



# UNIVERSIDAD DE GRANADA

## **Centro de Procesamiento de Datos**

### **Práctica 9 – Mecanismos de Seguridad para Acceso a un Servidor**

---

Arturo Alonso Carbonero

## Fail2ban

En primer lugar, instalamos *Fail2ban* en nuestra máquina servidor mediante el comando **apt install fail2ban**. Esto instala tanto el cliente como el servicio. En nuestro caso, apenas usaremos el cliente, pero permite consultar de forma directa el estado de las celdas, añadir direcciones a bloquear o desbloquear, entre otros. Comprobamos, mediante **systemctl**, que el servicio se encuentra activo.

```
vagrant@nodoP9: ~  
vagrant@nodoP9:~$ systemctl status fail2ban  
● fail2ban.service - Fail2Ban Service  
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: >  
   Active: active (running) since Fri 2023-12-01 11:57:49 UTC; 7min ago  
     Docs: man:fail2ban(1)  
    Main PID: 12146 (f2b/server)  
       Tasks: 5 (limit: 1117)  
      Memory: 12.8M  
     CGroup: /system.slice/fail2ban.service  
            └─12146 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
  
alonsoarturo@DESKTOP-UKJ4G5Q:~$
```

*Servicio instalado y activo*

Por defecto, la celda para el servicio SSH ya se encuentra activa. Podemos comprobar su configuración en el fichero **/etc/fail2ban/jail.conf**.

```
[sshd]  
  
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:  
# normal (default), ddos, extra or aggressive (combines all).  
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and detail>  
#mode    = normal  
port     = ssh  
logpath  = %(sshd_log)s  
backend  = %(sshd_backend)s  
  
alonsoarturo@DESKTOP-UKJ4G5Q:~$
```

*Celda de SSH*

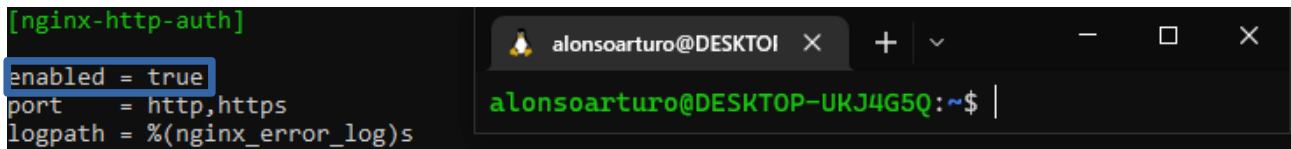
Mediante el cliente, usando el comando **fail2ban-client**, podemos comprobar el estado de dicha celda.

```
vagrant@nodoP9: ~  
vagrant@nodoP9:~$ sudo fail2ban-client status sshd  
Status for the jail: sshd  
- Filter  
  - Currently failed: 0  
  - Total failed: 0  
  - File list: /var/log/auth.log  
- Actions  
  - Currently banned: 0  
  - Total banned: 0  
  - Banned IP list:  
  
alonsoarturo@DESKTOP-UKJ4G5Q:~$
```

*Estado inicial de la celda de SSH*

Para poder realizar una supervisión del servicio *Nginx* mediante *Fail2ban*, es necesario habilitar una celda para dicho servicio. En el mismo fichero que en el ejemplo anterior, en la sección *[nginx-http-auth]*, añadimos la línea de la siguiente imagen para habilitar la celda.

