

Lab 2. Bastionado de sistemas:

Fecha: 14/10/2024

Elaborado por: Alvaro Rey Jimenez

- **INSTALACIÓN DE SERVICIOS MÍNIMOS:**

- Vamos a preparar un servidor de Ubuntu server con los servicios necesarios es decir. SSH, FTP, el servidor MySQL y el servidor Apache
- Hay que asegurarse que este servidor este accesible para poder hacer ping y que se puedan ver ambas máquinas
- Actualizaremos los paquetes del sistema antes de realizar todas las instalaciones
- Los comandos para instalar todos esos servicios son:

```
sudo apt-get install ssh
```

```
sudo apt-get install vsftpd
```

```
sudo apt-get install mysql-server
```

```
sudo apt-get install apache2
```

- Una vez instalados todos los servicios necesarios, tendremos que habilitarlos para que puedan funcionar
- Para ello, utilizaremos los siguientes comandos:

```
sudo systemctl enable ssh
```

```
sudo systemctl enable vsftpd
```

```
sudo systemctl enable mysql
```

```
sudo systemctl enable apache2
```

- Si utilizamos el comando que se verá en la siguiente imagen, podemos confirmar que estos servicios están habilitados

```
ciber_arj@especializacionciber:~$ sudo systemctl | grep running
proc-sys-fs-binfmt_misc.automount      loaded active running Arbitrary Executable File Formats File S
systemd-automount.service               loaded active running System and Service Manager
systemd-initrd.service                  loaded active running Session 1 of User ciber_arj
systemd-journald.service                 loaded active running The Apache HTTP Server
systemd-logind.service                   loaded active running Regular background program processing da
systemd-networkd.service                 loaded active running D-Bus System Message Bus
systemd-resolved.service                 loaded active running Fail2Ban Service
systemd-udev.service                     loaded active running Firmware update daemon
systemd-udevd.service                    loaded active running Getty on tty1
systemd-udisks2.service                  loaded active running Modem Manager
systemd-ufsd.service                     loaded active running Device-Mapper Multipath Device Controlle
systemd-usb-lcd.service                   loaded active running MySQL Community Server
systemd-usb-lcd.service                   loaded active running Authorization Manager
systemd-usb-lcd.service                   loaded active running System Logging Service
systemd-usb-lcd.service                   loaded active running Snap Daemon
systemd-usb-lcd.service                   loaded active running OpenBSD Secure Shell server
systemd-usb-lcd.service                   loaded active running Journal Service
systemd-usb-lcd.service                   loaded active running User Login Management
systemd-usb-lcd.service                   loaded active running Network Configuration
systemd-usb-lcd.service                   loaded active running Network Name Resolution
systemd-usb-lcd.service                   loaded active running Network Time Synchronization
systemd-usb-lcd.service                   loaded active running Rule-based Manager for Device Events and
udisks2.service                          loaded active running Disk Manager
unattended-upgrades.service               loaded active running Unattended Upgrades Shutdown
upower.service                           loaded active running Daemon for power management
vsftpd.service                           loaded active running User Manager for UID 1000
vsftpd.service                           loaded active running vsftpd FTP server
dbus.socket                              loaded active running D-Bus System Message Bus Socket
multipathd.socket                        loaded active running multipathd control socket
cni.socket                               loaded active running Socket activation for snappy daemon
ssh.socket                               loaded active running OpenBSD Secure Shell server socket
systemd-journald-dev-log.socket           loaded active running Syslog Socket
systemd-journald.socket                   loaded active running Journal Socket (/dev/log)
systemd-networkd.socket                   loaded active running Journal Socket
systemd-udev-control.socket               loaded active running Network Service Netlink Socket
systemd-udev-kernel.socket                loaded active running udev Control Socket
systemd-udev-kernel.socket                loaded active running udev Kernel Socket
```

- Se puede ver q todos los servicios están activos pero hay que iniciar todos los servicios habilitados.
- Para iniciarlos, habrá que utilizar los siguientes comandos

```
service ssh start
```

```
service vsftpd start
```

```
service mysql start
```

```
service apache2 start
```

