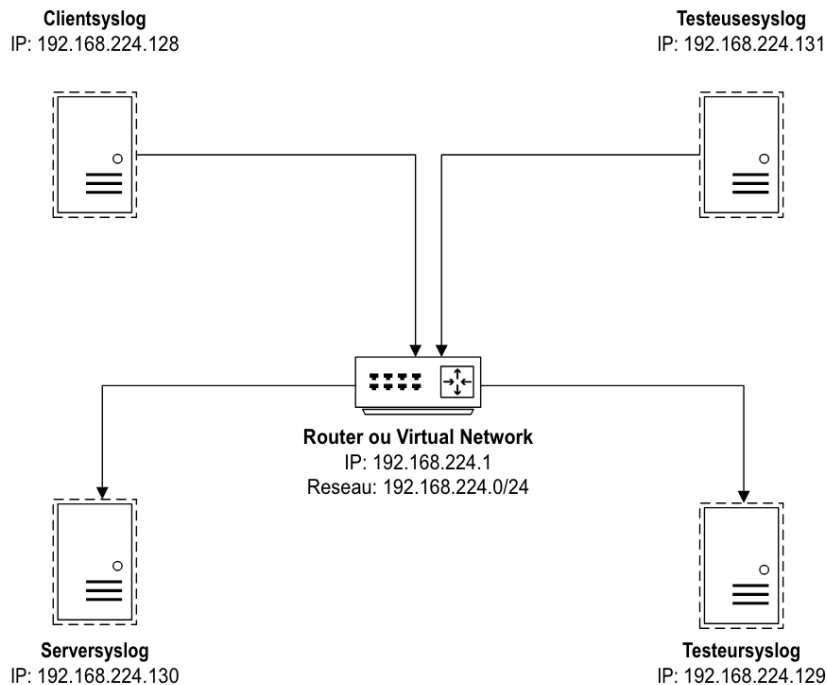


## Partie 1 : Mise en place de l'infrastructure

### 1. Construisez un schéma réseau d'une telle infrastructure



### 2. Réalisez ce schéma réseau avec les contraintes décrites ci-dessus.

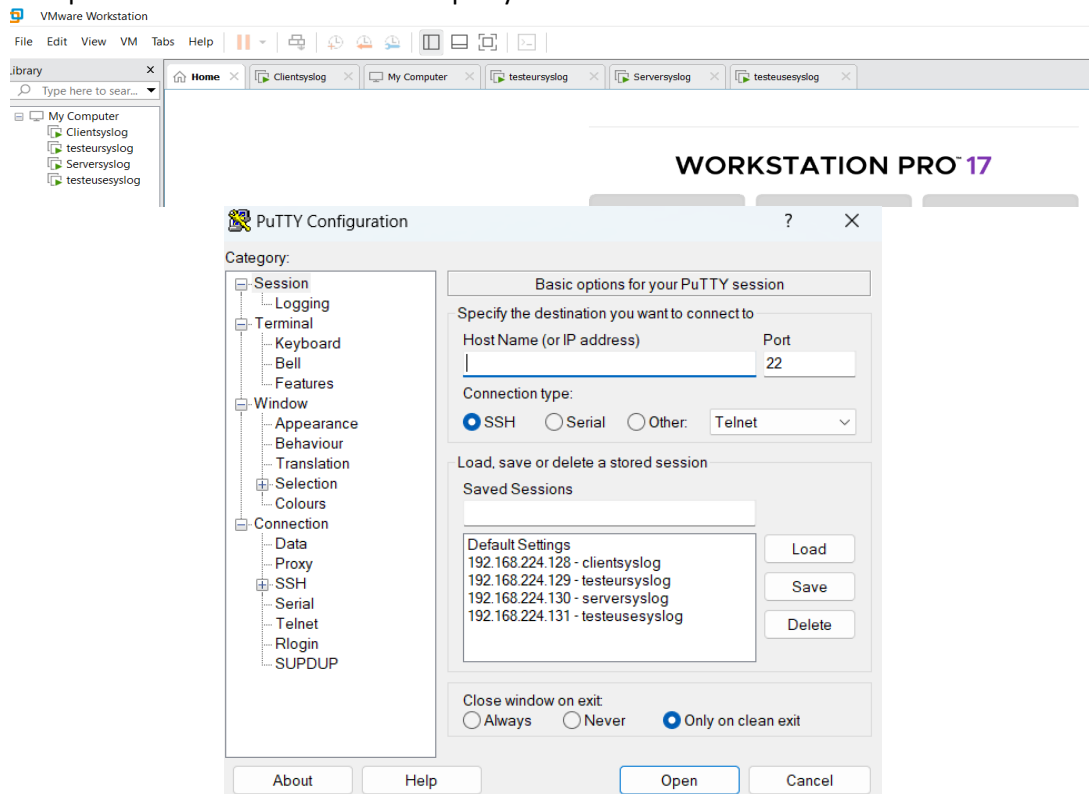
Pour réaliser ce TP, nous avons utilisé VMWare Workstation 17 comme hyperviseur de type 2. Les quatre machines virtuelles sont installées sous Ubuntu server 24. Nous avons choisi la dernière version pour être à jour et minimiser les risques de vulnérabilités. Le choix de la version serveur permet de mieux gérer les ressources matérielles (pas d'interface graphique) et d'utiliser uniquement terminal dans ce TP. Cela nous permettra de mieux nous habituer à cette interface.