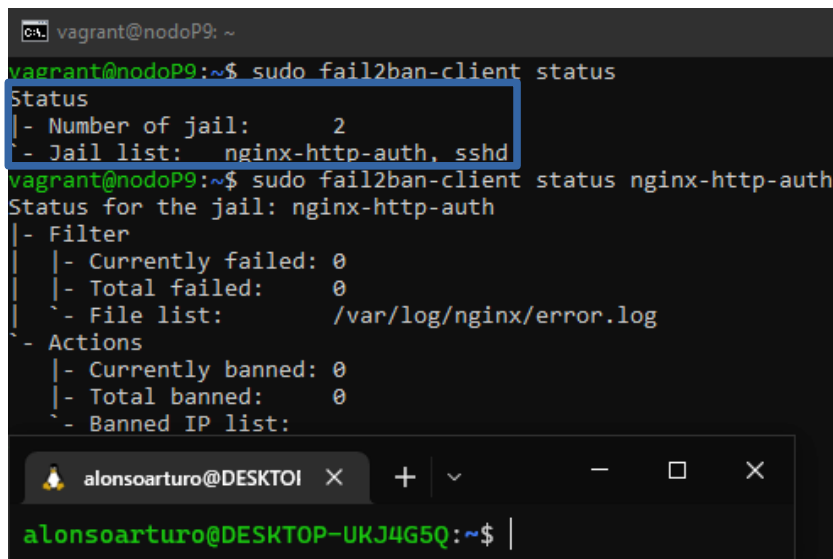
```
[nginx-http-auth]
enabled = true
port    = http,https
logpath = %(nginx_error_log)s
```



*Habilitación de celda de Nginx*

Podemos comprobar, mediante el comando **fail2ban-client status**, que la celda se ha añadido correctamente tras reiniciar mediante **systemctl**. Además, observamos su estado, tal y como en el ejemplo anterior.

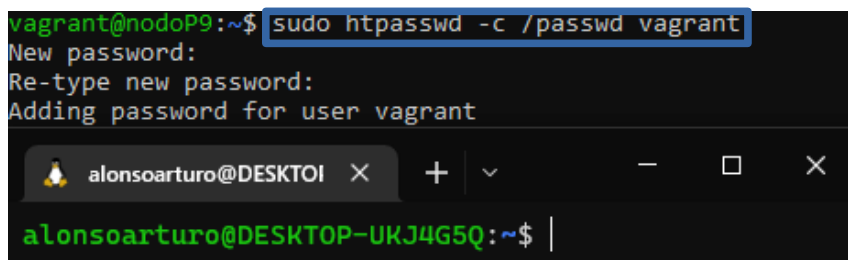
```
vagrant@nodoP9: ~
vagrant@nodoP9:~$ sudo fail2ban-client status
Status
- Number of jail:      2
- Jail list:  nginx-http-auth, sshd
vagrant@nodoP9:~$ sudo fail2ban-client status nginx-http-auth
Status for the jail: nginx-http-auth
- Filter
  - Currently failed: 0
  - Total failed:     0
  - File list:        /var/log/nginx/error.log
- Actions
  - Currently banned: 0
  - Total banned:     0
  - Banned IP list:
```



*Celda añadida y activa*

A continuación, habilitamos la autenticación mediante contraseña de *Nginx*. Para ello, hacemos uso del comando **htpasswd**, indicando el fichero para la contraseña y el usuario, tal y como se muestra en la siguiente imagen. La opción **-c** es precisamente para crear dicho fichero.

```
vagrant@nodoP9:~$ sudo htpasswd -c /passwd vagrant
New password:
Re-type new password:
Adding password for user vagrant
```



*Creación de la contraseña*

En el fichero de configuración de *Nginx*, añadimos el contenido de la siguiente imagen. Indicamos qué fichero debe emplearse para consultar la contraseña en el momento de la autenticación.

```
##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;
auth_basic "Restrined area";
auth_basic_user_file /passwd;
```

*Configuración de Nginx*

Reiniciamos el servicio, mediante **systemctl**, y comprobamos desde fuera, empleando un navegador o el comando **curl**, que es necesaria una contraseña para acceder a la página.

