Parcial 2 Parte 1 - DNS’s

Vagrantfile

```
Vagrant.configure("2") do |config|
  config.vm.define :maestro do |maestro|
    maestro.vm.box = "bento/ubuntu-22.04"
    maestro.vm.network :private_network, ip: "192.168.50.3"
    maestro.vm.hostname = "maestro"
  end

  config.vm.define :secundario do |secundario|
    secundario.vm.box = "bento/ubuntu-22.04"
    secundario.vm.network :private_network, ip: "192.168.50.2"
    secundario.vm.hostname = "secundario"
  end

  config.vm.define :firewall do |firewall|
    firewall.vm.box = "bento/ubuntu-22.04"
    firewall.vm.network :private_network, ip: "192.168.50.4"
    firewall.vm.hostname = "firewall"
  end
end
```

Paquetes y librerías

Instalar bind en `maestro` y `secundario`

En maestro y secundario:

```
sudo apt update
sudo apt install bind9 bind9utils bind9-doc -y
```

Crear clave TSIG para transferencia de zona

En `maestro`:

```
cd /etc/bind
sudo rndc-confgen -a -k clave -c /etc/bind/tsig.key
```

Este comando genera una clave TSIG y la guarda en `/etc/bind/tsig.key`. Vamos a usar esa clave para la transferencia de zonas.

Para poder visualizar esta clave se ejectuta:

```
sudo cat /etc/bind/tsig.key
```

Dando como resultado:

```
key "key" {
        algorithm hmac-sha256;
        secret "CLAVE";
};
```

En `secundario`:

```
sudo nano /etc/bind/tsig.key
```

Se pega el contenido del archivo `tsig.key` que teniamos en `maestro`

Configuración del DNS Maestro

Editar `/etc/bind/named.conf.local` en `maestro`:

```
sudo nano /etc/bind/named.conf.local
```

Pegar en el archivo de zonas:

```
key "key" {
   algorithm hmac-sha256;
   secret "CLAVE GENERADA";
};

zone "ejemplo.com" {
   type master;
   file "/etc/bind/db.key.com";
   allow-transfer { key key; };
};
```

Crear archivo de zona en `/etc/bind/db.ejemplo.com`:

```
$TTL    604800
@       IN      SOA     ns1.ejemplo.com. admin.ejemplo.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.ejemplo.com.
ns1     IN      A       192.168.50.3
```

Verificar la configuración:

```
sudo named-checkzone ejemplo.com /etc/bind/db.ejemplo.com
```

Reiniciar BIND:

```
sudo systemctl restart bind9
```

Configuración del DNS Secundario

En `secundario`, editar `/etc/bind/named.conf.local`:

```
key "key" {
    algorithm hmac-sha256;
    secret "AQUI VA LA CLAVE SECRETA QUE SE GENERA";
};

zone "ejemplo.com" {
    type slave;
    masters { 192.168.50.3 key key; };
    file "/var/cache/bind/db.ejemplo.com";
};
```

Reiniciar BIND:

```
sudo systemctl restart bind9
```

Verificación de la transferencia de zona

En `secundario`:

Revisar si `db.ejemplo.com` existe en la carpeta:

```
ls -l /var/cache/bind/
```

Configuración del Firewall (`firewall, maestro, secundario`)

En `firewall`

Habilitar UFW y permitir conexiones SSH:

```
sudo ufw allow ssh
```

Permitir las salientes desde el firewall hacia el DNS secundario para transferencia de zona:

```
sudo ufw allow out to 192.168.50.2 port 53 proto udp
sudo ufw allow out to 192.168.50.2 port 53 proto tcp
```

Agregar reglas desde cualquier VM al puerto DNS:

```
sudo ufw allow proto udp to any port 53
sudo ufw allow proto tcp to any port 5
```

Permitir trafico SOLO en la red interna

```
sudo ufw allow from 192.168.50.0/24 to any port 53 proto udp
sudo ufw allow from 192.168.50.0/24 to any port 53 proto tcp
```

En `/etc/ufw/before.rules` poner

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

# Redirirección de solicitudes al firewall
-A PREROUTING -p udp --dport 53 -j DNAT --to-destination 192.168.50.2:53
-A PREROUTING -p tcp --dport 53 -j DNAT --to-destination 192.168.50.2:53

# Masquerade para las respuestas de las solicitudes
-A POSTROUTING -p udp --dport 53 -d 192.168.50.2 -j MASQUERADE
-A POSTROUTING -p tcp --dport 53 -d 192.168.50.2 -j MASQUERADE

COMMIT
```

Descomentar la línea `net.ipv4.ip_forward=1` en el archivo `/etc/sysctl.conf` y aplicar los cambios con `sudo sysctl -p`

Reiniciamos el UFW:

```
sudo ufw enable
sudo ufw status
```

En la VM `secundario`

Permitir solo tráfico desde el `firewall` y desde el `maestro`:

De esta manera solo se podrá interactuar entre `firewall` y `maestro`.