Les cartes réseaux des machines sont configurées en mode NAT (Network Address Translation). Le mode NAT permet aux VMs de partager l'adresse IP de l'hôte pour accéder à Internet et de communiquer entre elles via le sous-réseau NAT.

VMs	os	IP	Services utilisés
Serversyslog	Ubuntu 24 server	192.168.224.130	Rsyslog, Iptables
Clientsyslog	Ubuntu 24 server	192.168.224.128	Rsyslog, SSH, Apache2, Fail2ban, Iptables/Nftables
Testeursyslog	Ubuntu 24 server	192.168.224.129	SSH

Testeusesyslog	Ubuntu 24 server	192.168.224.131	SSH
----------------	------------------	-----------------	-----

Voici des captures d'écran de vmware et de putty.



### 3. Vérifiez la connectivité entre toutes les machines à l'aide de la commande ping.

La commande ping sert à tester la connectivité entre les machines. Des machines Clientsyslog et Testeursyslog nous allons réaliser le ping vers Serveursyslog et Testeusesyslog. Les captures ci-dessous montrent les différents tests de connectivités entre les machines.

```

clientsyslog@clientsyslog:~$ ping 192.168.224.129
PING 192.168.224.129 (192.168.224.129) 56(84) bytes of data.
64 bytes from 192.168.224.129: icmp_seq=1 ttl=64 time=0.719 ms
64 bytes from 192.168.224.129: icmp_seq=2 ttl=64 time=0.745 ms
^C
--- 192.168.224.129 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.719/0.732/0.745/0.013 ms
clientsyslog@clientsyslog:~$ ^C
clientsyslog@clientsyslog:~$ ping 192.168.224.130
PING 192.168.224.130 (192.168.224.130) 56(84) bytes of data.
64 bytes from 192.168.224.130: icmp_seq=1 ttl=64 time=0.320 ms
64 bytes from 192.168.224.130: icmp_seq=2 ttl=64 time=0.605 ms
^C
--- 192.168.224.130 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.320/0.462/0.605/0.142 ms
clientsyslog@clientsyslog:~$ ping 192.168.224.131
PING 192.168.224.131 (192.168.224.131) 56(84) bytes of data.
64 bytes from 192.168.224.131: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 192.168.224.131: icmp_seq=2 ttl=64 time=0.324 ms
^C
--- 192.168.224.131 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1050ms
rtt min/avg/max/mdev = 0.324/0.338/0.353/0.014 ms
clientsyslog@clientsyslog:~$ 