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl 192.168.56.18
<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

*Comprobación con curl*

Para restringir el acceso mediante *Fail2ban*, hacemos uso del cliente para bloquear una determinada IP. En este caso, bloqueamos la dirección *gateway*, ya que se trata de una máquina virtual.

```
vagrant@nodoP9:~$ sudo fail2ban-client start nginx-http-auth
Jail started
vagrant@nodoP9:~$ sudo fail2ban-client status nginx-http-auth
Status for the jail: nginx-http-auth
- Filter
| - Currently failed: 0
| - Total failed: 0
| - File list: /var/log/nginx/error.log
- Actions
| - Currently banned: 0
| - Total banned: 0
| - Banned IP list:
vagrant@nodoP9:~$ sudo fail2ban-client -vvv set nginx-http-auth banip 192.168.56.1
+ 103 7F7DE13F4740 fail2ban.configreader INFO Loading configs for fail2ban under /etc/fail2ban
+ 107 7F7DE13F4740 fail2ban.configreader DEBUG Reading configs for fail2ban under /etc/fail2ban
+ 109 7F7DE13F4740 fail2ban.configreader DEBUG Reading config files: /etc/fail2ban/fail2ban.conf
+ 111 7F7DE13F4740 fail2ban.configparserinc INFO Loading files: ['/etc/fail2ban/fail2ban.conf']
+ 111 7F7DE13F4740 fail2ban.configparserinc TRACE Reading file: /etc/fail2ban/fail2ban.conf
+ 115 7F7DE13F4740 fail2ban.configparserinc INFO Loading files: ['/etc/fail2ban/fail2ban.conf']
+ 115 7F7DE13F4740 fail2ban.configparserinc TRACE Shared file: /etc/fail2ban/fail2ban.conf
+ 115 7F7DE13F4740 fail2ban INFO Using socket file /var/run/fail2ban/fail2ban.sock
+ 115 7F7DE13F4740 fail2ban INFO Using pid file /var/run/fail2ban/fail2ban.pid, [INFO] logging to /var/log/fail2ban.log
+ 116 7F7DE13F4740 fail2ban HEAVY CMD: ['set', 'nginx-http-auth', 'banip', '192.168.56.1']
+ 296 7F7DE13F4740 fail2ban HEAVY OK : 1
+ 296 7F7DE13F4740 fail2ban.beautifier HEAVY Beautify 1 with ['set', 'nginx-http-auth', 'banip', '192.168.56.1']
+ 296 7F7DE13F4740 fail2ban DEBUG Exit with code 0
vagrant@nodoP9:~$ sudo fail2ban-client status nginx-http-auth
Status for the jail: nginx-http-auth
- Filter
| - Currently failed: 0
| - Total failed: 0
| - File list: /var/log/nginx/error.log
- Actions
| - Currently banned: 1
| - Total banned: 1
| - Banned IP list: 192.168.56.1
```

Bloqueo de IP con Fail2ban

Comprobamos, de nuevo mediante el comando **curl**, que no es posible acceder al servicio *Nginx* de la máquina desde fuera.

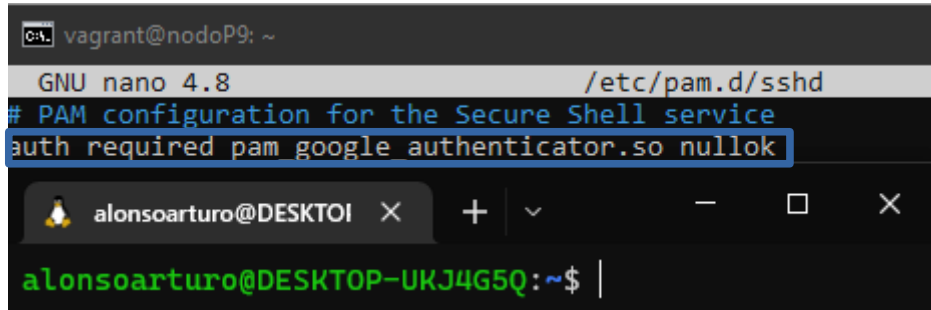
```
alonsoarturo@DESKTOP-UKJ4:~$ curl 192.168.56.18
curl: (7) Failed to connect to 192.168.56.18 port 80: Connection refused
```

Acceso bloqueado por Fail2ban

Podemos ver, al final de la imagen anterior a esta (en verde), que al comprobar el estado de la celda, se ha realizado un total de un bloqueo de acceso por parte de *Fail2ban*, mientras que al principio de la misma, el número de bloqueos era de 0.