```
sudo ufw allow from 192.168.50.4 to any port 53 proto udp
sudo ufw allow from 192.168.50.4 to any port 53 proto tcp
sudo ufw allow from 192.168.50.3 to any port 53 proto tcp
```

Reiniciar UFW:

```
sudo ufw enable
sudo ufw status verbose
```

En `maestro`

Solo permitir el `secundario` para transferencia de zona:

Asi, solo secundario puede solicitar la transferencia de zona por TSIG.

```
sudo ufw allow from 192.168.50.2 to any port 53 proto tcp
```

Activar UFW:

```
sudo ufw enable
sudo ufw status verbose
```

Instalar `dnsmasq`

Es necesario instalar `dnsmasq` para configurar el `firewall` como puerta de enlace y así enrutar todo el tráfico a través de ella.

En `firewall`

Detener y desactivar el `resolver` por defecto

```
sudo systemctl stop systemd-resolved
sudo systemctl disable systemd-resolved
```

Modificar `resolv.conf` para apuntar al `secundario`

```
 nameserver 192.168.50.2
```

Instalar `dnsmasq` :

```
sudo apt update
sudo apt install dnsmasq
```

En `/etc/dnsmasq.conf`

```
listen-address=192.168.50.4
bind-interfaces
no-resolv
server=192.168.50.2
log-queries
log-facility=/var/log/dnsmasq.log
```

Reiniciar `dnsmasq` :

```
sudo systemctl restart dnsmasq
sudo systemctl enable dnsmasq
```

Pruebas

Desde `firewall`

Al ejecutar `dig @192.168.50.4 ejemplo.com`, el nombre de dominio debería resolverse correctamente.

Al ejecutar `dig @192.168.50.3 ejemplo.com`, la conexión debería ser rechazada o devolver un error de **timeout**.

Al ejecutar `dig @192.168.50.3 ejemplo.com AXFR -y hmac-sha256:key:CLAVE`, también debería rechazar la conexión o mostrar un error de **timeout**. (bloqueos firewall)

Desde `secundario`

Al ejecutar `dig @192.168.50.3 prueba.com AXFR -y hmac-sha256:key:CLAVE`, se deberían obtener **todos los registros del dominio**.

🔒 ¿Por qué el comando `AXFR` devuelve toda la información y el `dig` común no?

Porque:

- `AXFR` realiza una **transferencia completa de zona**, la cual solo es posible si se usa TSIG o una IP autorizada.
- Un `dig` estándar hace una **consulta específica**, y si no existe un registro A para el dominio raíz (`prueba.com`), no se mostrará nada en la sección `ANSWER`.

Parte 2 FTP Seguro

VagrantFile

```
Vagrant.configure("2") do |config|
  config.vm.define :servidor1 do |servidor1|
    servidor1.vm.box = "bento/ubuntu-22.04"
    servidor1.vm.network :private_network, ip: "192.168.50.3"
    servidor1.vm.hostname = "servidor1"
```

```
  end

  config.vm.define :servidor2 do |servidor2|
    servidor2.vm.box = "bento/ubuntu-22.04"
    servidor2.vm.network :private_network, ip: "192.168.50.2"
    servidor2.vm.hostname = "servidor2"
  end
end
```

En `servidor1`

Actualizar e instalar librerías

```
sudo apt update
sudo apt install -y ufw iptables-persistent
```

Descomentar la línea `net.ipv4.ip_forward=1` en el archivo `/etc/sysctl.conf` y aplicar los cambios con `sudo sysctl -p`

Aplicar los cambios

`sudo sysctl -p`

Reglas NAT

En el archivo `/etc/ufw/before.rules`

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

-A PREROUTING -i eth1 -p tcp --dport 21 -j DNAT --to-destination 192.168.50.2:2
-A PREROUTING -i eth1 -p tcp --dport 990 -j DNAT --to-destination 192.168.50.2
-A PREROUTING -i eth1 -p tcp --dport 40000:50000 -j DNAT --to-destination 192
```

```
-A POSTROUTING -s 192.168.50.0/24 -o eth1 -j MASQUERADE

COMMIT
```

