

Práctica 8: Mecanismos de Seguridad para acceso a un servidor

Autor: Manuel Díaz-Meco Terrés

Fecha: 17 de noviembre 2024

Introducción

El objetivo de esta práctica es saber configurar elementos adicionales de seguridad para reducir el riesgo de ataques por accesos no autorizados mediante un sistema Host IDS y añadir autenticación de segundo factor.

Ejercicio 1: Fail2Ban

En primer lugar, inicializamos nuestra máquina de la práctica 5, la que usa *Jammy64* e instalamos **fail2ban** mediante `sudo apt install fail2ban`, tras esto hacemos lo siguiente:

```
vagrant@ubuntu-jammy:~$ sudo systemctl start fail2ban
vagrant@ubuntu-jammy:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-11-17 16:38:01 UTC; 7s ago
     Docs: man:fail2ban(1)
    Main PID: 2128 (fail2ban-server)
      Tasks: 5 (limit: 2309)
    Memory: 11.9M
       CPU: 86ms
    CGroup: /system.slice/fail2ban.service
            └─2128 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Nov 17 16:38:01 ubuntu-jammy systemd[1]: Started Fail2Ban Service.
Nov 17 16:38:01 ubuntu-jammy fail2ban-server[2128]: Server ready
vagrant@ubuntu-jammy:~$ sudo apt iptables -L -n
! Command line option '-L' (from -L) is not understood in combination with the other options.
vagrant@ubuntu-jammy:~$ sudo iptables -L -n
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
vagrant@ubuntu-jammy:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
  - Jail list:          sshd
vagrant@ubuntu-jammy:~$
```

Práctica 8: Mecanismos de Seguridad para acceso a un servidor

Autor: Manuel Díaz-Meco Terrés

Fecha: 17 de noviembre 2024

I) Fail2Ban

Existen programas más sencillos bloqueo de la IP de acceso. Sin e Además de SSH puede controlar Squid, Proftpd, Vsftpd, Postfix, S

Fail2ban establece "jaulas" (jail) posible ataque en el fichero estal al cortafuegos bloqueando la entr

Para instalar fail2ban en Ubuntu:

```
sudo apt-get install fail2ban
```

En CentOS/RPM debemos tener

```
yum install fail2ban fail2ban-sys
systemctl enable fail2ban
systemctl start fail2ban
```

```
sudo fail2ban-client status
```