## Google Authenticator

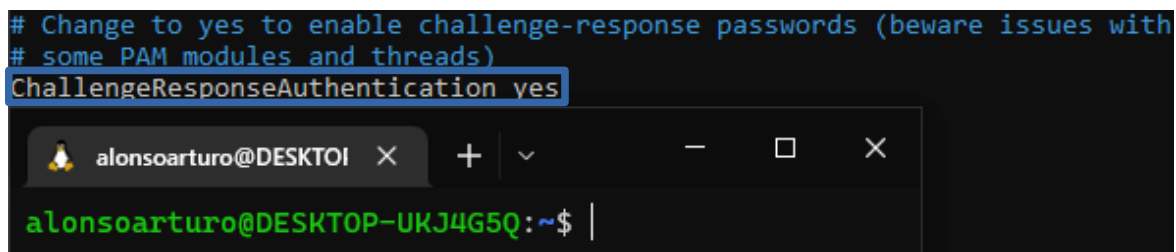
Para poder hacer uso de los servicios de *Google Authenticator*, debemos instalarlo en nuestra máquina haciendo uso del comando **apt**. A continuación, debemos activar el módulo en el servicio SSH. Para ello, añadimos la línea de la siguiente imagen al fichero **/etc/pam.d/sshd**.



```
vagrant@nodoP9: ~  
GNU nano 4.8 /etc/pam.d/sshd  
# PAM configuration for the Secure Shell service  
auth required pam google authenticator.so nullok  
alonsoarturo@DESKTOI  
alonsoarturo@DESKTOP-UKJ4G5Q:~$
```

*Activación del módulo*

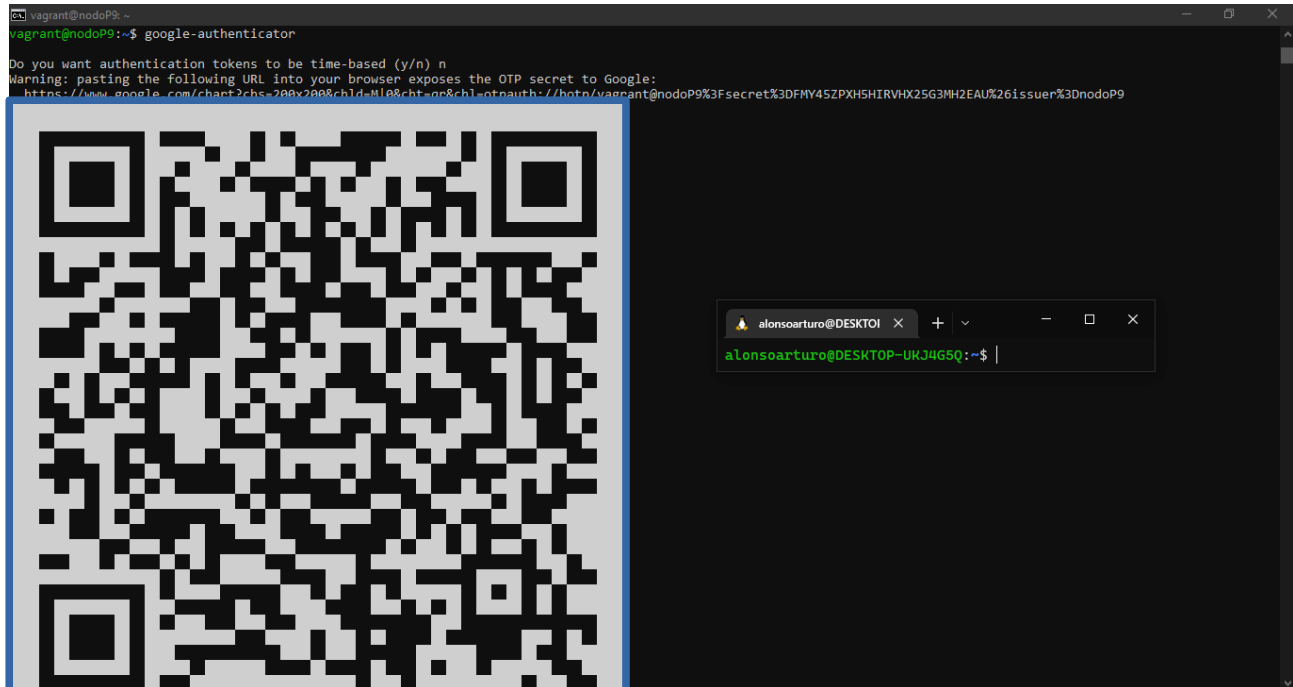
En el fichero de configuración de SSH, **/etc/ssh/sshd\_config**, deshabilitamos el acceso con contraseña y activamos, poniendo a 'yes', la línea de la siguiente imagen. Esto hace que, al intentar conectarnos a través de SSH a la máquina, el método de autenticación sea con un reto, que se corresponderá en nuestro caso con el código de verificación que hayamos generado. Además, reiniciamos el servicio SSH mediante **systemctl**.