```

```

testeusyslog@testeusyslog:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq_codel state UP group defa
link/ether 00:0c:29:45:dc:e3 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 192.168.224.129/24 metric 100 brd 192.168.224.255 scope global dynamic ens33
valid_lft 1705sec preferred_lft 1705sec
inet6 fe80::20c:29ff:fe45:dce3/64 scope link
valid_lft forever preferred_lft forever
testeusyslog@testeusyslog:~$ ping 192.168.224.128
PING 192.168.224.128 (192.168.224.128) 56(84) bytes of data.
64 bytes from 192.168.224.128: icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from 192.168.224.128: icmp_seq=2 ttl=64 time=0.410 ms
^C
--- 192.168.224.128 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1011ms
rtt min/avg/max/mdev = 0.410/0.411/0.412/0.001 ms
testeusyslog@testeusyslog:~$ ping 192.168.224.130
PING 192.168.224.130 (192.168.224.130) 56(84) bytes of data.
64 bytes from 192.168.224.130: icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from 192.168.224.130: icmp_seq=2 ttl=64 time=0.585 ms
64 bytes from 192.168.224.130: icmp_seq=3 ttl=64 time=0.448 ms
^C
--- 192.168.224.130 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2015ms
rtt min/avg/max/mdev = 0.448/0.765/1.264/0.356 ms
testeusyslog@testeusyslog:~$ ping 192.168.224.131
PING 192.168.224.131 (192.168.224.131) 56(84) bytes of data.
64 bytes from 192.168.224.131: icmp_seq=1 ttl=64 time=0.600 ms
64 bytes from 192.168.224.131: icmp_seq=2 ttl=64 time=0.316 ms
64 bytes from 192.168.224.131: icmp_seq=3 ttl=64 time=0.341 ms
^C
--- 192.168.224.131 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.316/0.419/0.600/0.128 ms
testeusyslog@testeusyslog:~$ 

```

#### 4. Installation des serveurs SSH (openssh-server) web (apache2) sur Clientsyslog

- **Installation, activation et démarrage du service SSH (openssh-server)**

```
sudo apt install ssh
```

```
clientsyslog@clientsyslog:~$ sudo apt install ssh
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ssh is already the newest version (1:9.6p1-3ubuntu13.8).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
```

```
sudo systemctl enable ssh
```

```
clientsyslog@clientsyslog:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
clientsyslog@clientsyslog:~$ _
```

```
sudo systemctl status ssh
```

```
clientsyslog@clientsyslog:~$ sudo systemctl status ssh
[sudo] password for clientsyslog:
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-20 23:31:00 UTC; 20min ago
 TriggeredBy: ● ssh.socket
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 1535 (sshd)
      Tasks: 1 (limit: 4552)
     Memory: 3.2M (peak: 4.1M)
        CPU: 128ms
     CGroup: /system.slice/ssh.service
            └─1535 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

- **Installation, activation et démarrage du service web (Apache2)**

```
sudo apt install apache2
```

```
clientsyslog@clientsyslog:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1
Suggested packages:
```

```
sudo systemctl enable apache2 && sudo systemctl status apache2
```

```
clientsyslog@clientsyslog:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
```

## Partie 2 : Test de la journalisation

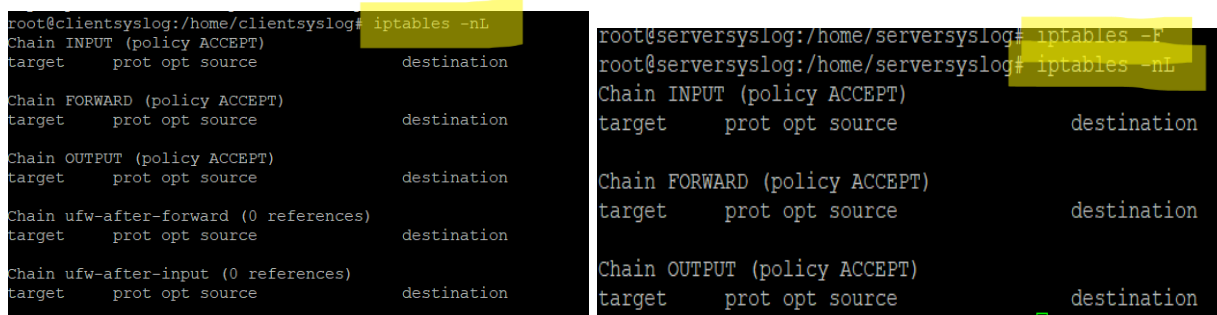
### 5. Suppression toutes les règles de firewall existantes sur Serversyslog et Clientsyslog.

Nous avons travaillé avec le pare-feu Iptables. Pour cela ufw est désactivé avec la commande ci-dessous:

```
sudo systemctl stop ufw && sudo systemctl disable ufw
```

Pour afficher les règles de firewall nous écrivons la commande : `iptables -nL`.  
Pour supprimer toutes les règles nous utilisons la commande: `iptables -F`

- Suppression des règles sur notre machine **Clientsyslog et Serversyslog**:



```
root@clientsyslog:/home/clientsyslog# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain ufw-after-forward (0 references)
target     prot opt source               destination

Chain ufw-after-input (0 references)
target     prot opt source               destination

root@serversyslog:/home/serversyslog# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

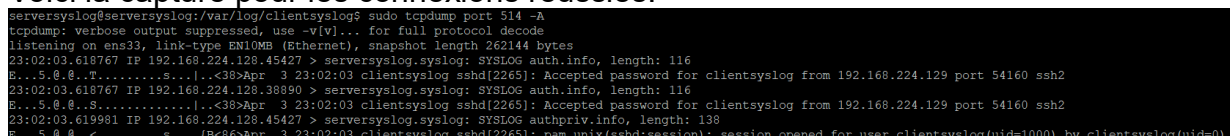
### 6. Interceptons uniquement les communications syslog entre Clientsyslog et Serversyslog durant les tentatives de connexion SSH de Testeursyslog (ou Testeusesyslog) à Clientsyslog, en incluant deux tentatives réussies et deux tentatives échouées.

Le protocole UDP est utilisé avec le port 514.

Depuis notre machine **serversyslog** (192.168.224.130) nous écrivons la commande :  
`sudo tcpdump udp port 514 -A`

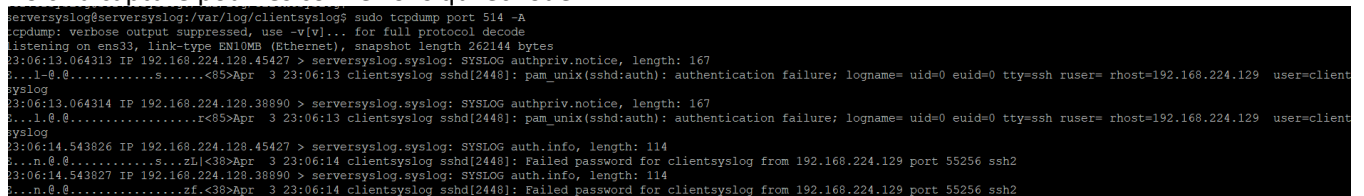
Cette commande nous permet d'intercepter les communications entre **Clientsyslog et Serveursyslog sur le port udp 514 lors de connexion ssh.**

Voici la capture pour les connexions réussies.



```
serversyslog@serversyslog:/var/log/clientsyslog$ sudo tcpdump port 514 -A
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:02:03.618767 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG auth.info, length: 116
E...5.0.0.T.....s...|<38>Apr  3 23:02:03 clientsyslog sshd[2265]: Accepted password for clientsyslog from 192.168.224.129 port 54160 ssh2
23:02:03.618767 IP 192.168.224.128.38890 > serversyslog.syslog: SYSLOG auth.info, length: 116
E...5.0.0.S.....s...|<38>Apr  3 23:02:03 clientsyslog sshd[2265]: Accepted password for clientsyslog from 192.168.224.129 port 54160 ssh2
23:02:03.619981 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG authpriv.info, length: 138
E...5.0.0.S.....s...|<38>Apr  3 23:02:03 clientsyslog sshd[2265]: pam_unix(sshd:session): session opened for user clientsyslog(uid=1000) by clientsyslog(uid=0)
```