Por defecto se activa la jaula de s

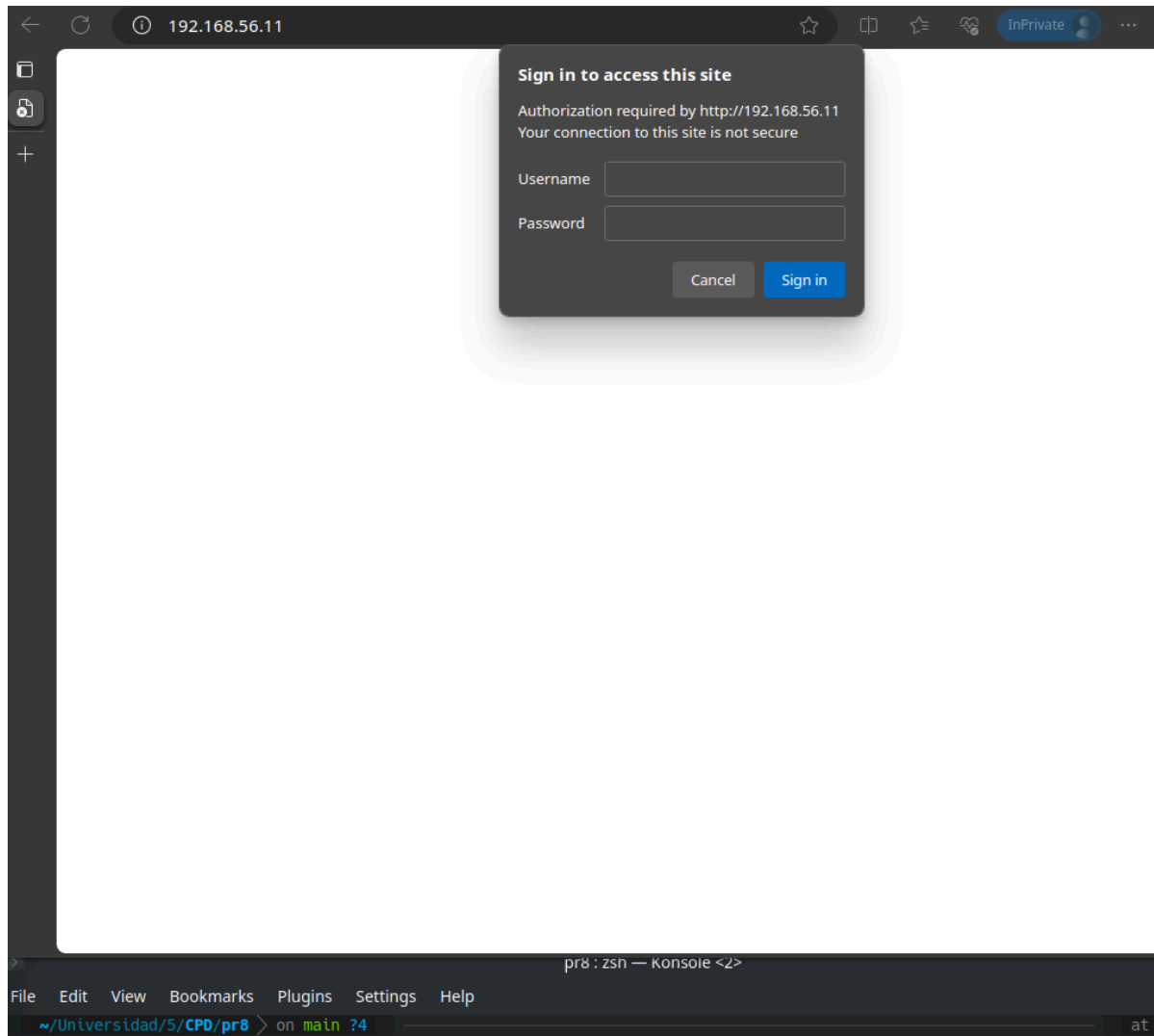
Podemos comprobar las reglas er

```
sudo iptables -L -n
```

Ahora, comprobamos que el servicio funciona intentando entrar sin éxito 3 veces desde nuestro host. Como se ve en la siguiente imagen, tras hacer esto se banea nuestra dirección IP, pudiendo comprobar eso con el comando `sudo iptables -L -n`:


```
location / {  
    try_files $uri $uri/ =404;  
    auth_basic "Restricted Content";  
    auth_basic_user_file /etc/nginx/.htpasswd;  
}
```

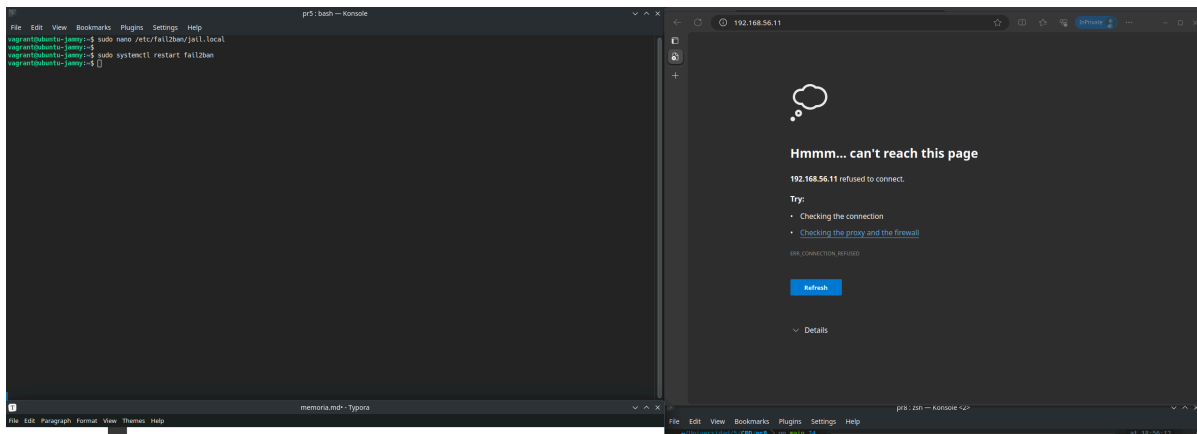
De esta forma cuando accedamos se nos pedirá usuario y contraseña. Recalcar que si se hace en el navegador se guarda la caché y ya no te lo vuelve a pedir. Por eso he accedido otra vez al servidor desde una página *InPrivate* o de incógnito.