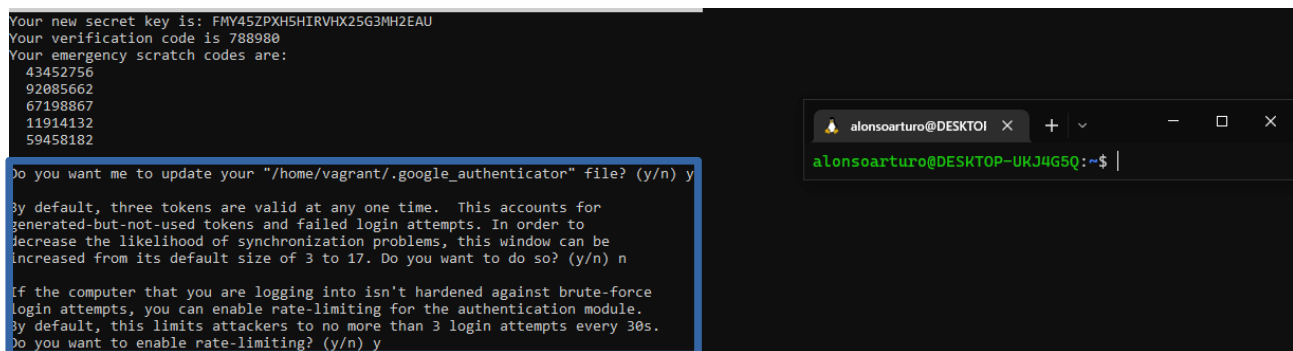
```
# Change to yes to enable challenge-response passwords (beware issues with  
# some PAM modules and threads)  
ChallengeResponseAuthentication yes  
alonsoarturo@DESKTOI  
alonsoarturo@DESKTOP-UKJ4G5Q:~$
```

*Autenticación con reto*

Una vez hemos configurado el entorno, ejecutamos el comando **google-authenticator**. Respondemos a una serie de preguntas en función de nuestro interés, para configurar opciones de seguridad, y obtendremos un código QR que, tras ser escaneado en la aplicación móvil de *Google Authenticator*, nos proporcionará el código de verificación para el acceso. Este código será volátil o no en función de la configuración que hayamos indicado. Para generar el código QR, hay que instalar previamente el paquete **qrencode** mediante **apt**.



Código QR



Configuración del código

Tras esto, podemos acceder mediante SSH desde fuera, proporcionando, además de las credenciales pertinentes, el código que hemos obtenido de la aplicación.