Configuración de UFW

```
sudo ufw allow ssh
sudo ufw allow 21/tcp              # FTP
sudo ufw allow 990/tcp             # FTP seguro
sudo ufw allow 40000:50000/tcp     # Modo Pasivo
```

Reenviar paquetes entre las interfaces de red

En el archivo `/etc/default/ufw` modificar esta linea a:

```
DEFAULT_FORWARD_POLICY="ACCEPT”
```

Persistir las reglas de IPTABLES

```
sudo netfilter-persistent save
sudo netfilter-persistent reload
```

Reiniciar UFW

```
sudo ufw disable
sudo ufw enable
```

Ver si ufw está activo

```
sudo ufw status
```

En servidor2

Instalar vsftpd en servidor2

```
sudo apt update && sudo apt install -y vsftpd openssl
```

Configurar vsftpd

En el archivo /etc/vsftpd.conf

```
#Opciones basicas
listen=YES
listen_ipv6=NO
listen_address=192.168.50.2
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES

chroot_local_user=YES
allow_writeable_chroot=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd

#configuracion SSL
ssl_enable=YES
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
```

```
force_local_data_ssl=YES
force_local_logins_ssl=YES


#CONFGURACION DE MODO PASIVO
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=50000
pasv_address=192.168.50.3  # Firewall
pasv_addr_resolve=NO
pasv_promiscuous=YES
```

Configuración de iptables para bloquear conexiones FTP directas desde cualquier origen, excepto desde el firewall.

```
sudo iptables -A INPUT -p tcp --dport 990 ! -s 192.168.50.3 -j DROP
sudo iptables -A INPUT -p tcp --dport 21 ! -s 192.168.50.3 -j DROP
sudo iptables -A INPUT -p tcp --dport 40000:50000 ! -s 192.168.50.3 -j DROP
```

Persistir estas reglas de iptables

```
sudo apt install -y iptables-persistent
sudo netfilter-persistent save
sudo netfilter-persistent reload
```

Generación el certificado SSL del servidor FTP

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/pr
```

Reiniciar y activar `vsftpd`

```
sudo systemctl restart vsftpd
sudo systemctl enable vsftpd
```

Pruebas

Si intentamos conectarnos a 192.168.50.2 con las credenciales desde FIleZilla, la conexión abortará por timeout, siendo efectivo el bloqueo del FTP directo.

Pero si conectamos al firewall directamente por FTP seguro con las siguientes credenciales:

- **IP:** 192.168.50.3
- **Usuario:** vagrant
- **Contraseña:** vagrant
- **Puerto:** 21

Lograremos iniciar sesión.

Parte 3 UFW avanzado

```
Vagrant.configure("2") do |config|
  config.vm.define :servidor do |servidor|
  servidor.vm.box = "bento/ubuntu-22.04"
  servidor.vm.network :private_network, ip: "192.168.90.3"
  servidor.vm.hostname = "servidor"
  servidor.vm.provider "virtualbox" do |vb|
     vb.memory = "1000"  # 500 MB de RAM
     vb.cpus = 1        # 1 núcleo de CPU
   end
  end
  config.vm.define :cliente do |cliente|
  cliente.vm.box = "bento/ubuntu-22.04"
  cliente.vm.network :private_network, ip: "192.168.90.2"
  cliente.vm.hostname = "cliente"
```

```
    cliente.vm.provider "virtualbox" do |vb|
      vb.memory = "1000"  # 1000 MB de RAM
      vb.cpus = 1       # 1 núcleo de CPU
    end
  end
  config.vm.define :atacante do |atacante|
  atacante.vm.box = "bento/ubuntu-22.04"
  atacante.vm.network :private_network, ip: "192.168.50.2"
  atacante.vm.hostname = "atacante"
  atacante.vm.provider "virtualbox" do |vb|
      vb.memory = "1000"  # 1000 MB de RAM
      vb.cpus = 1       # 1 núcleo de CPU
    end
  end
end
```

En `servidor`

Servicio FTP

Instalar vsftpd en `servidor`