Voici la capture pour les connexions qui échoue.



```
serversyslog@serversyslog:/var/log/clientsyslog$ sudo tcpdump port 514 -A
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:06:13.064313 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG authpriv.notice, length: 167
E...1.0.0.....s...|<85>Apr  3 23:06:13 clientsyslog sshd[2448]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.224.129 user=clientsyslog
23:06:13.064314 IP 192.168.224.128.38890 > serversyslog.syslog: SYSLOG authpriv.notice, length: 167
E...1.0.0.....s...|<85>Apr  3 23:06:13 clientsyslog sshd[2448]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.224.129 user=clientsyslog
23:06:14.543826 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG auth.info, length: 114
E...n.0.0.....s...zL|<38>Apr  3 23:06:14 clientsyslog sshd[2448]: Failed password for clientsyslog from 192.168.224.129 port 55256 ssh2
23:06:14.543827 IP 192.168.224.128.38890 > serversyslog.syslog: SYSLOG auth.info, length: 114
E...n.0.0.....s...zL|<38>Apr  3 23:06:14 clientsyslog sshd[2448]: Failed password for clientsyslog from 192.168.224.129 port 55256 ssh2
```

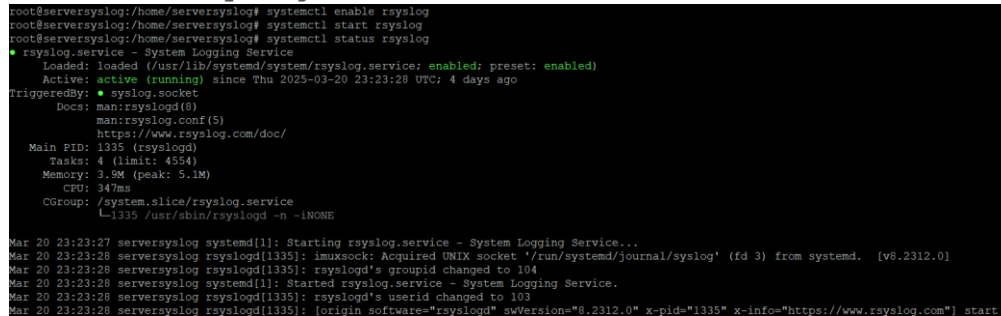
### 7. Configurons Serversyslog pour autoriser uniquement le flux syslog en provenance de Clientsyslog

Si Rsyslog n'est pas installé, il faut exécuter la commande suivante:

```
sudo apt install rsyslog -y
```

Dans notre cas, Rsyslog est installé. Nous allons activer, démarrer et voir le status du service

```
systemctl enable rsyslog
systemctl start rsyslog
systemctl status rsyslog
```

A terminal window showing the execution of systemctl commands to enable, start, and check the status of the rsyslog service. The output indicates that the service is active and running, with details about its configuration and resources.

```
root@serversyslog:/home/serversyslog# systemctl enable rsyslog
root@serversyslog:/home/serversyslog# systemctl start rsyslog
root@serversyslog:/home/serversyslog# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-20 23:23:28 UTC; 4 days ago
   TriggeredBy: ● syslog.socket
   Docs: man:rsyslogd(8)
         man:rsyslog.conf(5)
         https://www.rsyslog.com/doc/
  Main PID: 1335 (rsyslogd)
    Tasks: 4 (limit: 4554)
   Memory: 3.9M (peak: 5.1M)
      CPU: 347ms
   CGroup: /system.slice/rsyslog.service
           └─1335 /usr/sbin/rsyslogd -n -iNONE

Mar 20 23:23:27 serversyslog systemd[1]: Starting rsyslog.service - System Logging Service...
Mar 20 23:23:28 serversyslog rsyslogd[1335]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd. [v8.2312.0]
Mar 20 23:23:28 serversyslog rsyslogd[1335]: rsyslogd's groupid changed to 104
Mar 20 23:23:28 serversyslog systemd[1]: Started rsyslog.service - System Logging Service.
Mar 20 23:23:28 serversyslog rsyslogd[1335]: rsyslogd's userid changed to 103
Mar 20 23:23:28 serversyslog rsyslogd[1335]: [origin software="rsyslogd" swVersion="8.2312.0" x-pid="1335" x-info="https://www.rsyslog.com"] start
```