- Para confirmar que absolutamente todos los servicios están iniciados, utilizaremos los comandos que se ven en las siguientes

imagenes.

```
ciber_arj@especializacionciber:~$ systemctl status vsftpd
• vsftpd.service - vsftpd FTP server
  Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-10-07 16:31:44 UTC; 55min ago
  Main PID: 733 (vsftpd)
  Tasks: 1 (limit: 2276)
  Memory: 928.0K (peak: 1.5M)
  CPU: 13ms
  CGroup: /system.slice/vsftpd.service
          └─733 /usr/sbin/vsftpd /etc/vsftpd.conf
```

```
ciber_arj@especializacionciber:~$ systemctl status apache2
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-10-07 16:31:51 UTC; 56min ago
  Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 915 (apache2)
  Tasks: 55 (limit: 2276)
  Memory: 7.6M (peak: 7.7M)
  CPU: 339ms
  CGroup: /system.slice/apache2.service
          └─915 /usr/sbin/apache2 -k start
            └─917 /usr/sbin/apache2 -k start
              └─918 /usr/sbin/apache2 -k start
```

```
ciber_arj@especializacionciber:~$ systemctl status mysql
• mysql.service - MySQL Community Server
  Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-10-07 16:32:05 UTC; 56min ago
  Main PID: 1039 (mysqld)
  Status: "Server is operational"
  Tasks: 38 (limit: 2276)
  Memory: 423.0M (peak: 437.9M)
  CPU: 29.002s
  CGroup: /system.slice/mysql.service
          └─1039 /usr/sbin/mysqld
```

```
ciber_arj@especializacionciber:~$ systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Mon 2024-10-07 16:31:46 UTC; 57min ago
  TriggeredBy: • ssh.socket
  Docs: man:sshd(8)
         man:sshd_config(5)
  Main PID: 835 (sshd)
  Tasks: 1 (limit: 2276)
  Memory: 2.1M (peak: 2.4M)
  CPU: 34ms
  CGroup: /system.slice/ssh.service
          └─835 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

- Ahora hay que configurar el firewall de tal manera que pueda permitir solamente los puertos que se habiliten. Para ello, hay que ver en que puerto están los servicios activos. En mi caso al instala MySQL, el puerto por defecto de este es el 3306, el del SSH es el puerto 22, el de FTP en el puerto 21 y el Apache en el 80.
- Estos puertos se pueden ver con el comando:

```
nano /etc/services
```

- Para habilitar el firewall, habrá que utilizar los comandos de la siguiente imagen:

```
ciber_arj@especializacionciber:~$ sudo ufw allow 3306
Rules updated
Rules updated (v6)
ciber_arj@especializacionciber:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
ciber_arj@especializacionciber:~$ sudo ufw allow 443
Rules updated
Rules updated (v6)
ciber_arj@especializacionciber:~$ sudo ufw allow 21
Rules updated
Rules updated (v6)
ciber_arj@especializacionciber:~$ sudo ufw allow 80
Rules updated
Rules updated (v6)
ciber_arj@especializacionciber:~$
```

- Si hacemos un escaneo de nmap, podremos ver que no nos sale nuestro servicio mysql. Ya que por defecto el puerto sale cerrado

```
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
> nmap -p3306 192.168.129.31
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-07 20:05 CEST
Nmap scan report for 192.168.129.31
Host is up (0.00085s latency).

PORT      STATE SERVICE
3306/tcp  closed mysql
MAC Address: 08:00:27:01:0F:70 (Oracle VirtualBox virtual NIC)
```

- Por lo cual habrá que modificarlo con los parámetros de la siguiente imagen:

```
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# * IMPORTANT: Additional settings that can override those from this file!
#   The files must end with '.cnf', otherwise they'll be ignored.
#