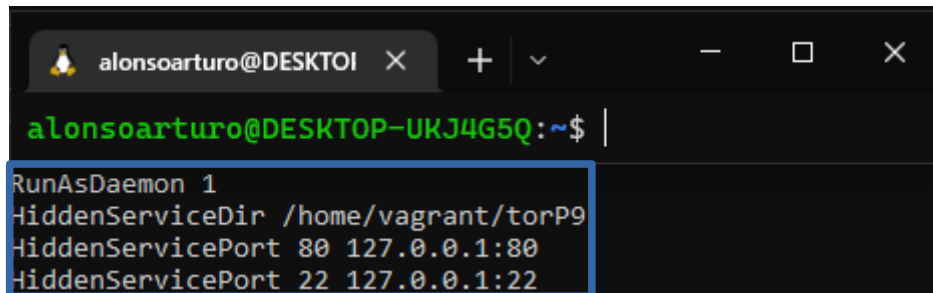
```
vagrant@nodoP9: ~  
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ssh vagrant@192.168.56.18  
Verification code:  
Password:  
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-164-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue Dec  5 11:33:20 UTC 2023  
  
System load:  0.11      Processes:            113  
Usage of /:   4.8% of 38.70GB   Users logged in:     1  
Memory usage: 29%      IPv4 address for enp0s3: 10.0.2.15  
Swap usage:   0%         IPv4 address for enp0s8: 192.168.56.18  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
  just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
1 additional security update can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Tue Dec  5 11:31:21 2023 from 192.168.56.1  
vagrant@nodoP9:~$ |
```

*Acceso con código de verificación*



## Tor

En primer lugar, instalamos Tor mediante **apt**. Posteriormente, editamos el fichero de configuración **/etc/tor/torrc** y añadimos las líneas de la siguiente imagen.



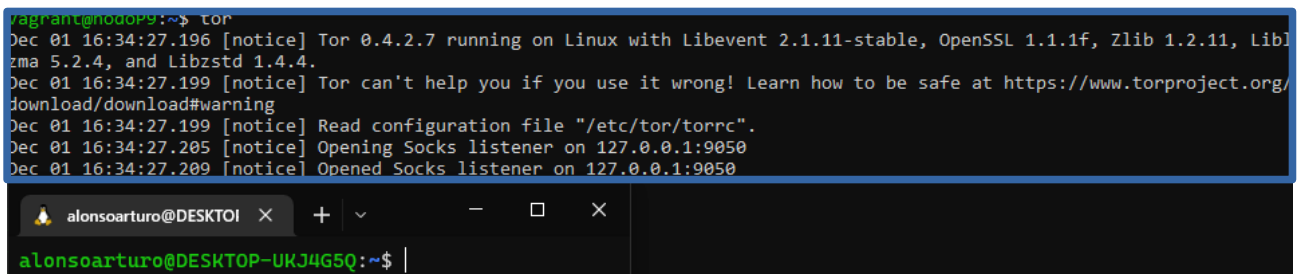
```
alonsoarturo@DESKTOI x + - □ x
alonsoarturo@DESKTOP-UKJ4G5Q:~$ |
RunAsDaemon 1
HiddenServiceDir /home/vagrant/torP9
HiddenServicePort 80 127.0.0.1:80
HiddenServicePort 22 127.0.0.1:22
```

*Configuración para Tor*

Con esto, estamos indicando que Tor se ejecute como demonio, esto es, en segundo plano. Especificamos también el directorio que contendrá los ficheros del *hostname* y la clave privada para nuestra red. Además, indicamos qué dirección, en este caso el *localhost* (*loopback*), y qué puertos debe emplear. Reiniciamos el servicio mediante **systemctl**.

Es necesario que el directorio que hemos especificado cuente con ciertos permisos concretos. Para ello, ejecutamos la orden **chmod go-rwx torP9**. Con ello, los permisos de escritura, lectura y ejecución, de los grupos distintos al del propietario, serán eliminados.

Para lanzar Tor y poder acceder a nuestra máquina desde una red Tor, ejecutamos la orden **tor**.

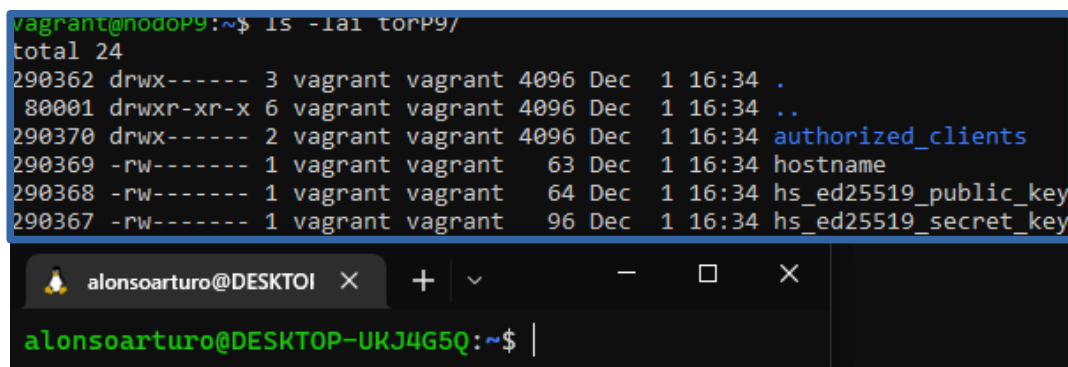


```
vagrant@nodoP9:~$ tor
Dec 01 16:34:27.196 [notice] Tor 0.4.2.7 running on Linux with Libevent 2.1.11-stable, OpenSSL 1.1.1f, Zlib 1.2.11, Libl
zma 5.2.4, and Libstd 1.4.4.
Dec 01 16:34:27.199 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://www.torproject.org/
download/download#warning
Dec 01 16:34:27.199 [notice] Read configuration file "/etc/tor/torrc".
Dec 01 16:34:27.205 [notice] Opening Socks listener on 127.0.0.1:9050
Dec 01 16:34:27.209 [notice] Opened Socks listener on 127.0.0.1:9050

alonsoarturo@DESKTOI x + - □ x
alonsoarturo@DESKTOP-UKJ4G5Q:~$ |
```

*Lanzar tor*

Si el proceso ha sido satisfactorio, dentro del directorio creado habrán aparecido una serie de archivos.



```
vagrant@nodoP9:~$ ls -la1 torP9/
total 24
290362 drwx----- 3 vagrant vagrant 4096 Dec  1 16:34 .
80001 drwxr-xr-x 6 vagrant vagrant 4096 Dec  1 16:34 ..
290370 drwx----- 2 vagrant vagrant 4096 Dec  1 16:34 authorized_clients
290369 -rw----- 1 vagrant vagrant  63 Dec  1 16:34 hostname
290368 -rw----- 1 vagrant vagrant  64 Dec  1 16:34 hs_ed25519_public_key
290367 -rw----- 1 vagrant vagrant  96 Dec  1 16:34 hs_ed25519_secret_key

alonsoarturo@DESKTOI x + - □ x
alonsoarturo@DESKTOP-UKJ4G5Q:~$ |
```

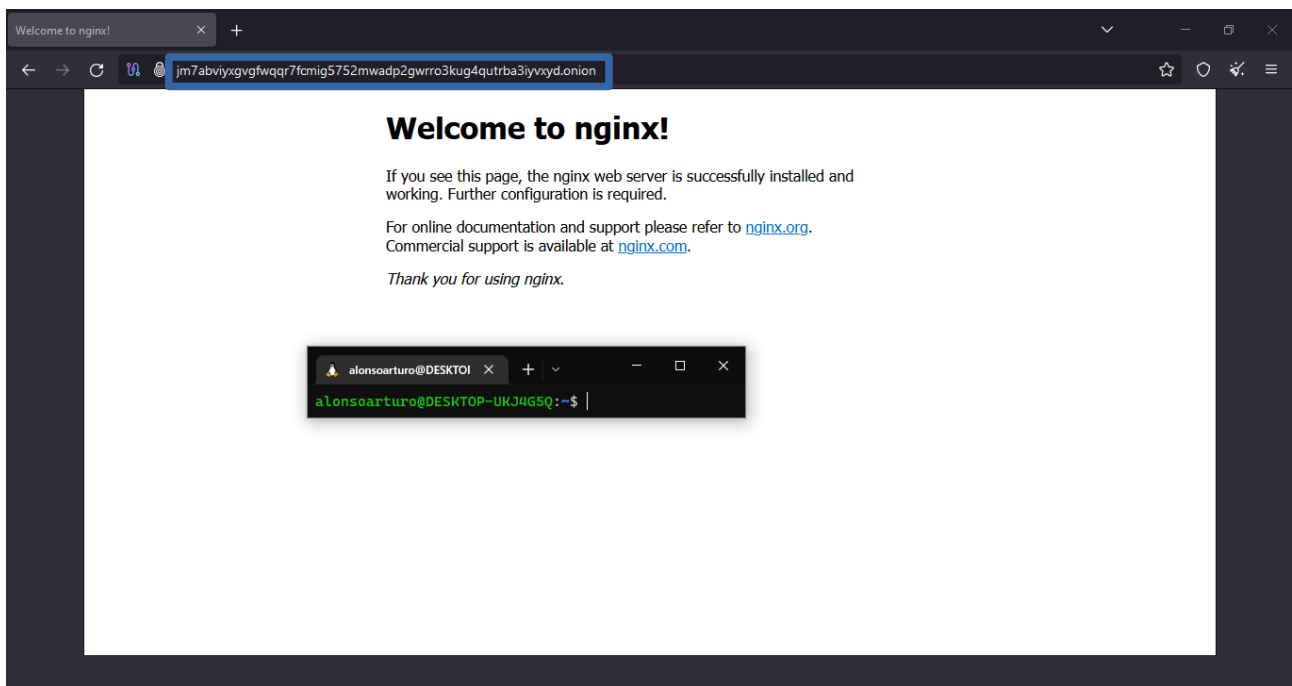
*Ficheros creados*

Entre ellos, el archivo con nombre “*hostname*” contiene la dirección ‘.onion’ que nos permitirá acceder a nuestra máquina usando una red Tor.

```
vagrant@nodoP9: ~  
vagrant@nodoP9:~$ cat torP9/hostname  
jm7abviyvgvgfwqqr7fcmig5752mwadp2gwrro3kug4qutrba3iyvxyd.onion  
alonsoarturo@DESKTOI x + v - □ x  
alonsoarturo@DESKTOP-UKJ4G5Q:~$ |
```

*Hostname*

Ahora, desde el cliente, es decir, el *host* anfitrión en mi caso, accedemos a dicha dirección desde un navegador Tor, permitiendo previamente la conexión a la red Tor.



*Acceso a la máquina desde Tor*

## Referencias

### Fail2ban

- <https://github.com/ArturoAcf/Servidores-Web-de-Altas-Prestaciones/blob/main/AlonsoCarboneroArturoP4.pdf>
- <https://github.com/ArturoAcf/Igenieria-De-Servidores/blob/main/Práctica 2 - Pila Lamp%2C Git y Servicios.pdf>
- <https://serverspace.io/es/support/help/install-configure-fail2ban-ubuntu-20-04/>
- <https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/>
- <https://kimerikal.com/bloquear-desbloquear-ips-manualmente-con-fail2ban/>
- <https://httpd.apache.org/docs/2.4/programs/htpasswd.html>

### Google Authenticator

- <https://www.entredvyops.es/posts/ssh-2fa.html>
- <https://www.techtarget.com/searchsecurity/definition/challenge-response-system>

### Tor

- <https://www.torproject.org/es/>