Ahora, editamos el archivo `/etc/fail2ban/jail.local`:

```
[nginx-http-auth]  
enabled = true  
port    = http,https  
logpath = %(nginx_error_log)s  
  
# To use 'nginx-limit-req' jail you should have 'ngx_http_limit_req_module'  
# and define 'limit_req' and 'limit_req_zone' as described in nginx documentation
```

Y, tras intentar unas cuantas veces acceder erróneamente al servidor **nginx** obtendremos lo siguiente:



Opcional 2: Utilizar la red TOR para acceder directamente a la máquina virtual

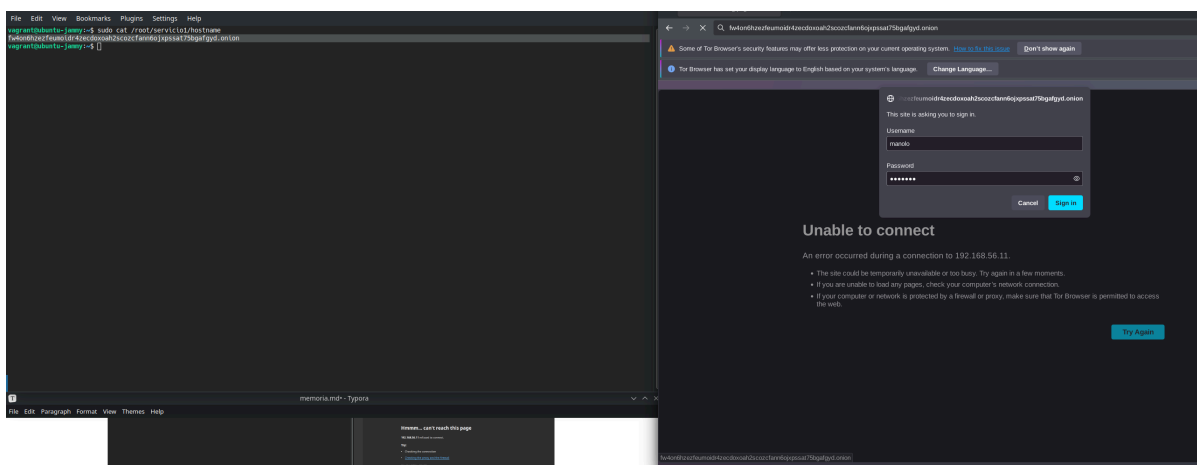
En primer lugar instalamos **tor** en la máquina *jammy* y modificamos el archivo `etc/tor/torrc` como se nos dice en el guión:

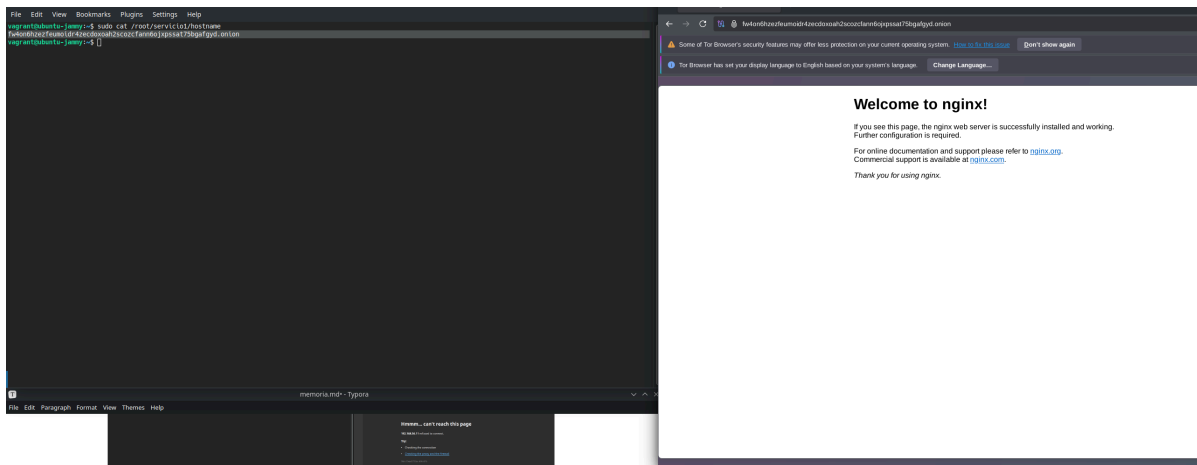
```
RunAsDaemon 1
HiddenServiceDir /root/servicio1
HiddenServicePort 22 127.0.0.1:22
HiddenServicePort 80 127.0.0.1:80
```