!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
user=mysql
port=3306
bind-address=0.0.0.0
```

• **ESCANEO DE PUERTOS ABIERTOS:**

- Ahora que ya hemos configurado absolutamente todo en nuestro Ubuntu server, llega la hora de realizar un escaneo en Nmap para encontrar vulnerabilidades. Para ello utilizaremos el siguiente comando:

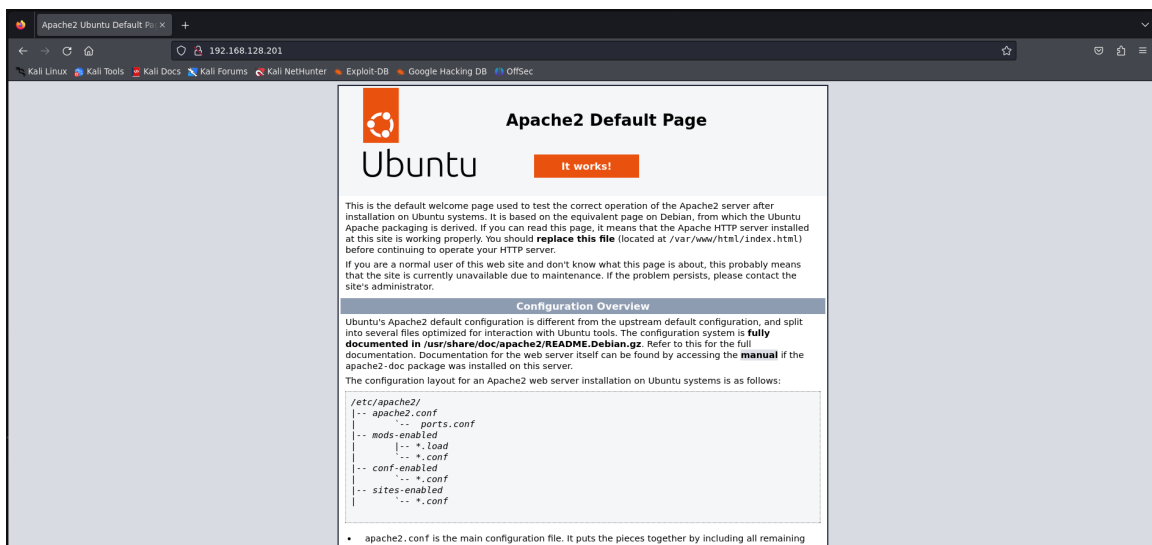
```
nmap -sS -sV -v -O 192.168.128.201
```

- -sS: Esto significa SYN Scan. Es una opción que te permite escanear puertos de una manera rápida y efectiva en servidores que no tenga un firewall con políticas estrictas
- -sV: Este comando te detecta la versión que esta utilizando el servicio que corre en el puerto abierto detectado
- -O: Este comando te detecta el tipo de sistema operativo que esta corriendo en ese puerto abierto
- -v: Este comando lo que hace es ir dando resultados a medida que los va encontrando para poder ir tocando herramientas incluso cuando no

haya acabado el escaneo.

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 3.0.5
22/tcp	open	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	Apache httpd 2.4.58 ((Ubuntu))
443/tcp	closed	https	
3306/tcp	open	mysql	MySQL (unauthorized)

- Esto es lo que nos ha salido con los parámetros utilizados. Como se puede ver tenemos el servicio ftp, ssh, http y mysql abiertos. Si entramos en la IP de Ubuntu Server, podremos ver que tenemos el servicio apache corriendo.



• **DESHABILITAR SERVICIOS INNECESARIOS:**

- Sabiendo los resultados que nos ha dado Nmap, podemos ver que tenemos el servicio ftp y ssh habilitados y corriendo. En este caso al tener ssh, el servicio ftp no nos servirá para nada ya que es un servicio mucho menos seguro y contiene mas vulnerabilidades por lo cual lo deshabilitaremos.
- Una vez deshabilitado, tendremos que configurar ssh para que el atacante no pueda acceder al servidor con autenticación root.
- En cuanto a MySQL, limitaremos el acceso solo a IPS de confianza
- Y en apache intentaremos que solo se tenga acceso a información limitada de tal manera que no se pueda ver información importante
- **DESHABILITAR FTP**

- Para deshabilitar ftp utilizaremos el siguiente comando:

```
sudo systemctl disable vsftpd
```

- Cuando este deshabilitado, tendremos que establecer las reglas del firewall. Ya que aunq el servicio este deshabilitado, el puerto sigue abierto por lo cual utilizaremos estos comandos:

```
sudo ufw deny 21/tcp
```

```
sudo ufw deny 21
```

- Estos comandos nos cerraran los puertos por los que ftp pueda ser visible. Pra comprobar que estan cerrados, yo he reiniciado el Ubuntu server y he hecho un escaneo de nmap rápido para que me detecte solo los puertos abiertos

```
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http      Apache httpd 2.4.58 ((Ubuntu))
443/tcp   closed https
3306/tcp  open  mysql     MySQL (unauthorized)
```

- Como se puede ver, ya no sale el servicio FTP en la máquina
- LIMITAR ACCESO MYSQL
 - En cuanto a MySQL, no nos interesa que se tenga acceso desde todas las IPS por lo cual configuraremos desde un archivo específico que solo tenga acceso mi maquina Kali Linux. Además queremos que funcione por lo cual no voy a deshabilitar el servicio de MySQL pero si voy a cerrar los puertos.
 - Este archivo se encuentra en el siguiente directorio