```
sudo apt update && sudo apt install -y vsftpd
```

Configuración de vsftpd

En el archivo `/etc/vsftpd.conf`

```
listen=YES
listen_ipv6=NO
anonymous_enable=NO
local_enable=YES
write_enable=YES
```

```
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
ftpd_banner=Bienvenido usuario.
chroot_local_user=YES
allow_writeable_chroot=NO

secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
chroot_local_user=YES

ssl_enable=NO
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key

max_per_ip=5
Crear archivo vsftpd.log usando:
sudo touch /var/log/vsftpd.log
sudo chmod 644 /var/log/vsftpd.log
sudo chown root:root /var/log/vsftpd.log
```

Crear el usuario

```
sudo adduser usuario
```

Directorio FTP

```
sudo mkdir -p /home/usuario/ftp
sudo chown nobody:nogroup /home/usuario/ftp
sudo chmod a-w /home/usuario/ftp
sudo mkdir -p /home/usuario/ftp/subida
sudo chown usuario:usuario/home/telematicos/ftp/subida
```

Configurar fail2ban para bloqueo de IP’s

Instalación

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

En `/etc/fail2ban/filter.d/vsftpd.conf`

```
[Definition]
failregex = . FAIL LOGIN: Client "<HOST>"

ignoreregex =
```

Reemplazar el contenido del archivo en `/etc/fail2ban/jail.local` por lo siguiente para manejar el **bloqueo de ip’s** y el **numero máximo de intentos permitidos**.

```
[vsftpd]
enabled = true
port = 21
filter = vsftpd
maxretry = 5
findtime = 120
bantime = 180
```

Reiniciar vsftpd y fail2ban

```
sudo systemctl restart fail2ban
sudo systemctl restart vsftpd
```

HTTP/HTTPS

Instalar apache2 en `servidor` y activar `rewrite`

```
sudo apt update y sudo apt install -y apache2
sudo a2enmod rewrite
```

Crear una página en `/var/www/html/index.html`

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba</title>
  </head>
    <body>
    <h1>Prueba</h1>
  </body>
</html>
```

Activar SSL, crear certificado y modificar permisos

```
sudo a2enmod ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/ssl/private/certificate.key -out /etc/ssl/certs/certificate.crt
sudo chmod 600 /etc/ssl/private/certificate.key
sudo chmod 644 /etc/ssl/certs/certificate.crt
```

Crear archivo de sitio en `/etc/apache2/sites-available/prueba.com.conf`

```
# Configuración del VirtualHost para HTTP (puerto 80)
<VirtualHost *:80>
  ServerName www.prueba.com
  DocumentRoot /var/www/html

  <Directory "/var/www/html">
```

```
        Require all granted

        # Activar el motor de reescritura
        RewriteEngine On

        # ==== REGLAS COMENTADAS DE HORARIO LABORAL ====
        # Estas reglas implementan el control de acceso solo de lunes a viernes,
        # entre 08:00 y 18:00 hora de Colombia (UTC-5).
        # Se comentan para efectos demostrativos de la práctica.

        # RewriteCond %{TIME_HOUR} ^(0[0-7]|1[8-9]|2[0-3])$ [OR]  # Fuera de 08
        # RewriteCond %{TIME_WDAY} ^(0|6)$                        # Domingo (0) o sába
        # RewriteRule ^ - [F]

        # ==== REGLAS ACTIVAS PARA BLOQUEO ENTRE 6:30pm - 8:30pm (hora C
        # Por efectos de la práctica, se bloquea únicamente entre 6:30pm y 8:30pm

        # Bloqueo entre 23:30 y 23:59 UTC
        RewriteCond %{TIME} ^23[3-5][0-9] [OR]

        # Bloqueo entre 00:00 y 00:59 UTC
        RewriteCond %{TIME} ^00[0-5][0-9] [OR]

        # Bloqueo entre 01:00 y 01:29 UTC
        RewriteCond %{TIME} ^01[0-2][0-9]
        RewriteRule ^ - [F]
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

# Configuración del VirtualHost para HTTPS (puerto 443)
<VirtualHost *:443>
    ServerName www.prueba.com
    DocumentRoot /var/www/html
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# Habilitación de SSL
SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache.crt
SSLCertificateKeyFile /etc/ssl/private/apache.key

<Directory "/var/www/html">
    Require all granted

    # Activar el motor de reescritura
    RewriteEngine On

    # ==== REGLAS COMENTADAS DE HORARIO LABORAL ====
    # Estas reglas implementan el control de acceso solo de lunes a viernes,
    # entre 08:00 y 18:00 hora de Colombia (UTC-5).
    # Se comentan para efectos demostrativos de la práctica.

    # RewriteCond %{TIME_HOUR} ^(0[0-7]|1[8-9]|2[0-3])$ [OR]
    # RewriteCond %{TIME_WDAY} ^(0|6)$
    # RewriteRule ^ - [F]

    # ==== REGLAS ACTIVAS PARA BLOQUEO ENTRE 6:30pm - 8:30pm (hora C

    # Bloqueo entre 23:30 y 23:59 UTC
    RewriteCond %{TIME} ^23[3-5][0-9] [OR]

    # Bloqueo entre 00:00 y 00:59 UTC
    RewriteCond %{TIME} ^00[0-5][0-9] [OR]

    # Bloqueo entre 01:00 y 01:29 UTC
    RewriteCond %{TIME} ^01[0-2][0-9]
    RewriteRule ^ - [F]
</Directory>
```

```
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>

    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>
</VirtualHost>
```