## • Configuration de Rsyslog sur **Serversyslog**

Finalement nous allons éditer le fichier `/etc/rsyslog.conf` pour activer les envois de logs UDP et TCP. Décommenter (enlever le #) les lignes suivantes :

```
module(load="imudp")
input(type="imudp" port="514")
module(load="imtcp")
input(type="imtcp" port="514")
```

Puis ajoutons un modèle qui permet de stocker les logs. Chaque machine qui envoie ses logs aura un répertoire avec son hostname. Dans ce répertoire se trouvera tous les logs de la machine. Dans notre cas, les logs envoyés par la machine **Clientsyslog** doivent se trouver sur le serveur dans le répertoire `/var/logs/Clientsyslog`.

```
$template remote-incoming-
logs, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.*?remote-incoming-logs
& stop
```

```
$AllowedSender UDP, 192.168.224.128
```

La ligne `AllowedSender` autorise la machine **Clientsyslog** à envoyer ses logs au **Serveursyslog**.

Sauvegarder le fichier et redémarrer le service : `systemctl restart rsyslog`

```

module(load="imuxsock") # provides support for local system logging
#module(load="imark") # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")

# provides kernel logging support and enable non-kernel klog messages
module(load="imklog" permitnonkernelfacility="on")

#####
### GLOBAL DIRECTIVES ###
#####

# Filter duplicated messages
$RepeatedMsgReduction on

#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog

#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog

#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
$template remote-incoming-logs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.?remote-incoming-logs
$ stop
$AllowedSender UDP, 192.168.224.128

```

8. Redirigeons tous les logs du serveur web vers Serversyslog. Interceptons le trafic réseau de deux tentatives d'accès web depuis Clientsyslog vers le serveur web. Vérifions si l'adresse IP de la machine Clientsyslog est présente dans les logs sur le serveur Serversyslog.

- Redirigeons tous les logs du serveur web vers la machine **Serversyslog**:  
Ajoutons la ligne ci-dessous au fichier /etc/rsyslog.conf sur Clientsyslog:

```
*.* @192.168.224.130:514
```

La ligne ci-dessus permet de rediriger les logs de **Clientsyslog** vers le **Serversyslog**. Nous devons redémarrer le service après modification du fichier de configuration. Sauvegarder le fichier et redémarrer le service : `systemctl restart rsyslog`

```

# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
*.? @192.168.224.130:514

```

- Configurons le fichier 000-default.conf sur le serveur **clientsyslog**.

Ajouter les lignes suivantes :

```
ErrorLog "|/usr/bin/logger -t apache_error -p local1.err"
CustomLog "|/usr/bin/logger -t apache_access -p local1.info" combined
```

```
clientsyslog@clientsyslog:/var/log/apache2$ sudo vi /etc/apache2/sites-available/000-default.conf
[sudo] password for clientsyslog:
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    #ErrorLog ${APACHE_LOG_DIR}/error.log
    #CustomLog ${APACHE_LOG_DIR}/access.log combined
    ErrorLog "|/usr/bin/logger -t apache_error -p local1.err"
    CustomLog "|/usr/bin/logger -t apache_access -p local1.info" combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

- Trafic réseau de deux tentatives d'accès web depuis **clientsyslog**:  
Sur **testeusesyslog**, nous avons exécuté la commande curl 2 fois.

```
testeusesyslog@testeusesyslog:~$ curl http://192.168.224.128
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
```

La capture réseau sur **serversyslog** prouve que nous recevons les logs apache de **clientsyslog** lors des connexions de **testeusesyslog**. Dans ce log sont présent les **adresses Ip** du **Clientsyslog** (serveur web: 192.168.222.128) et celle de la **Testeusesyslog** (192.168.222.131).



```

serversyslog@serversyslog:/var/log/clientsyslog$ sudo tcpdump port 514 -A
tcpdump: verbose output suppressed, use -v[... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:17:28.327851 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG local1.info, length: 141
E.....@. ....s.....K<142>Apr  3 23:17:28 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:23:17:28 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
23:17:28.327851 IP 192.168.224.128.38890 > serversyslog.syslog: SYSLOG local1.info, length: 141
E.....@. ....s.....K<142>Apr  3 23:17:28 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:23:17:28 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
23:17:29.847489 IP 192.168.224.128.45427 > serversyslog.syslog: SYSLOG local1.info, length: 141
E....v@.@.jy.....s.....I<142>Apr  3 23:17:29 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:23:17:29 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
23:17:29.847592 IP 192.168.224.128.38890 > serversyslog.syslog: SYSLOG local1.info, length: 141
E....w@.@.x.....<142>Apr  3 23:17:29 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:23:17:29 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"

```

Voici le fichier de log sur la machine **serversyslog**, nous pouvons observer les logs dans les 4 dernières lignes de la capture :

```

serversyslog@serversyslog:~$ cd /var/log/clientsyslog/
serversyslog@serversyslog:/var/log/clientsyslog$ ls
50-motd-news.log      apt.systemd.daily.log  fail2ban-client.log    lvm.log                polkit-agent-helper-1.log  snapd.log              systemd.log            systemd-timesyncd.log
apache_access.log     '(cron).log'          fail2ban-server.log    ModemManager.log       polkitd.log               sshd.log               systemd-logind.log     systemd-udevadm.log
apachectl.log         cron.log               fwupd.log              multipathd.log          python3.log               sudo.log               systemd-modules-load.log '(udev-worker).log'
apache_error.log      CRON.log              fwupdmgd.log           multipath.log           rsyslogd.log              systemd-fsck.log       systemd-networkd.log   udiskd.log
apparmor.systemd.log  dbus-daemon.log       kernel.log              PackageKit.log          snapd-apparmor.log        '(systemd).log'        systemd-resolved.log   VGAuthService.log
serversyslog@serversyslog:/var/log/clientsyslog$ tail -f apache_access.log

2025-04-03T22:26:03+00:00 clientsyslog apache_access: 192.168.224.1 - - [03/Apr/2025:22:26:03 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36"
2025-04-03T22:26:03+00:00 clientsyslog apache_access: 192.168.224.1 - - [03/Apr/2025:22:26:03 +0000] "GET / HTTP/1.1" 200 3460 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36"
2025-04-03T22:26:03+00:00 clientsyslog apache_access: 192.168.224.1 - - [03/Apr/2025:22:26:03 +0000] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://192.168.224.128/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36"
2025-04-03T22:26:03+00:00 clientsyslog apache_access: 192.168.224.1 - - [03/Apr/2025:22:26:03 +0000] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://192.168.224.128/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36"
2025-04-03T22:27:45+00:00 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:22:27:45 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
2025-04-03T22:27:45+00:00 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:22:27:45 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
2025-04-03T22:27:46+00:00 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:22:27:46 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"
2025-04-03T22:27:46+00:00 clientsyslog apache_access: 192.168.224.131 - - [03/Apr/2025:22:27:46 +0000] "GET / HTTP/1.1" 200 10926 "-" "curl/8.5.0"

```

### Partie 3 : Installation d'outils IDS/IPS

#### 9. Machine à configurer pour respecter la politique suivante :

La machine à configurer est **Clientsyslog**, car c'est elle qui héberge le service SSH. Fail2Ban doit être installé et configuré sur cette machine afin de surveiller les connexions SSH et appliquer la politique de blocage.

#### 10. Expliquez brièvement le fonctionnement de Fail2ban.

Fail2Ban est un outil de prévention des intrusions qui surveille les fichiers journaux (comme /var/log/auth.log) de divers services (comme SSH) à la recherche de tentatives de connexion suspectes. Lorsqu'il détecte un nombre excessif d'échecs de connexion provenant d'une même adresse IP, il bloque temporairement cette adresse en modifiant les règles du pare-feu du système (iptables ou nftables). Après une période de blocage prédéfinie, l'adresse IP est automatiquement débloquée. Fail2Ban utilise des filtres et des expressions régulières pour analyser les logs et identifier les schémas d'attaques potentielles.

#### 11. Implémentez la politique de blocage mentionnée à l'aide de Fail2ban.

##### • Installation de Fail2Ban

```
Sudo apt update && apt install fail2ban -y
```

```

clientsyslog@clientsyslog:/var/log/apache2$ sudo apt update && apt install fail2ban -y
[sudo] password for clientsyslog:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [711 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [960 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [136 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8,960 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [7,068 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [810 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [213 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [842 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [164 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [468 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [822 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [170 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [492 B]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,044 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:24 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]

```

- Créer le fichier de configuration : `vi /etc/fail2ban/jail.local`

- enabled = Activer la règle
- Logpath= Fichier de log à consulter
- maxretry = nombre de tentative de connexion avec échec
- findtime = Intervalle de temps de contrôles en second
- bantime = Temps de banissement en second

```

[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 2
findtime = 300
bantime = 3600
[DEFAULT]
allowip6 = auto

```

- Redémarrer le service : `systemctl restart fail2ban.`

## 12. Effectuons trois tentatives de connexion infructueuses depuis Testeursyslog (ou Testeusesyslog) vers la machine Clientsyslog.

- Après 3 tentatives infructueuses, vérifions l'état du service avec la commande :  
`fail2ban-client status sshd`

```

root@clientsyslog:/var/log/apache2# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed: 2
| `-- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`-- Actions
   |- Currently banned: 1
   |- Total banned: 1
   `-- Banned IP list: 192.168.224.129

```

En essayant de nous connecter au travers de la connexion SSH nous voyons qu'après trois fausses tentatives la connexion est refusée.

```

testeursyslog@testeursyslog:~$ ssh clientsyslog@192.168.224.128
ssh: connect to host 192.168.224.128 port 22: Connection refused

```

13. Affichons puis Désactivons des règles actives du firewall sur Clientsyslog.  
Désactivons les règles et réitérons la tentative de connexion.

Pour afficher les règles actives du firewall sur la machine Clientsyslog, nous utilisons la commande : `nft list ruleset`

```

root@clientsyslog:/var/log/apache2# nft list ruleset
table inet f2b-table {
    set addr-set-sshd {
        type ipv4_addr
        elements = { 192.168.224.129 }
    }

    chain f2b-chain {
        type filter hook input priority filter - 1; policy accept;
        tcp dport 22 ip saddr @addr-set-sshd reject with icmp port-unreachable
    }
}

```

Et lorsque nous désactivons Fail2Ban en écrivant la commande : `sudo nft delete table inet f2b-table`

```

root@clientsyslog:/var/log/apache2# sudo nft delete table inet f2b-table

```

Nous constatons qu'il n'y a plus de règles :

```

root@clientsyslog:/var/log/apache2# nft list ruleset
root@clientsyslog:/var/log/apache2#

```

Une fois la règle effacée dans le firewall, nous pouvons voir ci-dessous que la connexion fonctionne :

```

testeursyslog@testeursyslog:~$ ssh clientsyslog@192.168.224.128
clientsyslog@192.168.224.128's password:

```

Explication:

Fail2ban surveille les fichiers journaux pour détecter les tentatives de connexion échouées (SSH). Lorsqu'un nombre excessif de tentatives échouées est détecté, Fail2ban ajoute des règles à Iptables/Nftables pour bloquer l'adresse IP de la machine attaquante. Le flush des règles Iptables/Nftables supprime les blocages mis en place par fail2ban, ce qui permet à la testeusessyslog de se connecter à nouveau à Clientsyslog.