST>\]\)[: -]+ Maximum login attempts \([0-9]+\) exceeded, connection refused
            .* \s+$
            \(\S+\[<HOST>\]\)[: -]+ SECURITY VIOLATION: \S+ login attempted\.\s+$
            \(\S+\[<HOST>\]\)[: -]+ Maximum login attempts \(\d+\) exceeded\s+$

ignoreregex =
pierre@debian:~$
```

Source pour les regex : [https://hodari.be/posts/2019\\_10\\_17\\_setup\\_proftpd\\_protected\\_by\\_fail2ban/](https://hodari.be/posts/2019_10_17_setup_proftpd_protected_by_fail2ban/)

## Rendu

Sur un repository github, vous présentez votre démarche dans la réalisation de ce projet dans un document au format libre.

Vous y ajouterez les fichiers de configuration du serveur et les scripts/configurations utilisés.