Habilitar sitio prueba.com

```
sudo a2ensite prueba.com.conf
sudo systemstl reload apache2
```

Configuración del DNS

En `servidor`

```
sudo apt update && sudo apt install -y bind9 && sudo apt install -y dnsutils
```

Archivo de zonas en `/etc/bind/named.conf.local`

```
zone "prueba.com" {
    type master;
    file "/etc/bind/db.prueba.com";
};`

zone "90.168.192.in-addr.arpa" {
    type master;
```

```
    file "/etc/bind/db.90.168.192";
};
```

Archivo de zona directa en /etc/bind/db.prueba.com

```
;
; BIND data file for local loopback interface

;
$TTL    604800
@       IN      SOA     prueba.com. root.prueba.com. (

2         ; Serial
604800         ; Refresh
86400         ; Retry
2419200         ; Expire
604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.prueba.com.
ns      IN      A       192.168.90.3
www     IN      CNAME   ns
```

Archivo de zona inversa en `/etc/bind/` `db.90.168.192`

```
$TTL    604800
@       IN      SOA     ns.prueba.com. root.prueba.com. (
1         ; Serial
604800         ; Refresh
86400         ; Retry
2419200         ; Expire
604800 )       ; Negative Cache TTL
;
```

```
@      IN     NS     ns.
3      IN     PTR    ns.prueba.com.
```

Verificación de sintaxis

```
sudo named-checkzone telematica.com /etc/bind/db.prueba.com
sudo named-checkzone 90.168.192.in-addr.arpa /etc/bind/db.192
```

Rate limit en el archivo `/etc/bind/named.conf.options`

```
options {
    // Directorio donde se almacenan las cachés del servidor DNS
    directory "/var/cache/bind";

    // Validación automática de DNSSEC (protección contra suplantación de respu
    dnssec-validation auto;

    // Escuchar en todas las interfaces IPv6 disponibles
    listen-on-v6 { any; };

    // 🔒 Configuración básica de limitación de tasa (rate-limit) para mitigar ataque
    rate-limit {
        responses-per-second 10;   // Número máximo de respuestas por segundo p
        window 5;                  // Ventana de tiempo (en segundos) para evaluar el lím
        slip 2;                  // Por cada 2 respuestas que se rechazan, 1 se envía com
    };
};
```

UFW avanzado

Reglas granulares de bloqueo y acceso:

```
# 🔒 Bloquea todo el tráfico entrante desde la subred 192.168.50.0/24 (por ejemp
sudo ufw deny from 192.168.50.0/24

# 🔒 Bloquea una dirección IP específica (por ejemplo, una máquina potencialme
sudo ufw deny from 192.168.90.1

# ✅ Permite el tráfico entrante desde la subred 192.168.90.0/24 (por ejemplo, ur
sudo ufw allow from 192.168.90.0/24
```

Reglas para los servicios (DNS, HTTP/HTTPS, SSH y FTP):

```
# 🎯 Permitir conexiones FTP estándar y seguras desde la subred 192.168.90.0/2

# Puerto 21: Canal de control FTP (comandos)
sudo ufw allow from 192.168.90.0/24 to any port 21 proto tcp

# Puerto 20: Canal de datos FTP (transferencia activa)
sudo ufw allow from 192.168.90.0/24 to any port 20 proto tcp

# Puerto 990: FTPS implícito (FTP sobre TLS/SSL)
sudo ufw allow from 192.168.90.0/24 to any port 990 proto tcp


# 🌐 Habilitar tráfico web (HTTP/HTTPS) solo desde una IP específica

# Puerto 80: HTTP (no cifrado)
sudo ufw allow from 192.168.90.2 to any port 80 proto tcp

# Puerto 443: HTTPS (cifrado SSL/TLS)
sudo ufw allow from 192.168.90.2 to any port 443 proto tcp


# 🧭 Habilitar servicio DNS (consultas y transferencias) desde una IP específica
```

```
# Puerto 53 en TCP: Transferencias de zona y consultas largas
sudo ufw allow from 192.168.90.2 to any port 53 proto tcp

# Puerto 53 en UDP: Consultas DNS normales
sudo ufw allow from 192.168.90.2 to any port 53 proto udp


# 🔐 Permitir acceso remoto seguro por SSH

# Puerto 22: SSH (conexión remota segura)
sudo ufw allow 22/tcp
```

Activar y verificar reglas:

```
sudo ufw enable
# Verificar las reglas
sudo ufw status verbose
```

Monitoria y auditoria

Para registrar intentos de acceso bloqueado, use el siguiente comando:

```
sudo ufw logging on
sudo ufw logging high
```

Para filtrar los paquetes usar:

```
sudo grep "192.168.90.2" /var/log/ufw.log
```

Pruebas

Pruebas del firewall (UFW)

```
# FTP: Desde cliente
ftp 192.168.90.3

# Se espera el login exitoso
```

```
# FTP: Desde el intruso
ftp 192.168.90.3

# Se espera una denegación de conexión
```

```
# HTTP: Desde el cliente
curl 192.168.90.3

# Si se está dentro del horario permitido se espera ver la página,
# sino se espera ver un 401 Forbidden.
```

```
# HTTP: Desde el intruso
curl 192.168.90.3

# Se espera un rechazo de la conexión
```

```
# DNS: Desde el cliente
dig @192.168.90.3 prueba.com

# Se espera la resolución correcta del dominio
```

```
# DNS: Desde el intruso
dig @192.168.90.3 prueba.com
```

```
# Se espera el rechazo de la conexión
```

Pruebas contra ataques

Prueba de bloqueo de IP por intentos fallidos de login FTP

En la máquina cliente, ejecuta el siguiente comando para simular múltiples intentos de acceso con credenciales incorrectas:

```
for i in {1..6}; do echo -e "prueba\asdasda" | ftp 192.168.90.3; done
```

Respuesta esperada:

Después de varios intentos fallidos, la IP del cliente debería ser bloqueada y el siguiente mensaje debería aparecer al intentar conectarse nuevamente:

```
ftp: Can't connect to `192.168.90.3:21': Connection refused
```

Prueba de límite de conexiones simultáneas

Ejecuta el siguiente comando para simular múltiples conexiones simultáneas al servidor FTP:

```
for i in {1..6}; do ftp -n 192.168.90.3 & done
```

Respuesta esperada:

El servidor debería rechazar el exceso de conexiones con el mensaje:

```
421 There are too many connections from your internet address
```

Pruebas de Protección Contra Ataques (Apache y horarios)

Para comprobar que Apache responde solo en los horarios permitidos, ejecuta desde el cliente:

```
curl 192.168.90.3
```

> Asegúrate de que los relojes de las máquinas estén sincronizados con la hora internacional (NTP).

Comportamiento esperado:

- Si se accede en horario **permitido** ➜ se mostrará el contenido del `index.html`.
- Si se accede **fuera del horario permitido** o en fin de semana ➜ se mostrará un error:

```
HTTP/1.1 403 Forbidden
```

Prueba de protección DNS (rate-limit)

Ejecuta desde el cliente una prueba de múltiples consultas DNS para verificar la protección contra saturación:

```
for i in {1..200}; do dig @192.168.90.3 prueba.com +short & done
```

Resultado esperado:

El servidor debería dejar de responder debido a la limitación de tasa, mostrando mensajes como:

```
communications error to 192.168.90.3#53: timed out
```

Y en algunos casos:

```
[198]  Exit 9
```