```
/etc/mysql/mysql.conf.d/mysqld.cnf
```

- Lo editaremos como nano:

```

GNU nano 7.2 mysold.cnf *
# If MySQL is running as a replication slave, this should be
# changed. Ref: https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir
#   = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 192.168.129.37
mysqlx-bind-address     = 192.168.129.37
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K
#
# thread_cache_size     = -1
#
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options  = BACKUP
#
# max_connections       = 151
#
# table_open_cache      = 4000
#
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file       = /var/log/mysql/query.log
# general_log            = 1
#
# Error log - should be very few entries.
log_error               = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log          = 1
# slow_query_log_file    = /var/log/mysql/mysql-slow.log
# long_query_time         = 2
# log-queries-not-using-indexes
#
#
# Help      Write Out  Where Is  Cut      Execute  Location  Undo      Set Mark  To Bracket  Previous
# Exit     Read File  Replace  Paste    Justify  Go To Line Redo      Copy      Where Was  Next

```

- Lo que esta recuadrado, lo cambiaremos por la IP que queramos. En mi caso la IP de Kali linux
- Como se puede ver en la siguiente imagen, nuestro kali linux detecta el mysql abierto

Not shown: 990 filtered tcp ports (no response)				
PORT	STATE	SERVICE	VERSION	
22/tcp	open	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)	
80/tcp	open	http	Apache httpd 2.4.58 ((Ubuntu))	
443/tcp	closed	https		
3306/tcp	open	mysql	MySQL (unauthorized)	

- Ahora cerraremos el puerto de mysql. Para ello miraremos el status de nuestro firewall con los mismos comandos anteriores

```
sudo ufw deny 3306/tcp
```

```
sudo ufw deny 3306
```

Not shown: 990 filtered tcp ports (no response)				
PORT	STATE	SERVICE	VERSION	
22/tcp	open	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)	
80/tcp	open	http	Apache httpd 2.4.58 ((Ubuntu))	
443/tcp	closed	https		

- Ya no tendremos MYSQL en los puertos abiertos. Lo tenemos totalmente seguro

- CONFIGURACIÓN SSH

- SSH necesita los puertos abiertos ya que sin el, no podremos tener conexión remota para transferencia de archivos. Por lo cual necesitaremos aplicar una serie de reglas para que no pueda entrar cualquier equipo.
- Hay un programa que se llama fail2ban, que lo utilizaremos para establecer una serie de políticas pero el archivo de SSH también se podrá configurar.

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

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

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
```

- En esta imagen, hemos cambiado dos parámetros que son necesarios para que no entre cualquiera. Este archivo se encuentra en el directorio:

`/etc/ssh/sshd_config`

- Ahora utilizaremos la herramienta fail2ban para establecer políticas de ssh. Si no se tiene instalada la herramienta, se instalara con el siguiente comando:

`sudo apt-get install fail2ban`

- Crearemos un nuevo archivo con nano de tal manera que podamos escribir todas las políticas de seguridad. En ese archivo (que lo llamé jail.local) pondremos estos parámetros:

```
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
```

- Una vez hecho esto, reiniciaremos el servicio con el comando de siempre

```
sudo systemctl restart fail2ban
```

• CONFIGURACIÓN APACHE

- Apache es un servicio bastante seguro. No obstante, instalaremos una herramienta la cual nos lo proteja mas. Esta herramienta se llama "mod security"
- La instalaremos con este comando:

```
sudo apt-get install libapache2-mod-security2
```

- Una vez instalado, lo ejecutaremos y reiniciaremos el sistema

```
sudo a2enmod security2
```

- Por ultimo reiniciaremos el servicio como todos con el siguiente comando:

```
sudo systemctl restart apache2
```

• CONCLUSION

- Para el servicio FTP, lo he deshabilitado porque ya tenemos otra herramienta mucho mas segura para permitir la conexión remota
- En cuanto a MYSQL he limitado el acceso a solo la IP que he configurado en el archivo de configuración además de cerrar los puertos para que no se vea que esta corriendo este servicio
- SSH es un servicio que necesita tener los puertos abiertos para utilizar la conexión remota por lo cual, he actualizado las políticas de

seguridad tanto en fail2ban como en la propia configuración de SSH

- En Apache he instalado un servicio de las propias librerías de Apache para mantener el servicio https seguro
- En el firewall he deshabilitado los puertos innecesarios es decir el de mysql y el de ftp