Una vez hecho eso ejecutamos tor:

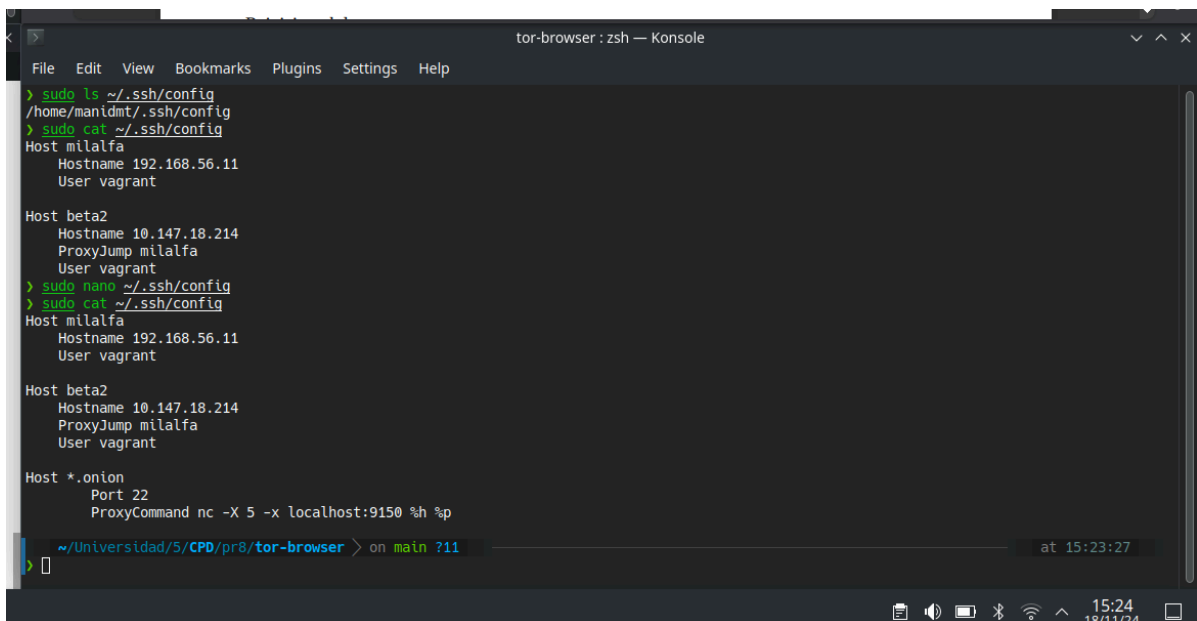
```
vagrant@ubuntu-jammy:~$ sudo tor
Nov 17 18:32:37.850 [notice] Tor 0.4.6.10 running on Linux with Libevent 2.1.12-stable, OpenSSL 3.0.2, Zlib 1.2.11, Liblzma 5.2.5, Libzstd 1.4.8 and Glibc 2.35 as libc.
Nov 17 18:32:37.850 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://support.torproject.org/faq/staying-anonymous/
Nov 17 18:32:37.850 [notice] Read configuration file "/etc/tor/torrc".
Nov 17 18:32:37.852 [notice] Opening Socks listener on 127.0.0.1:9050
Nov 17 18:32:37.852 [notice] Opened Socks listener connection (ready) on 127.0.0.1:9050
vagrant@ubuntu-jammy:~$ touch root/hostname
```

Con tor ya corriendo, instalamos en navegador de tor, **Tor Browser Bundle** desde su página web <https://www.torproject.org/download/>. Ejecutamos el navegador mediante el comando `./start-tor-browser`. En el navegador podremos acceder a nuestro servidor nginx poniendo lo que hay en el archivo `servicio1/hostname`:





Ahora modificamos el archivo creado en la práctica 5 para el *proxyjump* `~/.ssh/config` como se indica en el gui n:



De esta forma podremos acceder a nuestra m quina *jammy* con `ssh vagrant@hostname.onion` de la misma forma con la que estabamos accediendo a ella con `ssh vagrant@192.168.56.11`. Antes de conectarnos hemos de modificar el archivo `/etc/ssh/sshd_config` para cambiar el valor de **PasswordAuthentication** a yes:

