Institut Puig Castellar
Santa Coloma de Gramenet

JPA Cybersecurity Coop.

Projecte de desenvolupament i investigació
SMX2A

Resum del projecte:

Somos una empresa de ciberseguridad que ayuda a otras empresas a proteger sus sistemas y datos en línea. Trabajamos con ellos para identificar y solucionar posibles problemas de seguridad, brindándoles asesoramiento, instalando medidas de protección y ofreciendo capacitación para que puedan mantenerse seguros en un mundo digital cada vez más peligroso.

Paraules clau (entre 4 i 8):

Pentesting, Violación de infraestructuras de red, Seguridad informática, Auditorias de red, Evaluación de riesgos, Seguridad en la red,

Abstract:

Keywords (entre 4 i 8):

Pentesting, Violation of network infrastructures, Computer security, Network audits, Risk assessment, Network security.

Índex

Llista de figures

1. Introducción

"JPA Cybersecurity Coop" es una iniciativa colaborativa liderada por Pau Ojeda Gallego, Anass El Ouardi y Justin Álvarez, con la visión de establecer una empresa especializada en Pentesting (Pruebas de Penetración). Nuestra misión es ofrecer servicios de evaluación y fortalecimiento de la seguridad a empresas, instituciones y organizaciones diversas. A través de técnicas avanzadas de hacking ético, pretendemos identificar y remediar vulnerabilidades en sistemas y redes para garantizar un entorno digital más seguro.

Por ello en nuestro proyecto hemos decidido simular y recrear una empresa donde contratan nuestros servicios y tenemos que hacer un análisis de su infraestructura, ejecutar técnicas de pentesting, y proporcionar soluciones.

Contexto

El proyecto está recientemente comenzado y sólo tenemos la idea y las bases de lo que vamos a hacer, tenemos la idea muy desarrollada y clara, pero estamos a una etapa muy temprana. Lo que nos motiva para ello es aprender mucho sobre seguridad y mejorar nuestras habilidades con la informática.

Justificación

Es un tema que nos gusta a todos los integrantes del grupo y que también nos motiva para aprender más cosas relacionadas con la seguridad informática (pensando en el grado superior), pero también queremos hacer énfasis en la web para tener más bases en el Grado Superior

2. Objetivos

Objetivo general

Queremos ir más allá de lo que nos enseñan en clase y sumergirnos en el mundo de la informática de verdad. Estamos súper emocionados por aprender haciendo, metiéndonos en proyectos prácticos que nos preparen para nuestro futuro. Nos vemos creando nuestra propia empresa más adelante, así que estamos enfocados en aprender tanto las habilidades técnicas como las de negocios. Nos encanta explorar nuevas tecnologías y ver cómo pueden aplicarse en el mundo real, desde la seguridad cibernética hasta el desarrollo de software y más.

Objetivos específicos

Ver si estamos capacitados para hacer nuestra propia empresa y si vemos que el proyecto ha estado bien, mejorarlo de cara al futuro e intentar hacer nuestra propia empresa.

Estrategia y planificación del proyecto

Una de las estrategias que podemos llegar a utilizar es decir que podemos romper el servidor o página web de una persona, para después reparar los errores

Adaptaremos un producto (herramientas para hackear) para lanzar servidores y hacer “pentesting”, para después reparar el error para que el servidor tenga un correcto funcionamiento.

Metodología de trabajo

Hemos optado por utilizar las metodologías waterfall e incremental, ya que la idea del proyecto la hemos hecho estructurada de inicio a fin en orden y lo incremental es porque estamos avanzando con el proyecto empezando desde los principios de nuestras ideas hasta la última que hemos tenido, por lo que el proyecto es como si se fuera haciendo mayor cada vez que avanzamos.

3.Introducción al proyecto

Topologia de red principal

Departamentos de Clientes

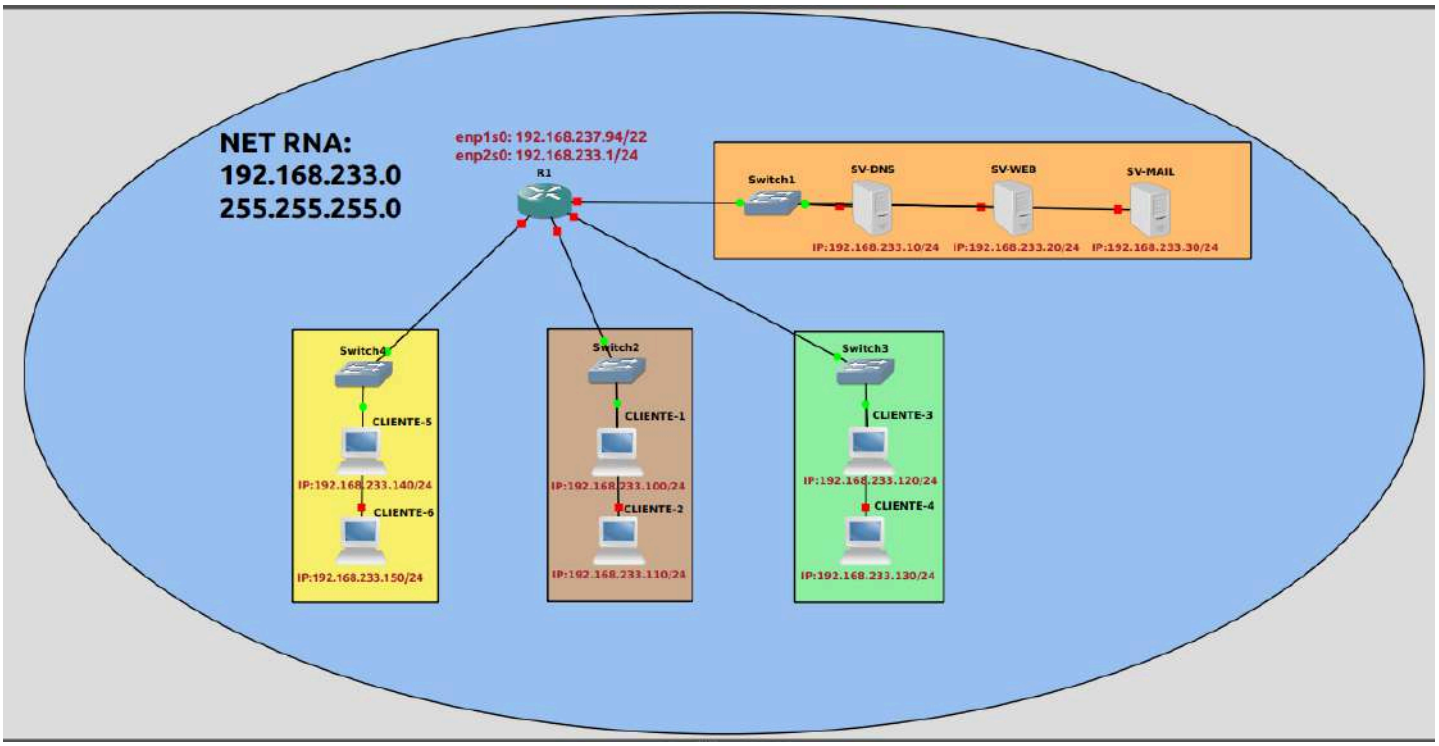
- **Departamentos Generales (3 en total)**: Cada uno de estos departamentos cuenta con dos clientes. Estos clientes pueden ser ordenadores de escritorio o estaciones de trabajo que los empleados utilizan para realizar sus tareas diarias. Los clientes de estos departamentos se comunican con los servidores centrales para acceder a servicios como correo electrónico, archivos compartidos, y aplicaciones de base de datos.

Departamento Central (Servidores y Gateway)

- **Servidores:** Este departamento alberga todos los servidores importantes que ofrecen diversos servicios a toda la organización. Entre estos servicios se pueden incluir:
    - **Servidor Web:** Maneja todas las solicitudes HTTP para la intranet de la empresa o sitios web externos administrados por la empresa.
    - **Servidor de Correos:** Administra todo el tráfico de correo electrónico, asegurando la comunicación interna y externa.
    - **Servidor DNS:** Resuelve los nombres de dominio en direcciones IP, esencial para la navegación por internet y el acceso interno a recursos.
- **Gateway:** Actúa como un punto de acceso entre la red interna de la empresa y el internet externo. Este dispositivo es crucial para la seguridad de la red, ya que puede implementar políticas de seguridad como firewalls, sistemas de prevención de intrusiones, y filtros de contenido para proteger la red de amenazas externas.

Conectividad de Red

- **Switches y Routers:** Los departamentos están interconectados a través de switches que facilitan la comunicación entre los clientes dentro de cada departamento y también con el departamento central de servidores. Un router puede ser utilizado para gestionar el tráfico entre los diferentes segmentos de la red y el gateway, asegurando que la comunicación entre los departamentos y el acceso a internet se realicen de manera eficiente.



Dispositivos de la red y sus servicios

Configuración General de la Red

- **Interfaz de Red 1:** puigcastellar1
  - IP Gateway: 192.168.237.94
- **Interfaz de Red 2**: puigcastellar2
  - IP Gateway: 192.168.233.1

Dispositivos

RNA-GW (Gateway/Router)
- **Servicio:** Actúa como gateway,DHCP.
- **Interfaz** puigcastellar2: IP 192.168.233.1 (como se mencionó, es el gateway)

SV-DNS (Servidor de DNS)
- **Servicio:** DNS
- **Interfaz** puigcastellar2: IP 192.168.233.10

SV-WEB (Servidor Web)
- **Servicio:** Web Hosting (HTTP/HTTPS)
- **Interfaz** puigcastellar2: IP 192.168.233.20

SV-MAIL (Servidor de Correo)

- **Servicio:** Correo Electrónico (SMTP, IMAP/POP3)
- **Interfaz** puigcastellar2: IP 192.168.233.30

CLIENTE-1 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.100

CLIENTE-2 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.101

CLIENTE-3 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.120

CLIENTE-4 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.130

CLIENTE-5 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.140

CLIENTE-6 (Cliente/Estación de trabajo)

- **Servicio:** Acceso a los servicios (DNS, Web, Mail)
- **Interfaz** puigcastellar2: IP 192.168.233.150

Del cliente 3 al 6 no son reales, simplemente son ordenadores ficticios para poder mostrar como quedaría la infraestructura debido a la demanda de recursos que se necesitan para sostener estos clientes.

Servicios y clientes

La configuración y servicios que hemos descrito forman un entorno de red empresarial típico, ofreciendo una amplia gama de servicios necesarios para las operaciones diarias de la empresa.

RNA-GW (Router/Gateway)

Funciones: Actúa como puerta de enlace (gateway) entre la red interna de la empresa y el exterior (Internet), además de proporcionar direcciones IP dinámicas a los dispositivos en la red a través de DHCP usando **KEA** y direcciones IP reservadas para los servidores, y un proxy transparente donde bloquean los anuncios emergentes y plataformas de streaming.

```
usuario@RNA-GW:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    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
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 52:54:00:2c:8d:93 brd ff:ff:ff:ff:ff:ff
    inet 192.168.237.94/22 metric 100 brd 192.168.239.255 scope global dynamic e
np1s0
       valid_lft 394sec preferred_lft 394sec
    inet6 fe80::5054:ff:fe2c:8d93/64 scope link
       valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 52:54:00:6a:ba:90 brd ff:ff:ff:ff:ff:ff
    inet 192.168.233.1/24 brd 192.168.233.255 scope global enp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe6a:ba90/64 scope link
       valid_lft forever preferred_lft forever
```

Configuración: Se le ha cambiado el nombre de host, se ha instalado el servicio DHCP, se ha activado el forwarding y se ha agregado una regla de iptables para poder reenviar paquetes a través de la interfaz de salida.

Forwarding:

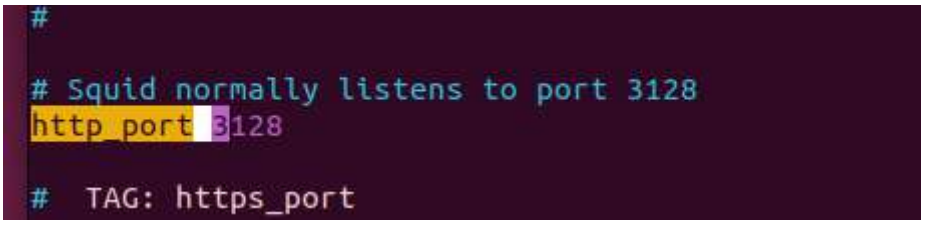
```
usuario@RNA-GW:~$ hostname
RNA-GW
usuario@RNA-GW:~$ cat /etc/rc.local
#!/bin/bash

iptables -t nat -A POSTROUTING -o enp1s0 -j MASQUERADE

exit 0
```

```
usuario@RNA-GW:~$ cat /proc/sys/net/ipv4/ip_forward
1
usuario@RNA-GW:~$
```

Config DHCP: Se ha configurado el servidor DHCP, dando direcciones IP entre un rango de 192.168.233.100-199 para clientes y reservando direcciones IP para los servidores.


usuario@RNA-GW: ~
{
"Dhcp4": {
"interfaces-config": {
"interfaces": [
"enp2s0"
],
"dhcp-socket-type": "raw"
},
"reservations-global": false,
"reservations-out-of-pool": true,
"valid-lifetime": 4000,
"renew-timer": 1000,
"rebind-timer": 2000,
"subnet4": [
{
"subnet": "192.168.233.0/24",
"match-client-id": false,
"option-data": [
{
"name": "routers",
"data": "192.168.233.1"
},
{
"name": "domain-name-servers",
"data": "192.168.233.10"
},
{
"name": "time-servers",
"data": "192.168.233.1"
},
{
"name": "domain-name",
"data": "netrna.domain"
}
],
"pools": [
{
"pool": "192.168.233.100-192.168.233.199"
}
],
"reservations": [
{

Config Proxy: Hemos implementado un proxy explícito en nuestra red con el objetivo de mejorar la productividad y seguridad. Este proxy está configurado para bloquear anuncios emergentes y el acceso a plataformas de streaming como Netflix, HBO y Amazon. Adicionalmente, hemos bloqueado el acceso a páginas con contenido para adultos.



(El puerto en el que funciona el proxy explícito).

Video del funcionamiento de los servicios:

https://youtu.be/ORDQq7NeEIM

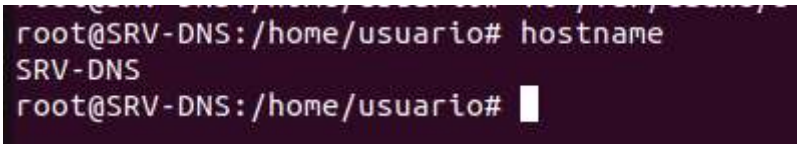
Rol en la red: Es el dispositivo central para el acceso a Internet y la asignación de IP, asegurando que los paquetes de datos lleguen a su destino correcto dentro y fuera de la red local.

SV-DNS (Servidor de DNS)

Funciones: Resuelve nombres de dominio a direcciones IP, permitiendo a los dispositivos en la red local encontrar servicios tanto internos como externos por nombre en lugar de por dirección IP.Clar

Configuración: Se ha instalado Bind9 y configurado una zona llamada netrna.domain, donde se declaran el servidor web y los servidores de correo.

Configuración DNS:



```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "netrna.domain" IN {
        type master;
        file "netrna.domain";
        };

~
~
```

```
$TTL 300
@       IN      SOA     ns admin.net.rna.domain. (
                        1       ; Serial
                        10800   ; Refresh
                        3600    ; Retry
                        604800  ; Expire
                        60      ; Minimum TTL
                        )

@       IN      NS      ns
ns      IN      A       192.168.233.10

@       IN      A       192.168.233.20

mail    IN      MX      10      mail

mail    IN      A       192.168.233.30
~
~
~
```

Rol en la red: Fundamental para la resolución de nombres, haciendo que la navegación web y el acceso a recursos internos sean más intuitivos.

Video del funcionamiento del servicio:

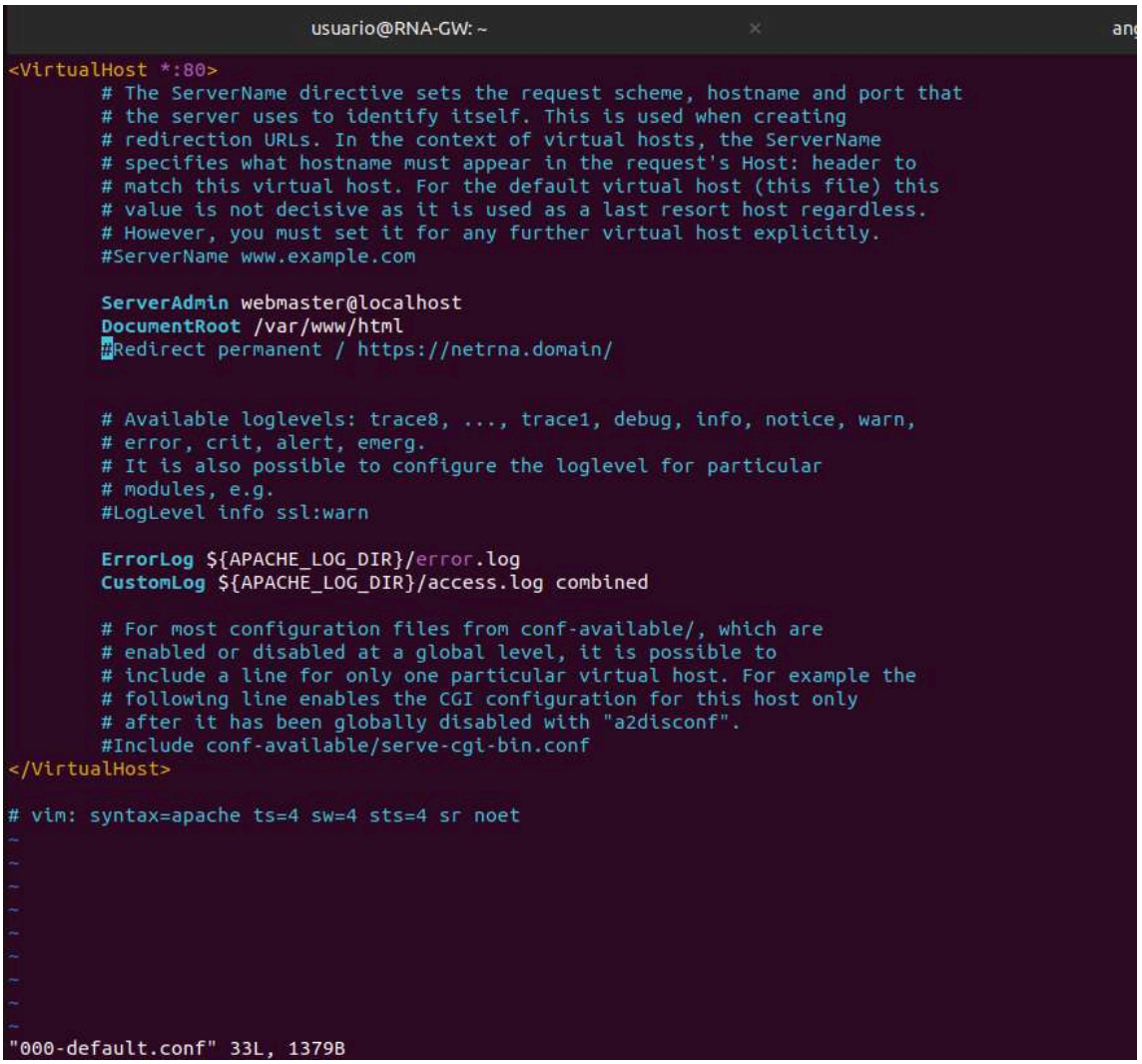
https://www.youtube.com/watch?v=mlmfzbHa0ss

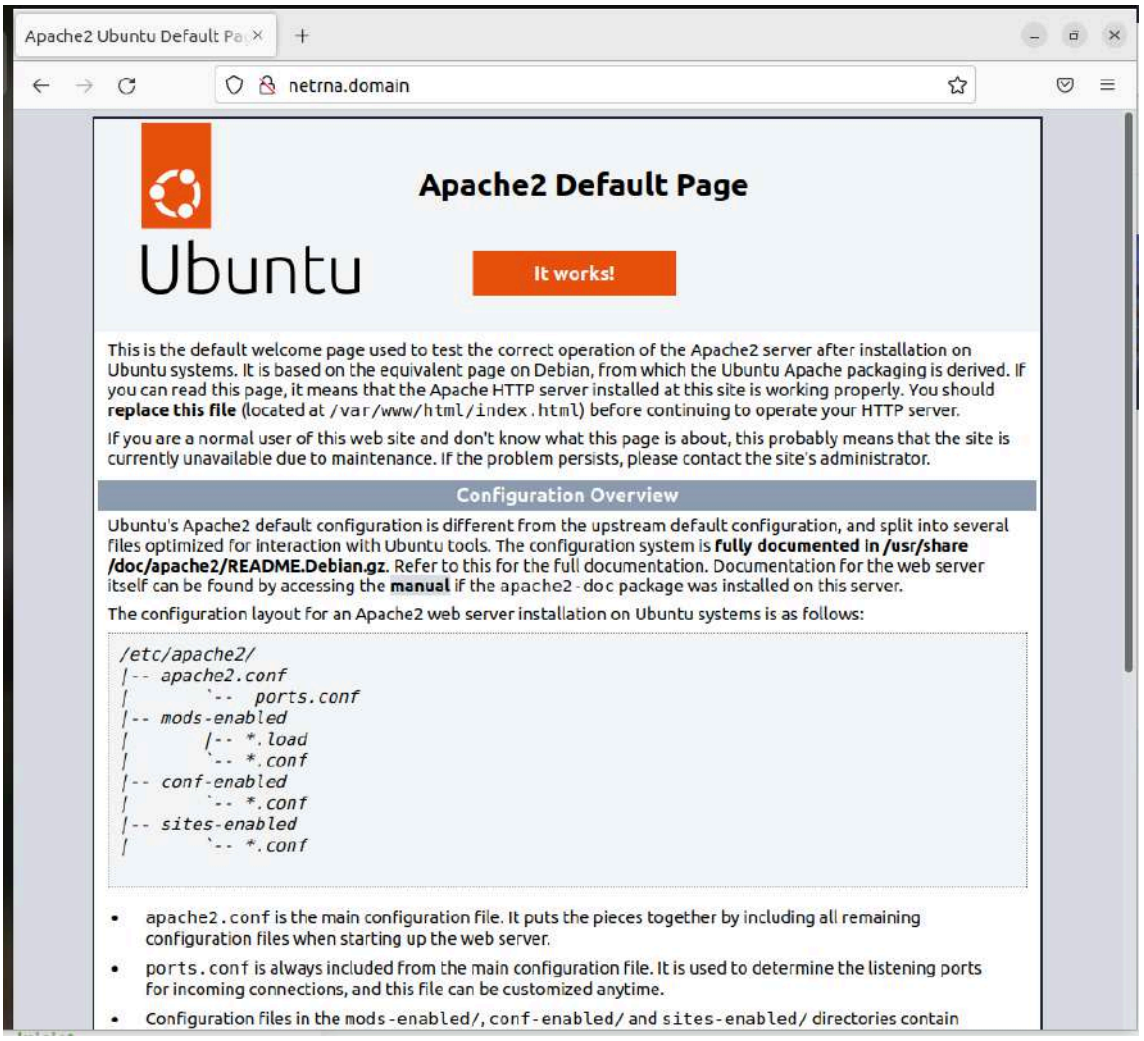
SRV-WEB (Servidor Web)

Funciones: Aloja la página web de la empresa, accesible tanto internamente como externamente si se configura correctamente.

Configuración: Se ha cambiado el nombre de host y actualizado el sistema.

Config apache2:


usuario@RNA-GW: ~
<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
#Redirect permanent / https://netrna.domain/
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
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>
vim: syntax=apache ts=4 sw=4 sts=4 sr noet
"000-default.conf" 33L, 1379B



Rol en la red: Provee información y servicios a empleados y posiblemente a clientes externos a través de la página web.

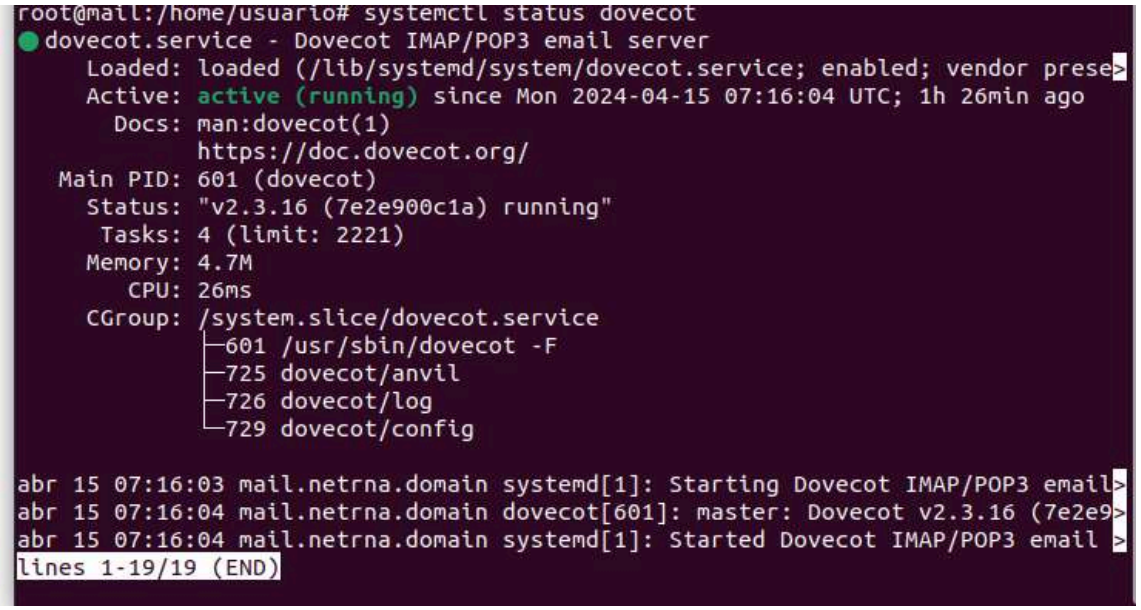
Video del funcionamiento de los servicio:

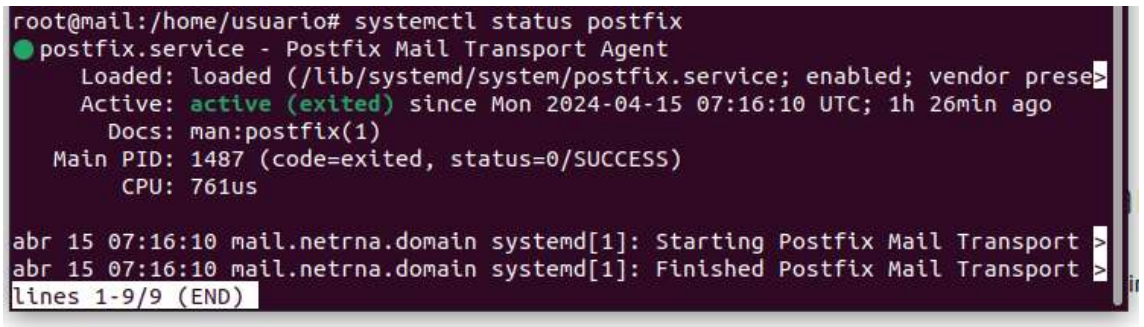
https://www.youtube.com/watch?v=qXzGfcQYRfQ

SRV-MAIL (Servidor de Correo)

Funciones: Gestiona el correo electrónico de la empresa (recepción y envío), utilizando Postfix como agente de transferencia de correo (MTA) y Dovecot como agente de entrega de correo (MDA).

Configuración: Configurado con el dominio netrna.domain, cambió a Maildir, permitiendo el relay de correos para clientes en la LAN y configuración de mailutils.

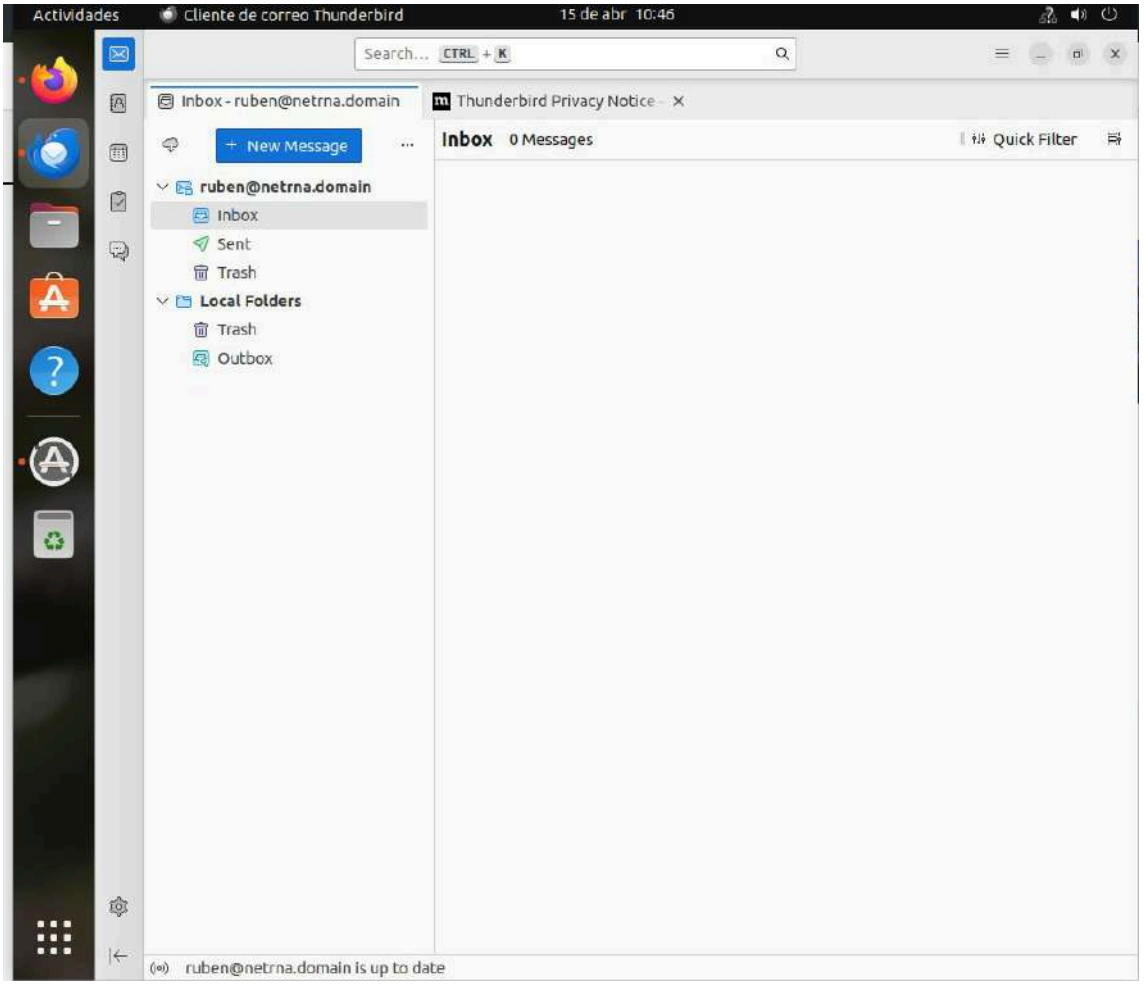


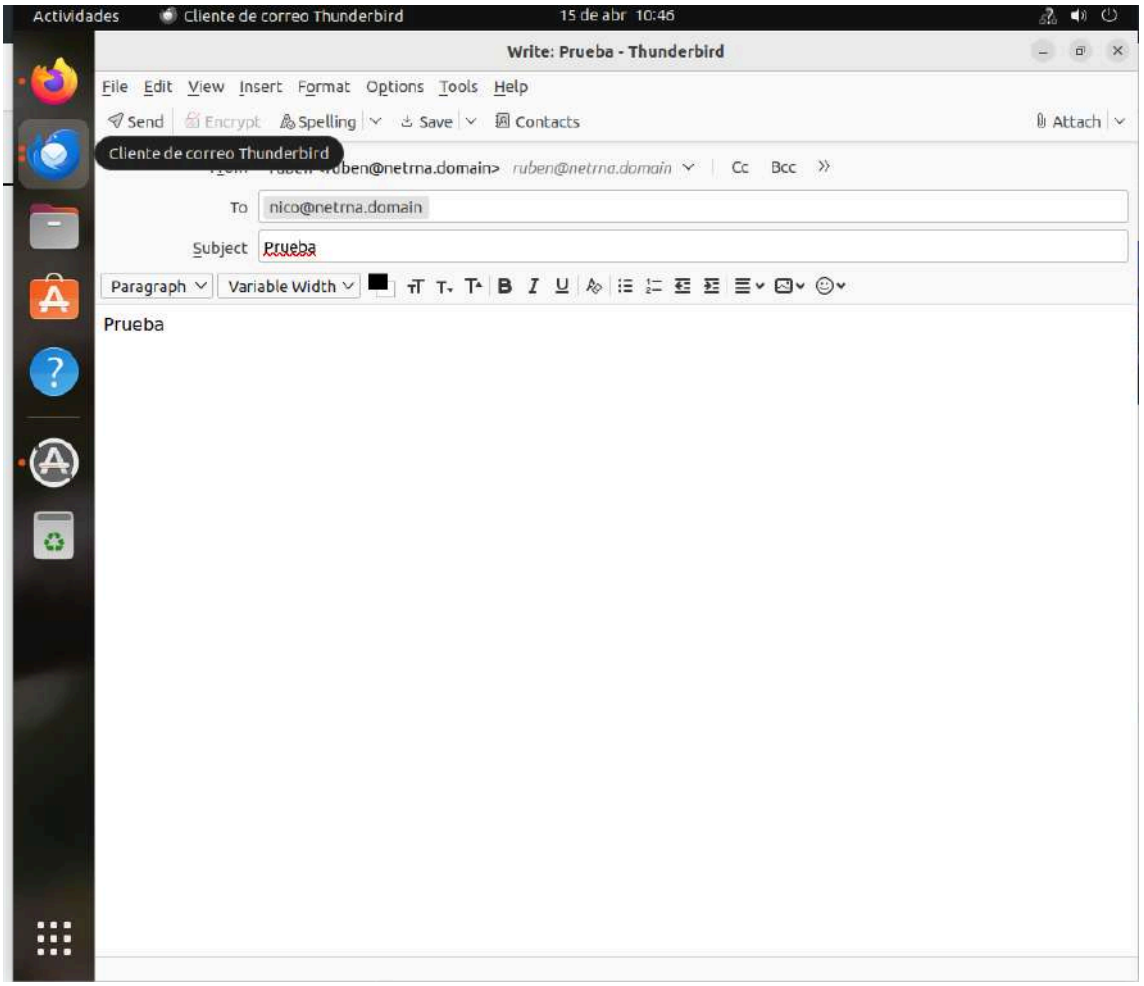


Permitimos el relay en nuestra red (192.168.233.0/24).

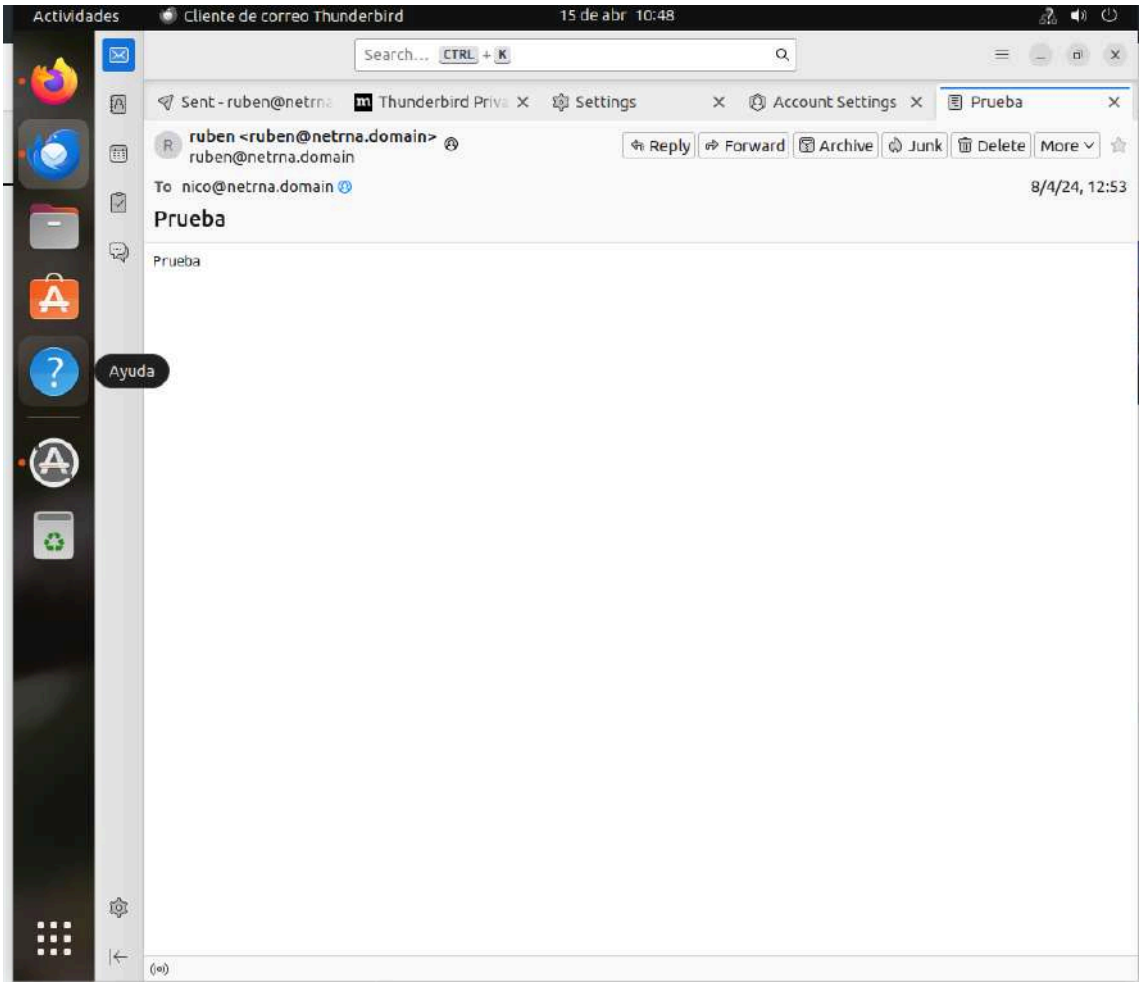
```
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, netrna.domain, mail.netrna.domain, localhost.netrna
.domain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.233.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
"/etc/postfix/main.cf" 47L, 1428B                    43,66         Final
```

Como podemos ver en el cliente, ya nos podemos conectar con el usuario y el dominio y enviar un correo.





Y como podemos ver, ha sido recibido por el destinatario.



Rol en la red: Esencial para la comunicación interna y externa, permitiendo el intercambio de correos de manera segura y eficiente.

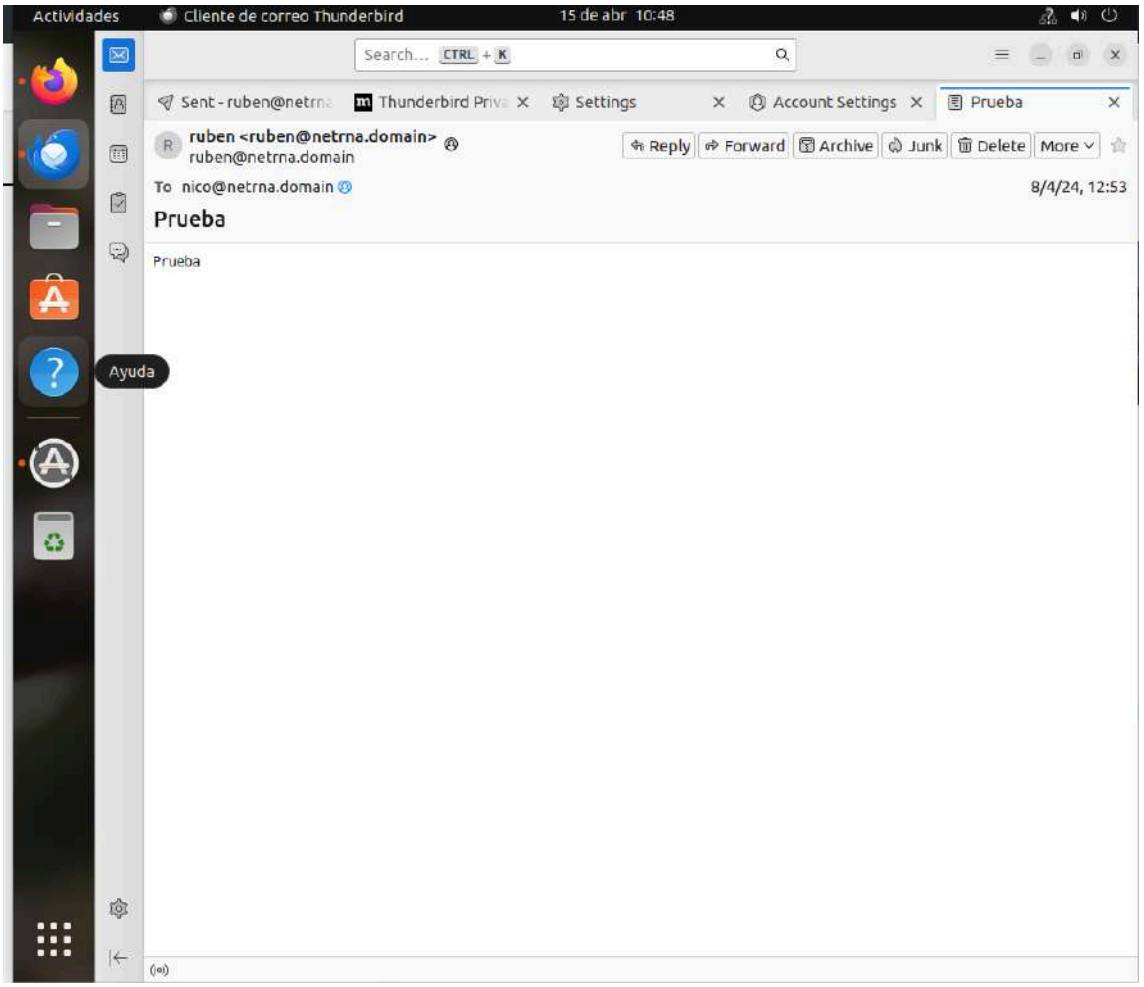
Video del funcionamiento del servicios:

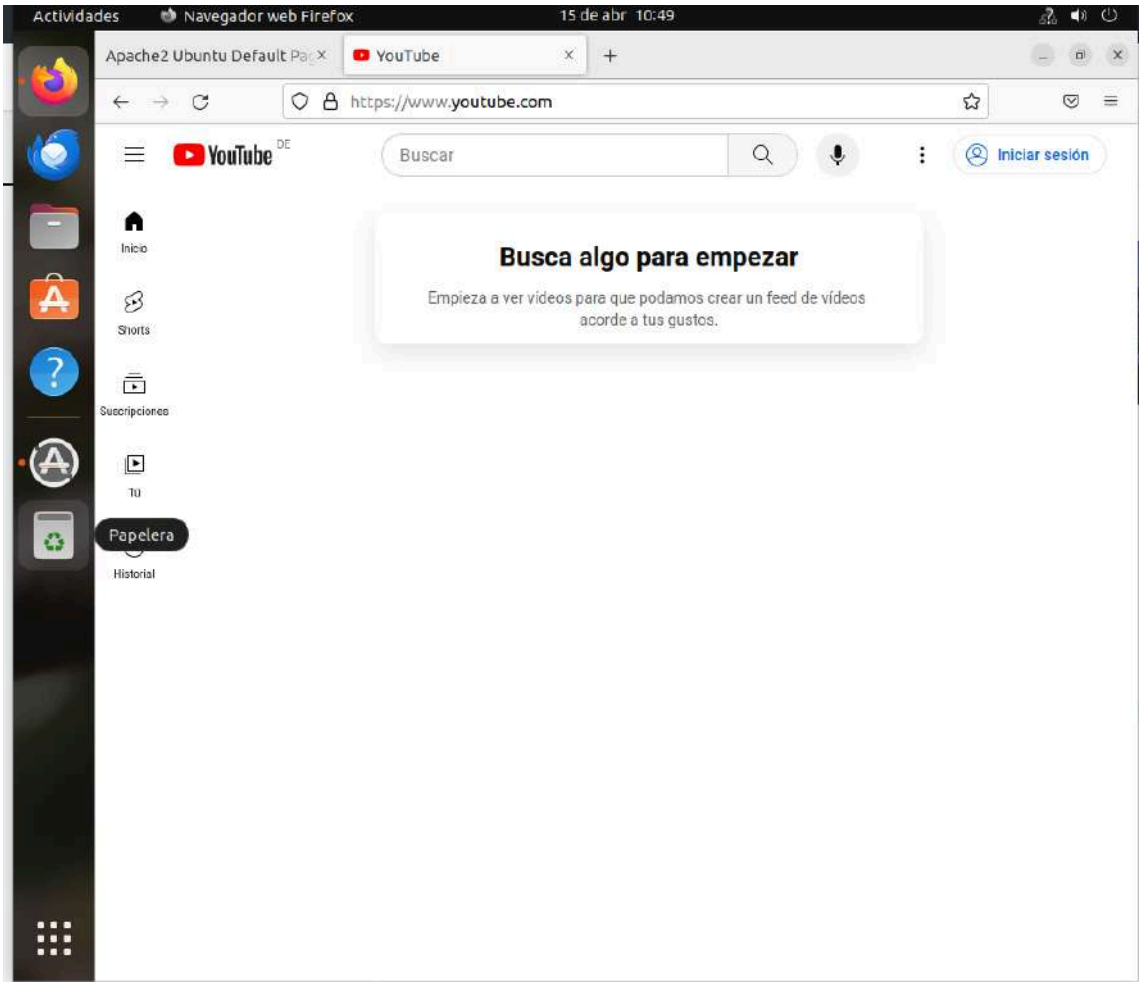
https://www.youtube.com/watch?v=-ssvauoyQ00

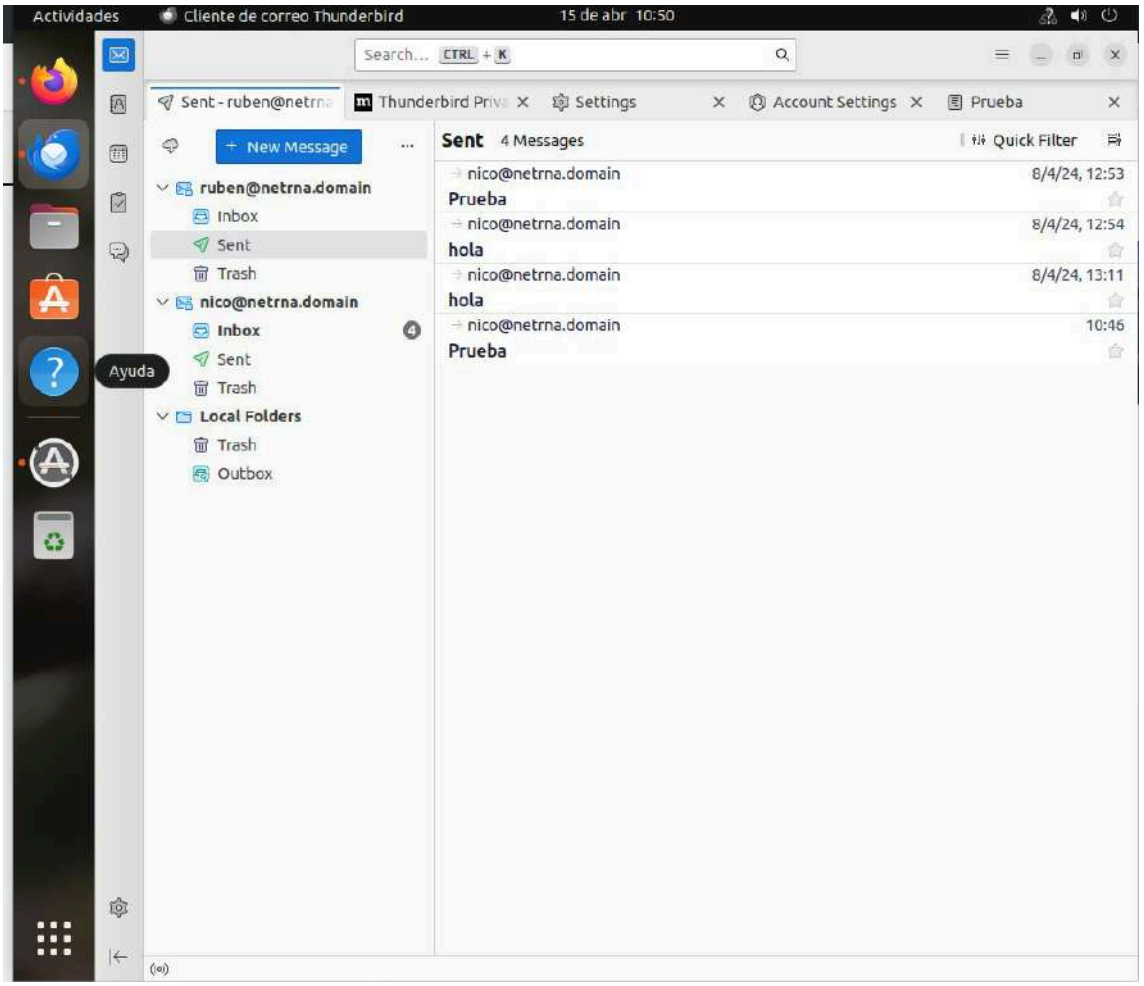
CLIENTE-1 y CLIENTE-2 (Estaciones de Trabajo)

Funciones: Permiten a los usuarios acceder a los servicios de la red (navegación web, correo electrónico, etc.).

Cliente 1:


Actividades
Cliente de correo Thunderbird
15 de abr 10:48
Search... CTRL + K
Sent - ruben@netrna
Thunderbird Priva
Settings
Account Settings
Prueba
ruben <ruben@netrna.domain>
ruben@netrna.domain
Reply
Forward
Archive
Junk
Delete
More
To nico@netrna.domain
8/4/24, 12:53
Prueba
Prueba
Ayuda


Actividades
Navegador web Firefox
15 de abr 10:49
Apache2 Ubuntu Default Pa
YouTube
https://www.youtube.com
YouTube DE
Buscar
Iniciar sesión
Inicio
Shorts
Suscripciones
Tú
Papelera
Historial
Busca algo para empezar
Empieza a ver vídeos para que podamos crear un feed de vídeos acorde a tus gustos.



Cliente 2:

Configuración: Cliente 1 corre Ubuntu Desktop y Cliente 2 Windows 7, ambos configurados para operar dentro de la red, utilizando los servicios DNS, web, y mail proporcionados.

Rol en la red: Son los usuarios finales de los servicios proporcionados, realizando actividades cotidianas como navegar por Internet y enviar/recibir correo electrónico.

Cada componente está configurado para desempeñar un rol específico dentro de la infraestructura de red, asegurando que la red sea funcional.

Inciso:

Es importante resaltar que, aunque los servicios han sido instalados y configurados en las máquinas mencionadas, la seguridad de estas no ha sido tratada con la profundidad necesaria. Esto no es un descuido, sino parte de un enfoque práctico con el objetivo de identificar y demostrar vulnerabilidades en una red con configuraciones básicas, para luego proceder a implementar las correspondientes mejoras y soluciones de seguridad.

Detección de posibles vulnerabilidades y solución.

Para asegurar una evaluación de seguridad integral, detallaremos métodos y técnicas adaptadas a cada tipo de ataque simulado. Esto nos permitirá identificar vulnerabilidades específicas en nuestra infraestructura de red y ajustar nuestras medidas de protección según sea necesario.

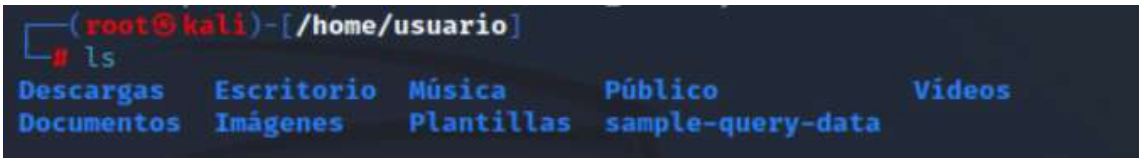
Ataque DoS al servidor web

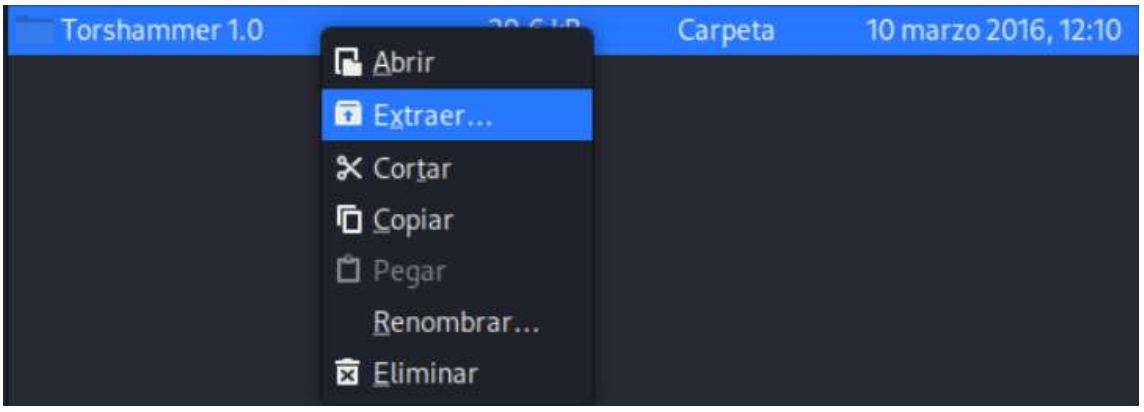
Este ataque prueba la capacidad del servidor para manejar un volumen excepcionalmente alto de tráfico. La técnica consiste en saturar el servidor con un flujo constante y masivo de datos, con el objetivo de agotar sus recursos hasta que se vuelva incapaz de procesar solicitudes legítimas. Este método nos ayudará a determinar la necesidad de implementar soluciones más robustas para la gestión de tráfico o la mejora de la infraestructura.

Realización del ataque:

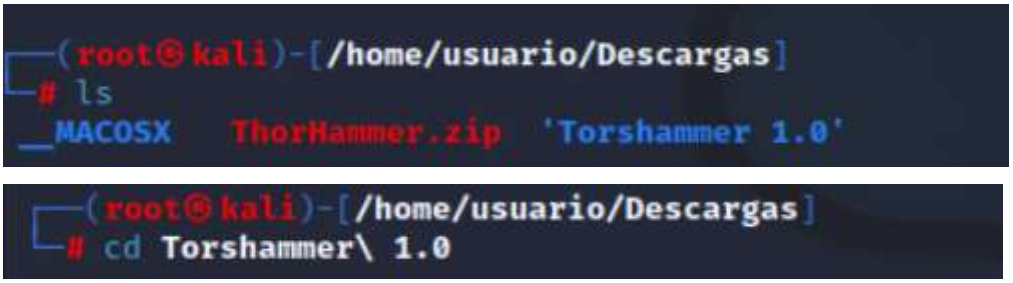
Hemos instalado una herramienta llamada Thor's Hammer. A continuación, procederemos a descomprimir los archivos de la herramienta y posteriormente ejecutarla. Este proceso es crucial para iniciar la utilización de la herramienta.

Hemos descargado el ataque desde la propia web del centro, gracias a la documentación del profesor Jordi Farrero, a continuación el link de la descarga:
https://elpuig.xeill.net/Members/jordifarrero/2014-15-seguretat-en-xarxes-sm2ab-diurn/uf2-scripts-demo/thor-hammer-python

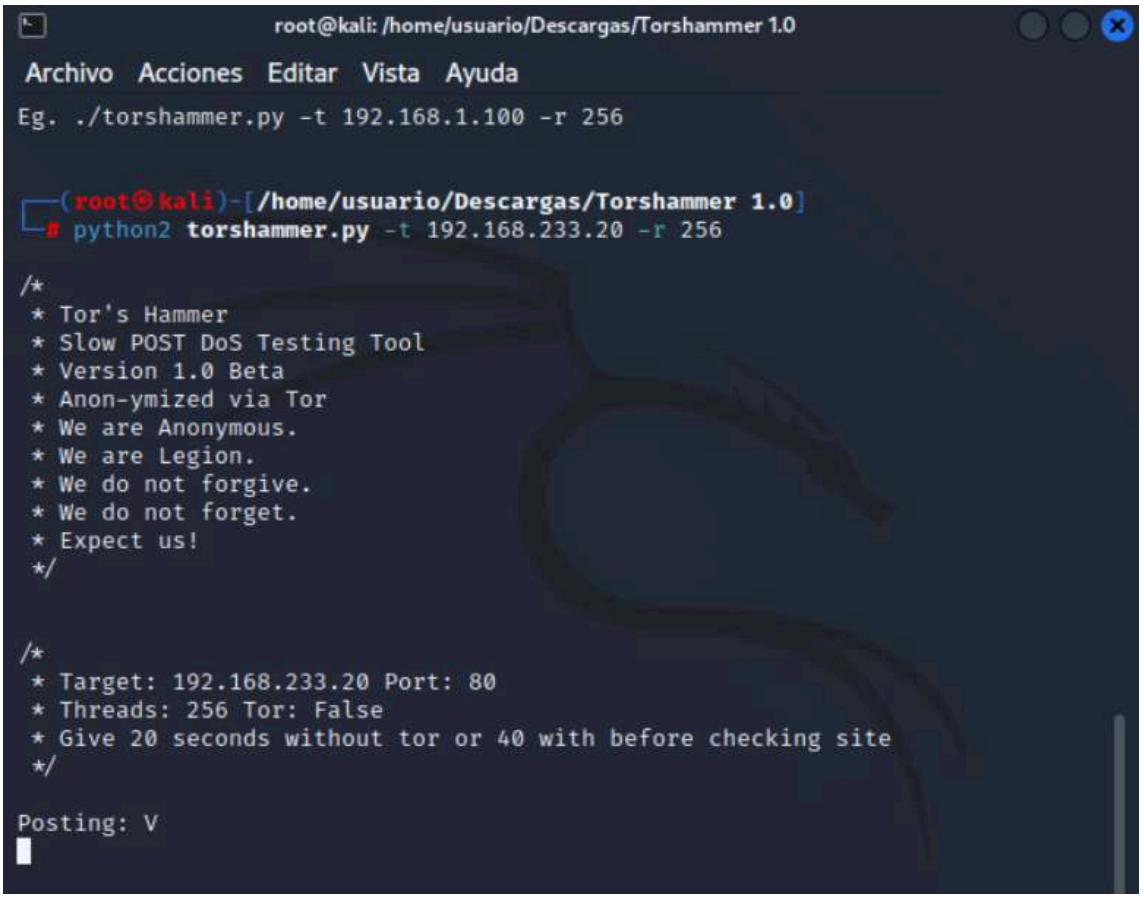




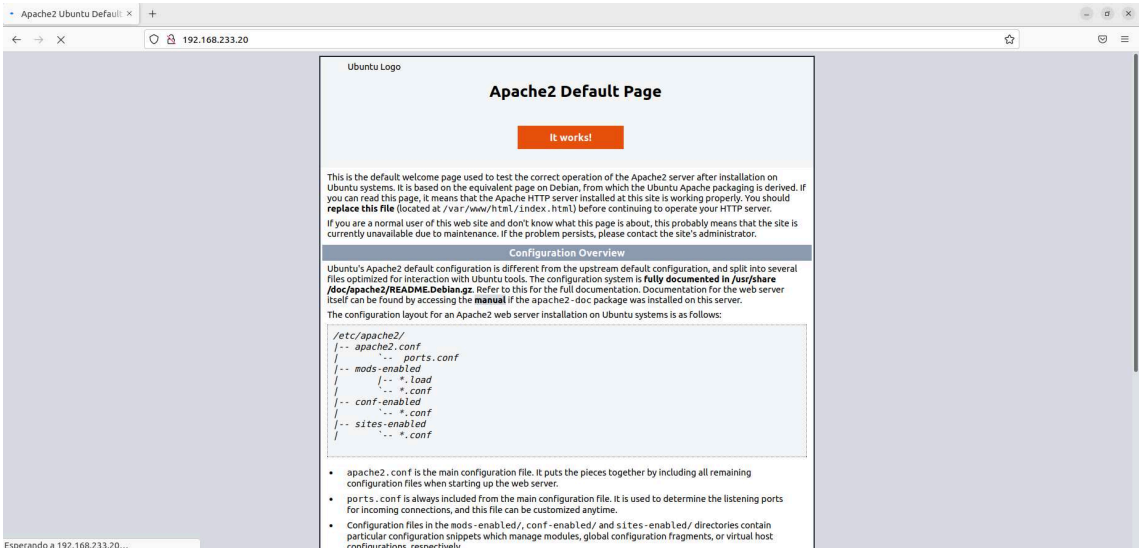
Una vez descomprimido el archivo, procederemos a entrar en el directorio correspondiente.



Finalmente, ejecutaremos la herramienta utilizando el comando mostrado en la captura. Es importante especificar en el comando la dirección IP de la máquina víctima, lo cual indica claramente el destino de nuestro ataque.



El resultado final del ataque es una notable disminución en la velocidad de carga del servidor Apache de la víctima. Este efecto se debe a la sobrecarga del servicio, intencionalmente provocada por el ataque, como se puede observar en las capturas de pantalla proporcionadas a continuación.



"Thor 's Hammer" es una herramienta diseñada para ejecutar ataques de denegación de servicio mediante la generación de numerosas solicitudes lentas a un servidor web. Este método aprovecha el uso de cabeceras HTTP para establecer múltiples conexiones con el servidor, las cuales se mantienen abiertas enviando datos a un ritmo extremadamente lento. De esta forma, el ataque sobrecarga los recursos del servidor, dificultando su capacidad de responder a los usuarios legítimos. Esto resulta en una ralentización significativa del servicio web, afectando potencialmente su operatividad y disponibilidad.

Video del ataque:

https://www.youtube.com/watch?v=3XtVxE_k3rw

Solucion:

El primer paso que vamos a utilizar para prevenir este tipo de ataques es utilizar en el servidor web unas reglas de firewall.

```
#!/bin/bash

sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT

sudo iptables -A INPUT -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 60 --hitcount 4 -j DROP
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

sudo iptables -A INPUT -i lo -j ACCEPT

sudo iptables -A INPUT -m conntrack --cstate ESTABLISHED,RELATED -j ACCEPT
```

Las iptables configuradas en el servidor web establecen políticas predeterminadas para bloquear todo el tráfico entrante y el reenvío de paquetes, mientras permiten todo el tráfico saliente. Se habilitan conexiones SSH con limitaciones para mitigar ataques de fuerza bruta, y se permiten conexiones HTTP y HTTPS para el acceso a servicios web. Además, se permite el tráfico desde la interfaz de loopback y se

aceptan conexiones ya establecidas y relacionadas para mantener sesiones activas sin interrupciones.

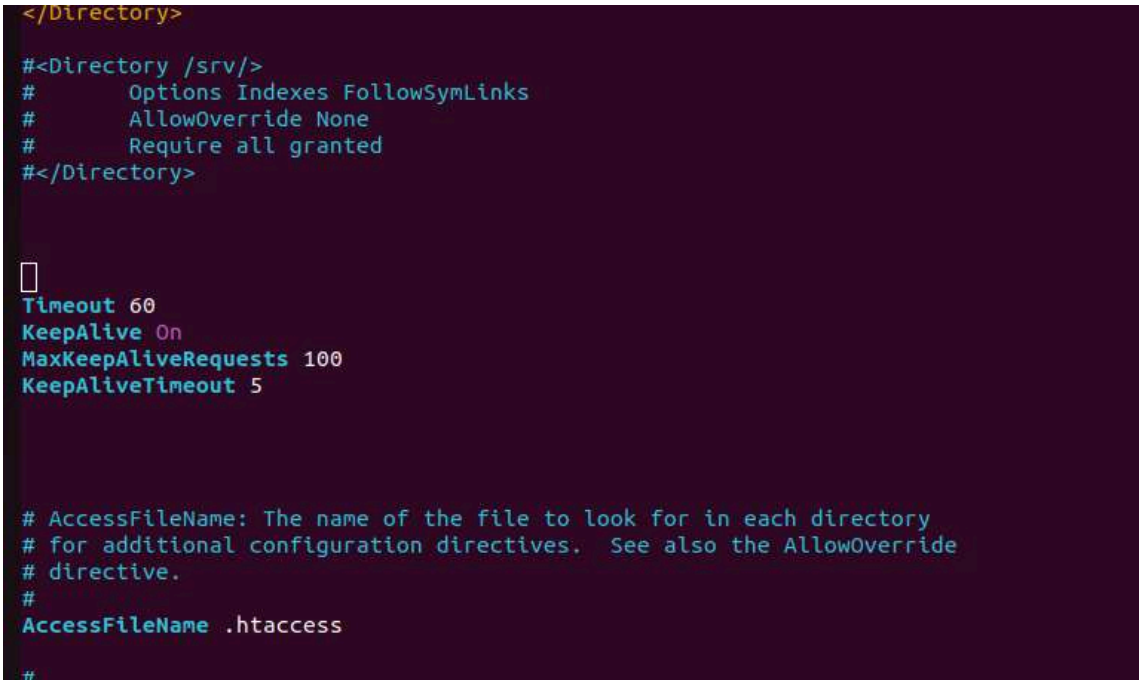
Lo siguiente que hacemos es instalar la herramienta **fail2ban** y le aplicamos la configuración.

```
root@srv-web:/home/usuario# apt install fail2ban_
```

```
bantime = 3600
findtime = 600
maxretry = 5

[sshd]
enabled = true

[apache]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache*/*error.log
maxretry = 3
```

La configuración de fail2ban bloquea una IP por una hora tras 5 intentos fallidos en 10 minutos. Protege el servicio SSH contra accesos no autorizados. Para Apache, monitorea intentos fallidos en los puertos HTTP y HTTPS, bloqueando IPs tras 3 intentos fallidos. Utiliza el filtro apache-auth para identificar estos intentos en los archivos de registro de error de Apache. Así, fail2ban asegura tanto SSH como Apache contra ataques de fuerza bruta.

Lo siguiente que se hace es la configuración del apache:



La configuración de mod_evasive en Apache ayuda a prevenir ataques DoS y DDoS. Limita las solicitudes a la misma página a 2 por segundo y al sitio completo a 50 por segundo, bloqueando IPs ofensivas por 10 segundos. Envía notificaciones al administrador y ejecuta un comando para bloquear la IP atacante usando iptables. Además, guarda los registros de ataques en /var/log/mod_evasive para análisis posterior.

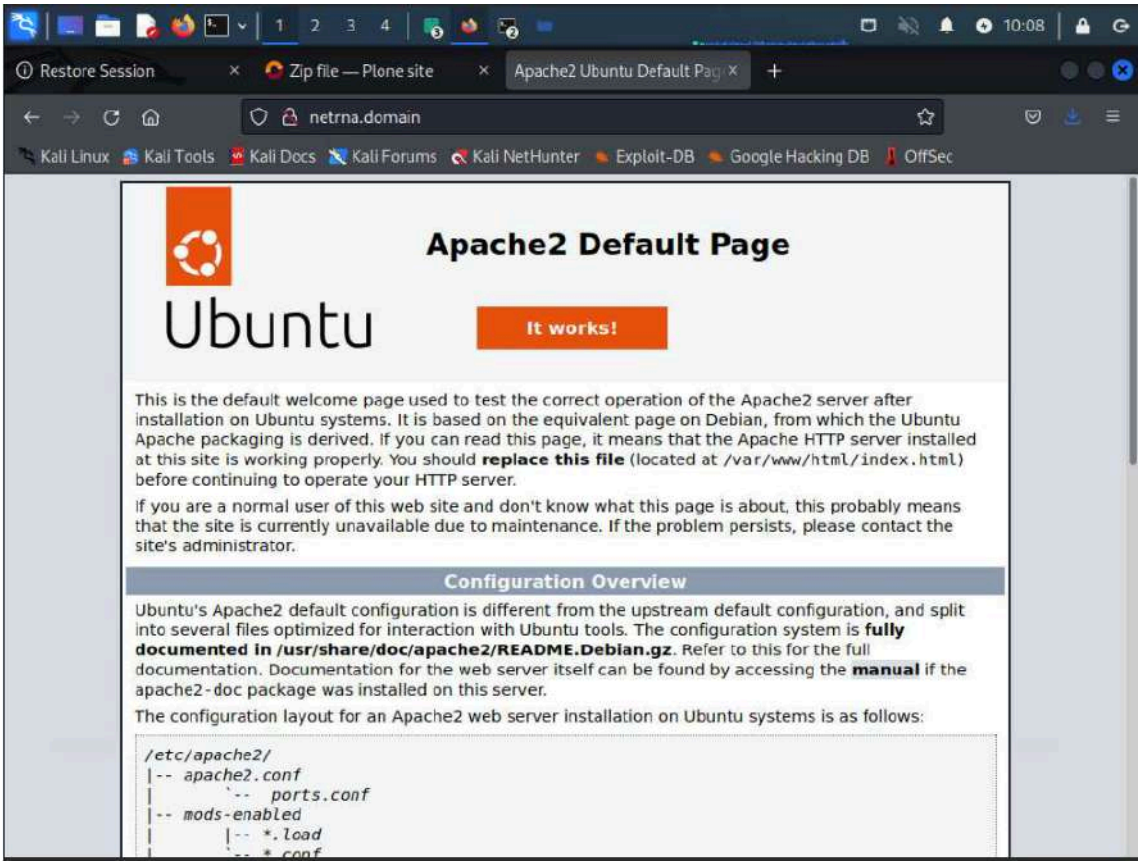
Se continúa instalando el módulo evasive de apache2.

```
root@srv-web:/home/usuario# sudo apt install libapache2-mod-evasive
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  bsd-mailx liblockfile-bin liblockfile1 postfix
Paquetes sugeridos:
  procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin
  | dovecot-common resolvconf postfix-cdb postfix-mta-sts-resolver postfix-doc
Se instalarán los siguientes paquetes NUEVOS:
  bsd-mailx libapache2-mod-evasive liblockfile-bin liblockfile1 postfix
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 104 no actualizados.
Se necesita descargar 1.351 kB de archivos.
Se utilizarán 4.507 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Y por último lo activamos.

```
root@srv-web:/home/usuario# sudo a2enmod evasive
Module evasive already enabled
root@srv-web:/home/usuario# sudo systemctl restart apache2
root@srv-web:/home/usuario#
```

Para acabar volvemos a ejecutar el ataque y comprobamos si la configuración que hemos aplicado a los servicios funciona.



Por último, implementaremos HTTPS en nuestra página para mejorar la seguridad. HTTPS proporciona cifrado para las comunicaciones entre el servidor y los clientes, protegiendo los datos sensibles contra interceptaciones y ataques. Para ello, obtendremos un certificado SSL/TLS de una autoridad de certificación, configuraremos Apache para utilizar este certificado, y añadiremos una regla para redirigir automáticamente todas las peticiones HTTP a HTTPS. Esto garantizará que todos los usuarios accedan a la versión segura de nuestro sitio web.

Video del ataque solucionado:

https://www.youtube.com/watch?v=5FFXTAguut0

```
<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
        Redirect permanent / https://netrna.domain/


        # 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

        # 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>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

En conclusión, para mitigar los ataques y proteger tu infraestructura, es esencial implementar una configuración de seguridad integral. Esto

incluye el uso de firewalls para filtrar el tráfico, mantener sistemas y aplicaciones actualizados con los últimos parches de seguridad, y segmentar la red para limitar el alcance de posibles ataques. Además, es crucial utilizar sistemas de monitoreo y detección de intrusos.

Ataque DNS Flood al servidor DNS

Mediante la simulación de una gran cantidad de solicitudes de resolución de nombres en un corto período de tiempo, podemos poner a prueba la capacidad del servidor DNS para manejar picos de carga. Este tipo de ataque busca identificar si el servidor DNS tiene configuraciones adecuadas para manejar altos volúmenes de tráfico, incluyendo la posibilidad de instaurar medidas preventivas como la limitación de tasa de peticiones.

Realización del ataque:

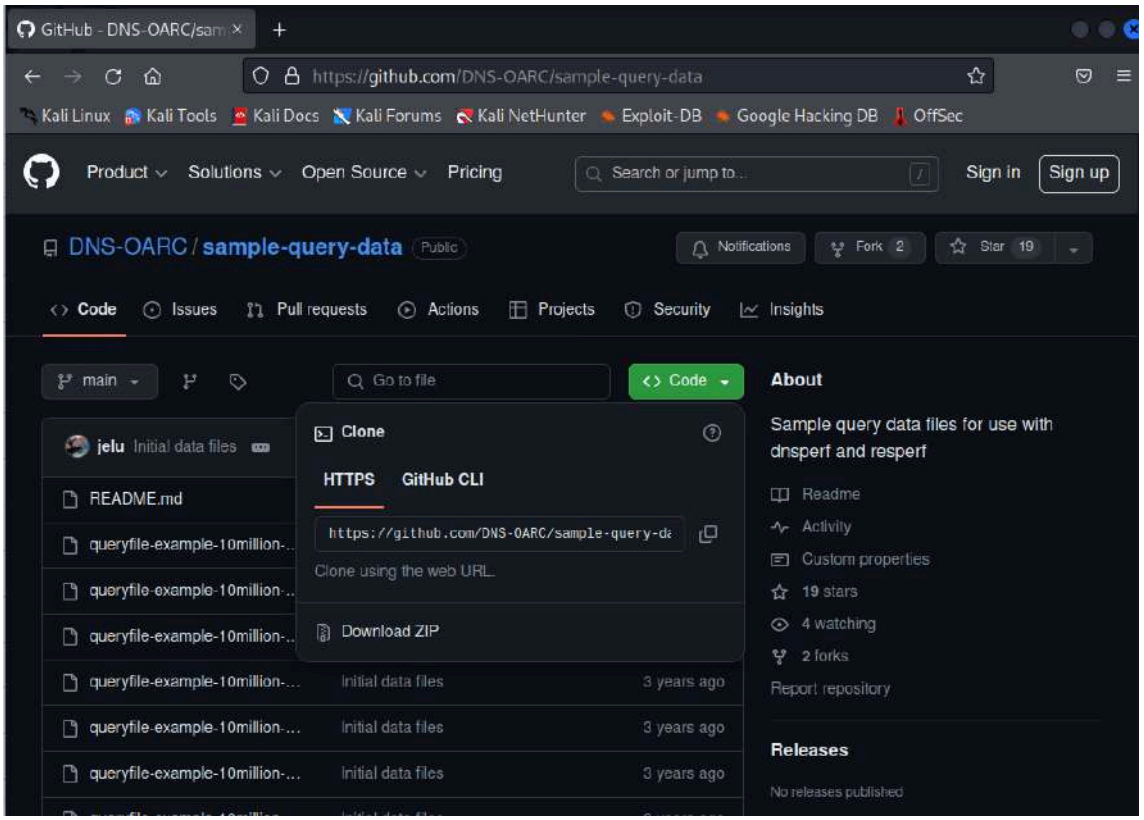
Para la realización del ataque, lo hemos sacado de la propia web del centro Institut Puig Castellar:
https://elpuig.xeill.net/Members/vcarceler/articulos/pruebas-de-rendimiento-de-un-servidor-dns-con-dnsperf-y-resperf

Empezando, descargamos el DNS-OARC via github que sirve para hacer pruebas de rendimiento a un servidor DNS podemos utilizar dnsperf y resperf.

El ataque tiene un funcionamiento simple

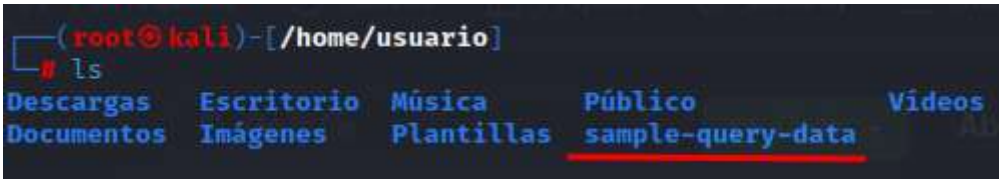
Cuando es lanzado, se envía 1 millón de consultas al DNS, básicamente pregunta por 1 millón de dominios a toda la velocidad que el DNS pueda

Para lanzar un test es necesario contar con el archivo de datos (el enlace de github) para hacer las consultas. El mismo DNS-OARC nos proporciona unos ficheros con 10 millones de consultas (1 millón por fichero).



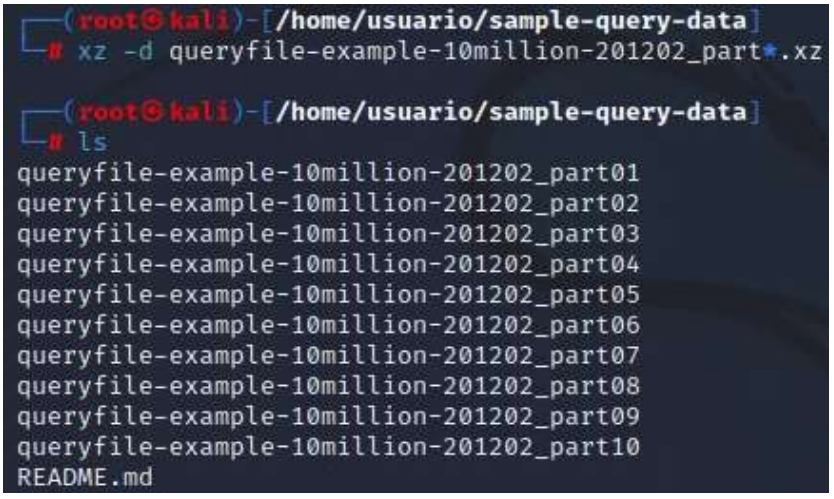
Hacemos un git-clone

Y ya lo tendremos ahí

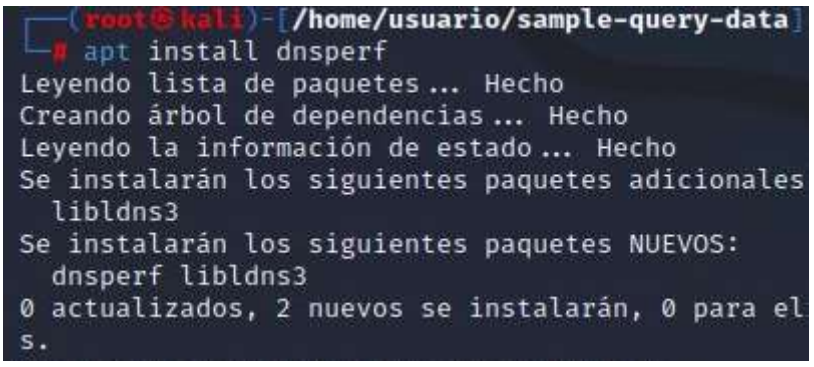


Tendremos aquí ya las 10 partes con 1 millón de ficheros en cada uno, osea 10 millones de posibles consultas.

Lo descomprimimos



Para poder continuar, habrá que instalar “dnsperf”



Y ya lanzamos el “Ataque”

```
┌──(root㉿kali)-[/home/usuario/sample-query-data]
└─# resperf -d queryfile-example-10million-201202_part01 -s 192.168.233.10
DNS Resolution Performance Testing Tool
Version 2.14.0

[Status] Command line: resperf -d queryfile-example-10million-201202_part01 -
s 192.168.233.10
[Status] Sending
```

Aquí vemos que con uno el servidor DNS empieza a ralentizarse

Sin el ataque:

```
top - 10:21:41 up 6 min,  2 users,  load average: 0,13, 0,20, 0,10
Tasks: 109 total,   2 running, 107 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2,0 us,  0,3 sy,  0,0 ni, 97,7 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   1963,9 total,    515,9 free,    846,7 used,    601,3 buff/cache
MiB Swap:   4096,0 total,   4096,0 free,      0,0 used.    961,9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    652 bind      20   0  933648 692948   7952 S   2,0  34,5   0:25.20 named
    366 root      19  -1  142176  89708  88604 S   0,3   4,5   0:06.33 systemd-journal
    917 root      20   0   11676   5548   4704 S   0,3   0,3   0:00.15 sudo
```

Con 1 millón de peticiones del ataque:

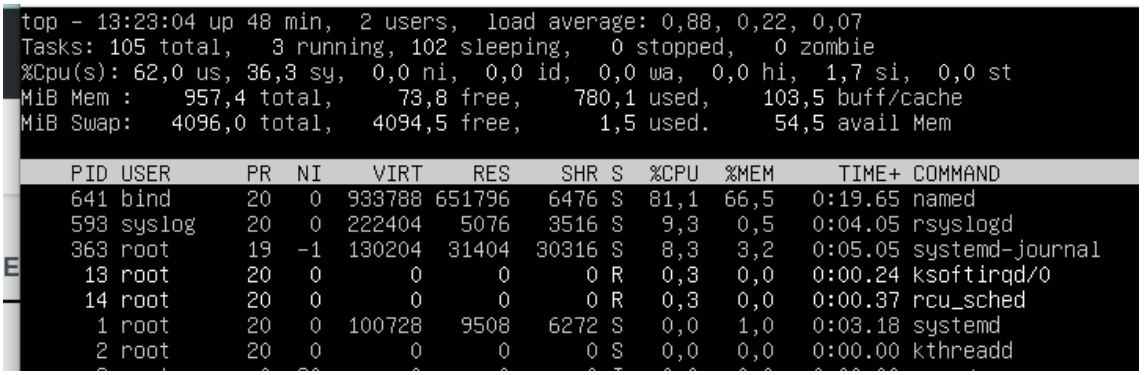
```
top - 10:23:04 up 8 min,  2 users,  load average: 0,82, 0,34, 0,15
Tasks: 109 total,   3 running, 106 sleeping,   0 stopped,   0 zombie
%Cpu(s): 62,8 us, 36,5 sy,  0,0 ni,  0,0 id,  0,0 wa,  0,0 hi,  0,7 si,  0,0 st
MiB Mem :   1963,9 total,    449,2 free,    873,3 used,    641,5 buff/cache
MiB Swap:   4096,0 total,   4096,0 free,      0,0 used.    934,7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    652 bind      20   0  933648 719344   7952 S  85,7  35,8   0:46.74 named
    601 syslog    20   0  222404   6024   4468 S   7,7   0,3   0:08.12 rsyslogd
```

Entonces, vamos intentar tirarlo, vamos a utilizar 3 partes (3 millones de peticiones)

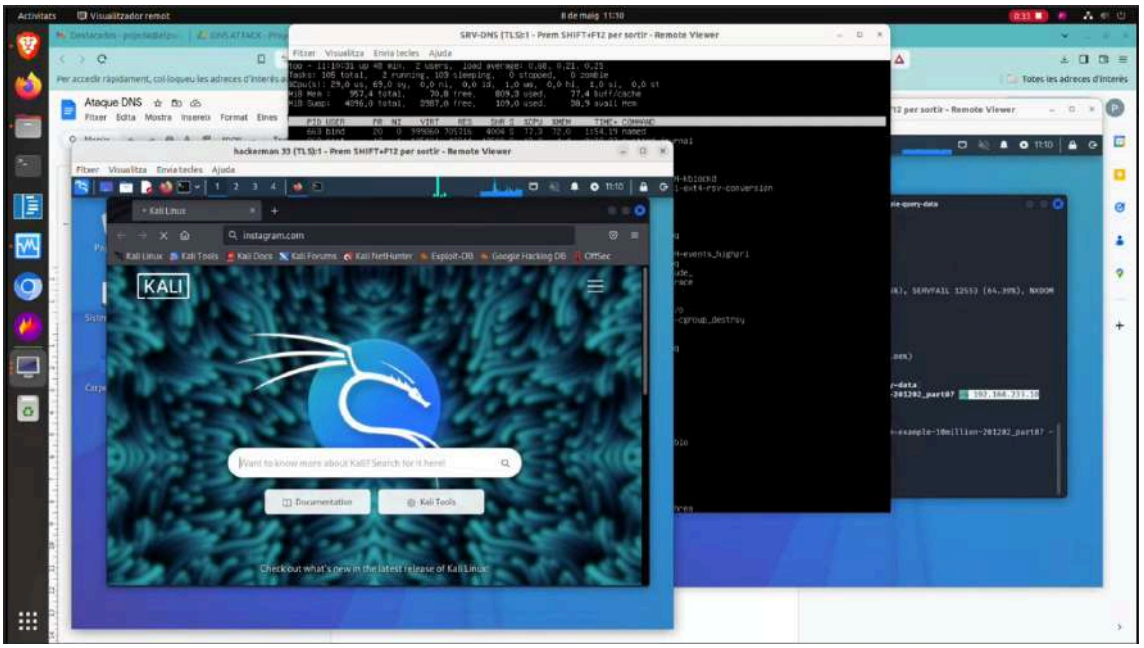
```
root@kali: /home/usuario/sample-query-data
Archivo  Acciones  Editar  Vista  Ayuda
root@kali:...query-data ×  root@kali:...query-data ×  root@kali:...query-data ×
┌──(usuario㉿kali)-[~]
└─$ sudo -s
[sudo] contraseña para usuario:
┌──(root㉿kali)-[/home/usuario]
└─# cd sample-query-data

┌──(root㉿kali)-[/home/usuario/sample-query-data]
└─# resperf -d queryfile-example-10million-201202_part01 -s 192.168.233.10
```

Con 3 tampoco puede tirar el DNS

Como no acaba de tirarlo, vamos a probar el ataque con 2 kali



Sorprendentemente el servidor DNS ha llegado a “resistir” (no lo hemos conseguido tirar) el ataque DNS Flood, aunque el servidor se ha ralentizado bastante, no ha llegado a romperse lo cúal no está nada mal, aunque si tratabas de buscar alguna página web mientras el ataque estaba siendo lanzado, no se podía entrar hasta que el ataque no estaba llegando a su fin (como se demuestra en el vídeo).



Video del ataque:

https://www.youtube.com/watch?v=8XtXhFmv9_4

Solucion:

La solución más sencilla para añadir una capa más de seguridad a nuestro servidor DNS estableciendo un límite de peticiones al DNS durante un minuto, para que esté así no colapse

Por ende vamos a establecer unas reglas de IPTables para tener más seguridad

```
root@bind:/home/usuario# sudo iptables -A INPUT -p udp --dport 53 -m recent
--set --name dnsquery
root@bind:/home/usuario# sudo iptables -A INPUT -p udp --dport 53 -m recent
--update --seconds 60 --hitcount 5 --name dnsquery -j DROP
```

Sin reglas de IPTables durante el ataque:

```
Received 166 bytes from 127.0.0.53#53 in 552 ms
usuario@srv-dns:~$ host -a wikipedia.es
Trying "wikipedia.es"
;; communications error to 127.0.0.53#53: timed out
;; communications error to 127.0.0.53#53: timed out
;; no servers could be reached

usuario@srv-dns:~$
```

Con reglas IPTables durante el ataque:

```
usuario@srv-dns:~$ host -a wikipedia.es
Trying "wikipedia.es"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23946
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 3

;; QUESTION SECTION:
;wikipedia.es.                  IN      ANY

;; ANSWER SECTION:
wikipedia.es.           3600    IN      HINFO   "RFC8482" ""
wikipedia.es.           86136   IN      NS      ns0.wikimedia.org.
wikipedia.es.           86136   IN      NS      ns2.wikimedia.org.
wikipedia.es.           86136   IN      NS      ns1.wikimedia.org.

;; ADDITIONAL SECTION:
ns0.wikimedia.org.      2971    IN      A       208.80.154.238
ns1.wikimedia.org.      2971    IN      A       208.80.153.231
ns2.wikimedia.org.      2971    IN      A       198.35.27.27

Received 166 bytes from 127.0.0.53#53 in 536 ms
```

Se consigue.

Ataque Man in The Middle (MITM)

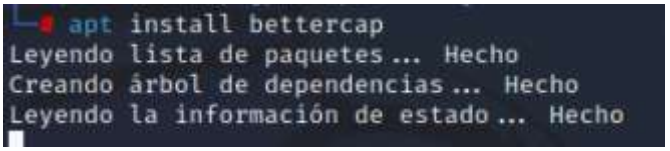
En este escenario, intentamos posicionarnos entre el servidor de correos y sus usuarios para interceptar y, potencialmente, modificar los

datos transmitidos. Este tipo de ataque nos permite evaluar la seguridad de las comunicaciones, incluyendo la efectividad del cifrado y la autenticación de las conexiones. La simulación buscará exponer cualquier fallo en el cifrado o en la validación de certificados que podría permitir a un atacante manipular o espiar las comunicaciones.

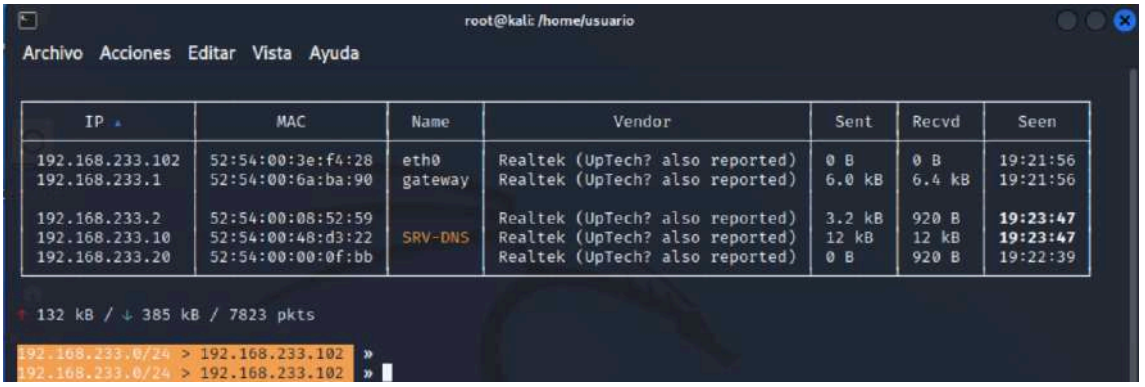
Realización del ataque:

La herramienta que utilizaremos para llevarlo a cabo será Bettercap.

Lo primero que haremos será descargar bettercap, que por defecto no está en el Kali Linux.

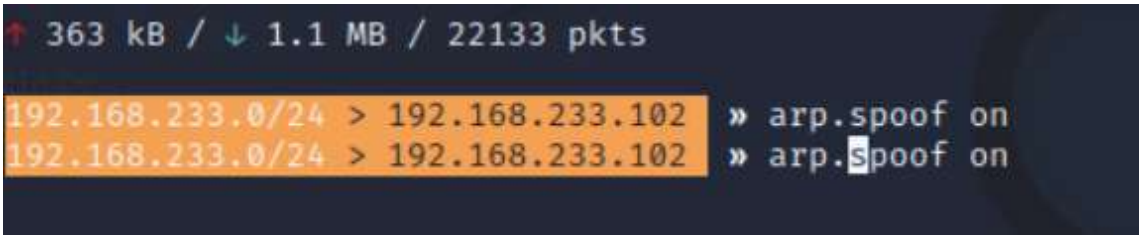


Haciendo un *net.probe* on, y un *ticker on* se logra hacer un escaneo de red, que se ve de esta manera (simplificando, es como hacer un nmap)

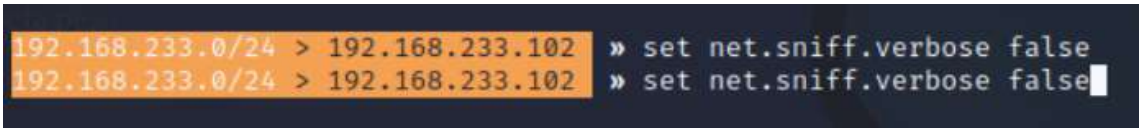


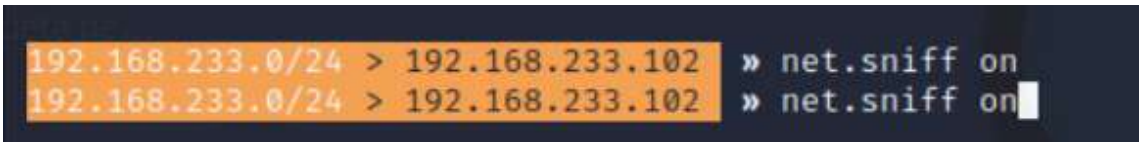
Declaramos los targets (el gateway)

Activamos el arp.spoof



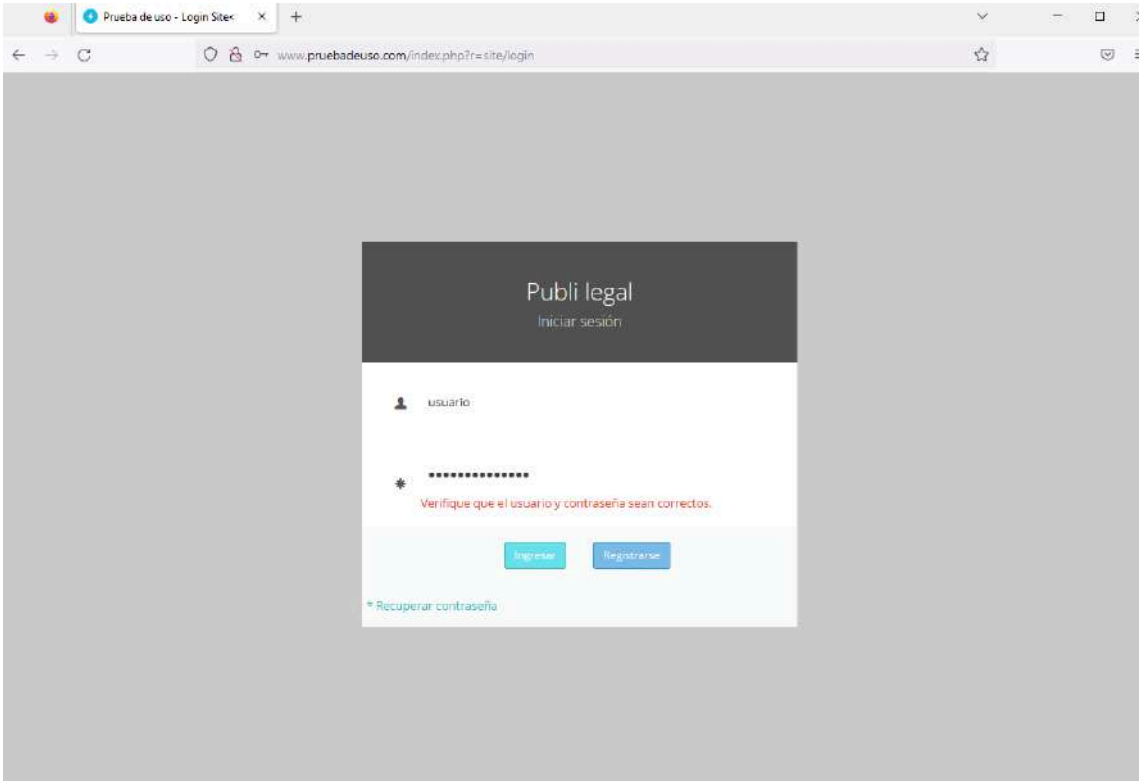
Y con estos comandos activamos el sniffeo de tráfico



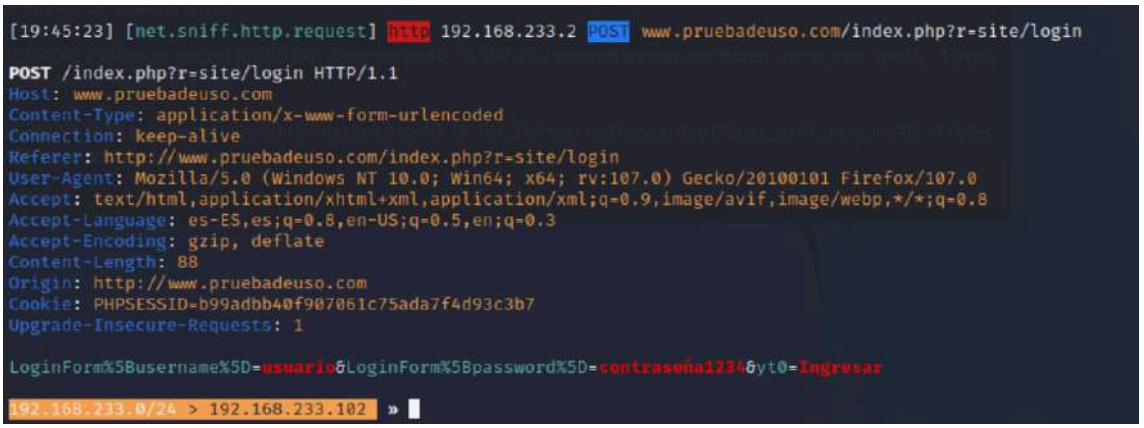


Con esto lo que hacemos es cambiar la tabla arp para poder hacerte pasar como router y ver todos los paquetes que van a pasar por el router, pudiendo ver, por ejemplo, usuarios y contraseñas en páginas no muy seguras.

Ahora, imaginemos que un cliente decide loguearse en un servicio mail poco seguro (que utiliza http).



Ahora, en bettercap vamos a ver qué ha sucedido



Podemos apreciar el usuario y contraseña que ha ingresado el cliente en la página web (en rojo la última línea)

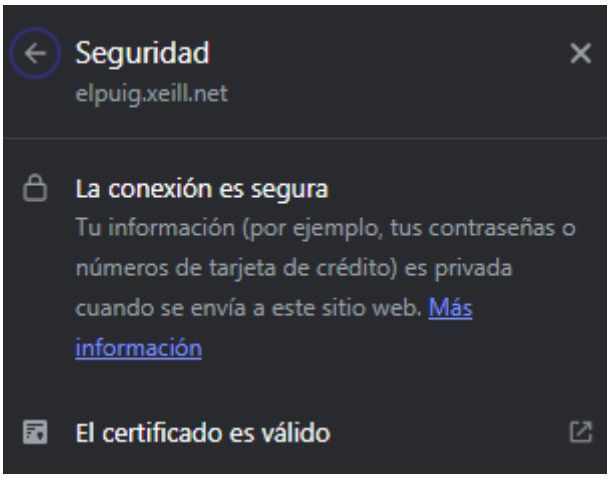
Con esto ya tenemos el Man In The Middle, que explicado así es muy simple,
Lo que hace es cambiar la tabla arp de la víctima, haciendo pasar al hacker como router, viendo así todo los paquetes antes de que lleguen al propio router, con eso puede ver contraseñas y usuarios con facilidad.

Video del ataque:

https://youtu.be/6Z2iDJEmvus

Solucion:

Para solucionar este ataque, lo que haremos es una cosa bastante sencilla, y es utilizar siempre conexiones con cifrado, en este caso en vez de logearnos vía http, debemos utilizar https, para que así la conexión vaya cifrada y nadie que nos espíe pueda ver nuestras contraseñas.



Con HTTPS ya no tendríamos ese problema.

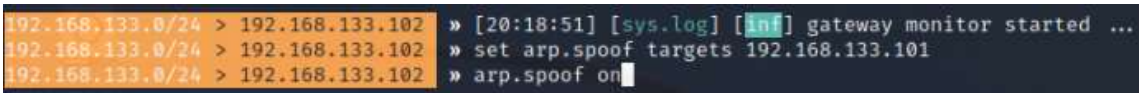
Ataque DNS Spoofing

El ataque DNS Spoofing implica que el atacante introduce información falsa en la caché de un servidor DNS. Cuando los usuarios consultan para obtener la dirección IP de un dominio, reciben una dirección IP incorrecta proporcionada por el atacante. Esto redirige a los usuarios a sitios maliciosos en lugar de los sitios legítimos que pretendían visitar. Este ataque explota la confianza en la infraestructura DNS, permitiendo al atacante desviar el tráfico web sin que los usuarios lo detecten.

Realización del ataque:

Ahora vamos a hacer un DNS SPOOFING a una máquina en concreto (192.168.133.101)

Vamos a usar otra vez bettercap, pero ahora de target vamos a poner a la máquina víctima en concreto en vez de al gateway, y activamos el spoof



Podemos ver que ha funcionado viendo que en la máquina cliente se repite la MAC de dos ip diferentes
(la del gateway 192.168.133.1 y la de abajo que es la kali 192.168.133.102 tienen la misma MAC)

```
usuario@cliente-2:~$ arp -a
? (192.168.133.100) en 52:54:00:60:67:83 [ether] en enp1s0
_gateway (192.168.133.1) en 52:54:00:15:c2:4f [ether] en enp1s0
? (192.168.133.102) en 52:54:00:15:c2:4f [ether] en enp1s0
```

Antes de nada, descargamos un apache2 y hacemos un html, para poder hacernos pasar por la web de netrna.domain

```
└─# apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

```
└─# cd /var/www/html

┌──(root㉿kali)-[/var/www/html]
└─# nano index.html
```

```
  GNU nano 6.3                                  index.html
<head><meta charset="utf-8" /></head>
<h1>Has sido hackeado!! 💀 💀 💀 💀 💀 </h1>
```

```
┌──(root㉿kali)-[/var/www/htm
└─# systemctl restart apache2
```

Una vez ya con el apache2 montado ponemos lo siguiente

```
192.168.133.0/24 > 192.168.133.102  » set dns.spoof.domains netrna.domain
192.168.133.0/24 > 192.168.133.102  » set dns.spoof.address 192.168.133.102
192.168.133.0/24 > 192.168.133.102  » dns.spoof on
```

Que básicamente lo que hace es que haga un spoof al dominio de netrna.domain, y en vez de poner la ip de ese dominio, ponga la que nosotros queramos (en este caso la de nuestra máquina hacker con apache2)

```
21:05:02] [sys.log] [inf] dns.spoof netrna.domain → 192.168.133.102
```

Así se ve en el cliente cuando busca netrna.domain

```
192.168.133.0/24 > 192.168.133.102 » [23:03:49] [sys.log] [inf] dns.spoof sending spoofed DNS reply for netrna.do
main (→192.168.133.102) to 192.168.133.100 : 52:54:00:60:67:83 (Realtek (UpTech? also reported)).
192.168.133.0/24 > 192.168.133.102 » [23:04:05] [sys.log] [inf] dns.spoof sending spoofed DNS reply for wpad.netr
na.domain (→192.168.133.102) to 192.168.133.100 : 52:54:00:60:67:83 (Realtek (UpTech? also reported)).
```

Y así se ve en la máquina hacker

Aquí se puede apreciar verdaderamente la magnitud del ataque, porque un simple html diciendo Has sido hackeado, no hace nada, pero por ejemplo, haciéndoonos pasar por paypal.com, podemos

conseguir las credenciales bancarias de la persona que esté en la máquina víctima.

Video del ataque:

https://youtu.be/Ii4JrwpTWSw

Solucion:

Para solucionar este ataque, lo que haremos será dejar la tabla arp estática, para que así no se pueda cambiar

Veamos en la tabla arp, la ip del gateway, que vamos a hacerlo permanente.

```
root@pasarela2:/home/usuario# arp -a
_gateway (192.168.233.1) en 52:54:00:6a:ba:90 [ether] en enp1s0
? (192.168.233.20) en 52:54:00:00:0f:bb [ether] en enp1s0
? (192.168.133.101) en 52:54:00:35:12:18 [ether] en enp2s0
? (192.168.233.30) en 52:54:00:45:b1:7b [ether] en enp1s0
? (192.168.233.23) en <incompleto> en enp1s0
? (192.168.133.23) en 52:54:00:7f:51:f0 [ether] en enp2s0
? (192.168.133.102) en 52:54:00:15:c2:4f [ether] en enp2s0
? (192.168.233.10) en 52:54:00:48:d3:22 [ether] en enp1s0
? (192.168.133.100) en 52:54:00:60:67:83 [ether] en enp2s0
? (192.168.233.10) en <incompleto> en enp2s0
```

Simplemente ponemos que la ip y la mac junto un arp -s, y como se aprecia abajo ya se quedará permanentemente

```
root@pasarela2:/home/usuario# arp -s 192.168.233.1 52:54:00:6a:ba:90
root@pasarela2:/home/usuario# arp -a
_gateway (192.168.233.1) en 52:54:00:6a:ba:90 [ether] PERM en enp1s0
? (192.168.233.20) en 52:54:00:00:0f:bb [ether] en enp1s0
? (192.168.133.101) en 52:54:00:35:12:18 [ether] en enp2s0
? (192.168.233.30) en 52:54:00:45:b1:7b [ether] en enp1s0
? (192.168.233.23) en <incompleto> en enp1s0
? (192.168.133.23) en 52:54:00:7f:51:f0 [ether] en enp2s0
? (192.168.133.102) en 52:54:00:15:c2:4f [ether] en enp2s0
? (192.168.233.10) en 52:54:00:48:d3:22 [ether] en enp1s0
? (192.168.133.100) en 52:54:00:60:67:83 [ether] en enp2s0
? (192.168.233.10) en <incompleto> en enp2s0
```

```
_gateway (192.168.233.1) en 52:54:00:6a:ba:90 [ether] PERM en enp1s0
```

(PERM significa que es un arp estático y permanente)

Esto va bien para hacerlo al servidor web por ejemplo, el cúal es importante que no “hackeen” porque es donde más los hosts de la red irán a visitar.

```
root@pasarela2:/home/usuario# arp -s 192.168.233.30 52:54:00:45:b1:7b
root@pasarela2:/home/usuario# arp -a
_gateway (192.168.233.1) en 52:54:00:6a:ba:90 [ether] PERM en enp1s0
? (192.168.233.20) en 52:54:00:00:0f:bb [ether] en enp1s0
? (192.168.133.101) en 52:54:00:35:12:18 [ether] en enp2s0
? (192.168.233.30) en 52:54:00:45:b1:7b [ether] PERM en enp1s0
? (192.168.233.23) en <incompleto> en enp1s0
? (192.168.133.23) en 52:54:00:7f:51:f0 [ether] en enp2s0
? (192.168.133.102) en 52:54:00:15:c2:4f [ether] en enp2s0
? (192.168.233.10) en 52:54:00:48:d3:22 [ether] en enp1s0
? (192.168.133.100) en 52:54:00:60:67:83 [ether] en enp2s0
? (192.168.233.10) en <incompleto> en enp2s0
```

Así que también la hemos hecho permanente
Y así deberíamos hacer con todas las ip de los servidores, cuyas ip sepamos que no van a cambiar, porque sino sería perder el tiempo.

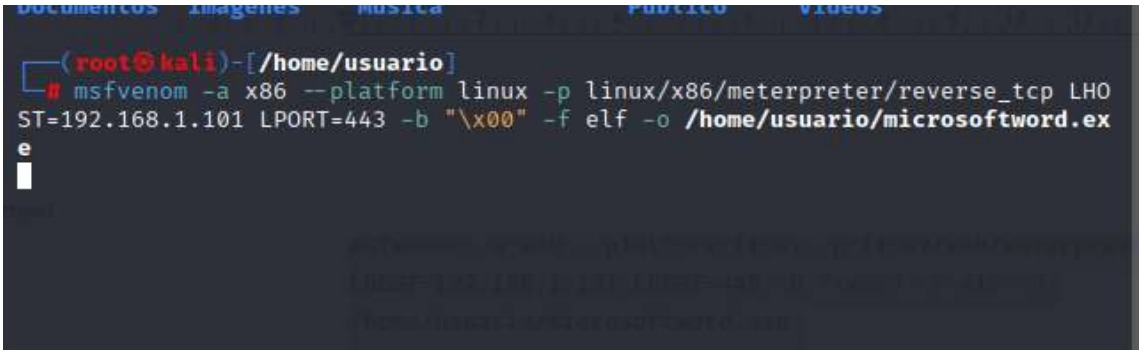
Ataque con Metasploit

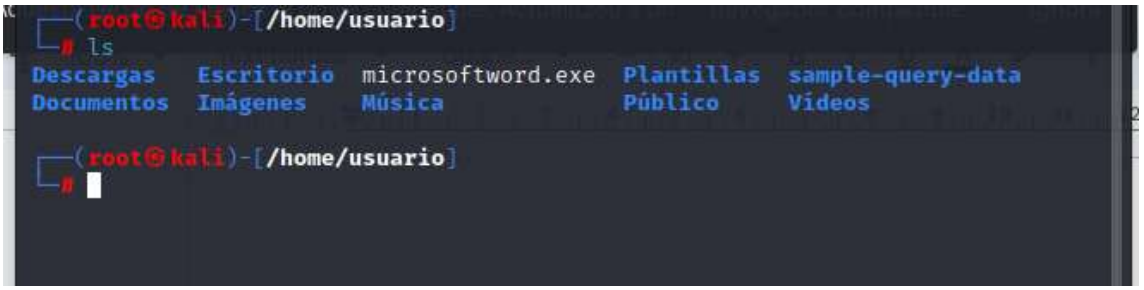
Este ataque puede dirigirse tanto a sistemas operativos Windows como Linux y consiste en establecer una conexión TCP reversa. Esto permite que un sistema comprometido inicie una conexión hacia el atacante, facilitando el control remoto del sistema afectado y evadiendo firewalls y otras medidas de seguridad. Esta técnica subraya la importancia crítica de implementar medidas de seguridad robustas y mantener todos los sistemas y aplicaciones protegidos frente a vulnerabilidades potenciales.

Realización del ataque:

1. Creación del troyano: Se utiliza `msfvenom` para generar un archivo ejecutable con un payload de conexión inversa TCP, apto tanto para sistemas Linux como Windows, que permite al atacante controlar remotamente la máquina infectada.

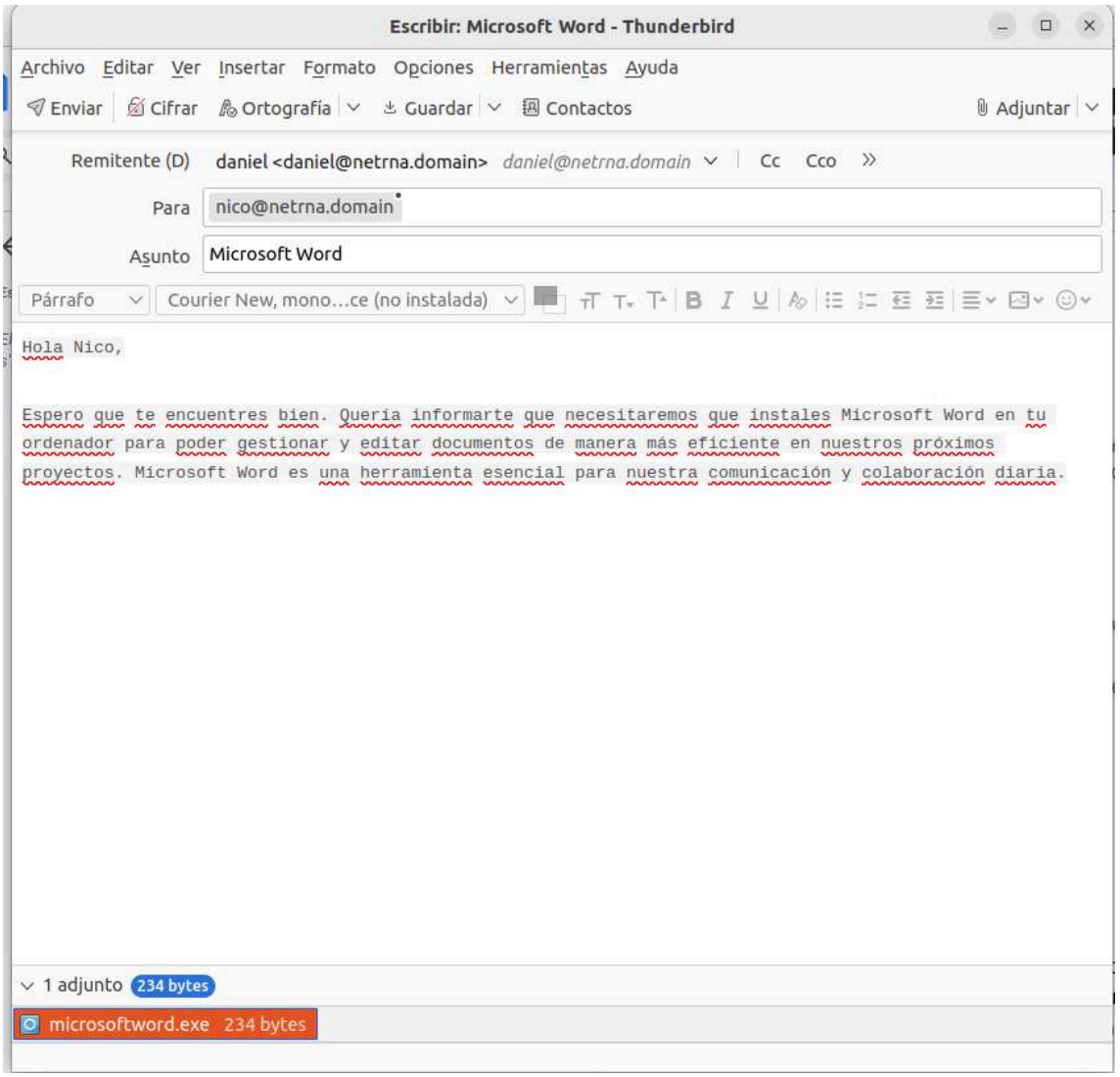
Dentro del archivo ejecutable proporcionamos nuestra ip y puerto (Máquina Atacante)



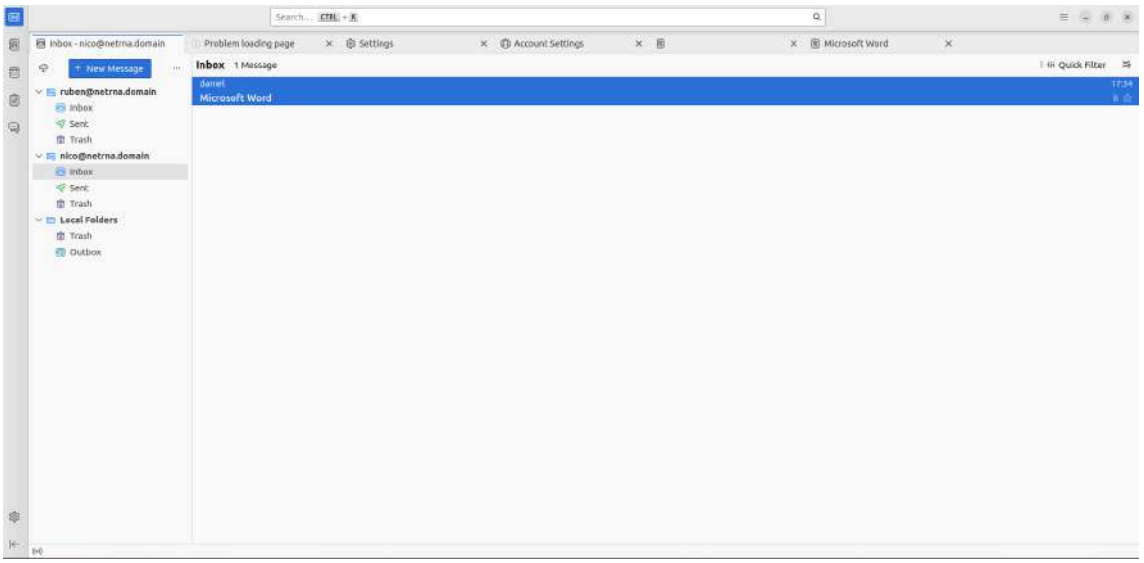


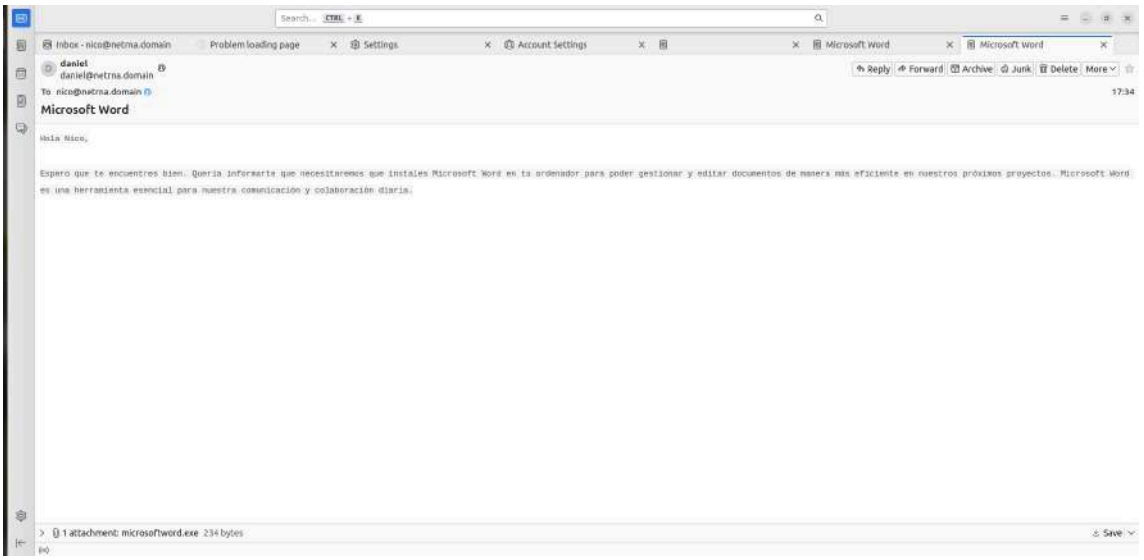
2. Envío del archivo a la víctima: El archivo malicioso se transfiere a la máquina objetivo.

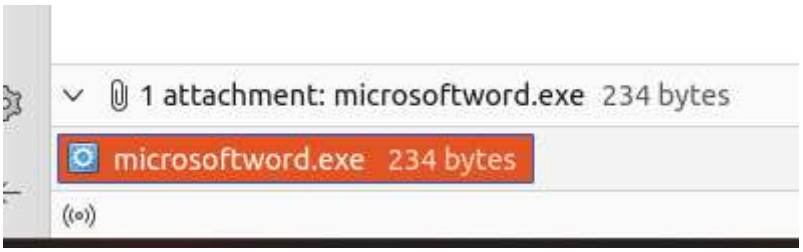
(Atacante)


Escribir: Microsoft Word - Thunderbird
Archivo Editar Ver Insertar Formato Opciones Herramientas Ayuda
Enviar Cifrar Ortografía Guardar Contactos Adjuntar
Remitente (D) daniel <daniel@netrna.domain> daniel@netrna.domain Cc Cco
Para nico@netrna.domain
Asunto Microsoft Word
Párrafo Courier New, mono...ce (no instalada)
Hola Nico,
Espero que te encuentres bien. Quería informarte que necesitaremos que instales Microsoft Word en tu ordenador para poder gestionar y editar documentos de manera más eficiente en nuestros próximos proyectos. Microsoft Word es una herramienta esencial para nuestra comunicación y colaboración diaria.
1 adjunto 234 bytes
microsoftword.exe 234 bytes

(Cliente)

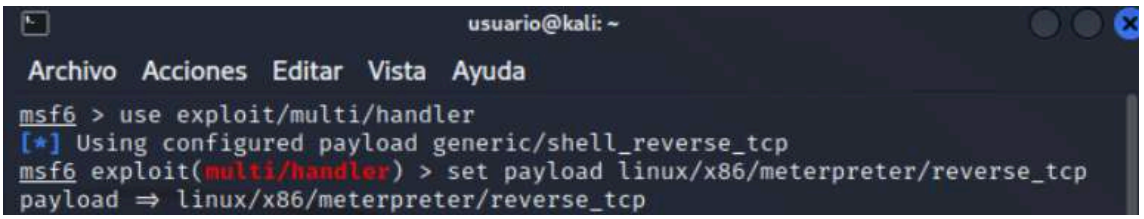


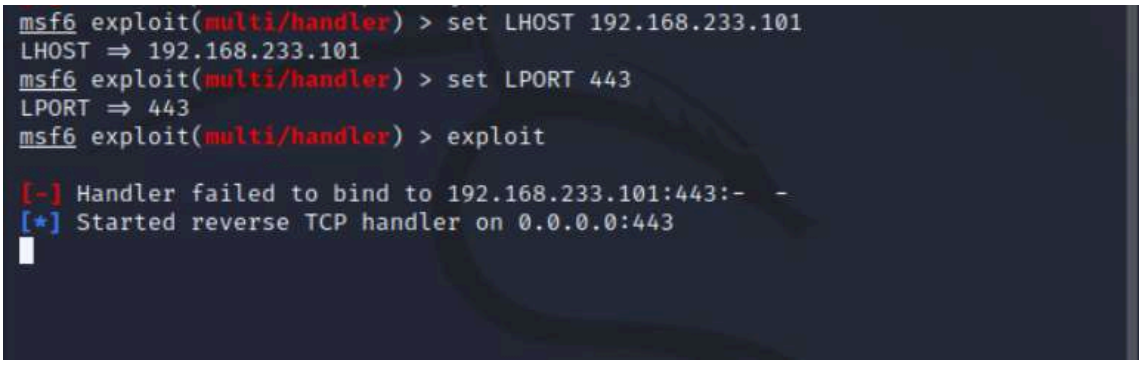




Una vez generado se lo tenemos que enviar a nuestra víctima, en este caso vamos a probar a enviarlo mediante un correo diciendo que se lo descargue , así que de ese modo cuando se descargue y se ejecute, estará esperando a la conexión del atacante para poder acceder a la máquina.

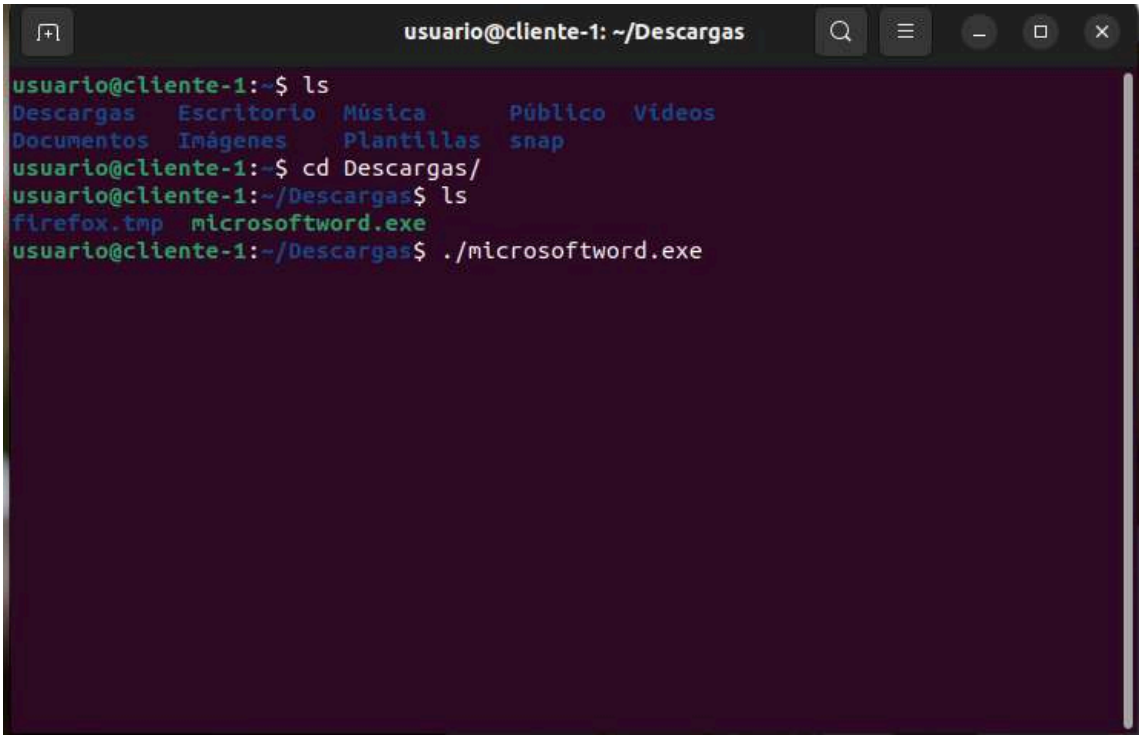
3. Configuración y activación del listener: Se prepara un listener en Metasploit (`msfconsole`), configurado para esperar conexiones en la dirección IP y puerto definidos, acordes al payload del troyano.



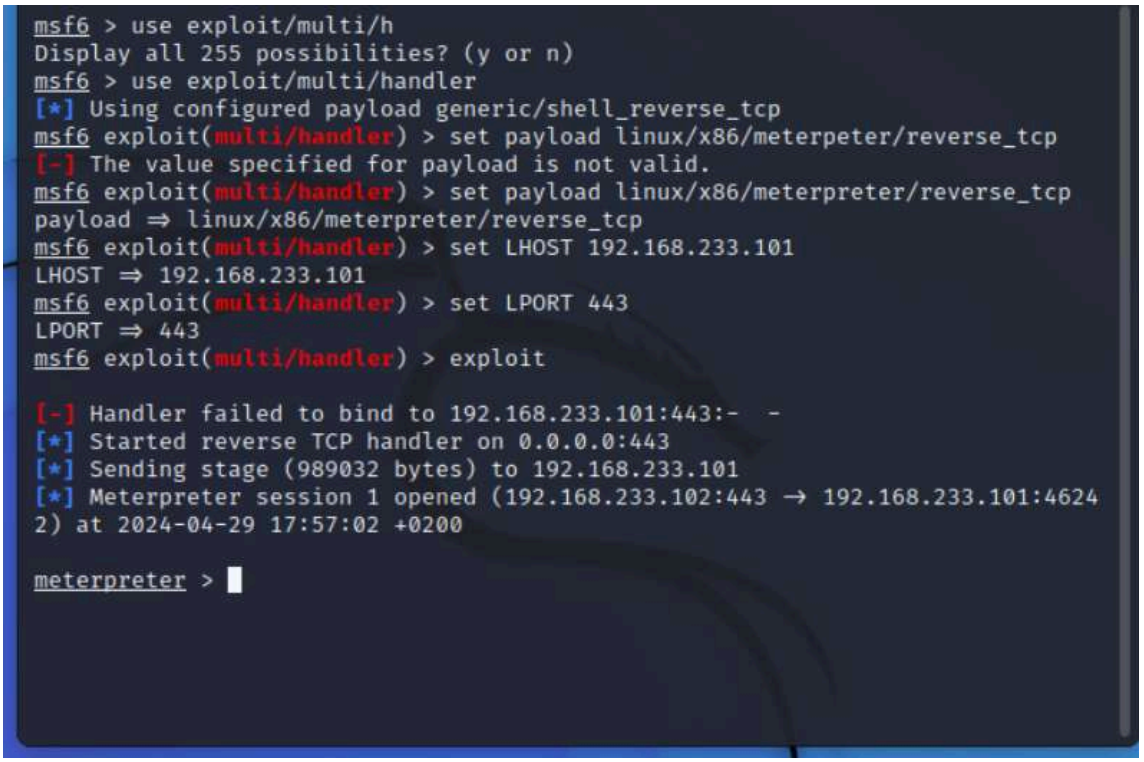


4. Control de la máquina víctima: Una vez ejecutado el troyano en la máquina objetivo, se establece la conexión con el listener, proporcionando al atacante control remoto del sistema.

(Cliente)



(Atacante)

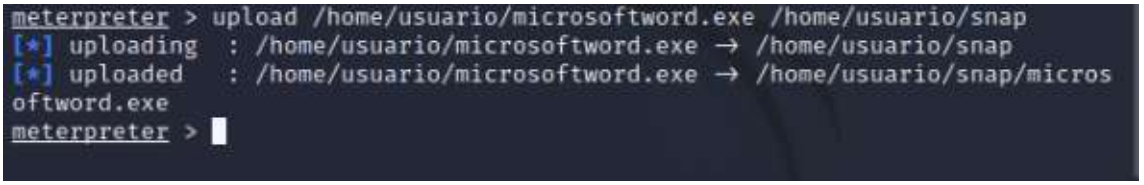


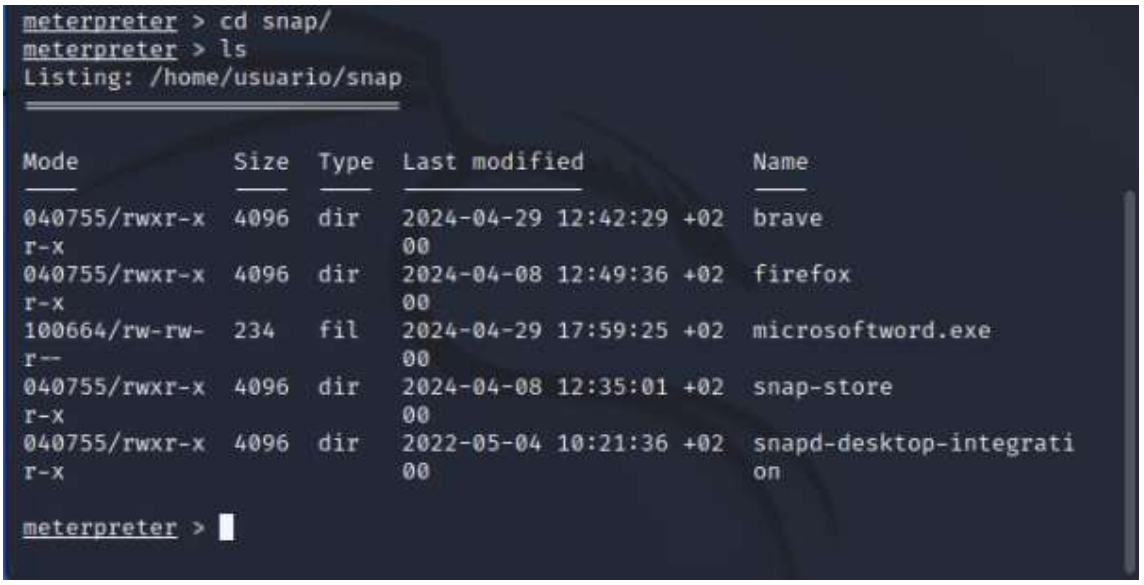
Ya tenemos control del sistema.

5. Reubicación del troyano para evitar detección: Si se detecta que el archivo original no ejecuta acciones visibles o se sospecha que puede ser descubierto, se sube de nuevo pero a una ubicación diferente donde sea menos probable que el usuario lo encuentre.

(Atacante)

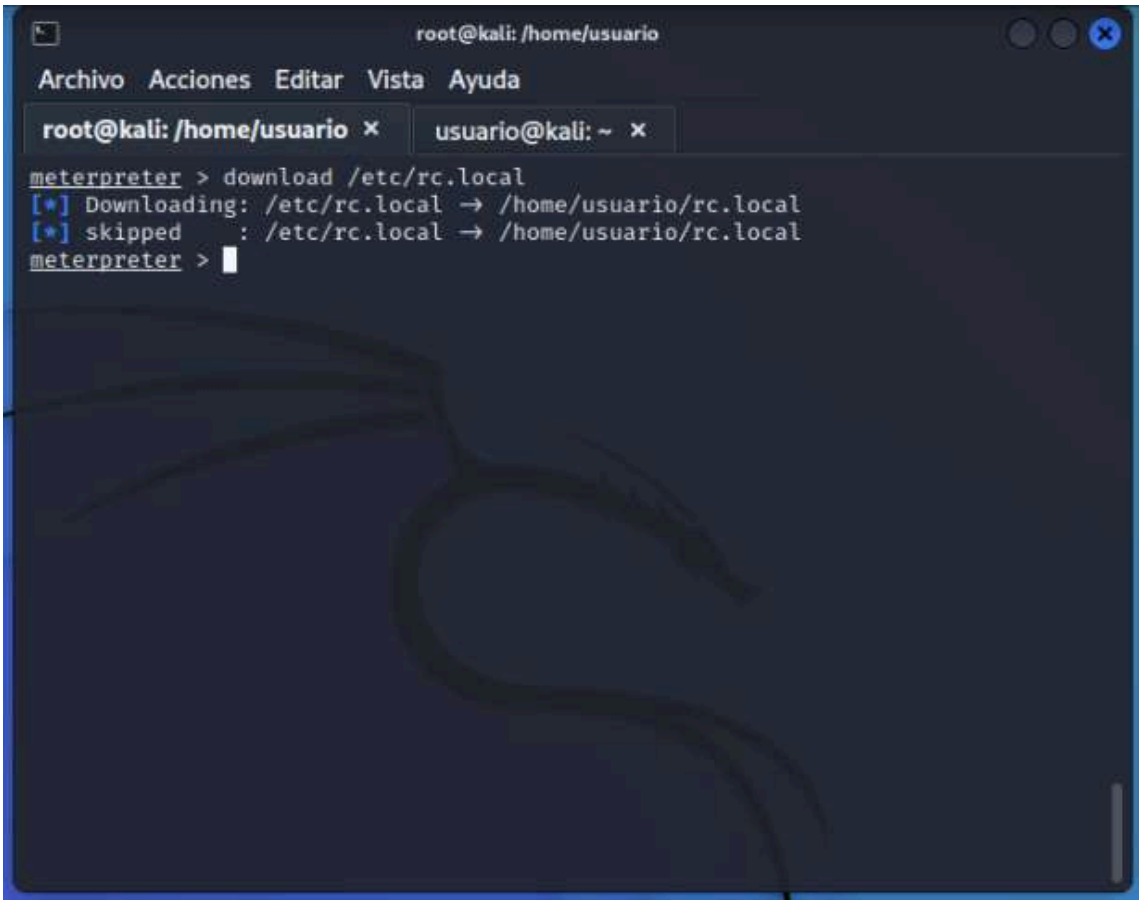
Lo subimos otra vez el fichero y lo subimos en *"/home/usuario/snap"*



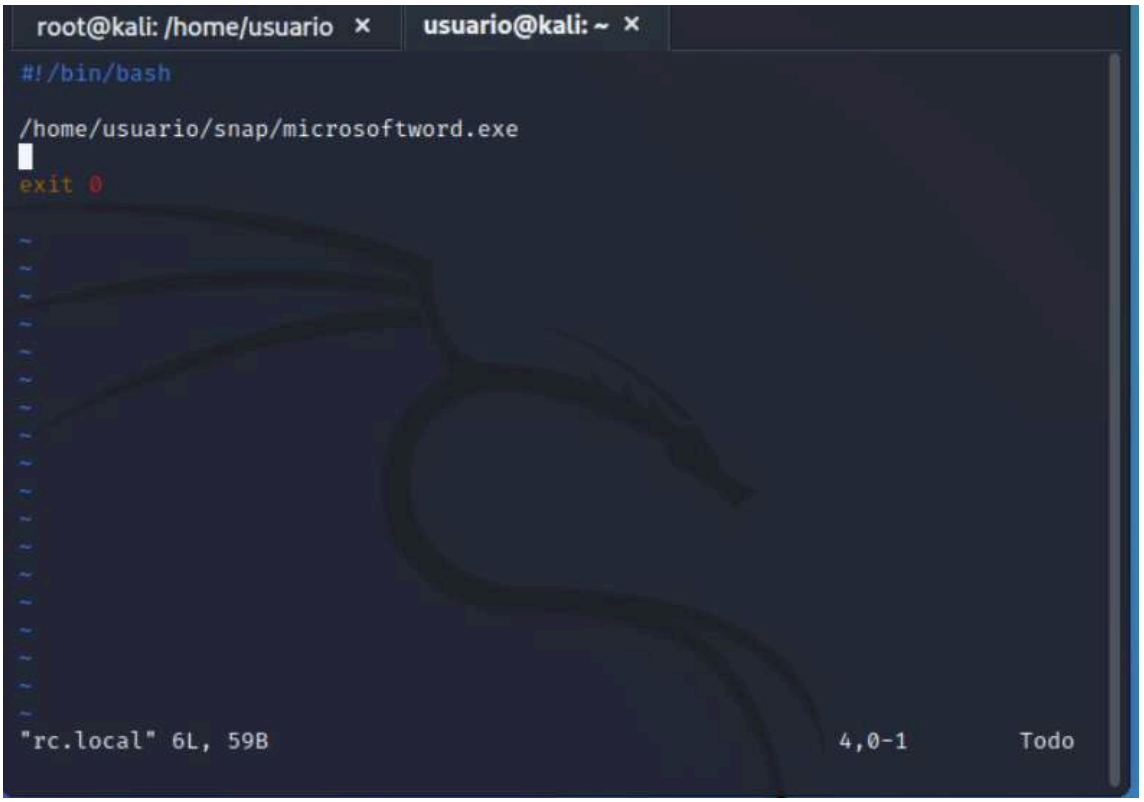


6. Modificación para persistencia: Para asegurar que el troyano se ejecute automáticamente en cada inicio del sistema, se descarga el archivo `/etc/rc.local` de la máquina víctima, se modifica añadiendo la ejecución del troyano, y se vuelve a subir. Esto garantiza la activación automática del troyano, manteniendo el acceso remoto de manera persistente.

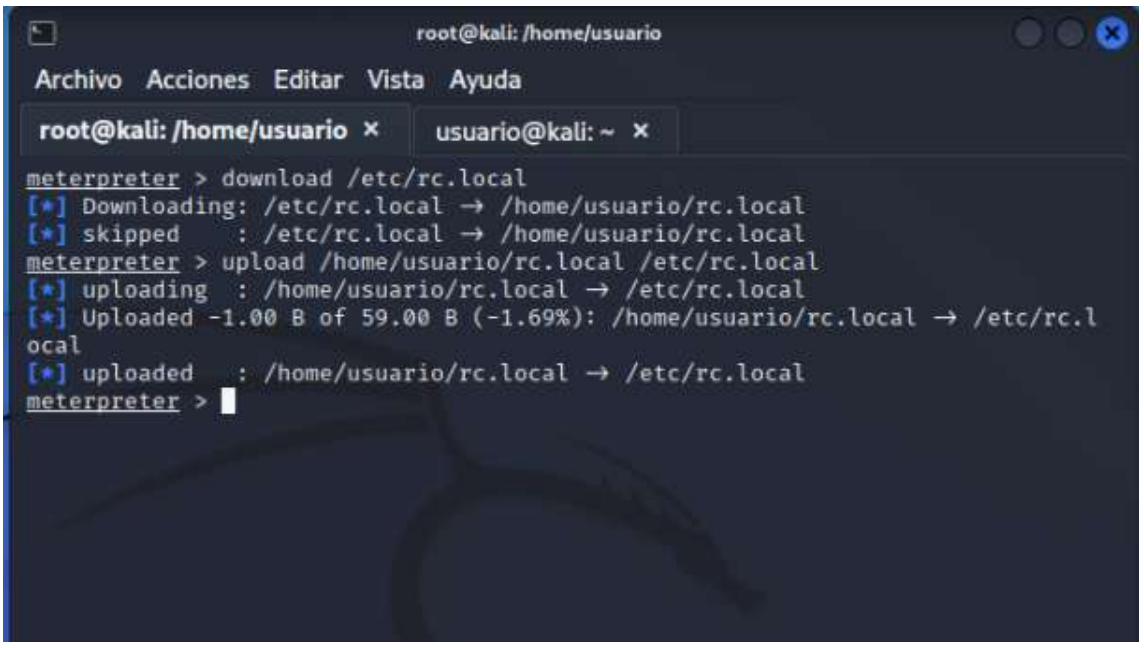
Nos descargamos el archivo “/etc/rc.local” de la máquina víctima



Lo modificamos para que inicie al iniciar el ordenador automáticamente con la siguiente configuración

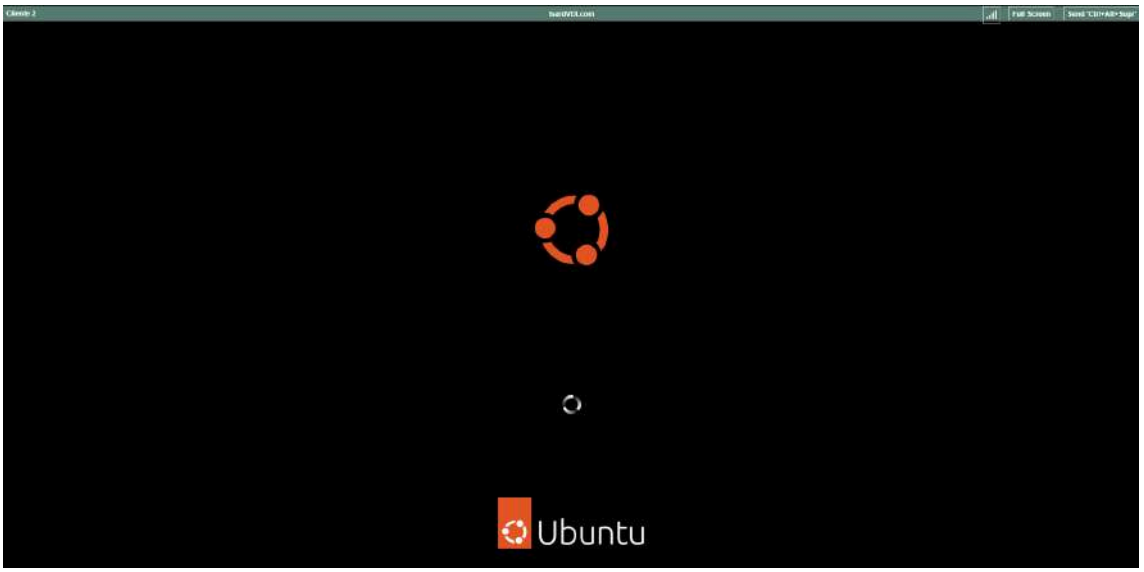


Posteriormente lo guardamos y lo subimos a la máquina objetivo:

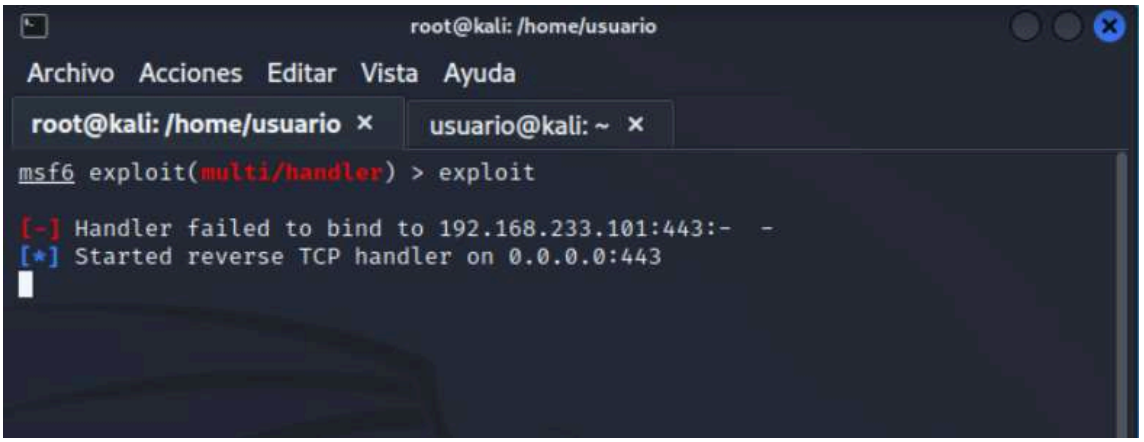


Ahora reiniciamos la maquina objetivo y podremos tener acceso de manera persistente

(Cliente)

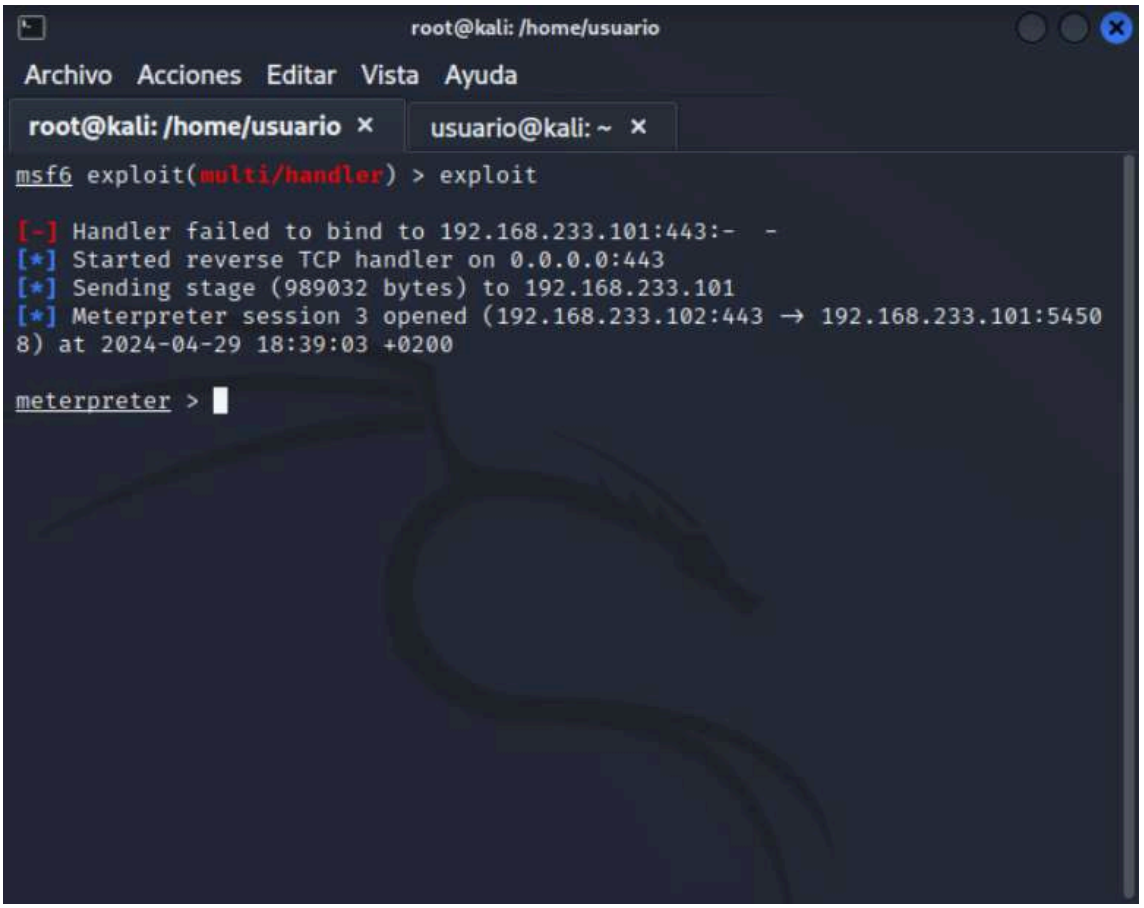


(Atacante esperando con el listener)



(Cliente al iniciar)

(Atacante con la sesión ya establecida)



Video del ataque:

https://www.youtube.com/watch?v=CBp-nheAV1I

Solucion:

Para abordar este problema, es fundamental tomar varias medidas preventivas. En primer lugar, se recomienda actualizar el sistema a la versión más reciente, que incluya los últimos parches de seguridad disponibles. Estas actualizaciones suelen abordar vulnerabilidades conocidas y proporcionar defensas adicionales contra amenazas emergentes.

Además, es crucial implementar un software antivirus confiable en todos los dispositivos de la red. Un buen software antivirus puede detectar y eliminar malware, incluidos los tipos utilizados en estafas en

línea. Esto actúa como una capa adicional de protección, ayudando a prevenir la instalación de programas maliciosos y a mantener seguros los datos y sistemas de la red.

Por último, pero no menos importante, es importante proporcionar formación y concienciación a los usuarios de la red. Los empleados deben ser educados sobre las tácticas comunes utilizadas en estafas en línea, como el phishing y la ingeniería social. Con una comprensión sólida de cómo identificar y evitar estas amenazas, los usuarios pueden ser menos propensos a caer en trampas y comprometer la seguridad de la red.

En resumen, una combinación de actualizaciones de software, software antivirus y formación del usuario puede ayudar a proteger eficazmente contra este tipo de estafas en línea y mantener la integridad y seguridad de la red

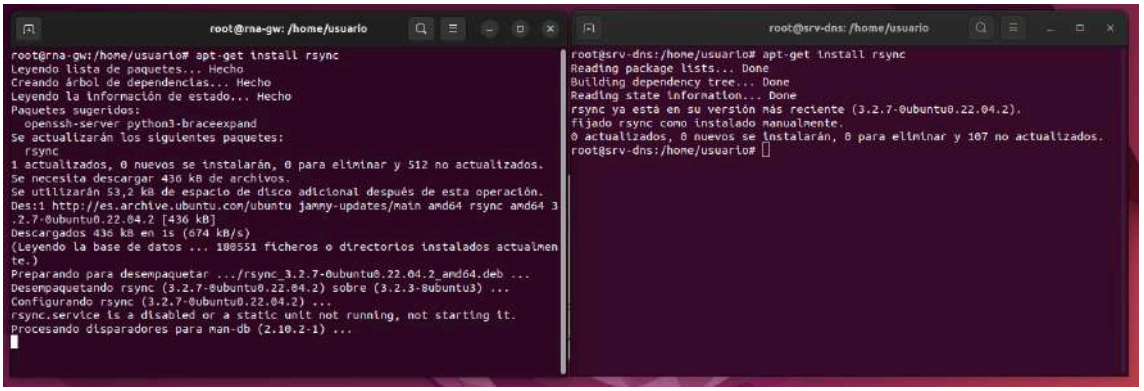
Implementación de software para realizar copias de seguridad

Vamos a usar una herramienta llamada rsync para hacer copias de seguridad de dos servicios importantes: el DNS y el gateway. Lo haremos enviando los datos a un servidor Ubuntu Desktop preparado especialmente para esto. Después, configuraremos todo para que se haga una copia de seguridad automática que se guardará en Google Drive usando una aplicación llamada Déjà Dup. Esto nos ayudará a mantener segura nuestra información de forma sencilla y automática.

Primero, instalaremos rsync en los dispositivos de los cuales queremos hacer copias de seguridad. Luego, configuraremos estos dispositivos

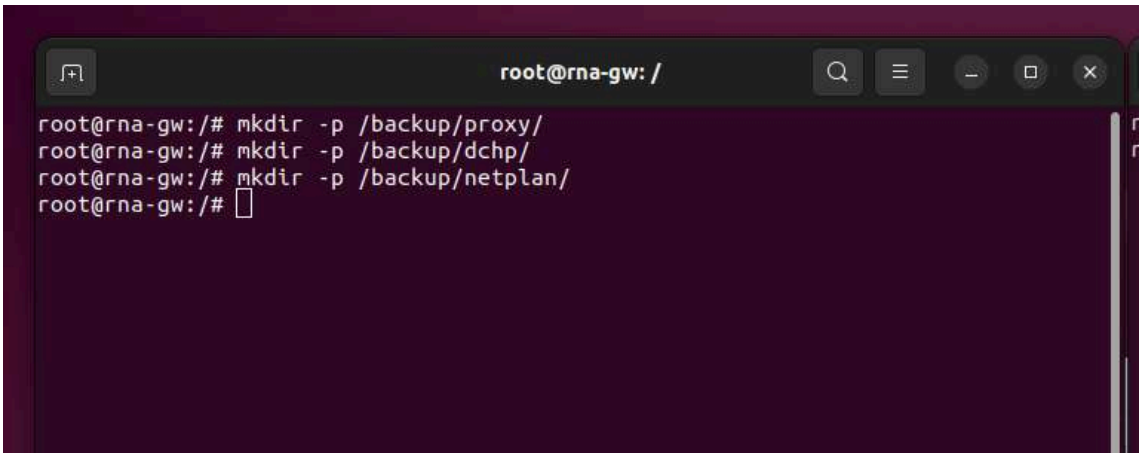
para que compartan los archivos a un directorio local. A continuación, usaremos Samba para compartir ese directorio con el servidor de backups. Finalmente, desde este servidor, haremos una copia de seguridad que se subirá automáticamente a Google Drive. Así tendremos todo organizado y seguro.

Instalacion de rsync:

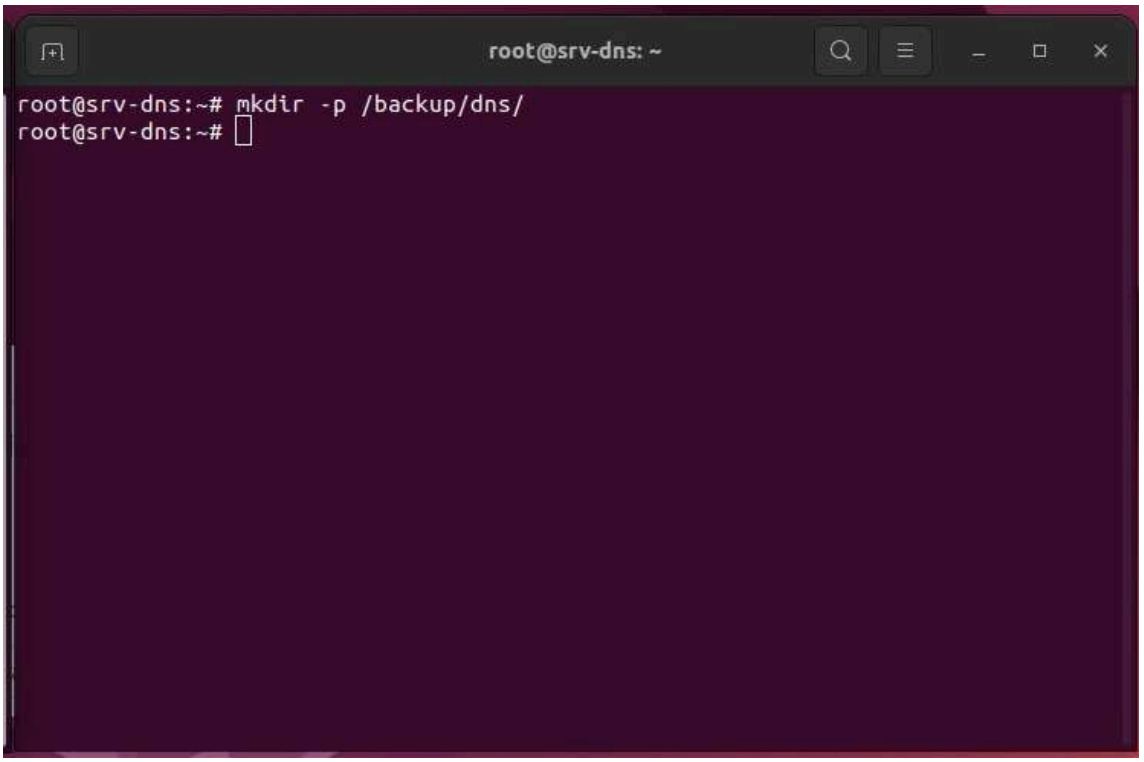


Una vez hecho esto, crearemos los directorios respectivos donde almacenaremos las copias de seguridad.

GATEWAY:



DNS:



Una vez creados los directorios, procederemos a realizar las copias de seguridad de nuestros archivos de configuración en el directorio establecido.

En el caso del gateway, realizaremos copias de seguridad de los archivos de configuración del netplan, del proxy y del DHCP. Para el DNS, nos centraremos en hacer una copia de seguridad de los archivos de configuración propios del DNS.

GATEWAY:

(netplan)

```
root@rna-gw:/home/usuario# rsync -av /etc/netplan/01-network-manager-all.yaml /b
ackup/netplan/
sending incremental file list
01-network-manager-all.yaml

sent 289 bytes  received 35 bytes  648,00 bytes/sec
total size is 165  speedup is 0,51
root@rna-gw:/home/usuario#
```

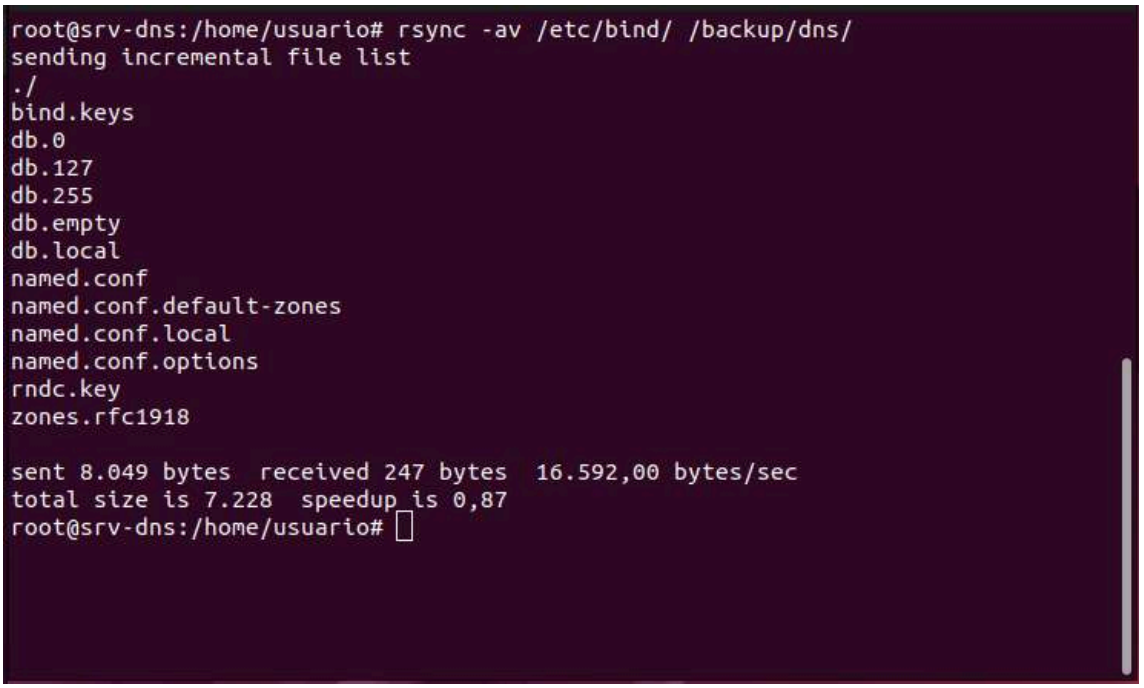
(dhcp)

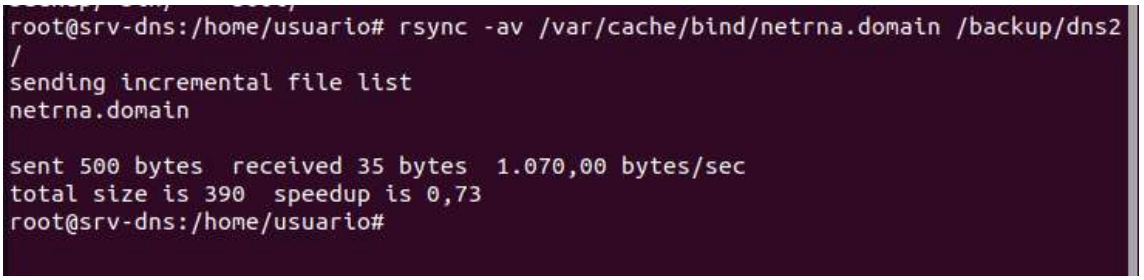
```
root@rna-gw:/home/usuario# rsync -av /etc/kea/kea-dhcp4.conf /backup/dhcp/
sending incremental file list
kea-dhcp4.conf

sent 1.510 bytes  received 35 bytes  3.090,00 bytes/sec
total size is 1.400  speedup is 0,91
root@rna-gw:/home/usuario#
```

(proxy)

```
root@rna-gw:/home/usuario# rsync -av /etc/squid/ /backup/proxy/
sending incremental file list
./
ad_block.txt
blocked_domains.txt
errorpage.css
squid.conf
conf.d/
conf.d/debian.conf

sent 439.114 bytes  received 122 bytes  878.472,00 bytes/sec
total size is 438.574  speedup is 1,00
root@rna-gw:/home/usuario#
```

DNS:

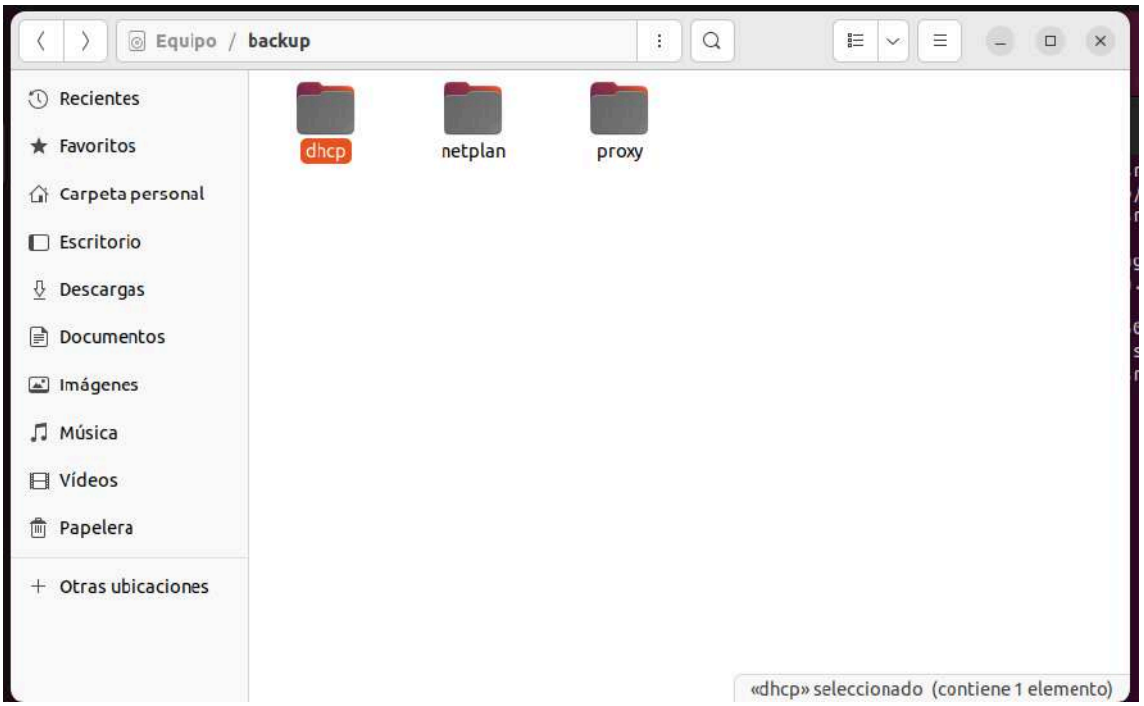


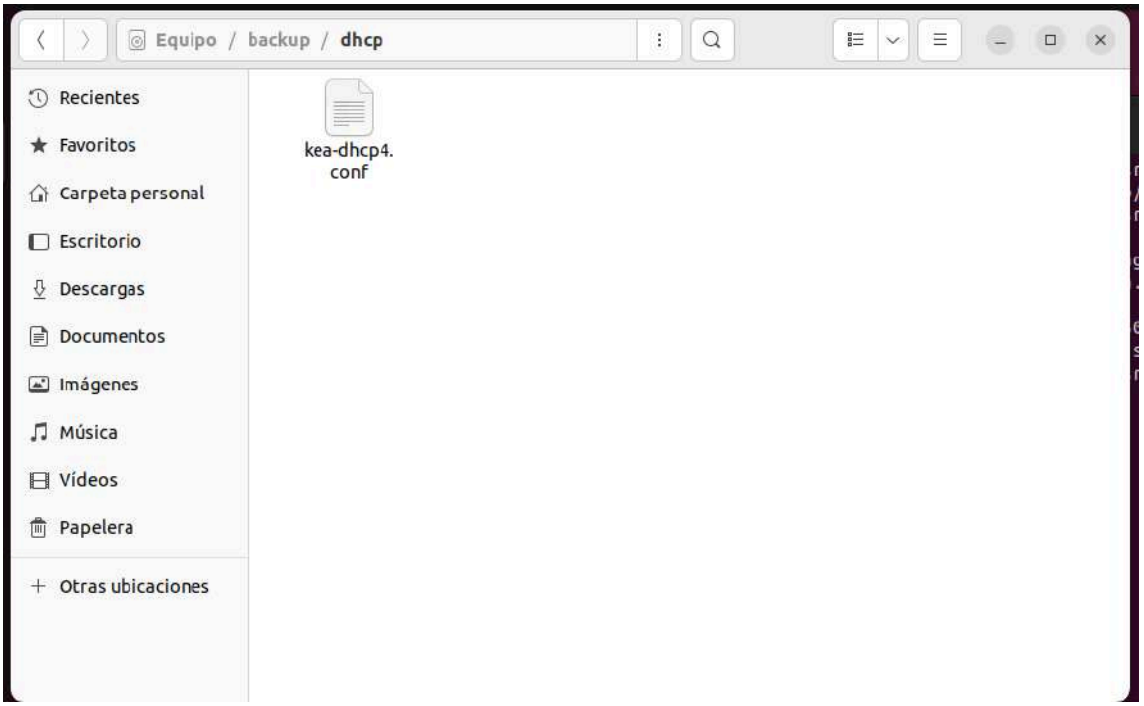


Después de realizar estos pasos, podremos comprobar que los archivos de configuración han sido correctamente copiados al directorio destinado para las copias de seguridad.

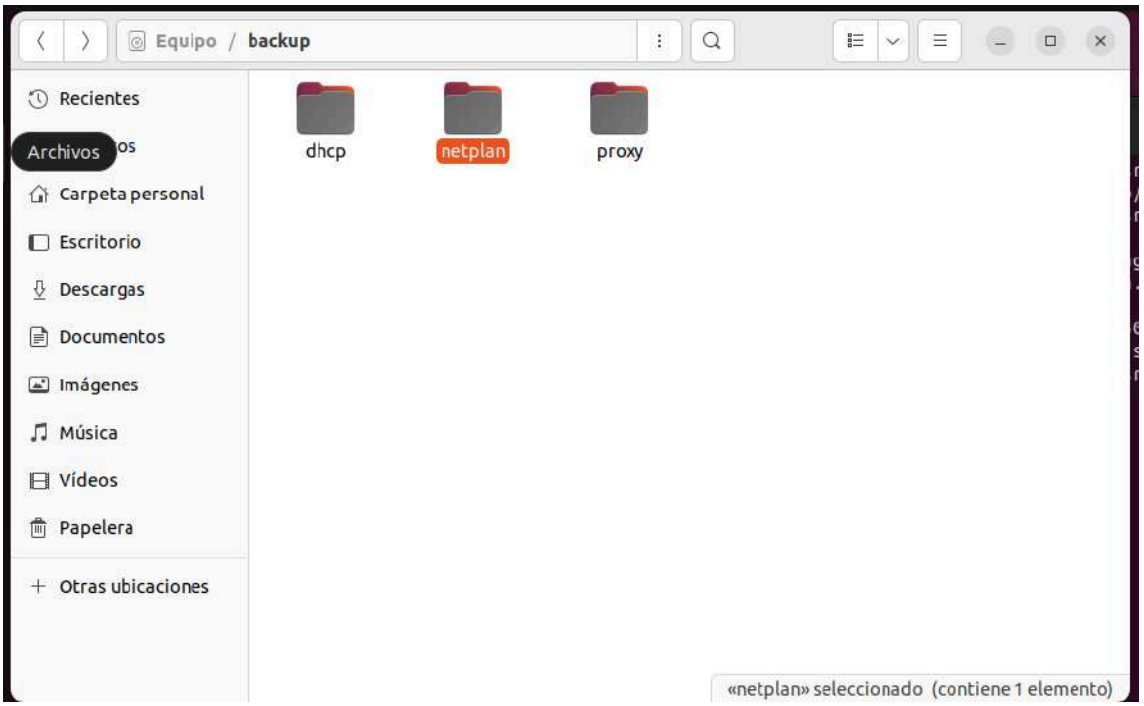
GATEWAY:

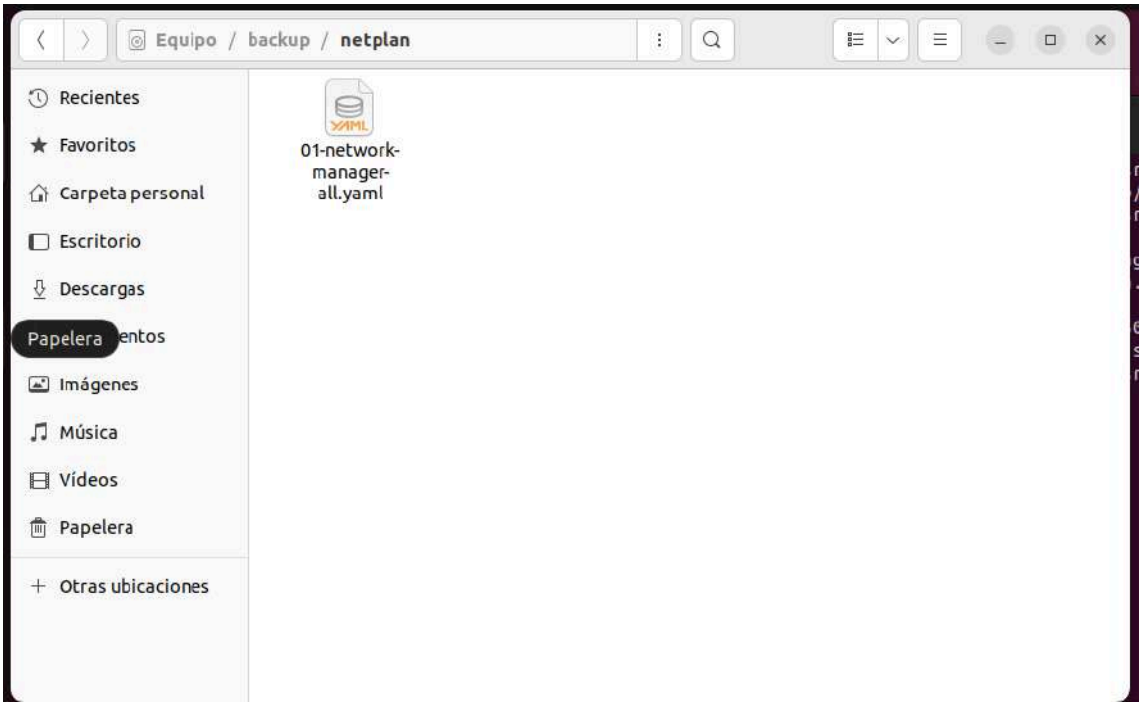
(dhcp)


Equipo / backup
Recientes
Favoritos
Carpeta personal
Escritorio
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
dhcp
netplan
proxy
«dhcp» seleccionado (contiene 1 elemento)

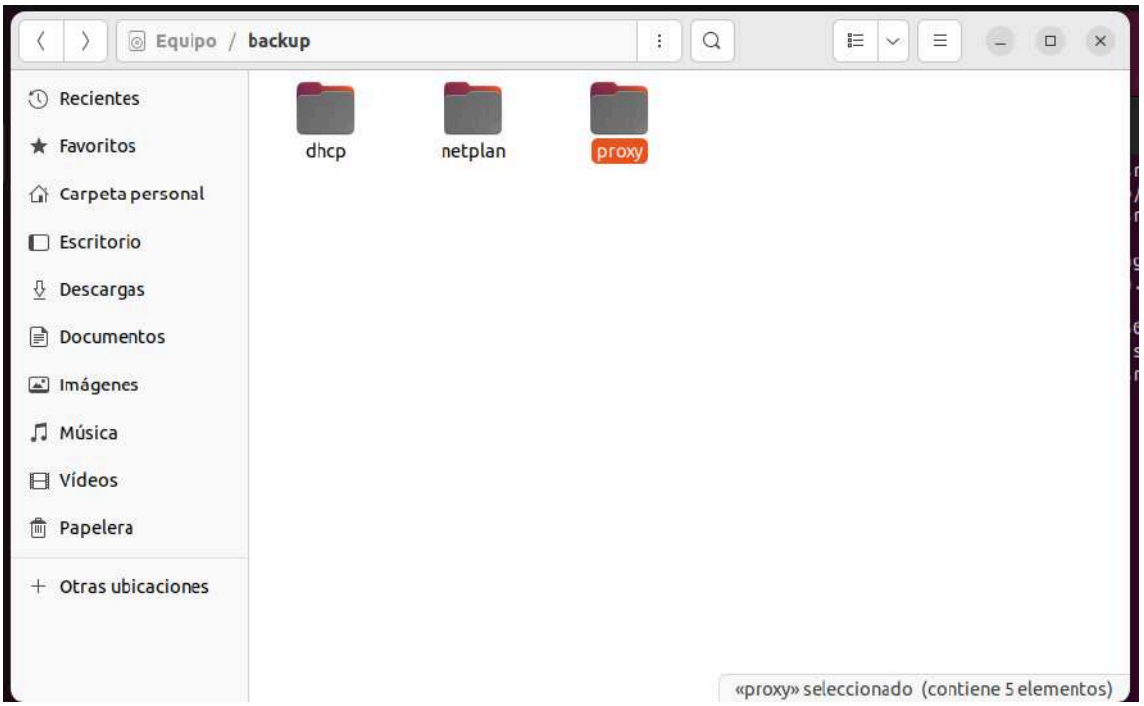

Equipo / backup / dhcp
Recientes
Favoritos
Carpeta personal
Escritorio
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
kea-dhcp4.
conf

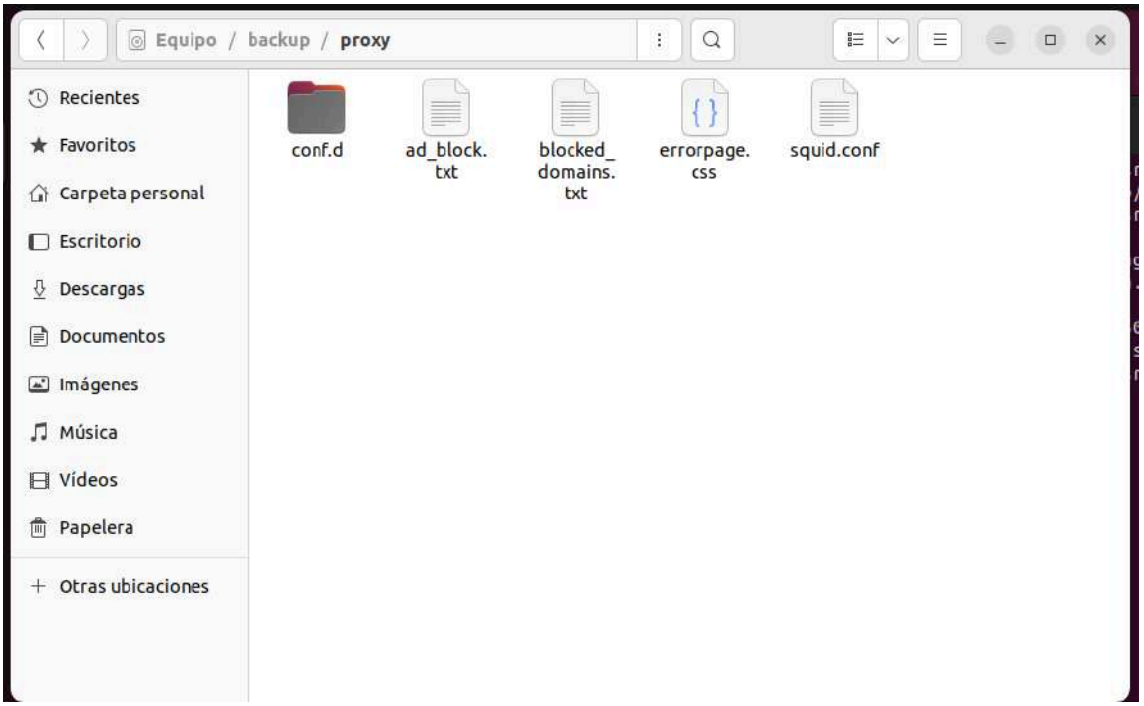
(netplan)


Equipo / backup
Recientes
Archivos
Carpeta personal
Escritorio
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
dhcp
netplan
proxy
«netplan» seleccionado (contiene 1 elemento)


Equipo / backup / netplan
Recientes
Favoritos
Carpeta personal
Escritorio
Descargas
Papelera
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
01-network-manager-all.yaml

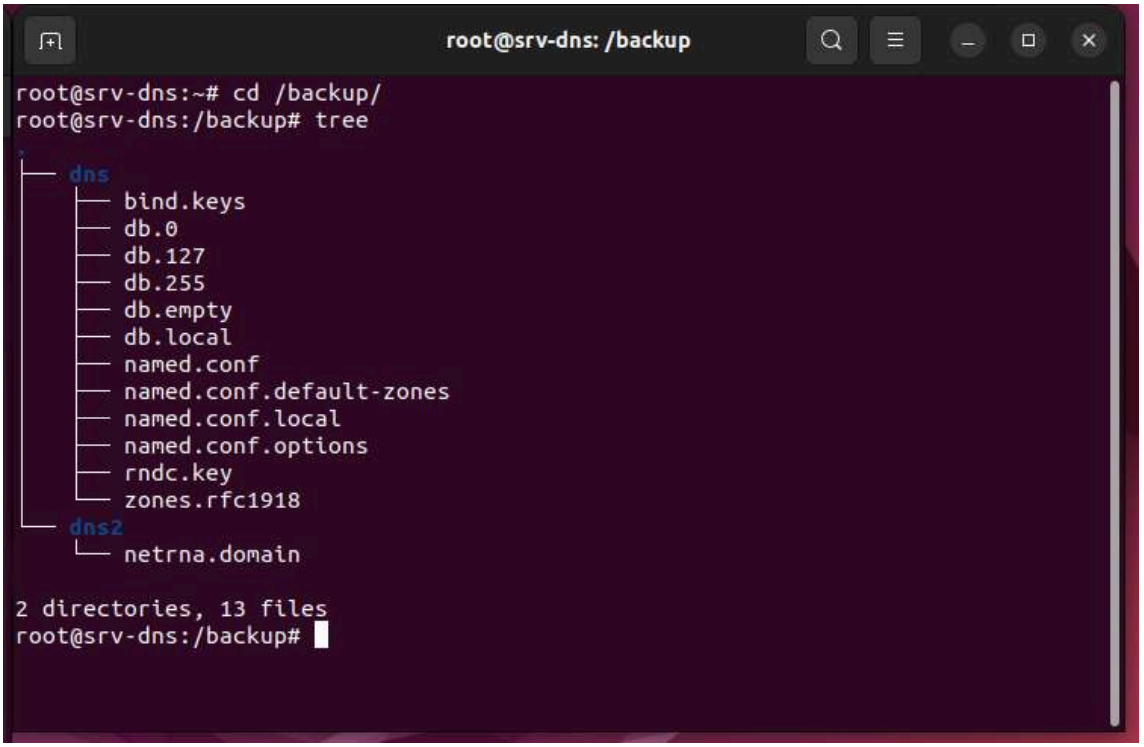
(proxy)


Equipo / backup
Recientes
Favoritos
Carpeta personal
Escritorio
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
dhcp
netplan
proxy
«proxy» seleccionado (contiene 5 elementos)


Equipo / backup / proxy
Recientes
Favoritos
Carpeta personal
Escritorio
Descargas
Documentos
Imágenes
Música
Vídeos
Papelera
Otras ubicaciones
conf.d
ad_block.txt
blocked_domains.txt
errorpage.css
squid.conf

DNS:

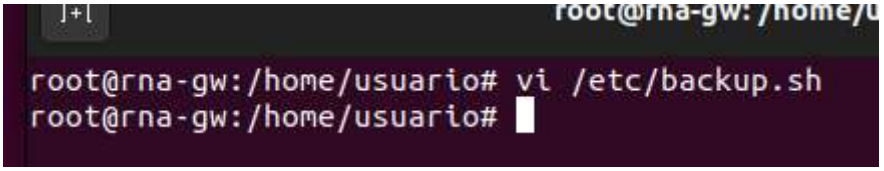
(no se puede mostrar en interfaz gráfica debido a que es un servidor.)



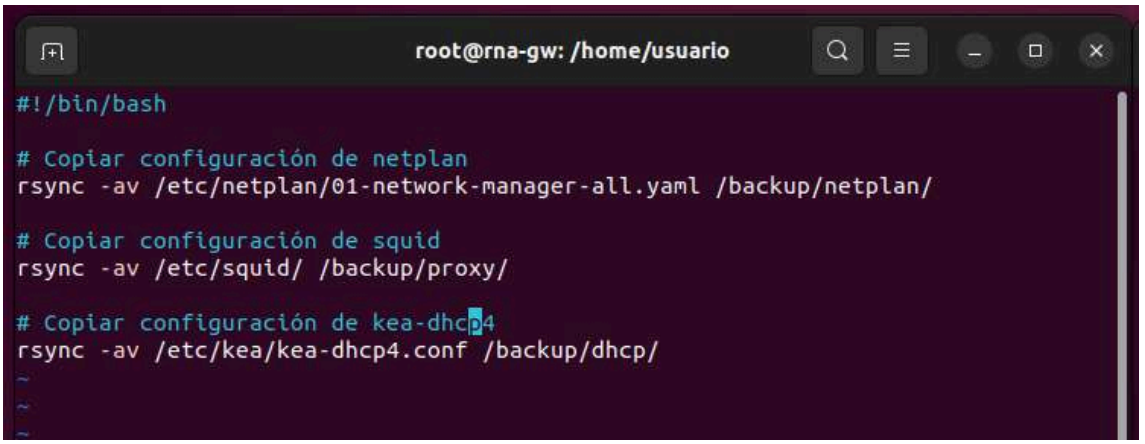
Una vez que hemos verificado que los directorios están correctamente creados y que la copia de los archivos funciona, procederemos a automatizar esta tarea utilizando crontab. Programaremos crontab para que realice las copias de seguridad automáticamente cada día. Con esto, aseguraremos que nuestros archivos de configuración siempre estén actualizados y seguros en el directorio de copias de seguridad sin necesidad de intervención manual.

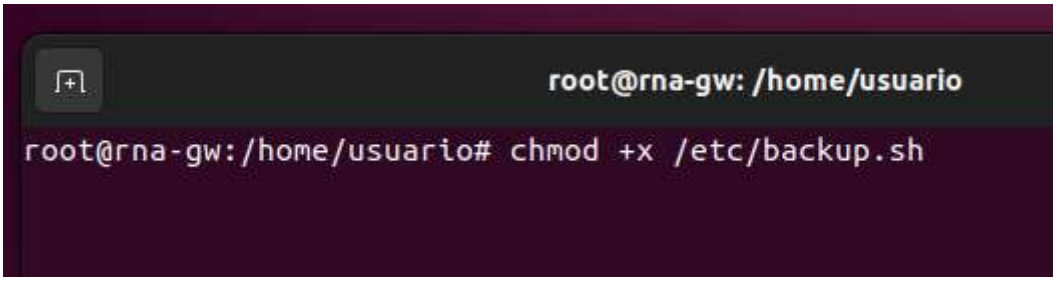
Vamos a realizar un ejemplo en la máquina GATEWAY.

Primero, creamos un archivo de script:

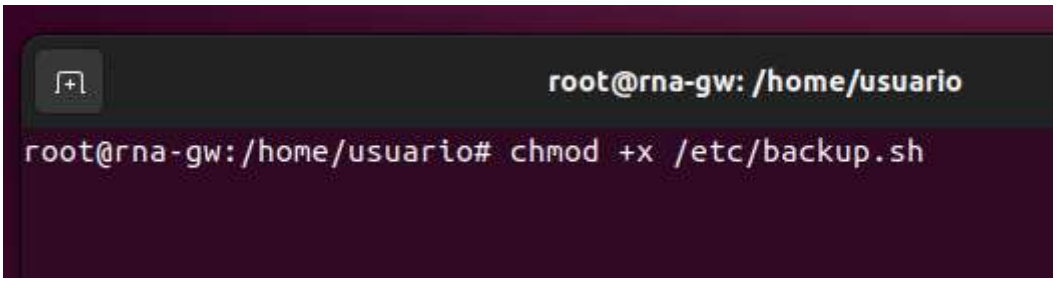


Creamos un archivo llamado backup.sh. Abrimos un editor de texto y escribimos el siguiente contenido:

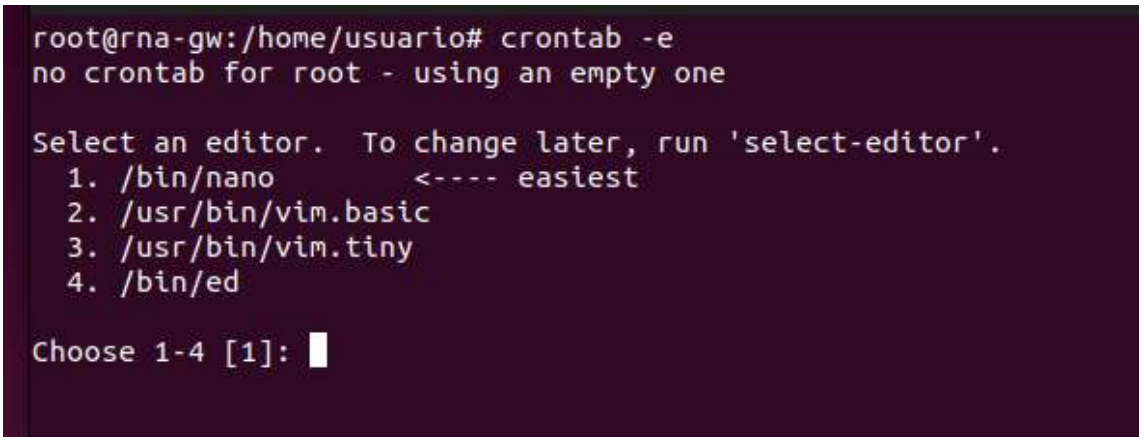




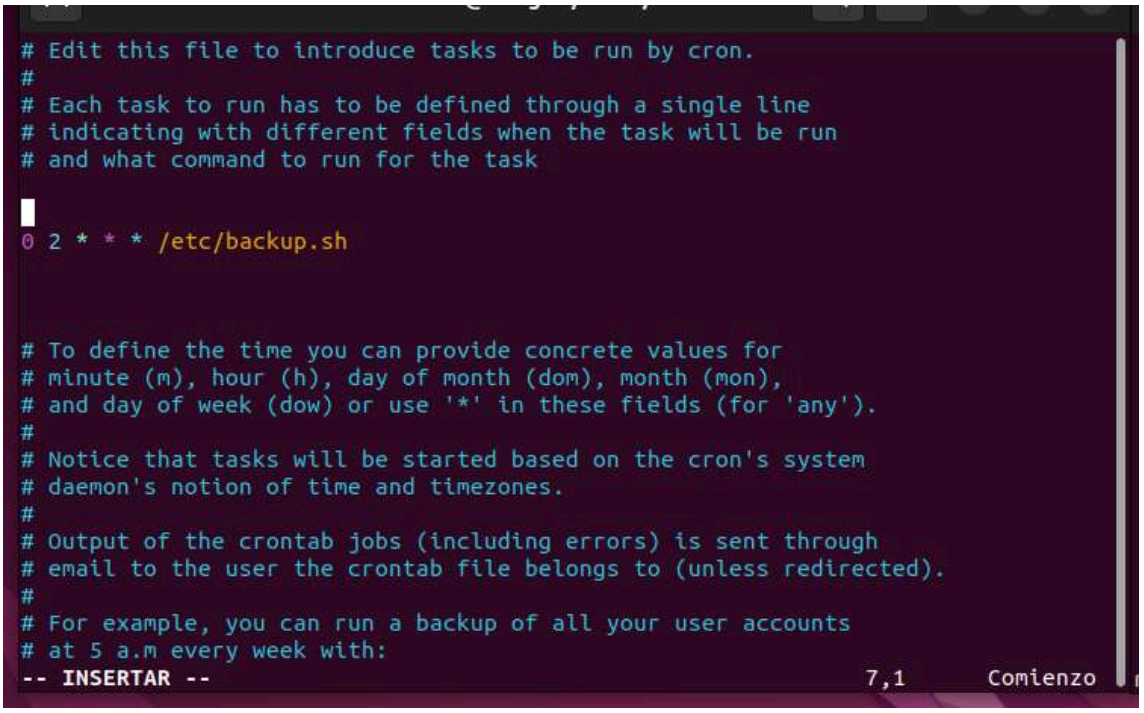
Ejecutamos el siguiente comando para hacer el script ejecutable:



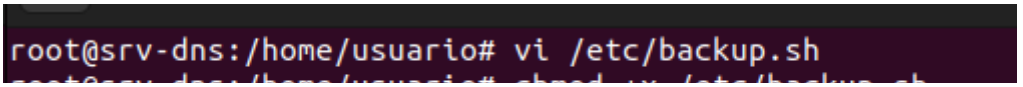
Abrimos crontab en modo de edición con el siguiente comando y elegimos nuestro editor favorito:

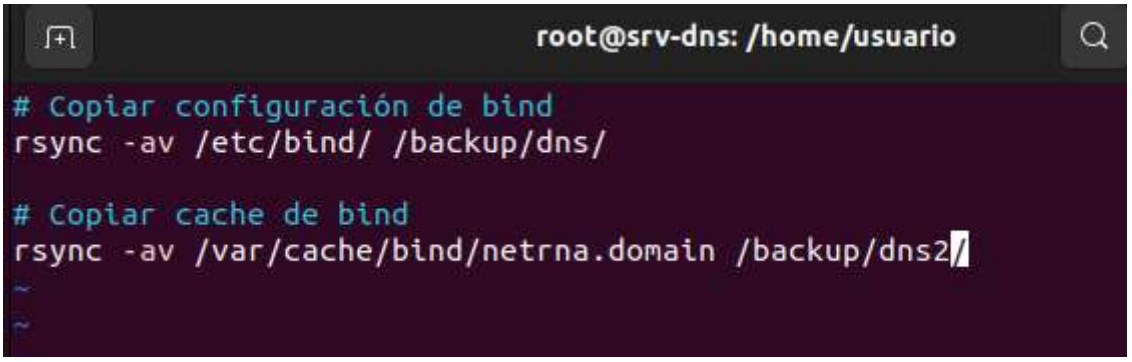


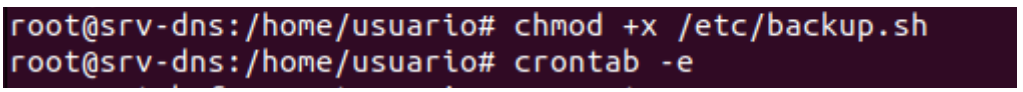
Agregamos esta línea para que el script se ejecute todos los días a las 2:00 AM:



Con esto, ya tenemos programado que cada día a las 2 se haga la copia. Ahora haremos lo mismo en el servidor de DNS.



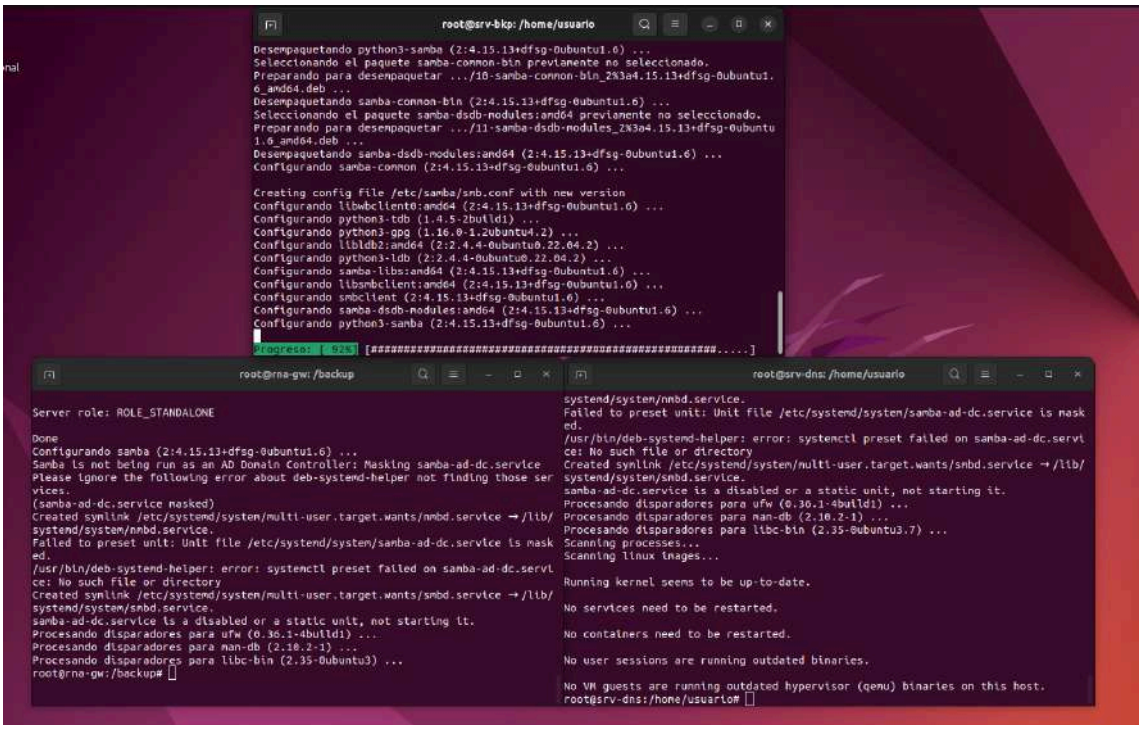




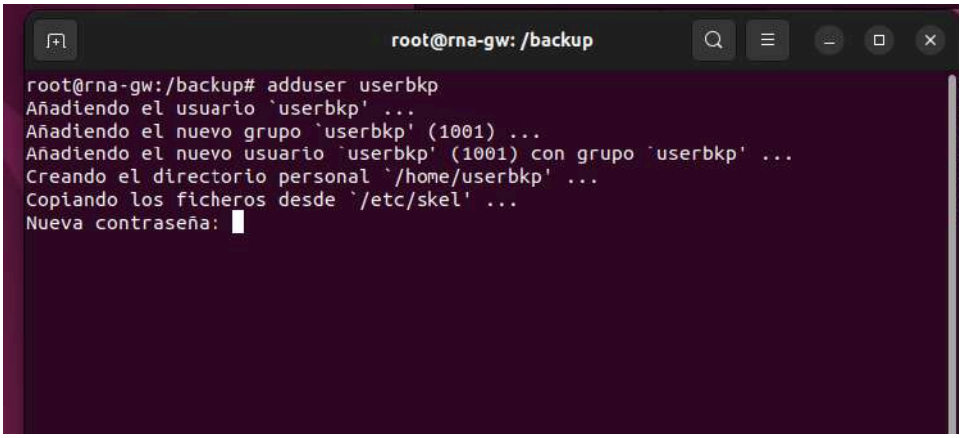
```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task

0 2 * * * /etc/backup.sh


# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
```

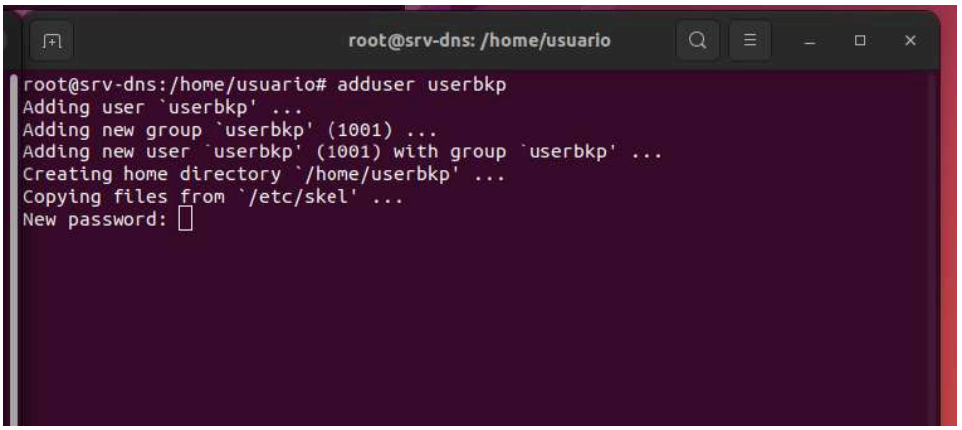
Con esto, tendremos las copias de seguridad que se harán automáticamente cada día a las 2 al directorio local. Ahora, lo que tenemos que hacer es compartir ese directorio con Samba al servidor de backups.

Instalamos samba en las máquinas

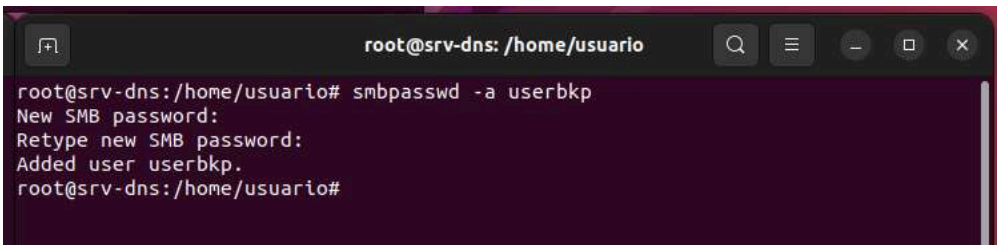


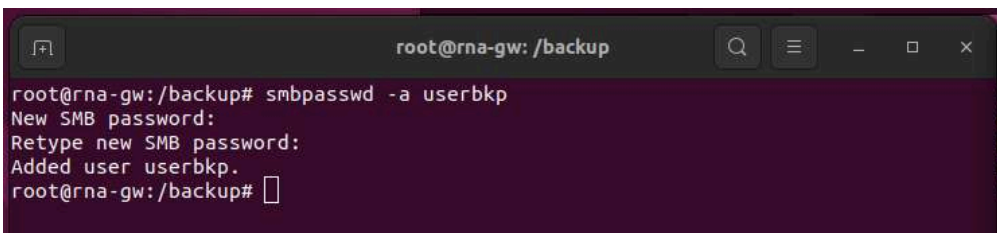
Ahora vamos a crear usuarios para que el servidor de Backups pueda acceder a las máquinas que contienen las copias.





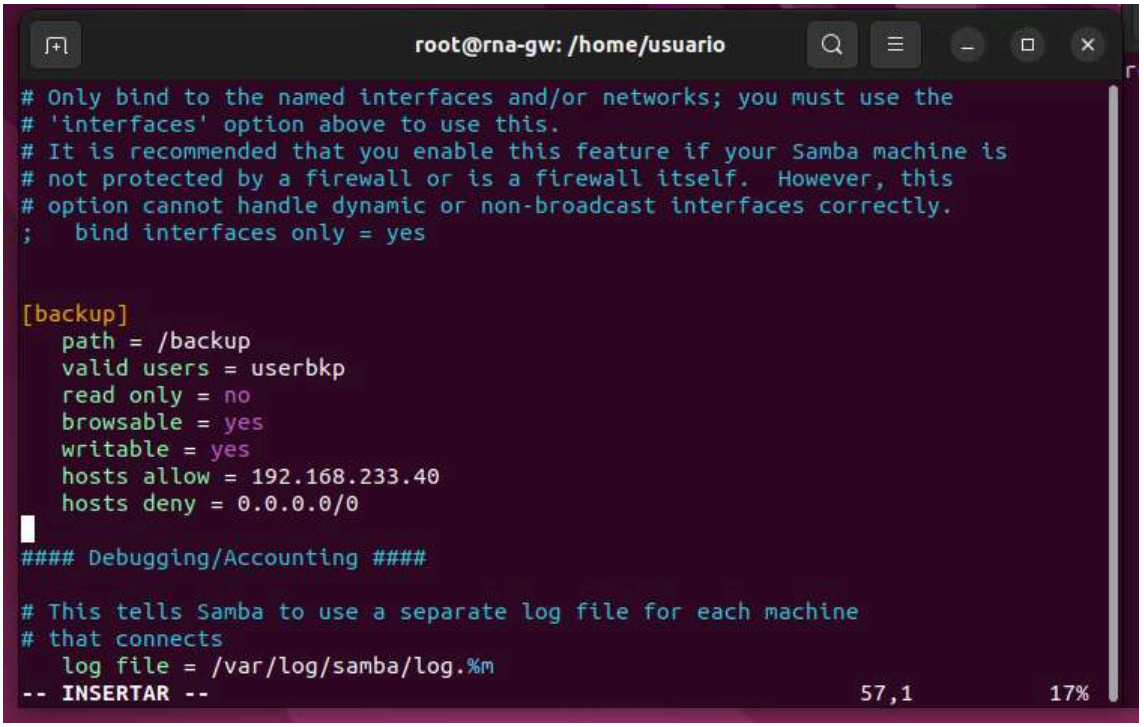
Ahora creamos los usuarios samba



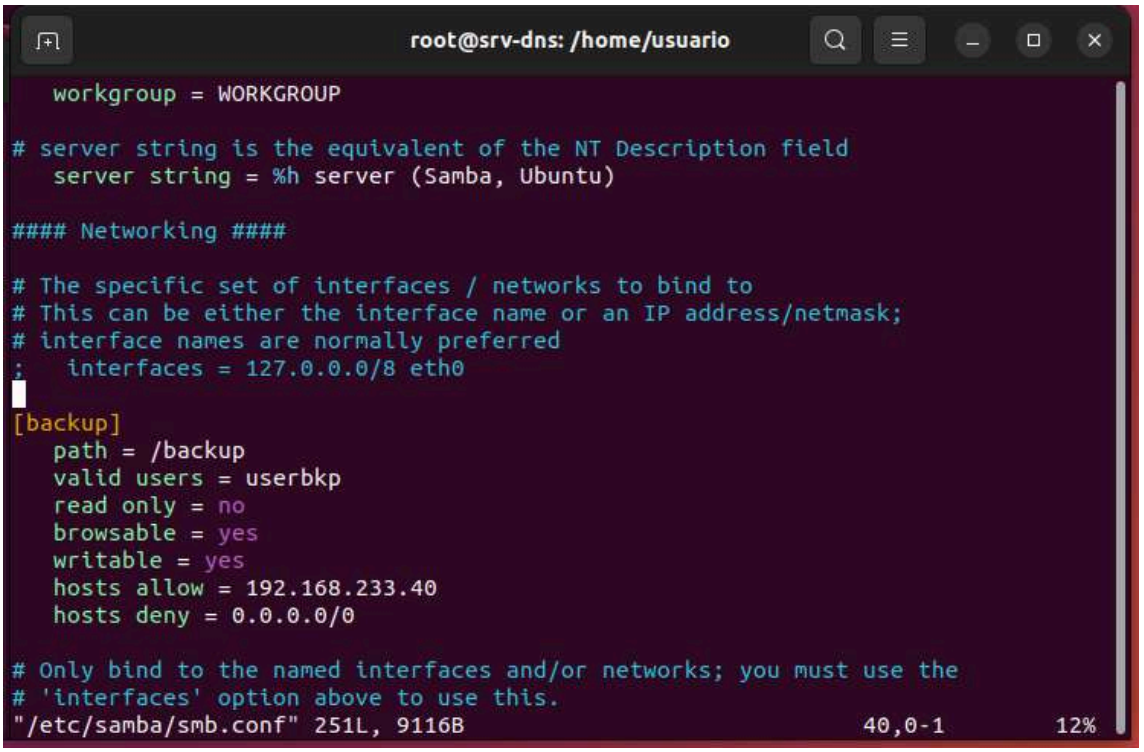


Posteriormente creamos el “trabajo” en samba en ambos servidores para que pueda acceder el servidor de copias.

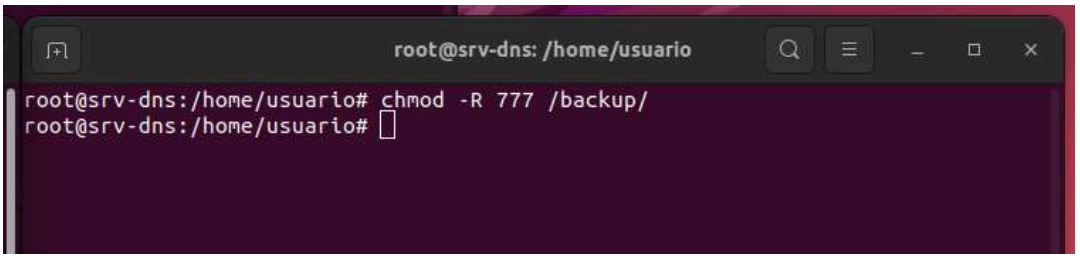
GATEWAY:

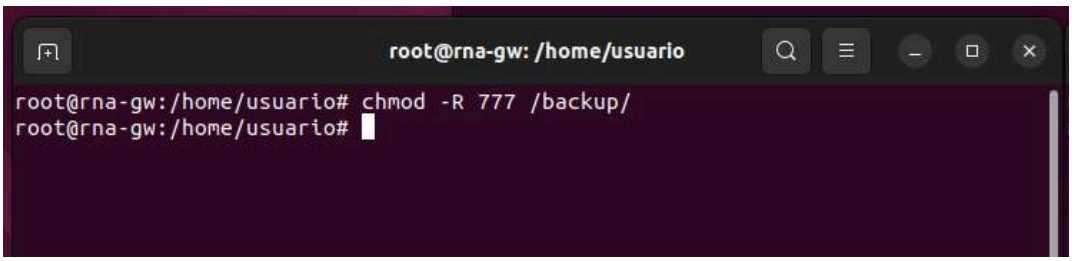


DNS:



Después habrá que darle los permisos a la carpeta /backup en cada servidor

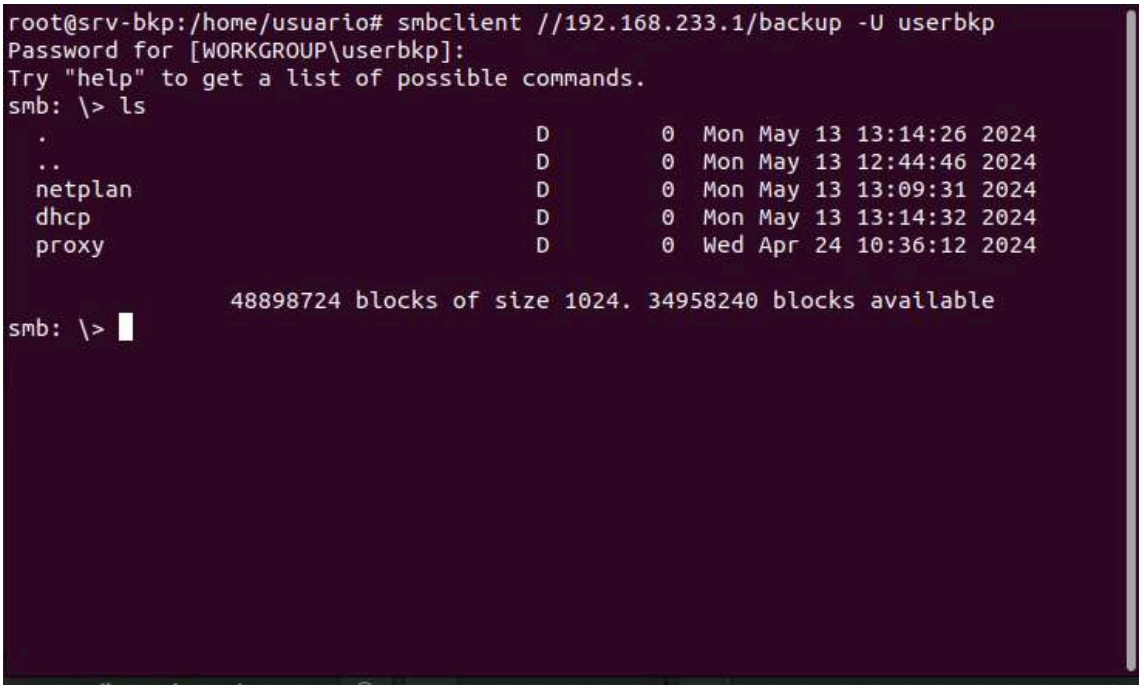




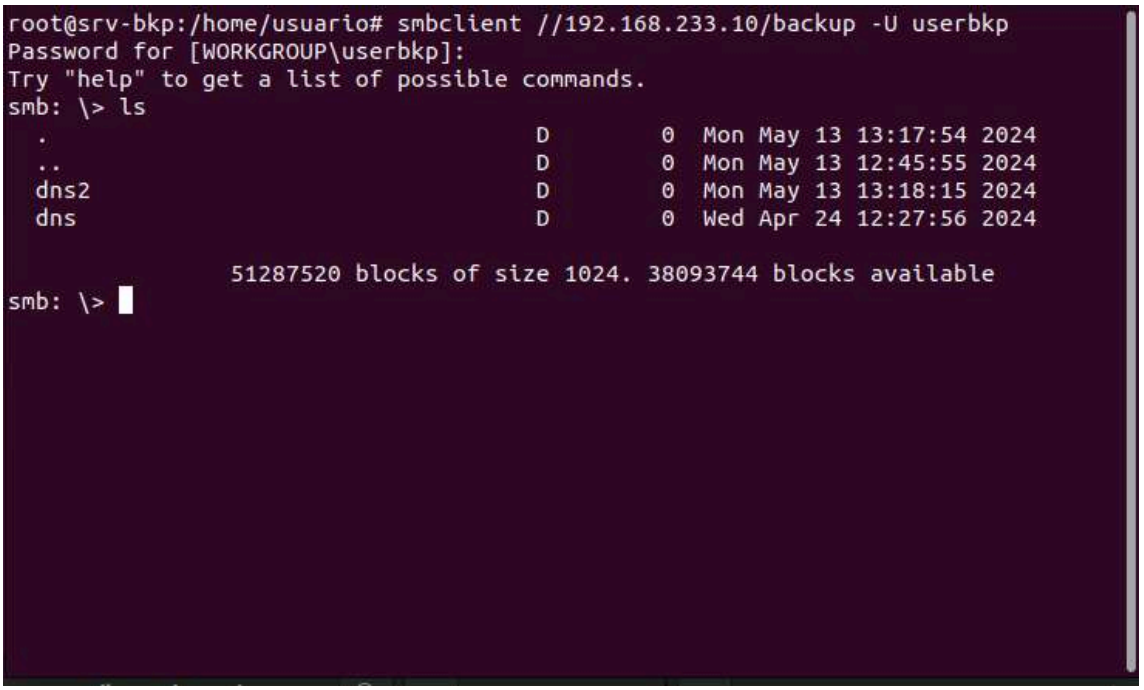
Después restablecemos el servicio y probamos a ver si puede acceder el servidor de copias.

Y como podemos ver podemos acceder a las carpetas de los servidores donde se alojan las copias de seguridad

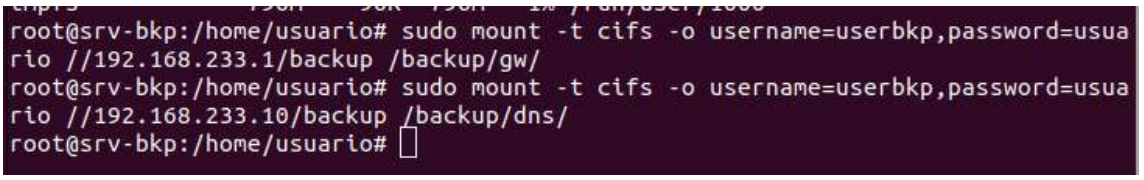
GATEWAY:

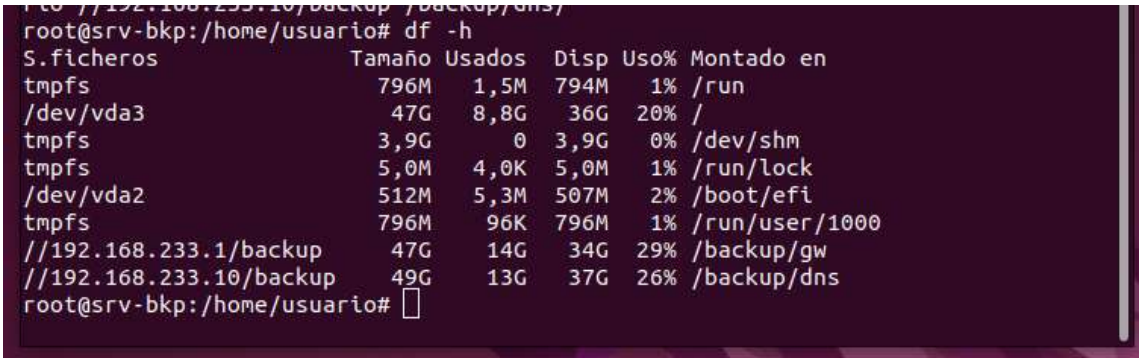


DNS:

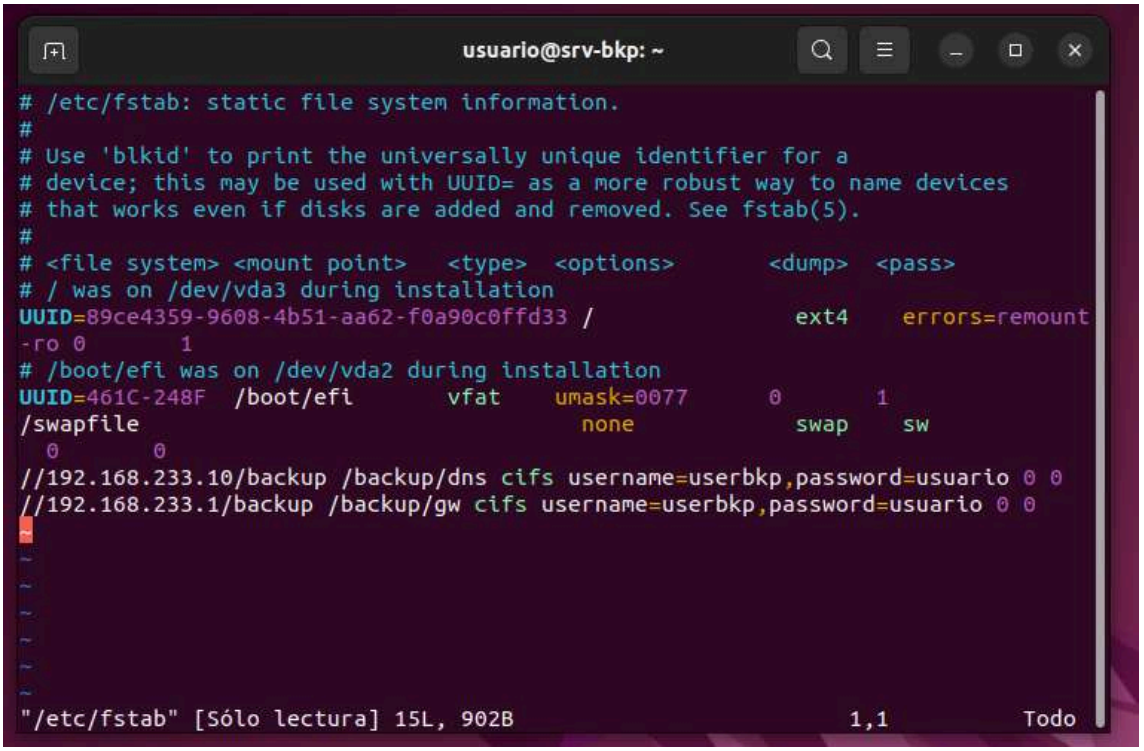


Ahora lo que deberemos hacer es montar los directorios para que sean persistentes usando el mount

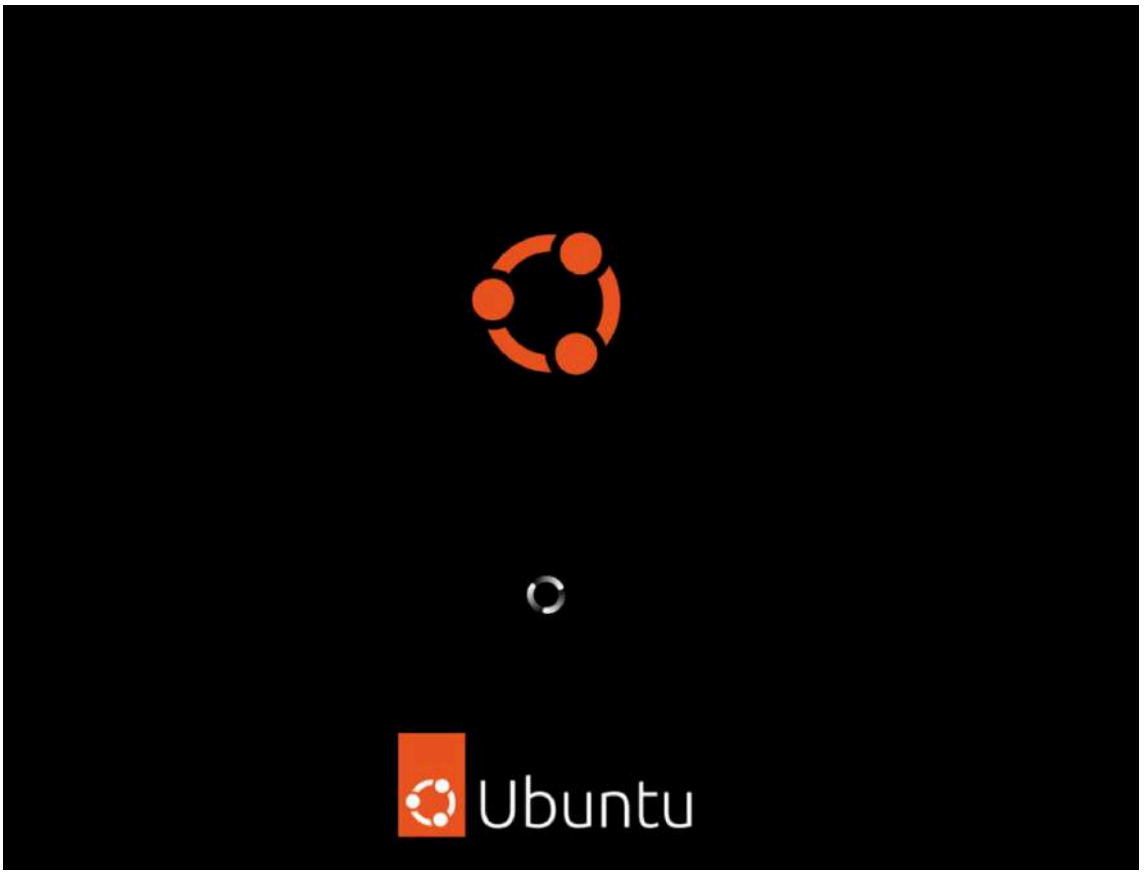


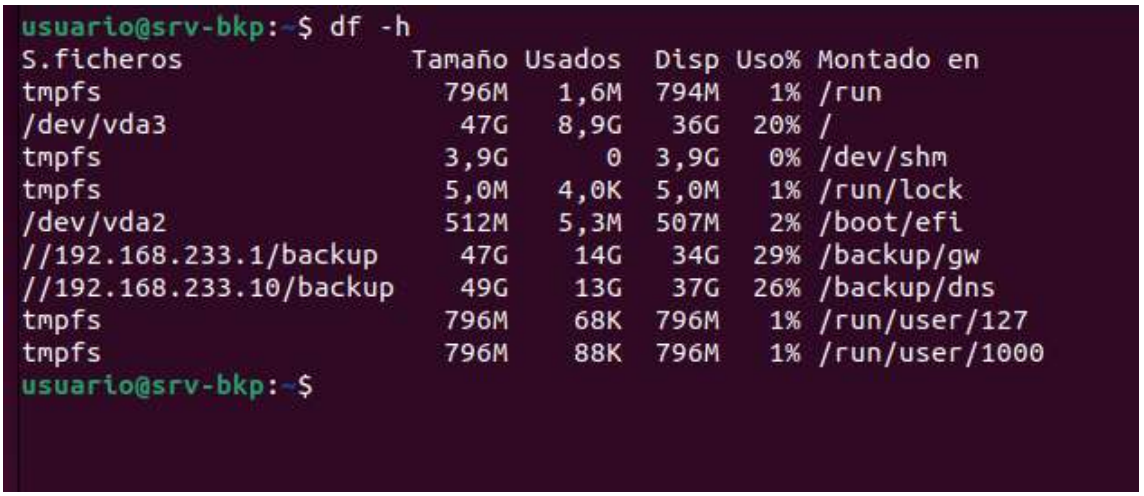


De esta manera podemos ver que ya los tenemos montados pero para que sean persistentes al reiniciar debemos modificar el fichero **/etc/fstab**



Ahora al reiniciar el ordenador seguirá estando montado



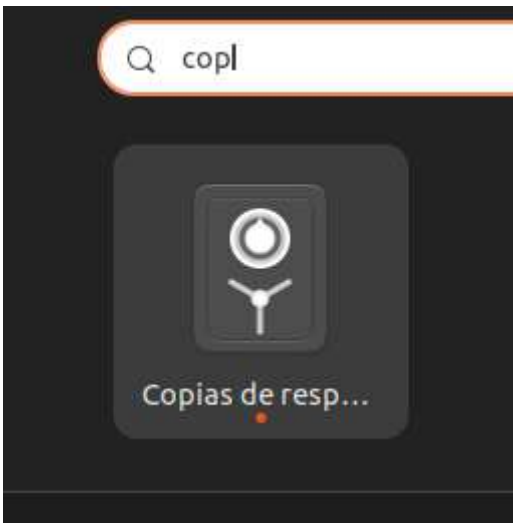


Ahora ya que lo tenemos montado y es un directorio accesible, lo que deberemos hacer es con la herramienta **deja-dup,** que haga una copia externalizada al drive para aumentar la seguridad.

Instalamos deja-dup

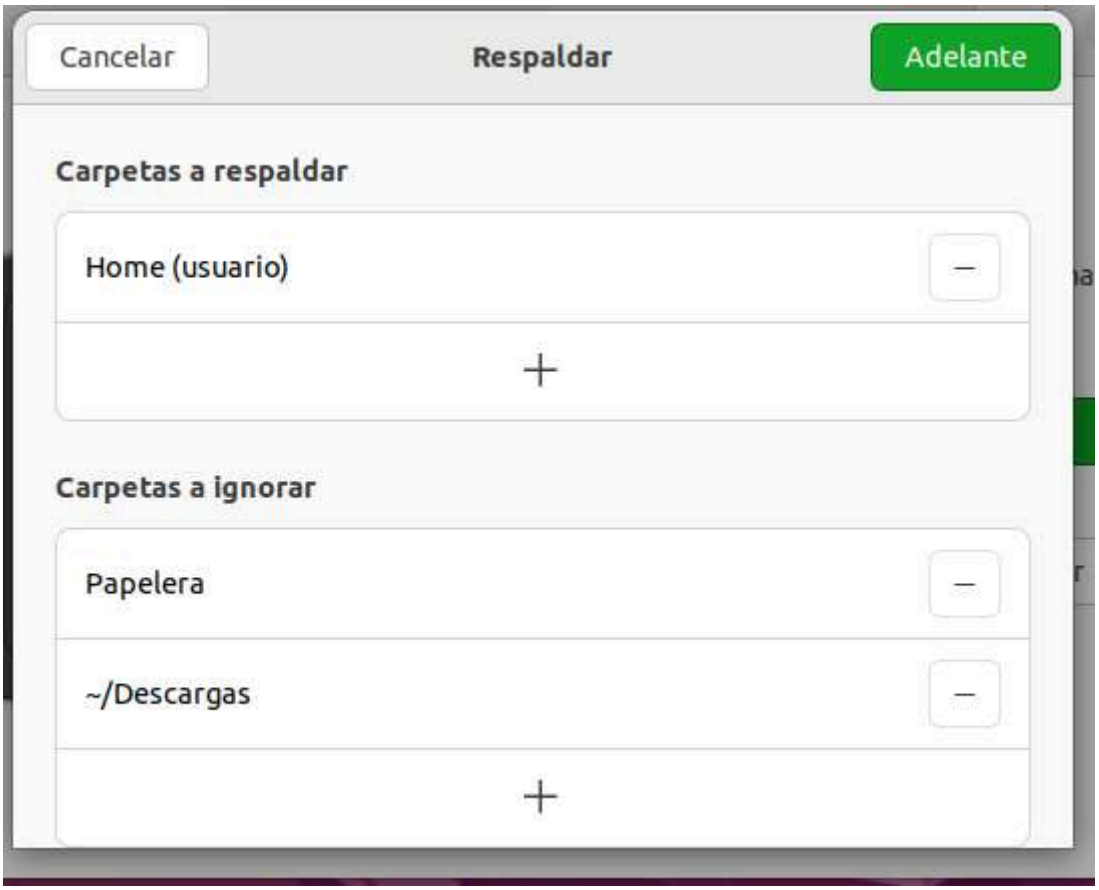
```
root@srv-bkp:/home/usuario# apt install deja-dup
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  duplicity libpython3-stdlib libpython3.10 libpython3.10-minimal
  libpython3.10-stdlib librsync2 python3 python3-bcrypt python3-fasteners
  python3-future python3-lib2to3 python3-lockfile python3-minimal
  python3-monotonic python3-paramiko python3.10 python3.10-minimal
Paquetes sugeridos:
  python3-pydrive python3-boto ncftp lftp tahoe-lafs python3-swiftclient
  python3-pip par2 python3-doc python3-tk python3-venv python-future-doc
  python-lockfile-doc python3-gssapi python3-invoke python3.10-venv
  python3.10-doc binutils binfmt-support
Se instalarán los siguientes paquetes NUEVOS:
  deja-dup duplicity librsync2 python3-bcrypt python3-fasteners python3-future
  python3-lib2to3 python3-lockfile python3-monotonic python3-paramiko
Se actualizarán los siguientes paquetes:
  libpython3-stdlib libpython3.10 libpython3.10-minimal libpython3.10-stdlib
  python3 python3-minimal python3.10 python3.10-minimal
8 actualizados, 10 nuevos se instalarán, 0 para eliminar y 508 no actualizados.
Se necesita descargar 8.593 kB de archivos.
Se utilizarán 6.160 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

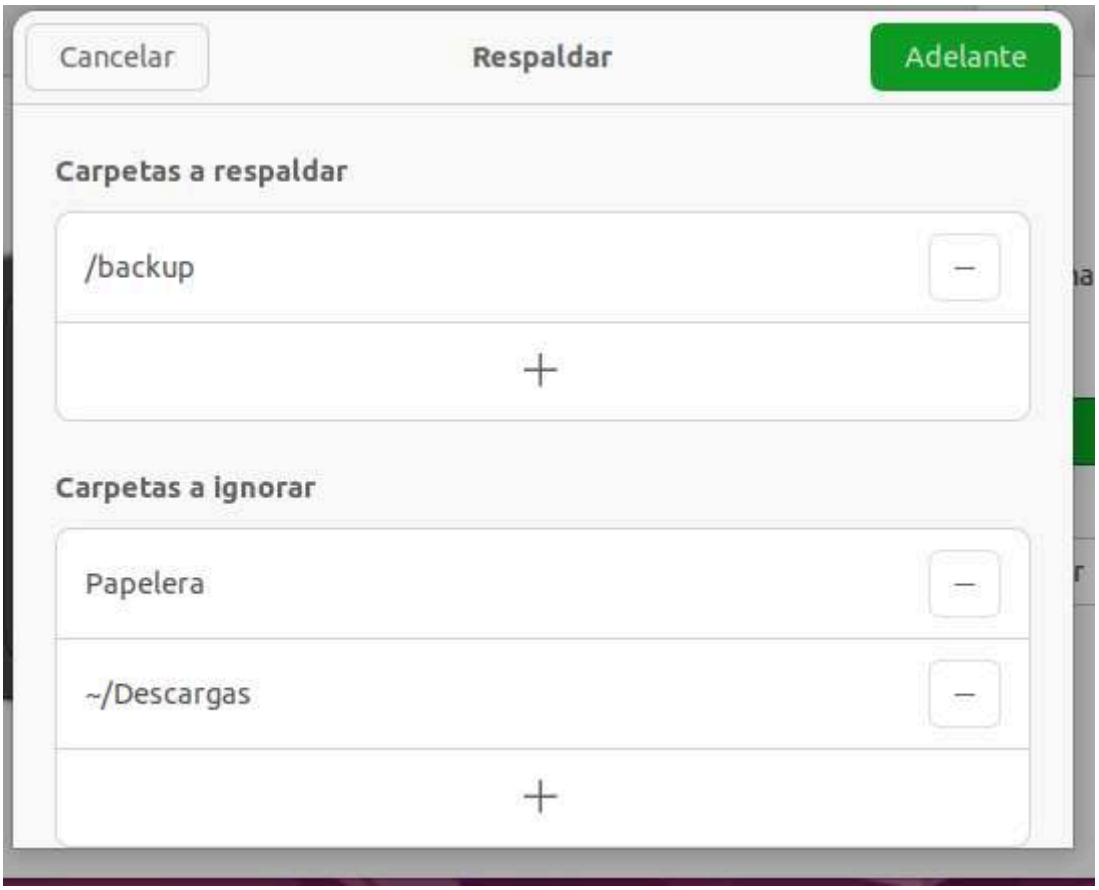
Una vez instalado accedemos a la aplicación que estará en nuestro escritorio.



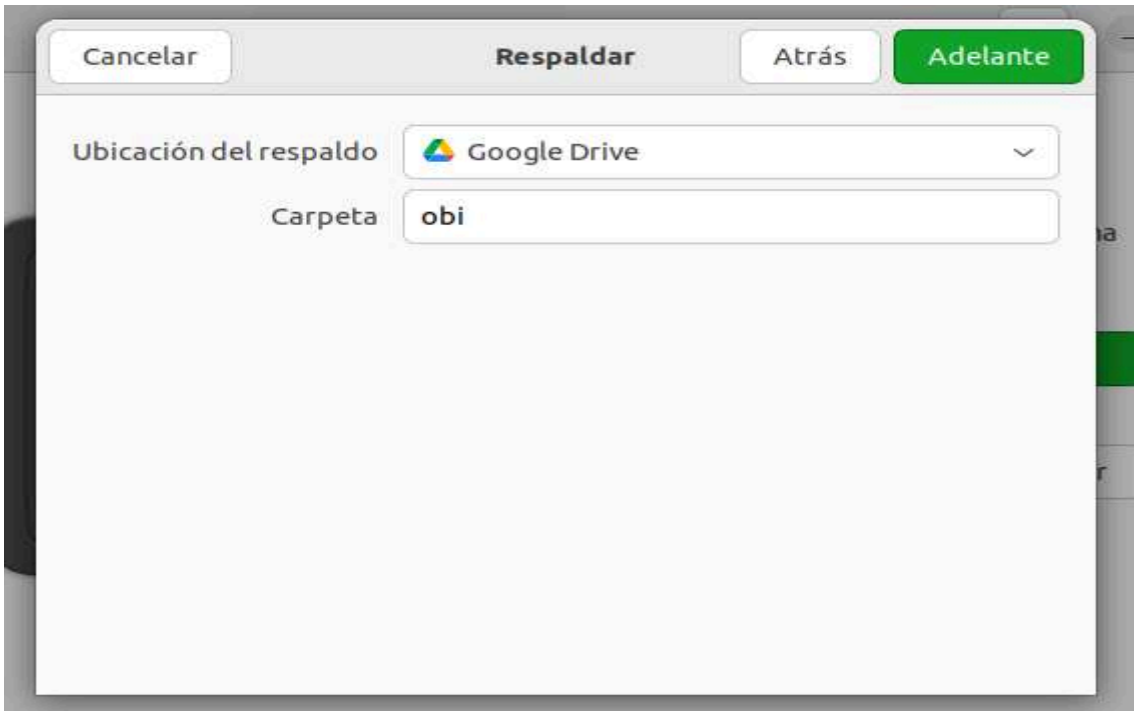
Le damos a crear mi primer respaldo

Y aquí debemos de respaldar las carpetas montadas donde contienen las copias de seguridad, le damos al botón de “Adelante”


Cancelar
Respaldar
Adelante
Carpetas a respaldar
Home (usuario)
Carpetas a ignorar
Papelera
~/Descargas

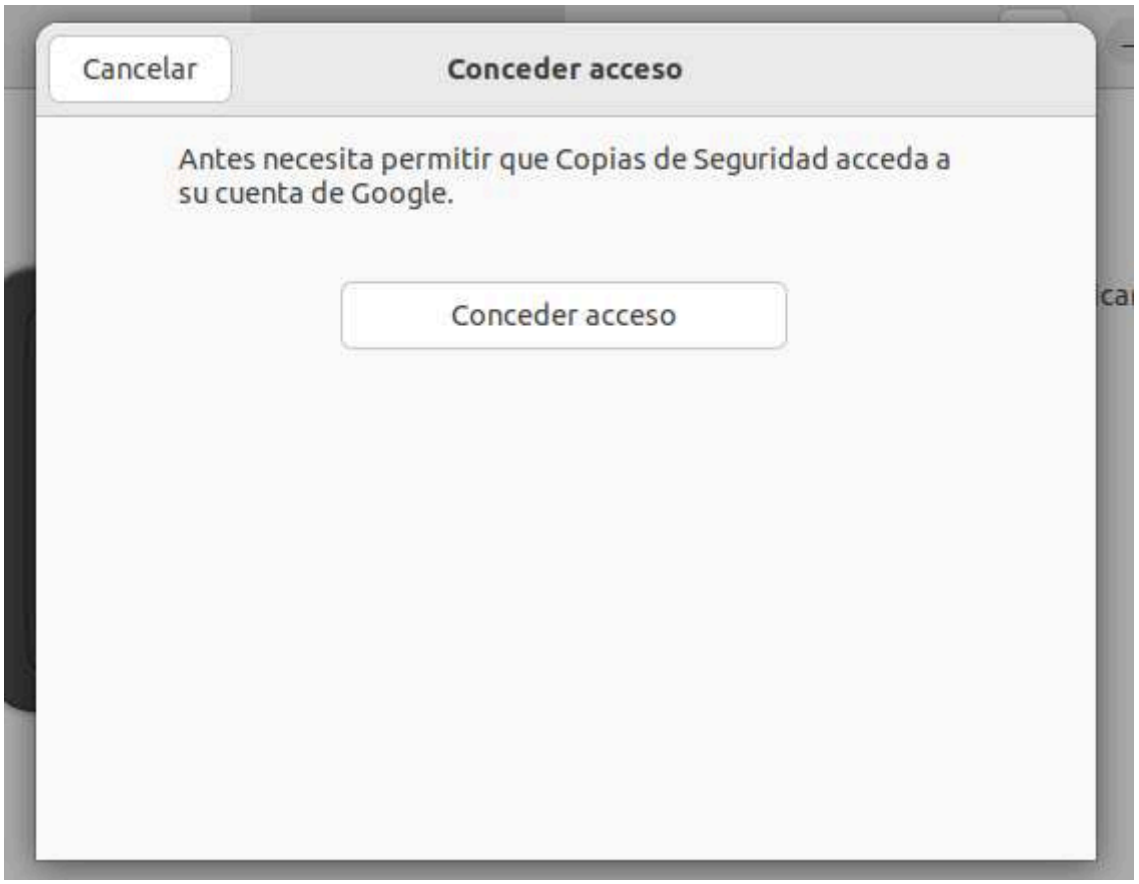


Una vez elegido nuestro directorio que en este caso es el de los backups iremos al siguiente paso donde podremos externalizar la copia en la nube usando google drive.

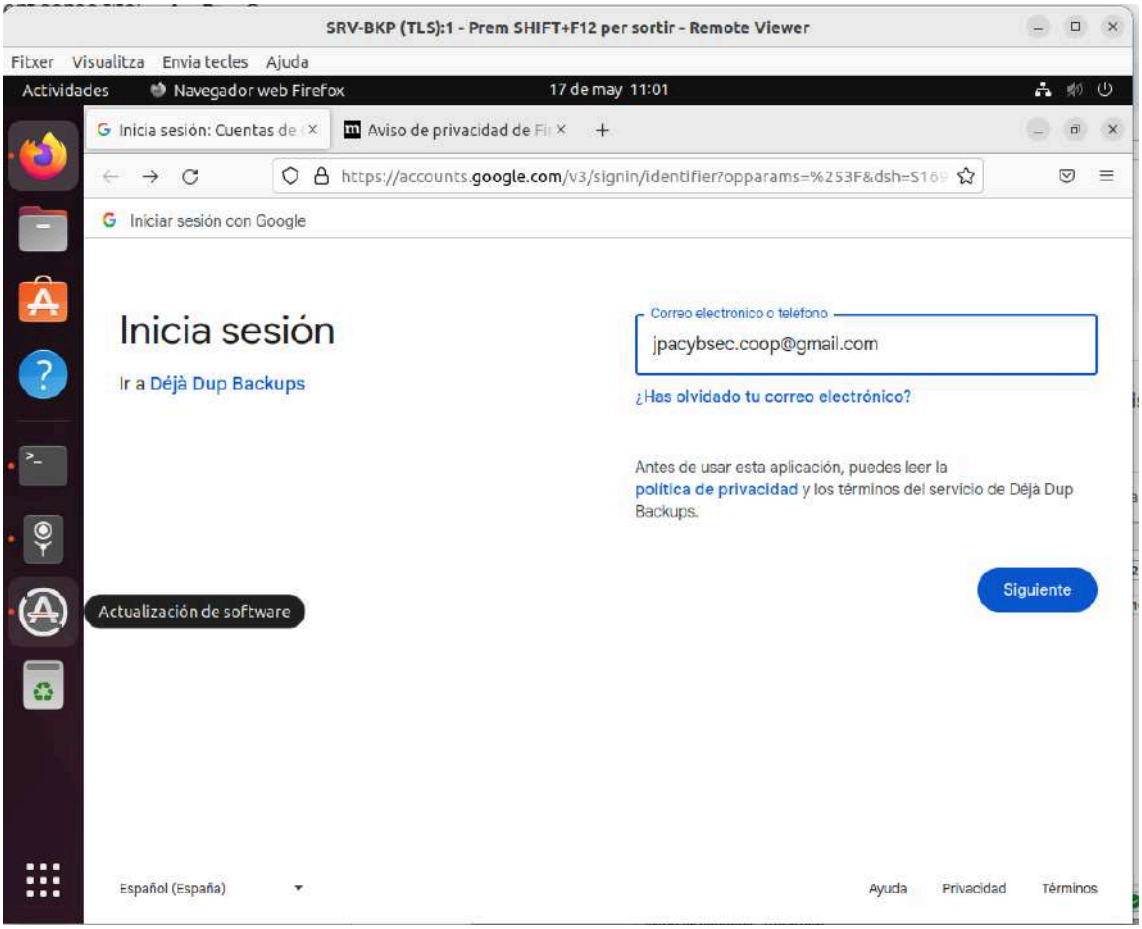


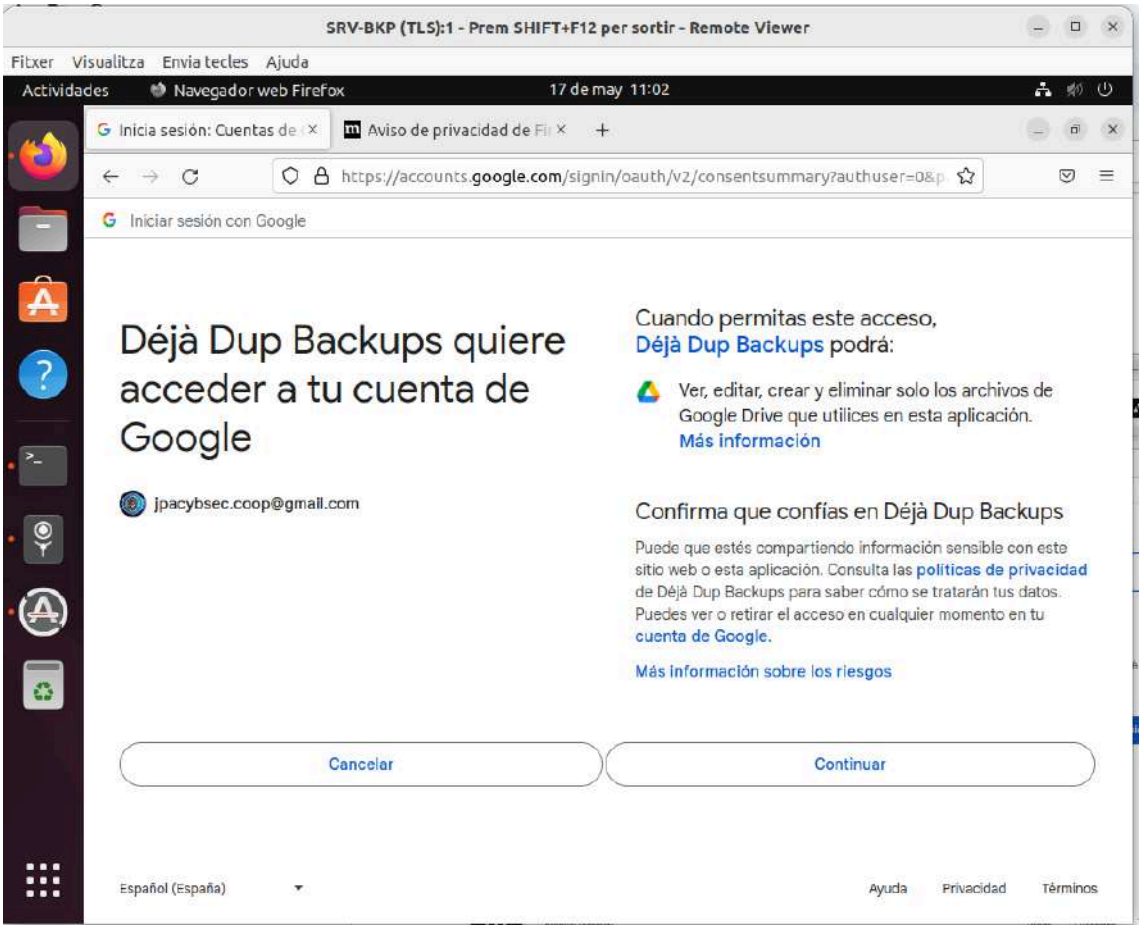
Instalamos los paquetes necesarios.

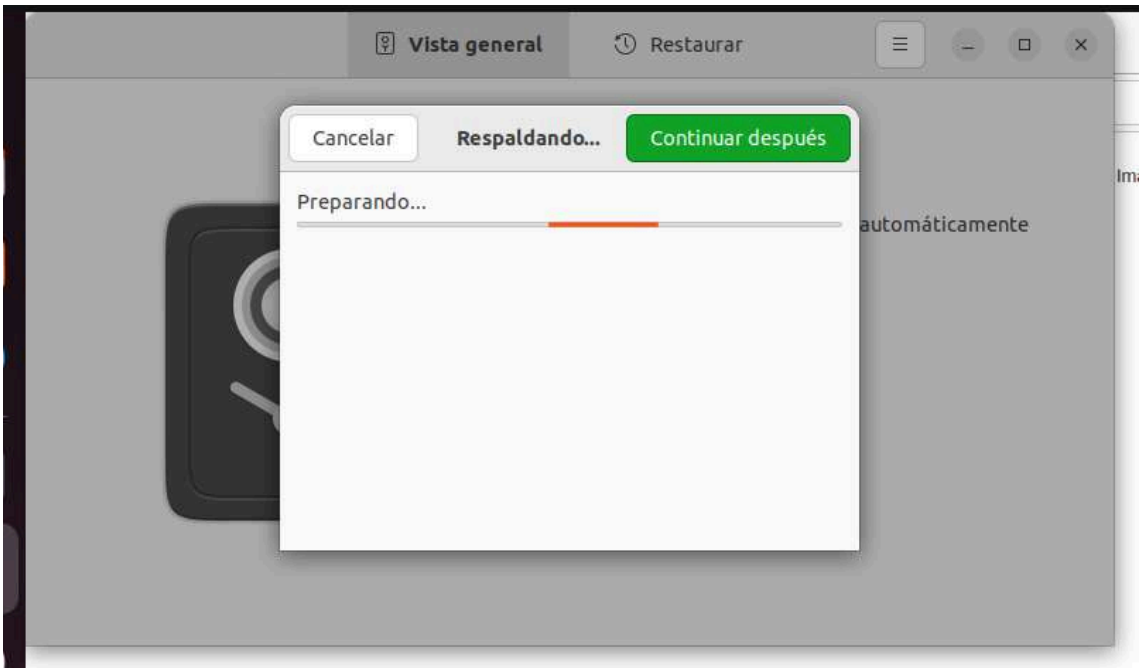
Por últimos le concedemos el acceso



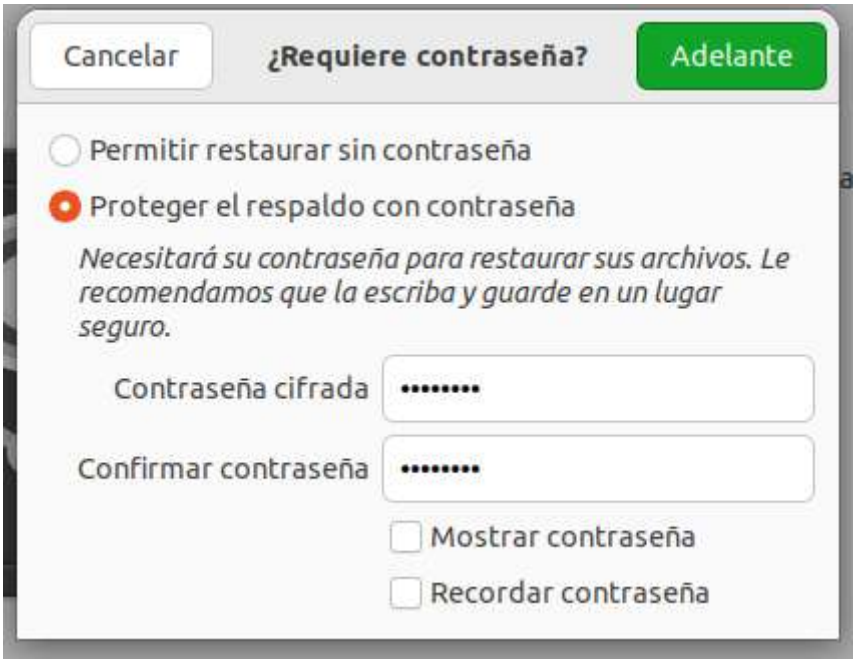
Y por último iniciamos sesion y lo subimos al drive

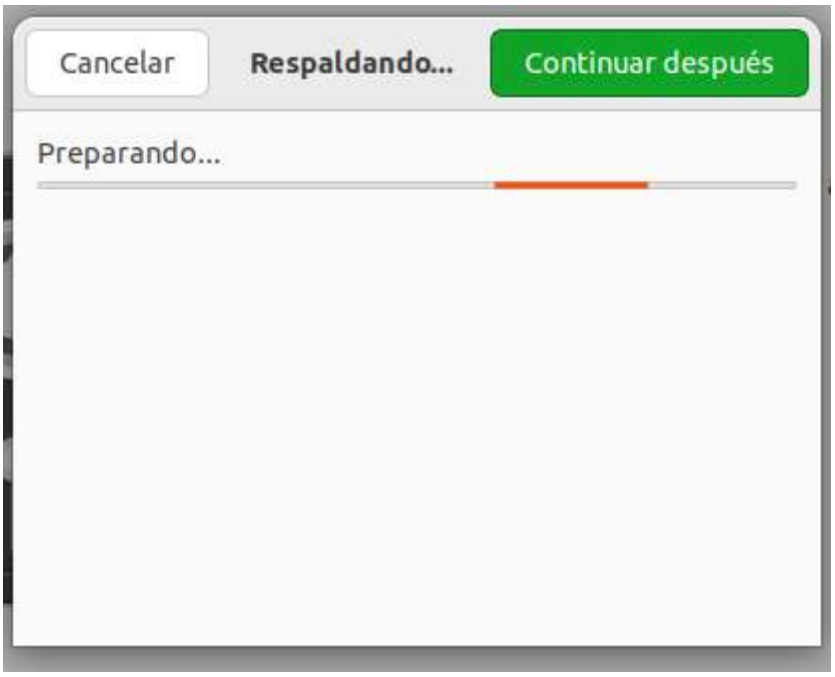

SRV-BKP (TLS):1 - Prem SHIFT+F12 per sortir - Remote Viewer
Fitxer Visualitza Envia tecles Ajuda
Actividades
Navegador web Firefox
17 de may 11:01
Inicia sesión: Cuentas de
Aviso de privacidad de Fi
https://accounts.google.com/v3/signin/identifier?opparams=%253F&dsh=S16
Iniciar sesión con Google
Inicia sesión
Ir a Déjà Dup Backups
Correo electrónico o teléfono
jpacybsec.coop@gmail.com
¿Has olvidado tu correo electrónico?
Antes de usar esta aplicación, puedes leer la política de privacidad y los términos del servicio de Déjà Dup Backups.
Siguiente
Actualización de software
Español (España)
Ayuda
Privacidad
Términos

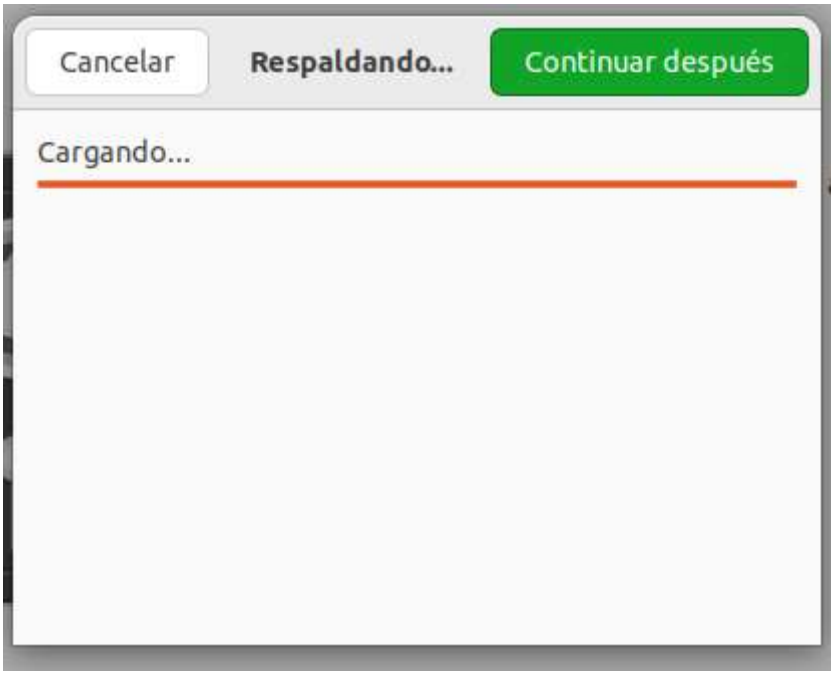

SRV-BKP (TLS):1 - Prem SHIFT+F12 per sortir - Remote Viewer
Fitxer Visualitza Envia tecles Ajuda
Actividades
Navegador web Firefox
17 de may 11:02
Inicia sesión: Cuentas de
Aviso de privacidad de Fi
https://accounts.google.com/signin/oauth/v2/consentsummary?authuser=0&p
Iniciar sesión con Google
Déjà Dup Backups quiere acceder a tu cuenta de Google
jpacybsec.coop@gmail.com
Cuando permitas este acceso, Déjà Dup Backups podrá:
Ver, editar, crear y eliminar solo los archivos de Google Drive que utilices en esta aplicación.
Más información
Confirma que confías en Déjà Dup Backups
Puede que estés compartiendo información sensible con este sitio web o esta aplicación. Consulta las políticas de privacidad de Déjà Dup Backups para saber cómo se tratarán tus datos. Puedes ver o retirar el acceso en cualquier momento en tu cuenta de Google.
Más información sobre los riesgos
Cancelar
Continuar
Español (España)
Ayuda
Privacidad
Términos

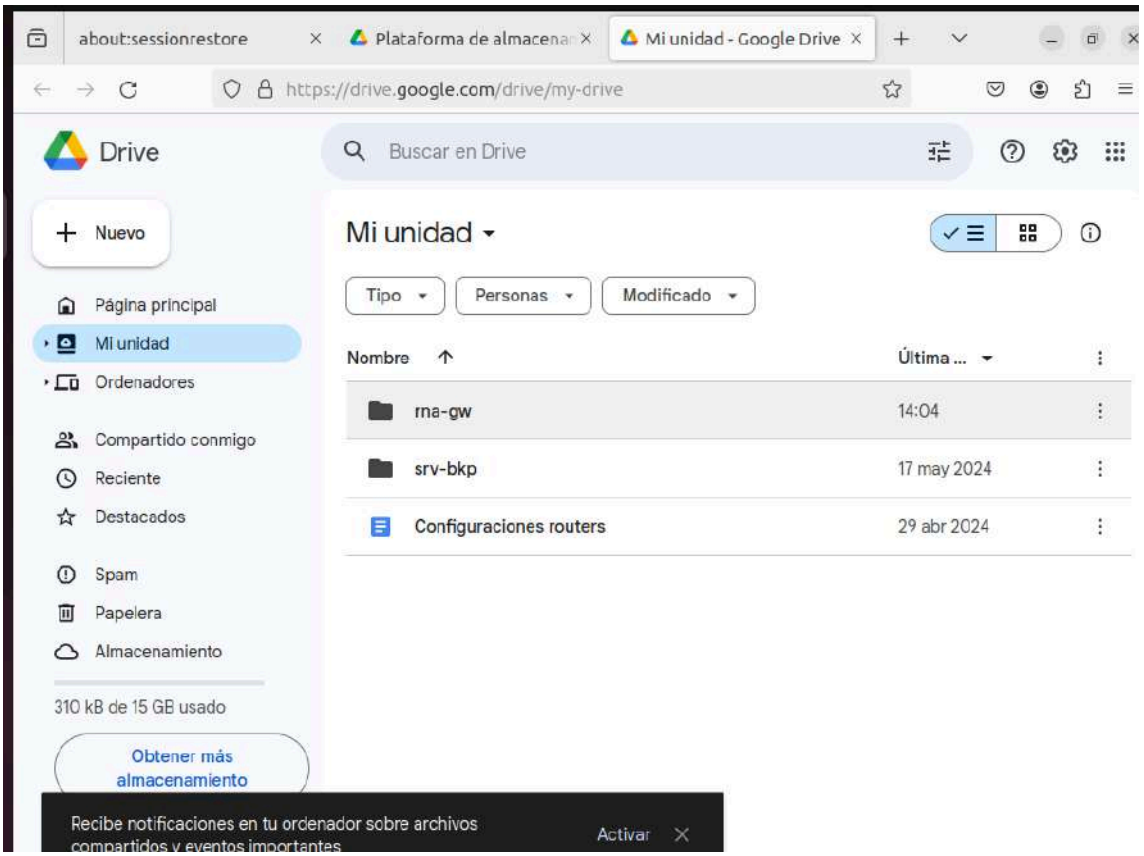

Vista general
Restaurar
Cancelar
Respaldando...
Continuar después
Preparando...
automáticamente

Aquí nos da la opción de poder cifrar nuestra copia de seguridad




Cancelar
Respaldando...
Continuar después
Preparando...


Cancelar
Respaldando...
Continuar después
Cargando...

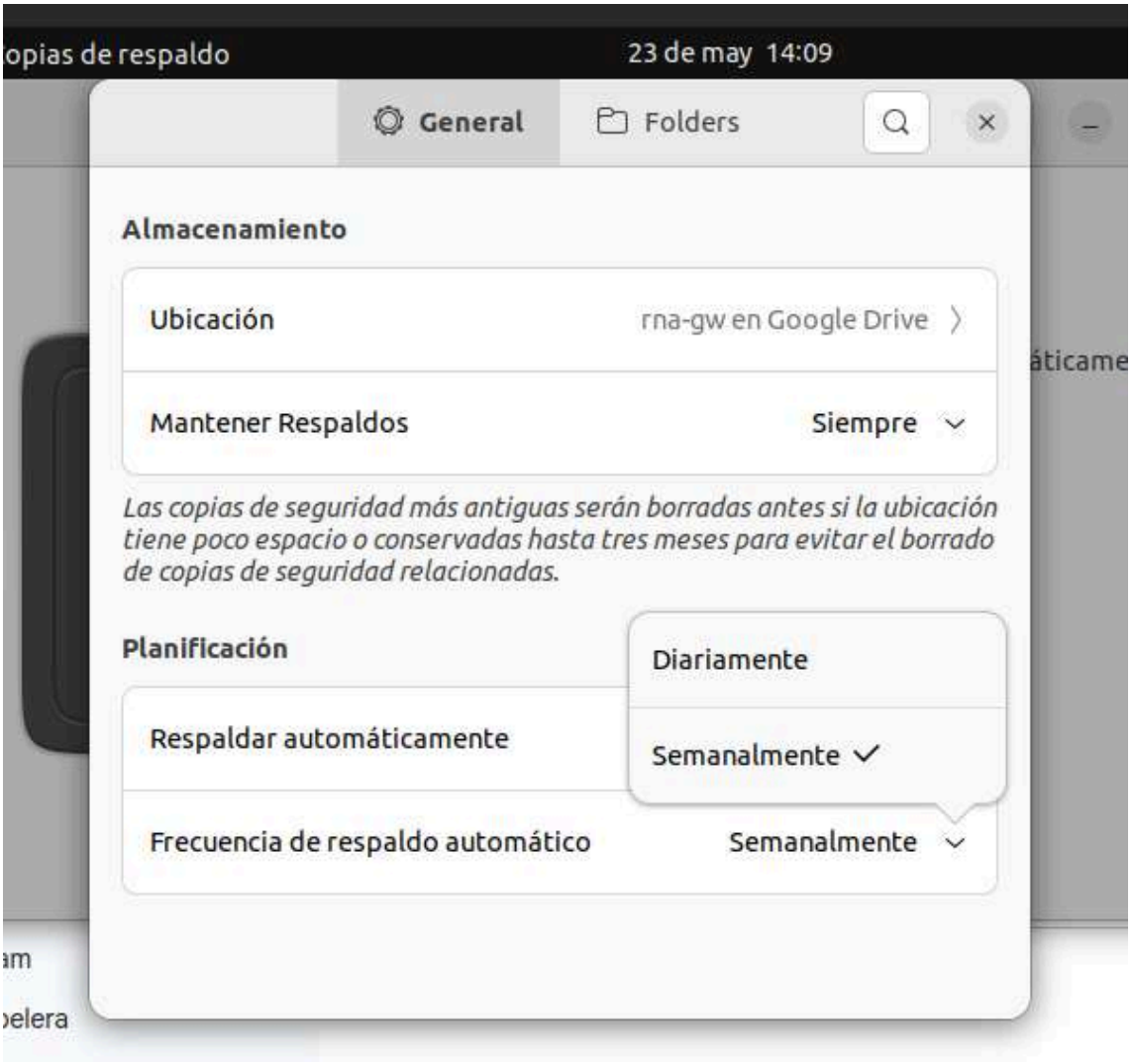


De esta manera ya tenemos nuestro servidor enviando backups al drive, este ha sido de manera manual pero posteriormente nos da la opción de poder enviar copia automáticamente.

Para ellos volvemos a acceder a la herramienta y nos saldrá la opción de realizar copias de manera automática.

Activamos esta opción y automáticamente enviará copias de seguridad cada 7 días.

Podemos cambiar la frecuencia con la que se realizan las copias de seguridad, para ello nos dirigimos a ajustes en preferencias y elegimos la frecuencia de respaldo.



En conclusión la estrategia descrita implica realizar copias de seguridad automáticas diariamente en un servidor de backups local, y luego externalizar estas copias a la nube. Esto maximiza la eficiencia al mejorar la seguridad, la accesibilidad y la escalabilidad del

almacenamiento de datos, protegiéndolos contra pérdidas y facilitando su recuperación en caso de desastres.

Implementación de la DMZ

Para garantizar la seguridad y eficiencia de nuestra red, vamos a configurar dos pasarelas con diferentes funciones y restricciones.

Pasarela 1

- **Red:** 192.168.233.0/24
- **Servicios:** DNS, web y correo.
- **Políticas de acceso:**
    - **Puertos permitidos:** Se permite el acceso desde el exterior a través de los puertos 443 (HTTPS), 80 (HTTP), 53 (DNS), 25 (SMTP) ,143 (IMAP) y ICMP.
    - **Conexiones entrantes (input):** Se rechazan todas las conexiones directas entrantes a la pasarela 1 que no sean por los puertos permitidos mencionados. Esto ayuda a proteger la pasarela de accesos no autorizados.
    - **Conexiones reenviadas (forward):** Se permite el reenvío de conexiones bajo las condiciones especificadas. Esto significa que las conexiones que llegan a la pasarela 1 y necesitan ser dirigidas a otras partes de la red se permiten siempre que cumplan con las políticas de puertos permitidos.
    - **Conexiones salientes (output)**: Se permite que todas las conexiones salientes de la pasarela 1 puedan salir sin restricciones. Esto es necesario para que los servicios dentro de la DMZ puedan comunicarse hacia el exterior según sea necesario.
    - **Política general:** Todo el tráfico restante que no cumpla con las reglas anteriores se bloquea (dropea) para evitar accesos no deseados.

Pasarela 2

- **Red:** 192.168.133.0/24

- **Servicios:** Clientes, servidor de backups y servidor de monitorización.
- **Políticas de acceso**:
    - **Conexiones solicitadas:** Sólo se permite la información que haya sido solicitada previamente desde la red interna. Esto significa que solo se permiten conexiones establecidas y relacionadas, reforzando así la seguridad interna.
    - **Conexiones entrantes (input):** Se rechazan todas las conexiones entrantes no solicitadas a la pasarela 2, previniendo así accesos no autorizados desde el exterior.
    - **Conexiones salientes (output):** Se permite que todas las conexiones salientes de la pasarela 2 puedan salir sin restricciones, asegurando que los dispositivos internos puedan comunicarse con el exterior según sea necesario.
    - **Política general:** Todo el tráfico restante que no cumpla con las reglas anteriores se bloquea (dropea) para mantener la seguridad de la red interna.

Resumen

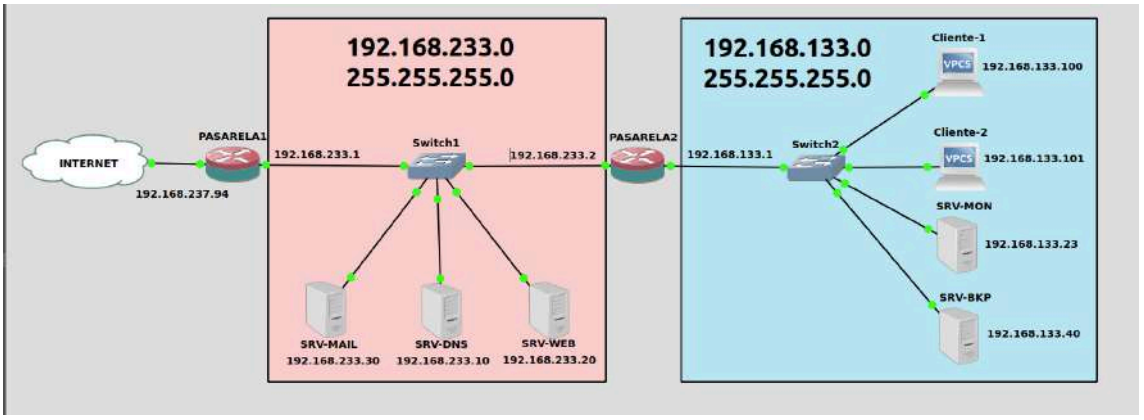
- La Pasarela 1 actúa como una DMZ, donde se alojan servicios que deben ser accesibles desde el exterior bajo restricciones específicas para proteger la red.
- La Pasarela 2 está destinada a la red interna, asegurando que solo las respuestas a solicitudes previas sean permitidas y protegiendo así los datos y dispositivos internos de accesos no autorizados.

Topología de la Red

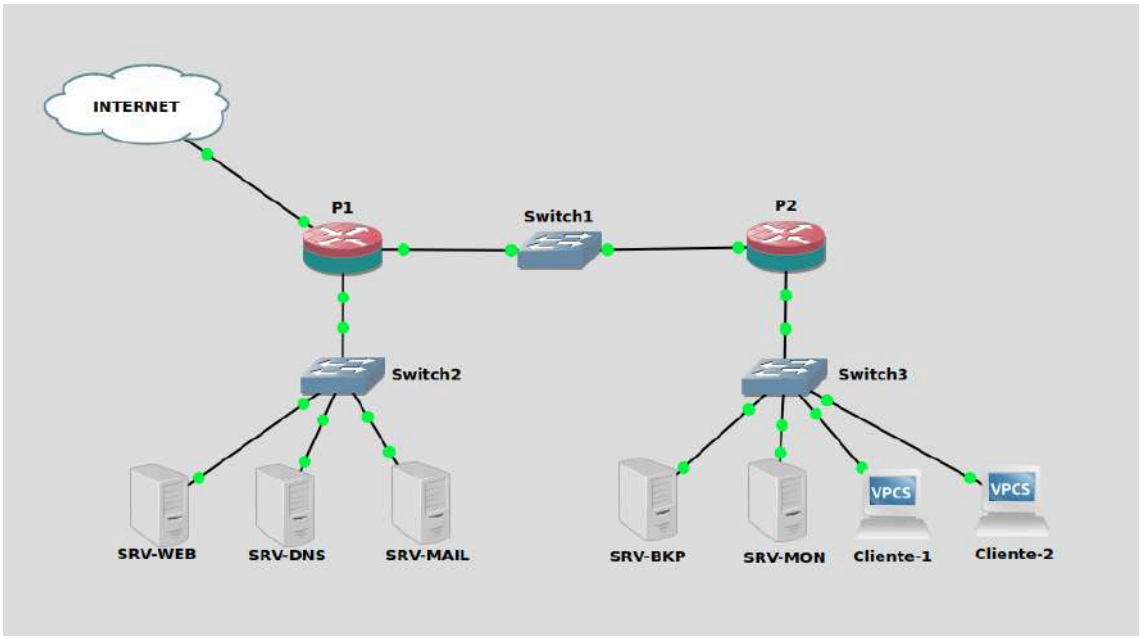
La topología de la red mostrará cómo están interconectados física y lógicamente los dispositivos y servicios mencionados en las dos pasarelas.

Ahora os mostraremos la topología física y lógica.

TOPOLOGIA LOGICA:



TOPOLOGIA FISICA:



Ahora procederemos a montar las pasarelas en nuestra infraestructura. Primero, utilizaremos el GATEWAY como Pasarela 1. Luego, crearemos una nueva pasarela que será la Pasarela 2. Para ello, tendremos que cambiar las direcciones IP del servidor de backup, monitorización, cliente 1 y cliente 2.

La Pasarela 2 tendrá dos interfaces:

1. Una interfaz que conecta con la red de la Pasarela 1.
2. Otra interfaz que conecta con la red interna.

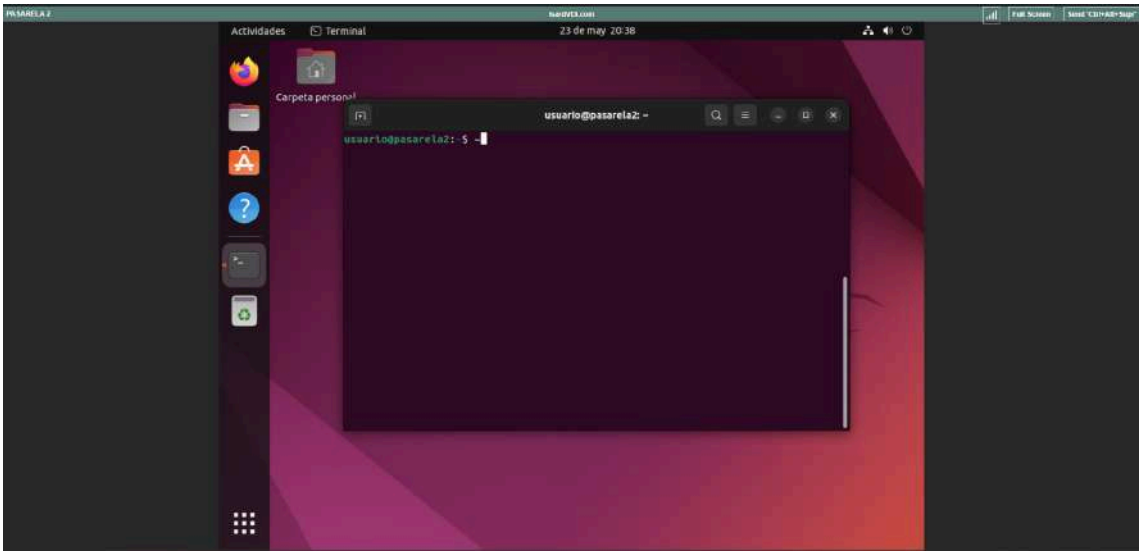
La Pasarela 1 ya está configurada con dos interfaces:

1. Una interfaz hacia la red externa.
2. Otra interfaz hacia la red interna.

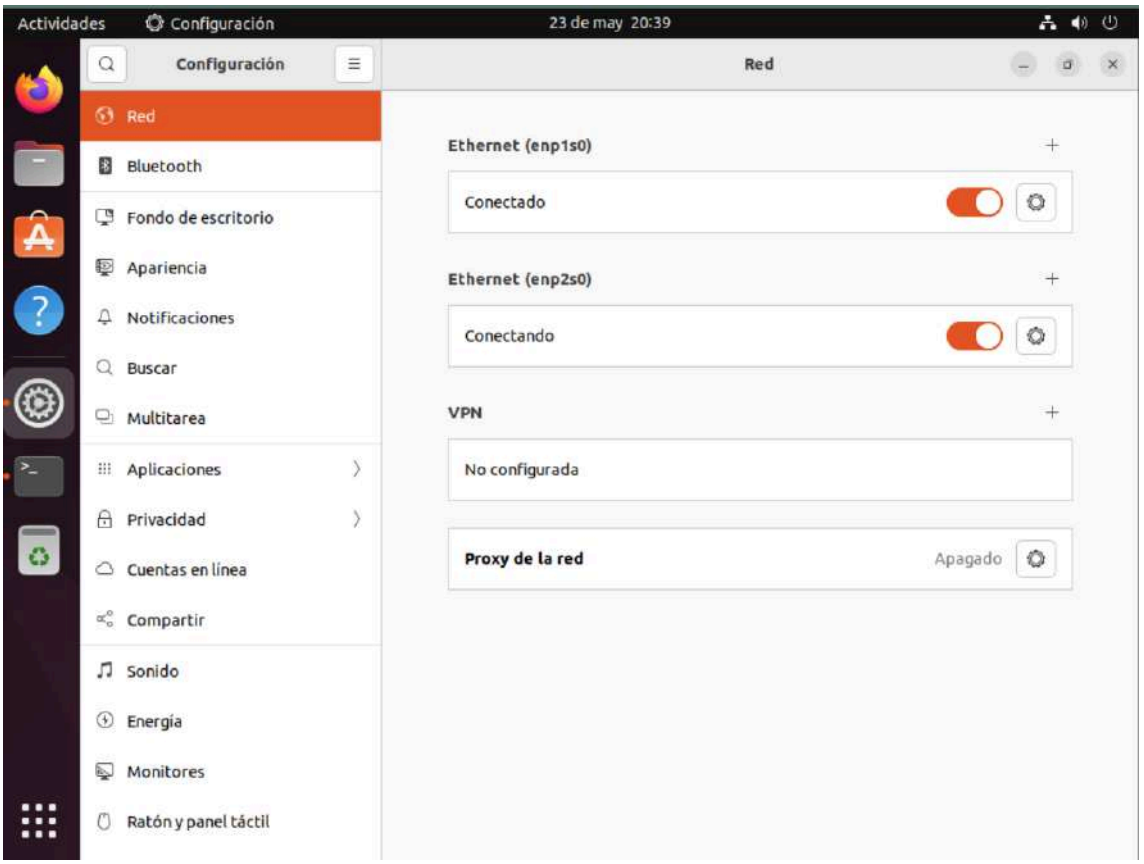
Con esta configuración, aseguraremos que cada pasarela maneje adecuadamente el tráfico según las políticas de seguridad establecidas.

Procederá a la creación de la pasarela 2 y la implementación de las 2 interfaces necesarias.

También hemos realizado la configuración estándar de la pasarela como actualizar los paquetes, hostname,zona horario,etc.



Como tenemos dos interfaces, tendremos que configurar la red interna y la red de la pasarela 1.



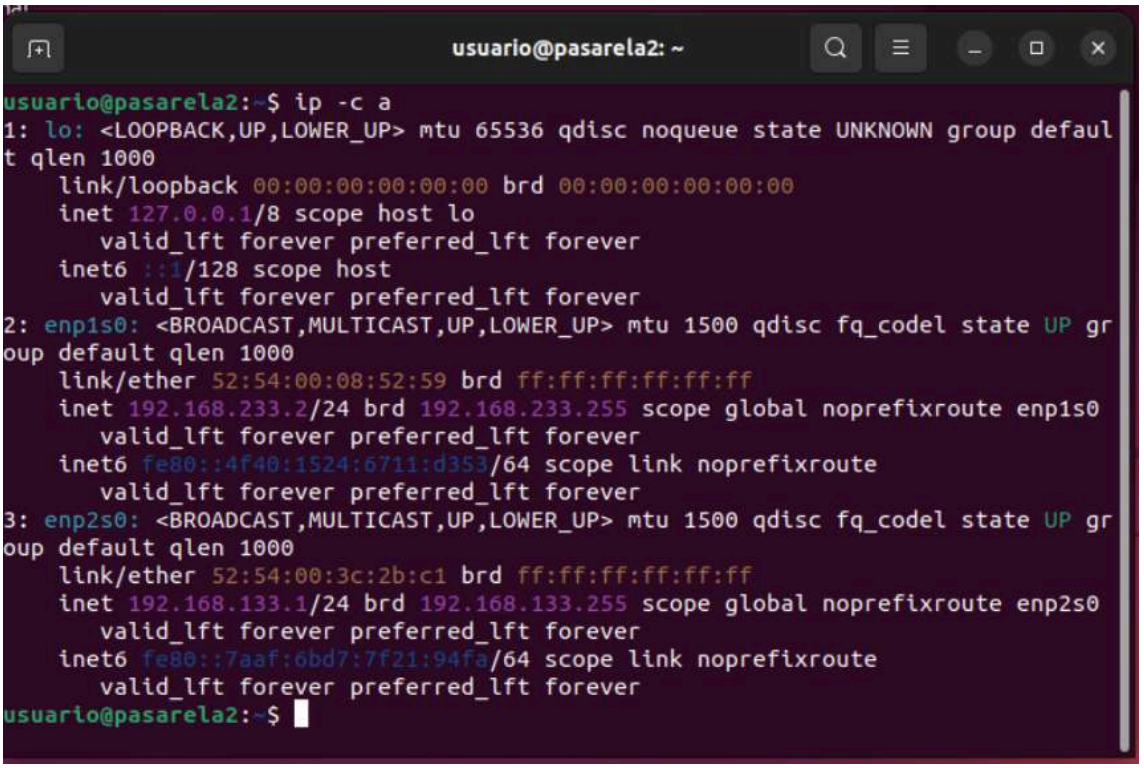
(enp1s0 es la red de la pasarela 1 que habrá que cambiarla a la 192.168.233.2 debido a que el DHCP ha dado concesión de ip)

(enp2s0 es la red interna que habrá que configurar)

Configuración de red de la pasarela 1:

Configuración de red de la red interna (pasarela 2):

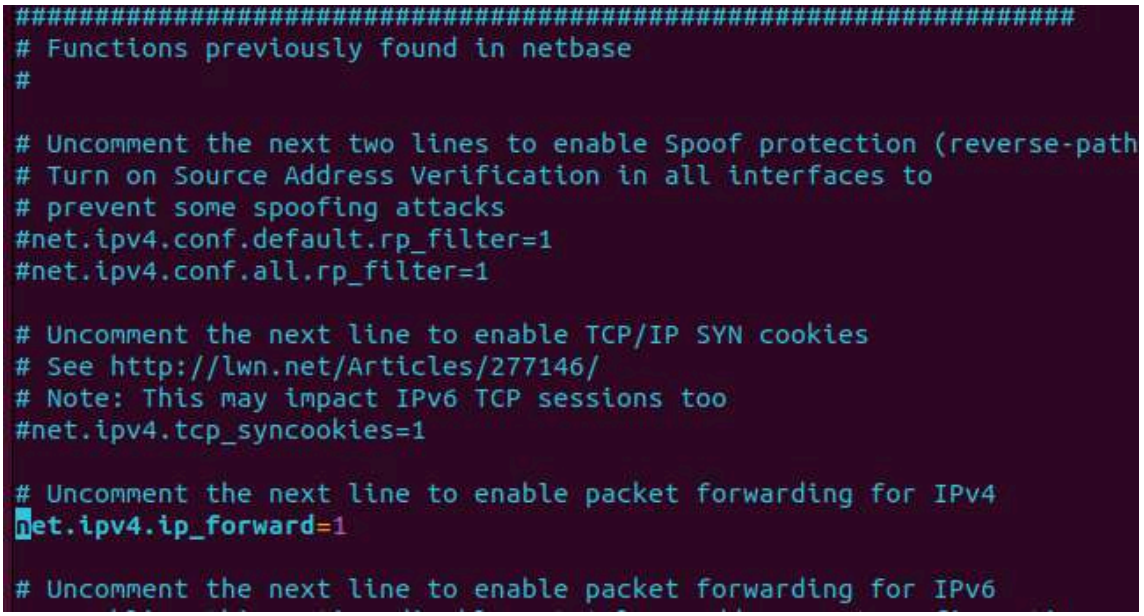
La pasarela 2 funcionará como Gateway para la red interna y tendrá la ip 192.168.133.1.



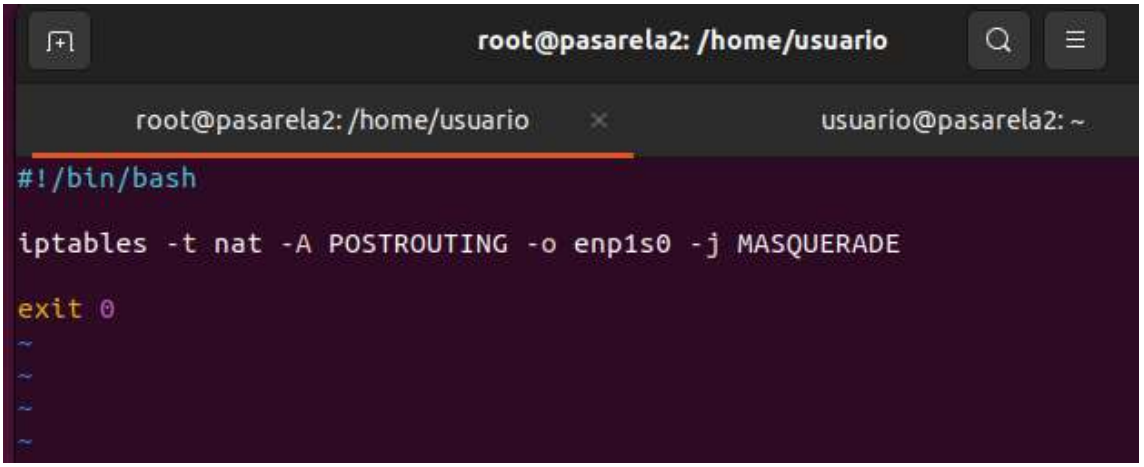
Antes que nada tenemos activar el reenvío de paquetes y utilizar una regla iptables la cual nos permite reenviar tráfico

Activar Forwarding:

En el fichero de configuración /etc/sysctl.conf descomentamos la línea #net.ipv4.ip_forward=1 para que haga forwarding de manera recursiva

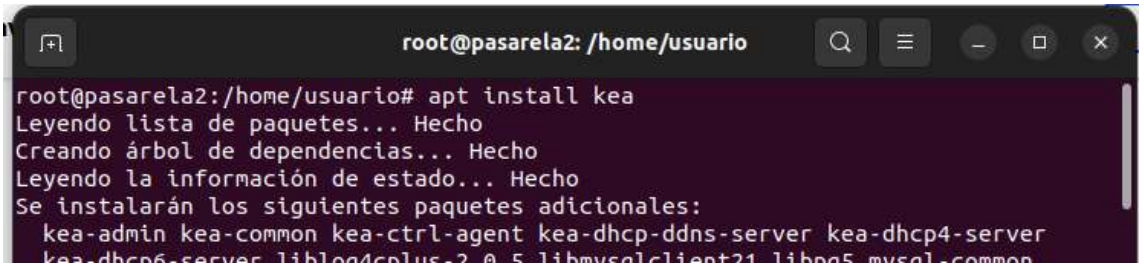


Iptables:



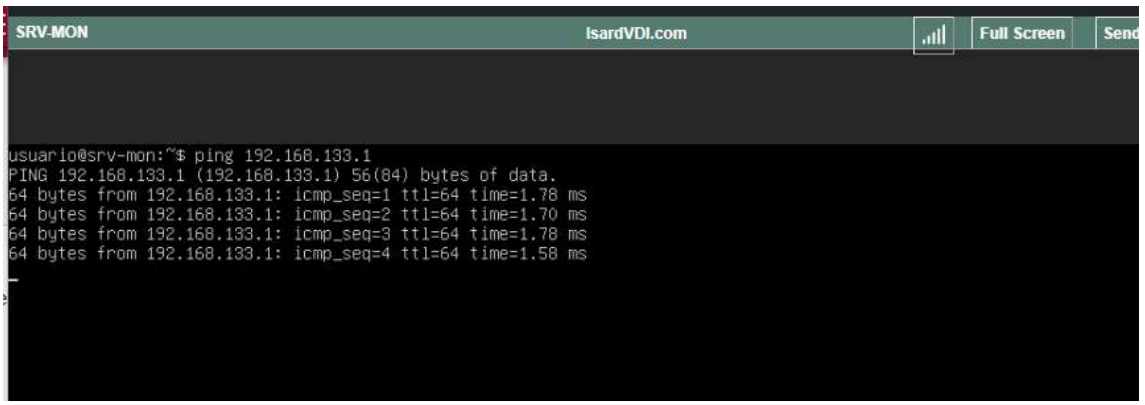
Una vez activado el reenvío de paquete tendremos que cambiar las ip de lo servidores que son el de Backup ,Monitorización y lo clientes ya que nos interesa que esto estén al red interna (Pasarela 2) y para

facilitar el proceso vamos a instalar un servicio DHCP y haremos un ping para comprobar la conectividad.



Esta es la configuración del dhcp y ahora haremos pruebas de conectividad.

Servidor Monitorización:



Servidor Backup:

```
root@srv-bkp:/home/usuario# ping 192.168.133.1
PING 192.168.133.1 (192.168.133.1) 56(84) bytes of data.
64 bytes from 192.168.133.1: icmp_seq=1 ttl=64 time=5.25 ms
64 bytes from 192.168.133.1: icmp_seq=2 ttl=64 time=1.70 ms
64 bytes from 192.168.133.1: icmp_seq=3 ttl=64 time=1.59 ms
```

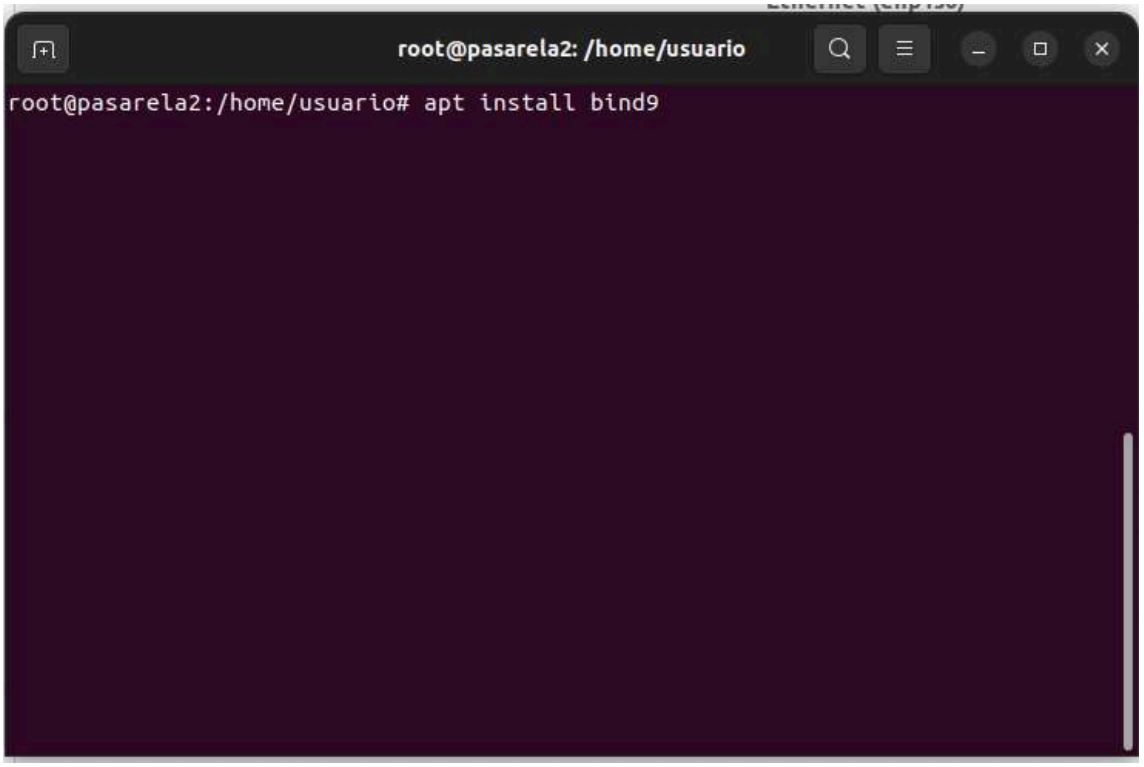
Cliente 1:

```
C:\Users\isard>ping 192.168.133.1

Haciendo ping a 192.168.133.1 con 32 bytes de datos:
Respuesta desde 192.168.133.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.133.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.133.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.133.1: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.133.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\isard>
```

Cliente 2:

Una vez configurado y cambiando las ip a los servidores y cliente que queremos que estén en la red interna tenemos que instalar un servidor dns dentro de la pasarela 2 ya que los dispositivos dentro de la red están configurados con el dns de la pasarela 2.

Instalaremos un DNS modo recursivo en la pasarela 2 utilizando bind9



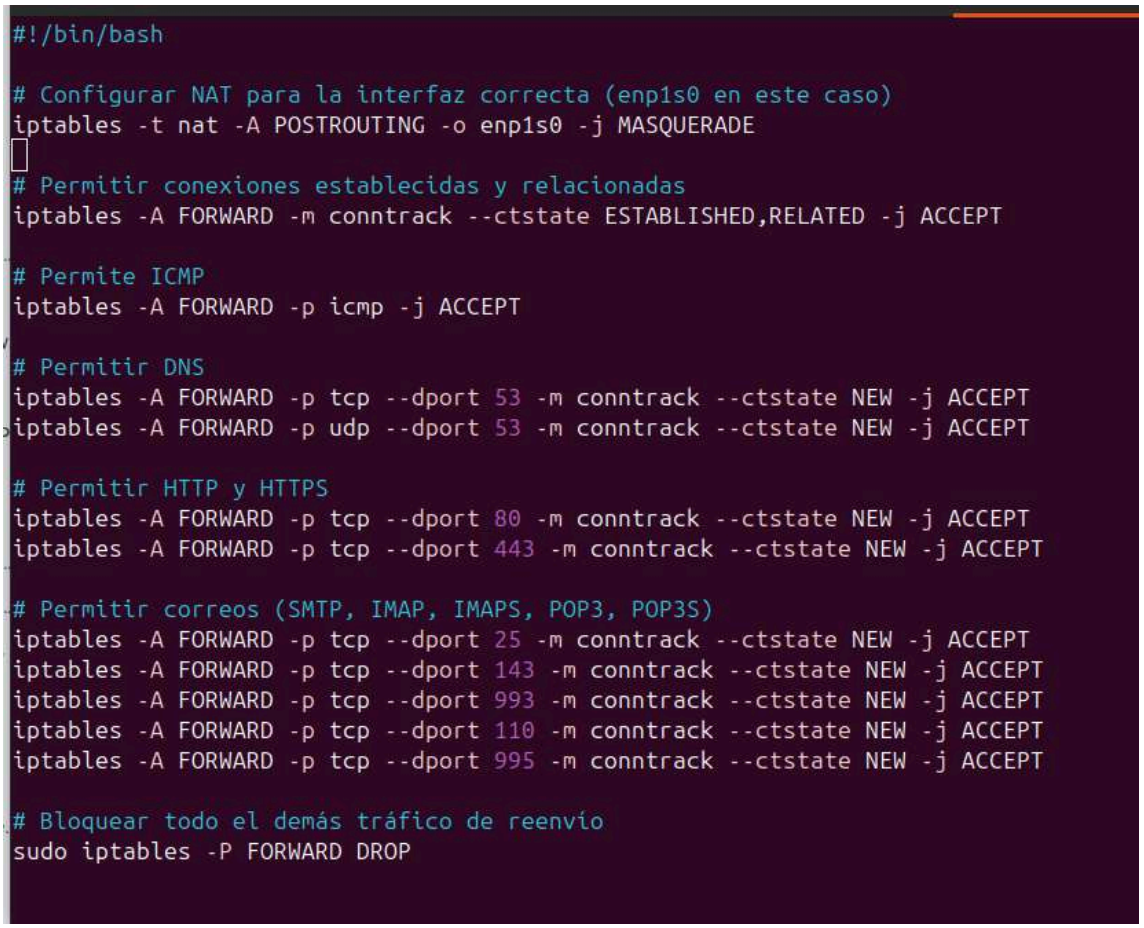
Después de haber instalado el DNS tenemos que declarar una ruta en la pasarela 1 y que la pasarela 1 no conoce a la red interna de la pasarela 2 por ello accederemos a la configuración del netplan y así podremos acceder a la red interna y viceversa.

Netplan Pasarela 1:

```
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp1s0:
      dhcp4: true
    enp2s0:
      addresses: [192.168.233.1/24]
      routes:
        - to: 192.168.133.0/24
          via: 192.168.233.2
```

Una vez creada la pasarela 2 y configurados la IP y DHCP en la pasarela, los servidores y clientes, y el DNS en la pasarela 2, además de haber activado el forwarding y configurado la regla de iptables para permitir NAT, y después de haber establecido una ruta en la pasarela 1 para llegar a la pasarela 2, el último paso será configurar nuestras políticas de firewall mediante iptables, tal como mencionamos anteriormente.

Configuración firewall Pasarela 1:



```
#!/bin/bash

# Configurar NAT para la interfaz correcta (enp1s0 en este caso)
iptables -t nat -A POSTROUTING -o enp1s0 -j MASQUERADE

# Permitir conexiones establecidas y relacionadas
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Permite ICMP
iptables -A FORWARD -p icmp -j ACCEPT

# Permitir DNS
iptables -A FORWARD -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

# Permitir HTTP y HTTPS
iptables -A FORWARD -p tcp --dport 80 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

# Permitir correos (SMTP, IMAP, IMAPS, POP3, POP3S)
iptables -A FORWARD -p tcp --dport 25 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p tcp --dport 143 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p tcp --dport 993 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p tcp --dport 110 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -p tcp --dport 995 -m conntrack --ctstate NEW -j ACCEPT

# Bloquear todo el demás tráfico de reenvio
sudo iptables -P FORWARD DROP
```

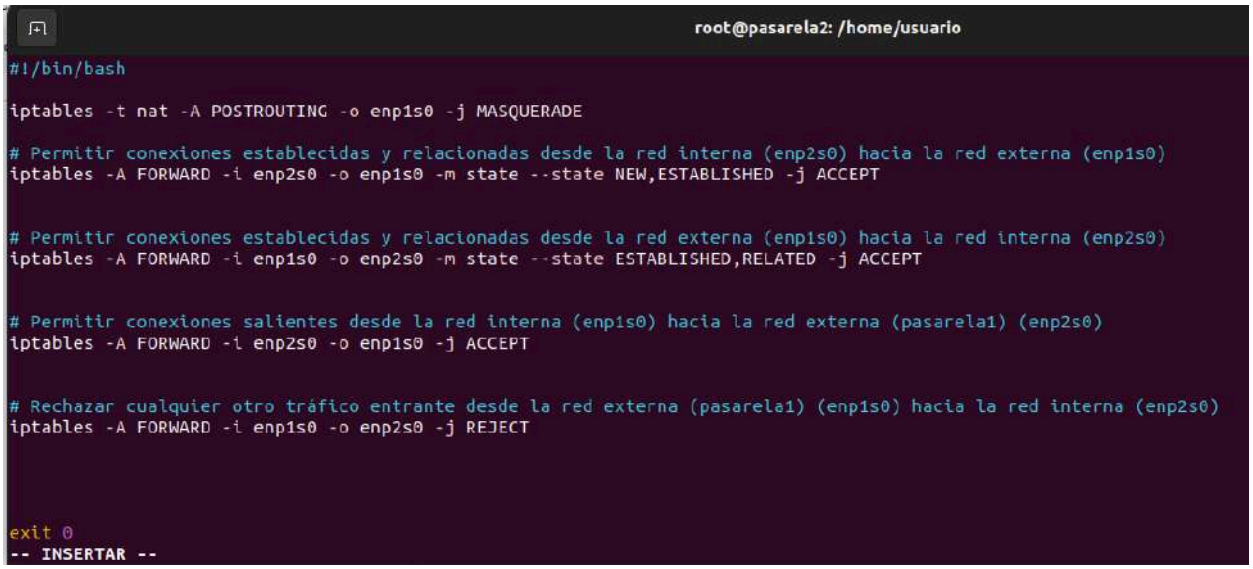
Configurar NAT para la interfaz correcta (enp1s0 en este caso)

iptables -t nat -A POSTROUTING -o enp1s0 -j MASQUERADE

Permitir conexiones establecidas y relacionadas

```
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Permite ICMP

iptables -A FORWARD -p icmp -j ACCEPT

# Permitir DNS

iptables -A FORWARD -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

# Permitir HTTP y HTTPS

iptables -A FORWARD -p tcp --dport 80 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

# Permitir correos (SMTP, IMAP, IMAPS, POP3, POP3S)

iptables -A FORWARD -p tcp --dport 25 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p tcp --dport 143 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p tcp --dport 993 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p tcp --dport 110 -m conntrack --ctstate NEW -j ACCEPT

iptables -A FORWARD -p tcp --dport 995 -m conntrack --ctstate NEW -j ACCEPT

# Bloquear todo el demás tráfico de reenvío

sudo iptables -P FORWARD DROP
```

Configuración firewall Pasarela 2 (Red Interna):



Permitir todo el tráfico desde la red 192.168.233.0/24 hacia cualquier destino

iptables -A FORWARD -s 192.168.233.0/24 -j ACCEPT

Permitir todo el tráfico hacia la red 192.168.233.0/24 desde cualquier origen

iptables -A FORWARD -d 192.168.233.0/24 -j ACCEPT

Regla de NAT para permitir enmascaramiento de las conexiones salientes en la interfaz enp1s0

iptables -t nat -A POSTROUTING -o enp1s0 -j MASQUERADE

Permitir conexiones establecidas y relacionadas desde la red interna (enp2s0) hacia la red externa (enp1s0)

iptables -A FORWARD -i enp2s0 -o enp1s0 -m state --state NEW,ESTABLISHED -j ACCEPT

Permitir conexiones establecidas y relacionadas desde la red externa (enp1s0) hacia la red interna (enp2s0)

iptables -A FORWARD -i enp1s0 -o enp2s0 -m state --state ESTABLISHED,RELATED -j ACCEPT

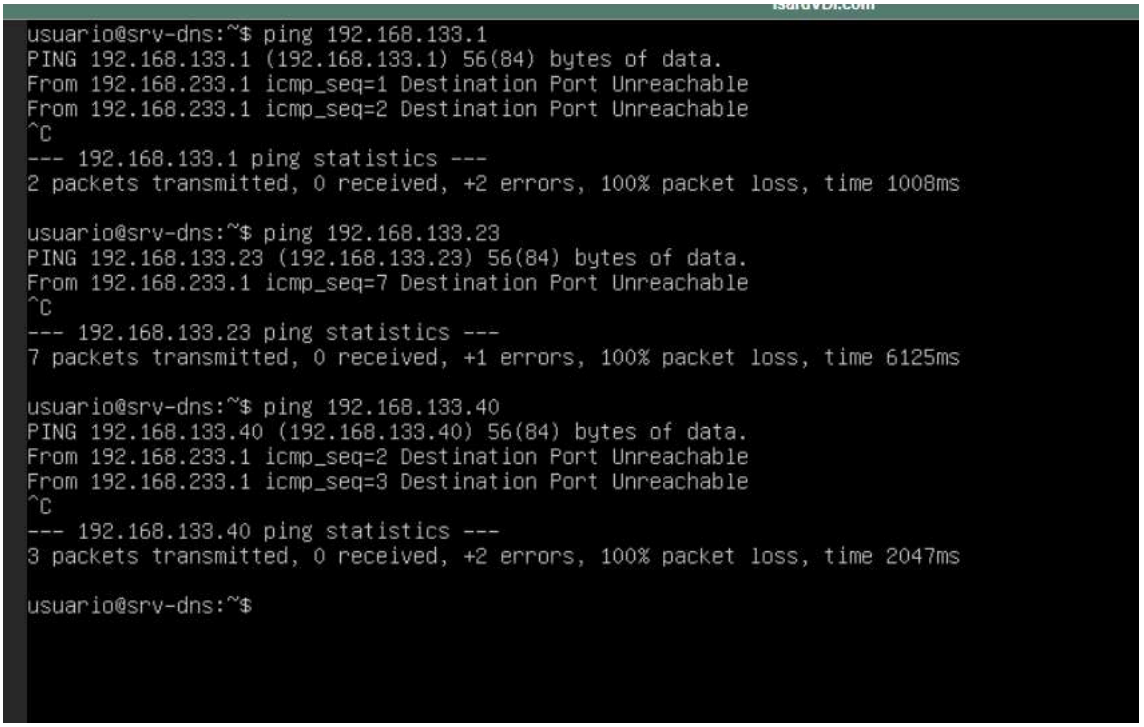
Permitir conexiones salientes desde la red interna (enp1s0) hacia la red externa (pasarela1) (enp2s0)

iptables -A FORWARD -i enp2s0 -o enp1s0 -j ACCEPT

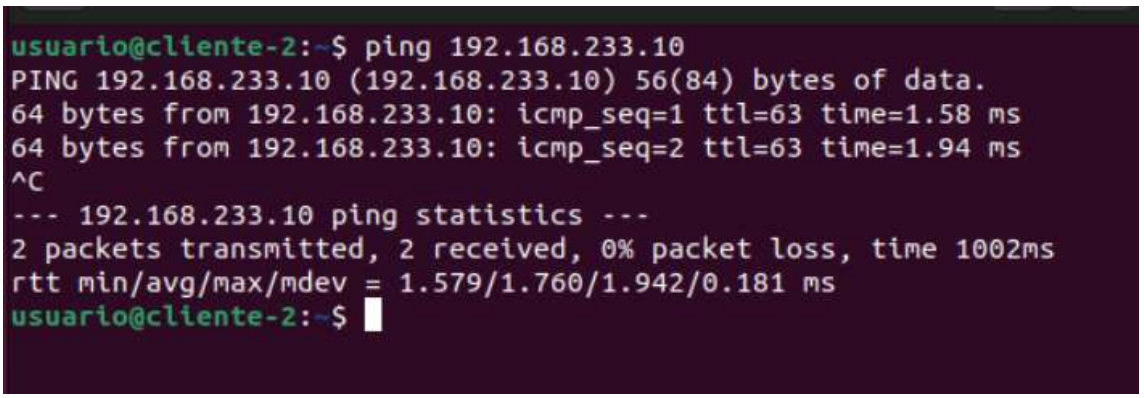
Rechazar cualquier otro tráfico entrante desde la red externa (pasarela1) (enp1s0) hacia la red interna (enp2s0)

iptables -A FORWARD -i enp1s0 -o enp2s0 -j REJECT

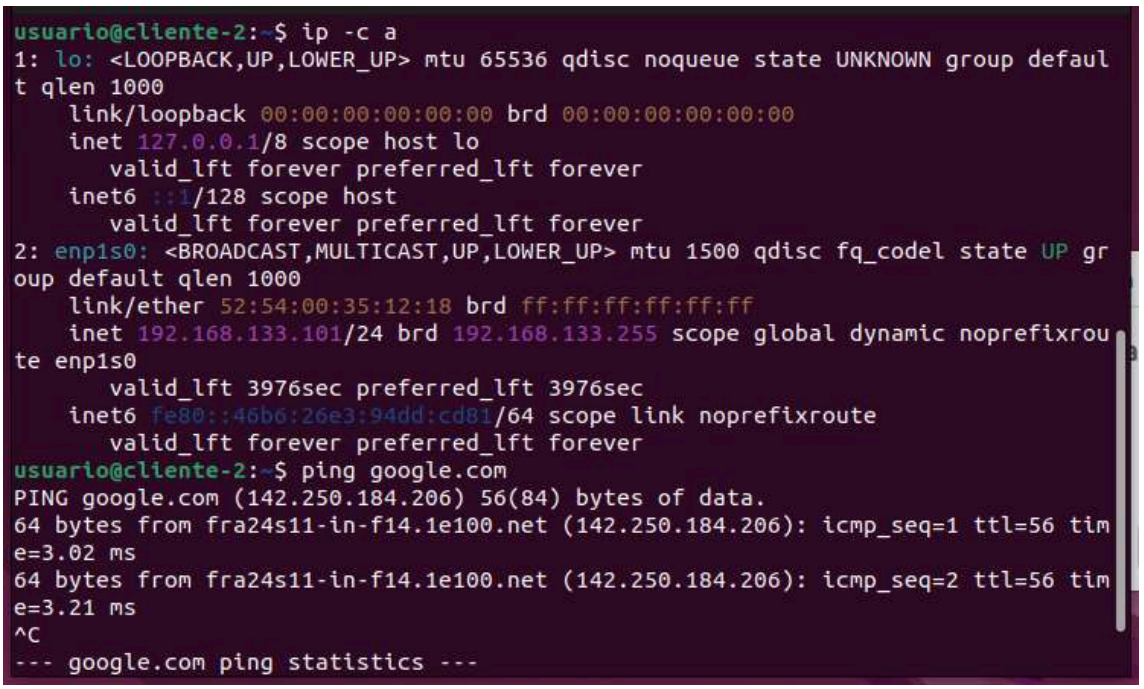
Después de esto podemos ver que desde la pasarela 1 ya no nos permite hacer un ping o acceder a cualquier dispositivo de la red debido a que no permite conexiones entrantes a menos de que se haya pedido antes así que sabemos que la principal función de la pasarela 2 es correcta.



En cambio desde la red interna si que deja realizar un ping al servidor que forma parte de la pasarela 1



También el cliente que está en la red interna nos permite salir hasta el exterior que como podemos ver nos deja hacer un ping a google.



Con el uso de reglas de iptables relativamente simples, hemos logrado mejorar la seguridad de nuestras redes de manera significativa. Nuestro enfoque se ha centrado en proteger la red interna de la pasarela 2. Para lograr esto, hemos implementado una arquitectura de red que incluye una pasarela entre la red interna y Internet.

Esta pasarela actúa como un punto de control que filtra y dirige el tráfico entre la red interna y externa. Además, hemos dejado en medio de esta arquitectura los servicios menos críticos. Esto significa que los servicios como el correo electrónico, la web y DNS se encuentran en una zona intermedia, accesibles tanto desde el exterior como desde el interior de la red, pero están protegidos por las reglas de iptables que hemos establecido.

En resumen, al implementar estas medidas de seguridad, hemos logrado crear un entorno donde el acceso desde el exterior está controlado y restringido según sea necesario, al mismo tiempo que garantizamos que los servicios esenciales estén disponibles para los usuarios internos. Esto evita que personas no autorizadas accedan a nuestra red interna, brindando un nivel de seguridad adicional a nuestra infraestructura.

Monitorización de los equipos de la red utilizando Grafana y Prometheus

Servicios

Para monitorizar toda la red vamos a usar los siguientes servicios.

- **Grafana**: Es una plataforma de visualización de datos que permite crear dashboards personalizados para monitorear y analizar métricas y registros. Se integra con múltiples fuentes de datos como Prometheus y Loki.
- **Prometheus**: Es un sistema de monitoreo que recopila y almacena métricas de sistemas y aplicaciones en tiempo real. Analizar y alerta sobre el estado de los sistemas monitoreados.

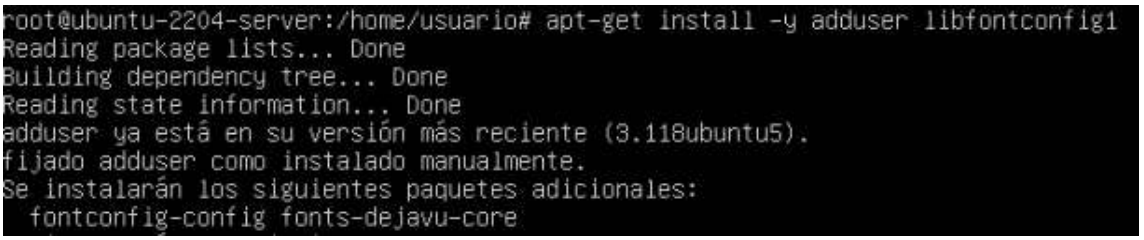
- **Loki**: Es un sistema de almacenamiento y búsqueda de registros, lee los logs y permite subirlos a tu grafana. Permite almacenar grandes volúmenes de registros de manera eficiente y realizar búsquedas rápidas para la resolución de problemas.

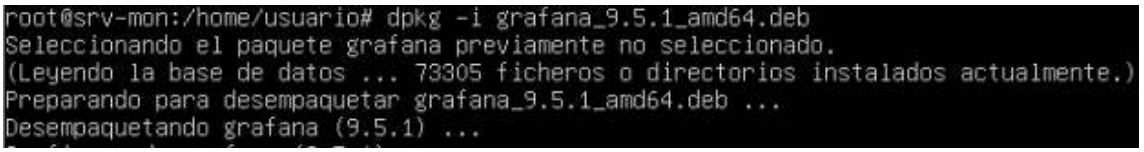
Instalación

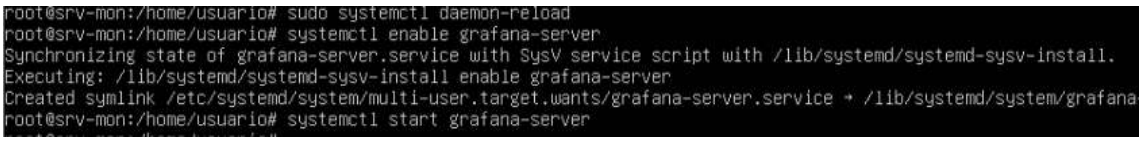
Comenzaremos con Grafana, después con Prometheus, y por último seguiremos con Loki

Instalación Grafana

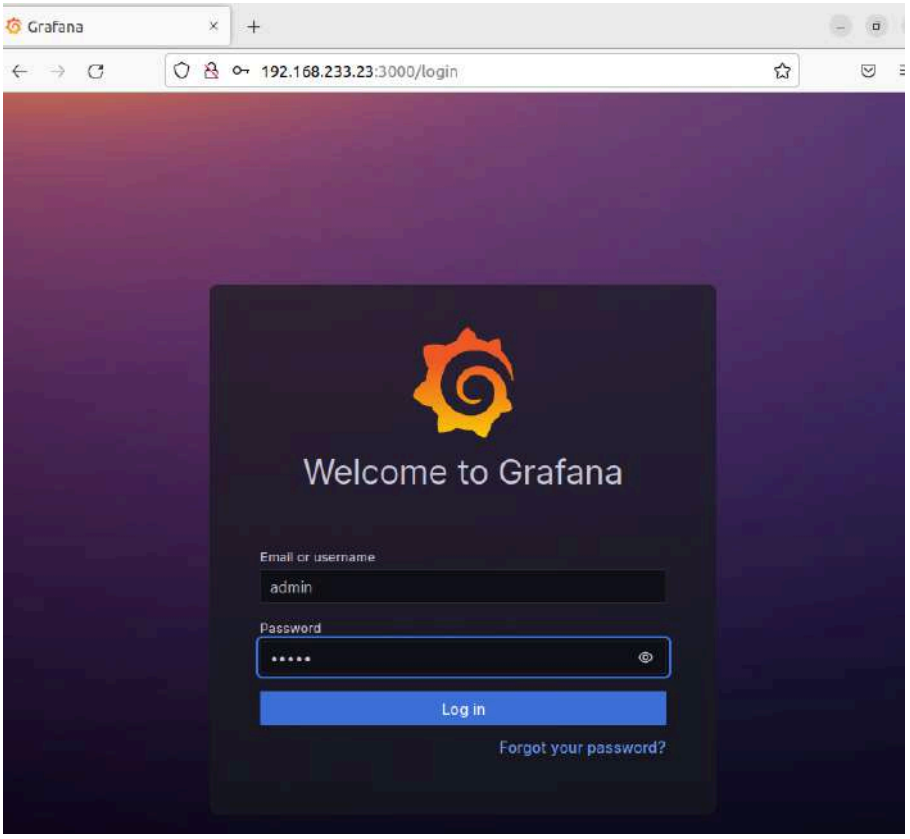
Debido a que no está en los repositorios de Ubuntu, hay que descargarlo mediante comandos.



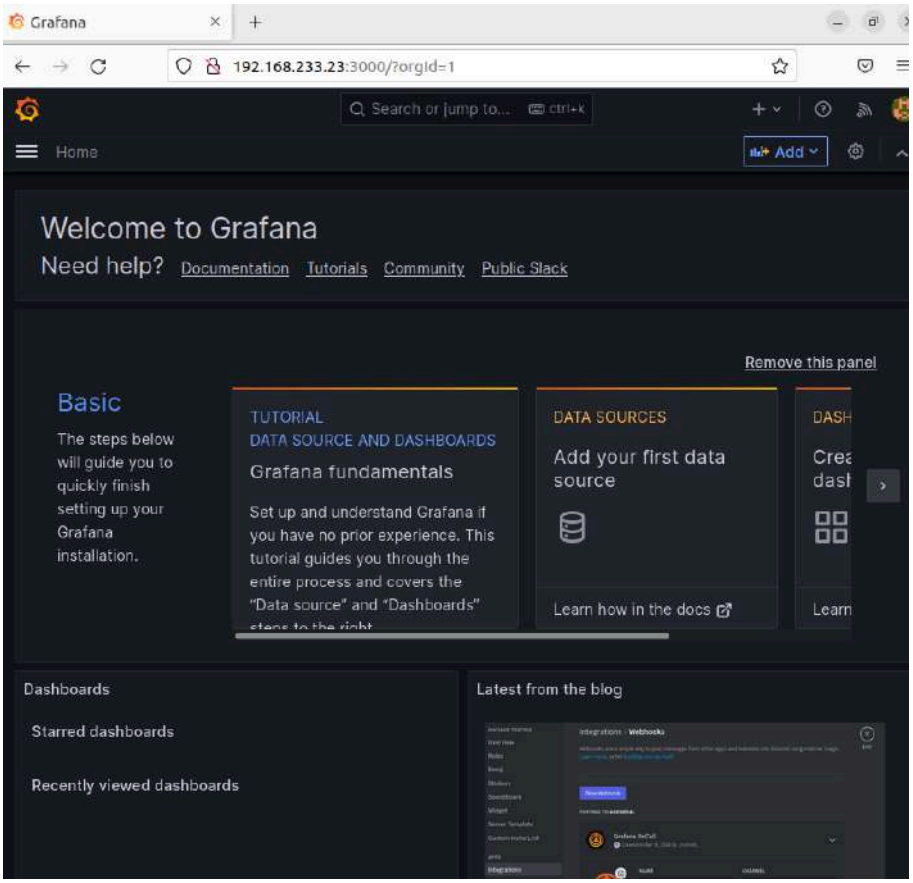




Encontraremos la interfaz web de la aplicación en el puerto 3000, en nuestro caso http://192.168.233.23:3000/ (Está IP se queda momentáneamente, está susceptible a cambios en el futuro)
Por defecto la contraseña es admin admin.



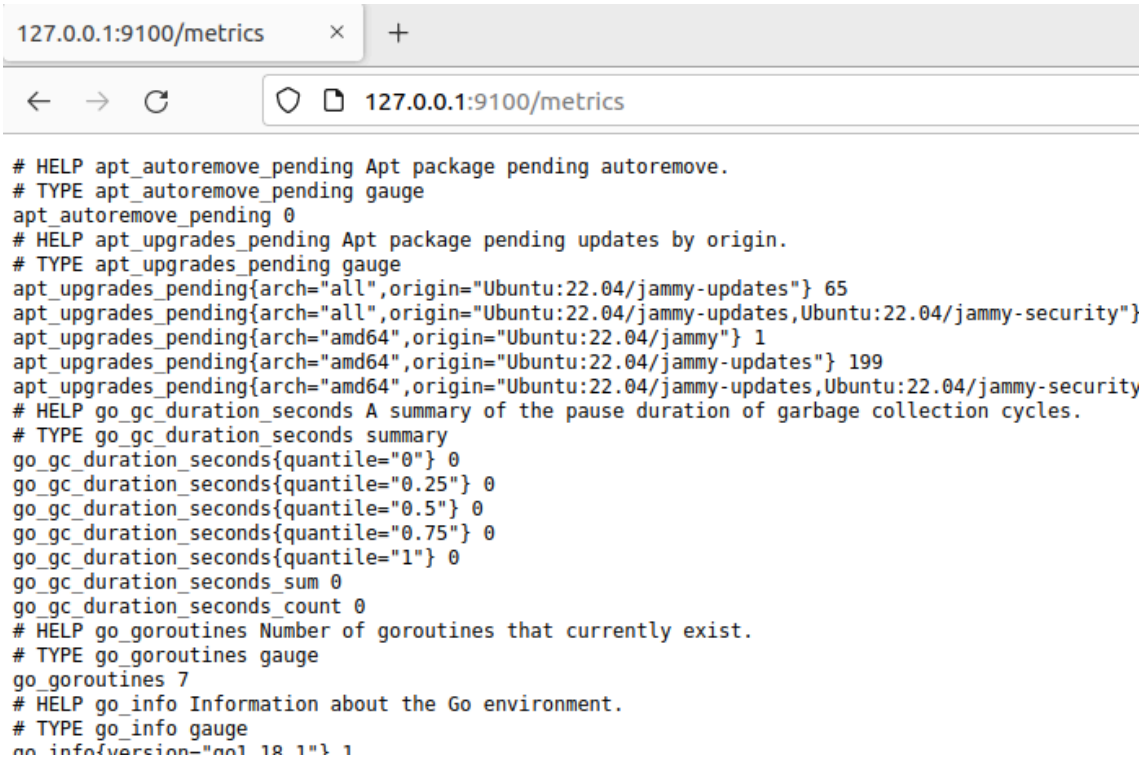
Y ya tendremos el Grafana concluido satisfactoriamente



Prometheus y prometheus-node-exporter

Empezamos descargando prometheus-node-exporter en las máquinas donde queramos recoger las métricas (por defecto las métricas estarán en el puerto 9100 de esa máquina)

```
root@cliente-1:/home/usuario# apt install prometheus-node-exporter
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libio-pty-perl libipc-run-perl libtime-duration-perl moreutils
  prometheus-node-exporter-collectors smartmontools
```



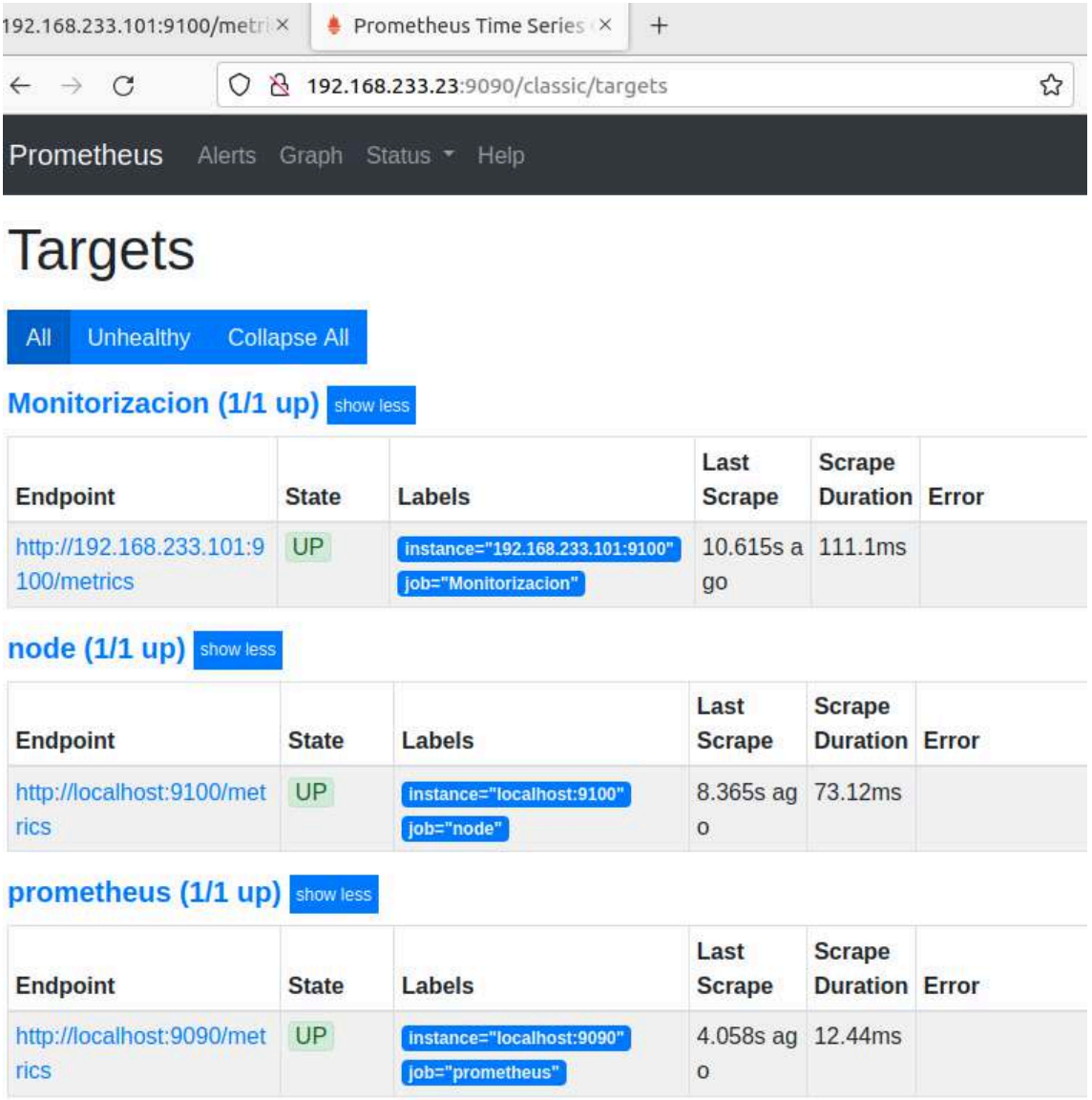
Ahora continuamos con prometheus, que se descarga de una manera bastante sencilla en la máquina servidor

```
root@srv-mon:/home/usuario# apt install prometheus
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se instalarán los siguientes paquetes adicionales:
  fonts-glyphicons-halflings javascript-common libio-pty-pe
```

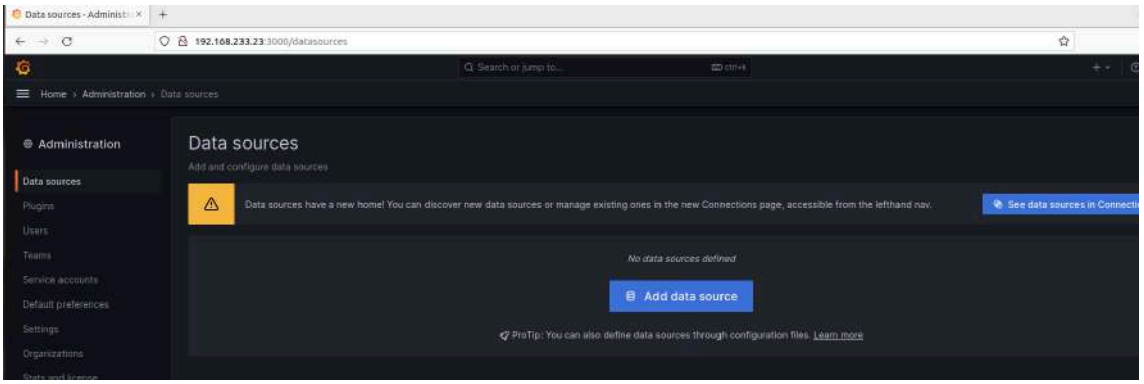
Una vez descargado, entramos en el archivo de configuración (/etc/prometheus/prometheus.yml) y creamos un nuevo trabajo dentro de “scrape:_configs” en el cual ponemos los hosts que queramos recoger sus métricas

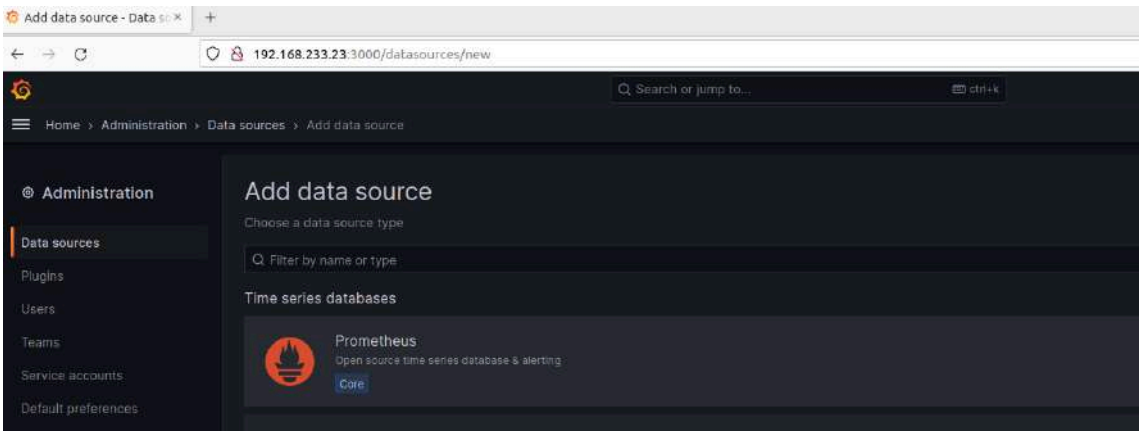
```
- job_name: 'Monitorizacion'
  static_configs:
    - targets: ['192.168.233.101:9100']
```

Y ya lo podemos comprobar

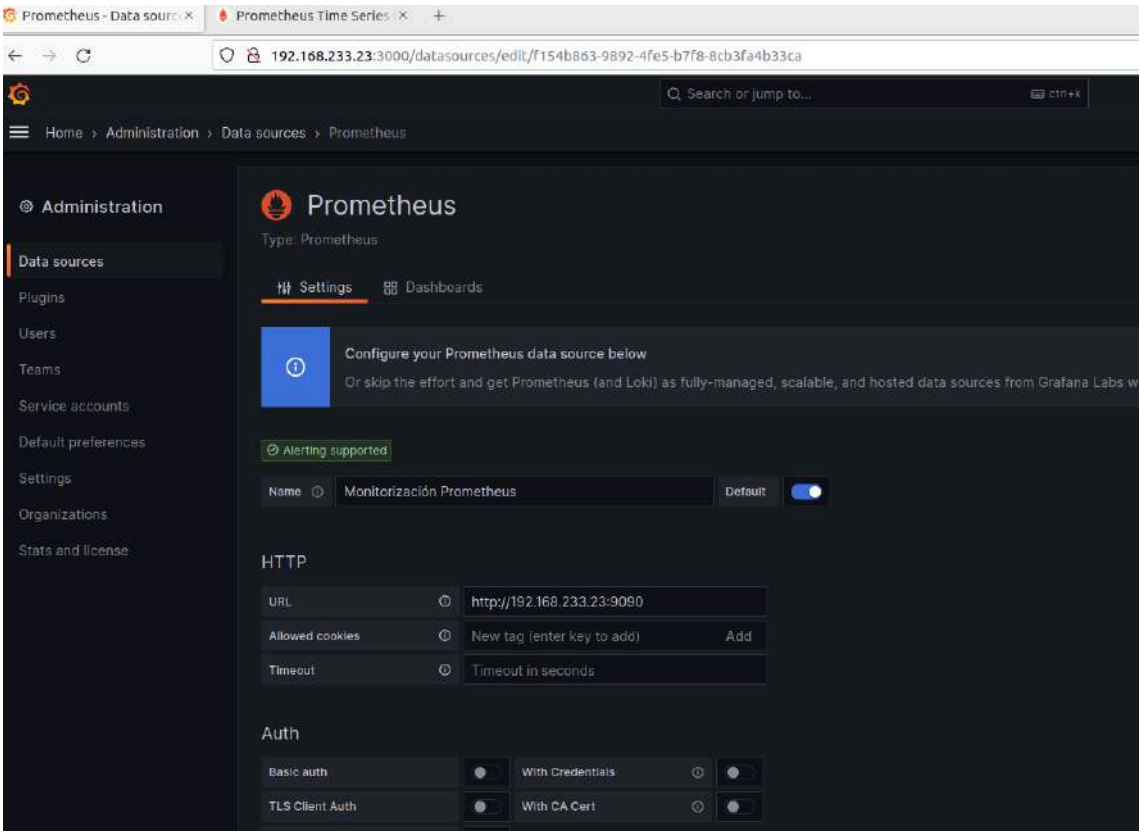


Ahora, dentro de la configuración del grafana, vamos a añadir prometheus de la siguiente manera:
Entramos en Home> Administration > Data sources y clickeamos en Add data source





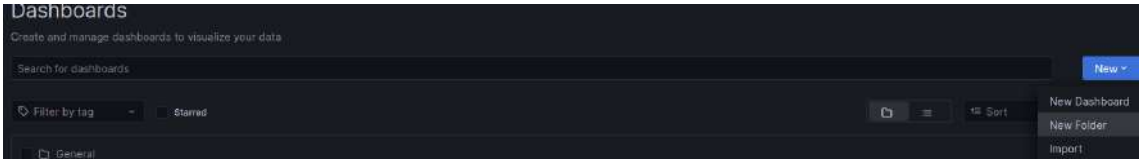
Y pulsamos en prometheus



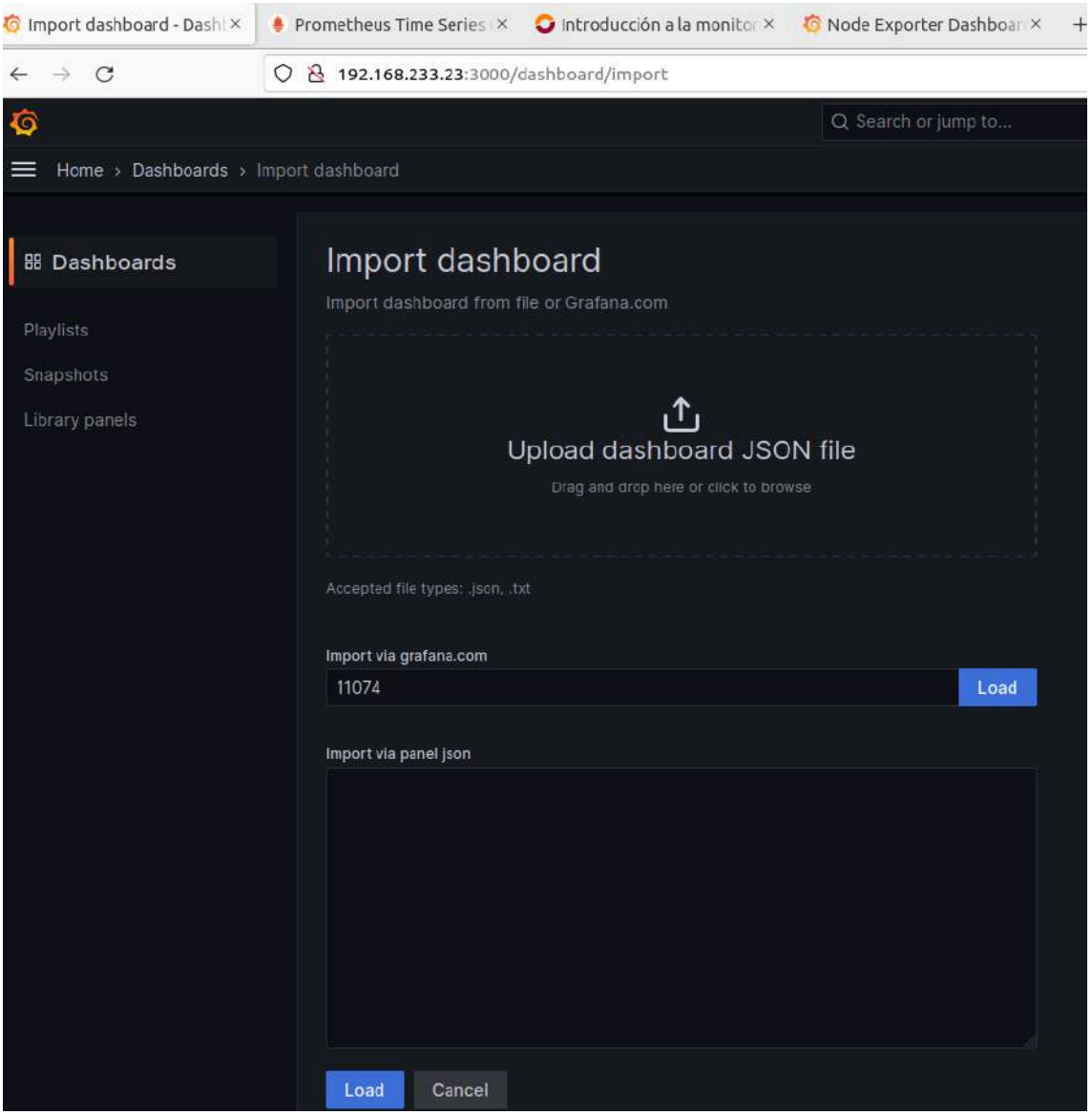
Y ponemos los ajustes que veamos necesarios (en este caso la URL solamente, ya que sin la URL no funcionaria)

Ahora vamos a importar un Dashboard ya hecho. (https://grafana.com/grafana/dashboards/11074-node-exporter-for-prometheus-dashboard-en-v20201010/)

Vamos a Dashboard

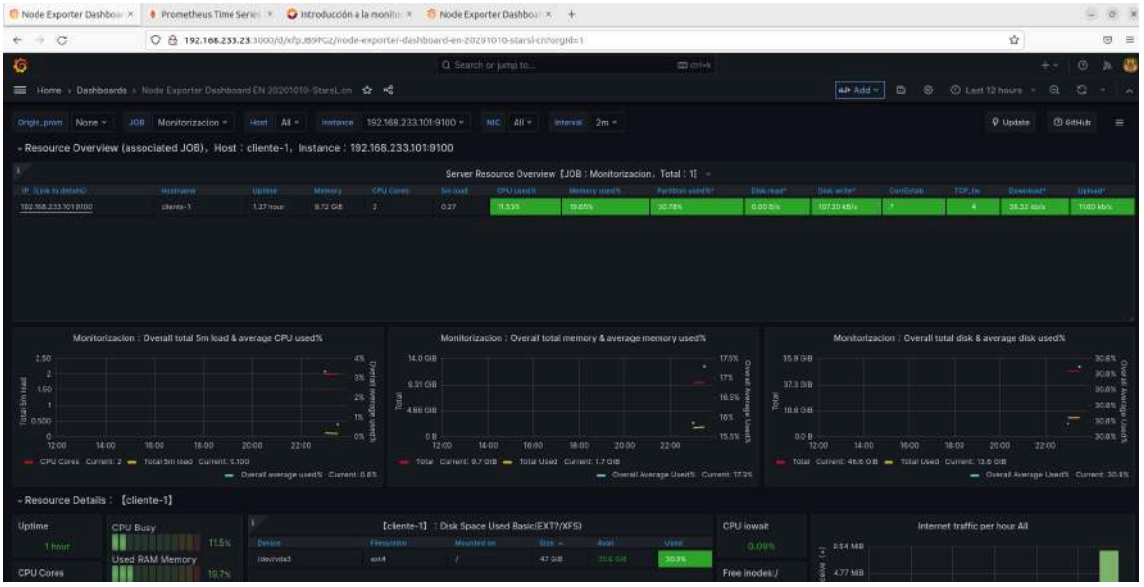


Le damos a Import

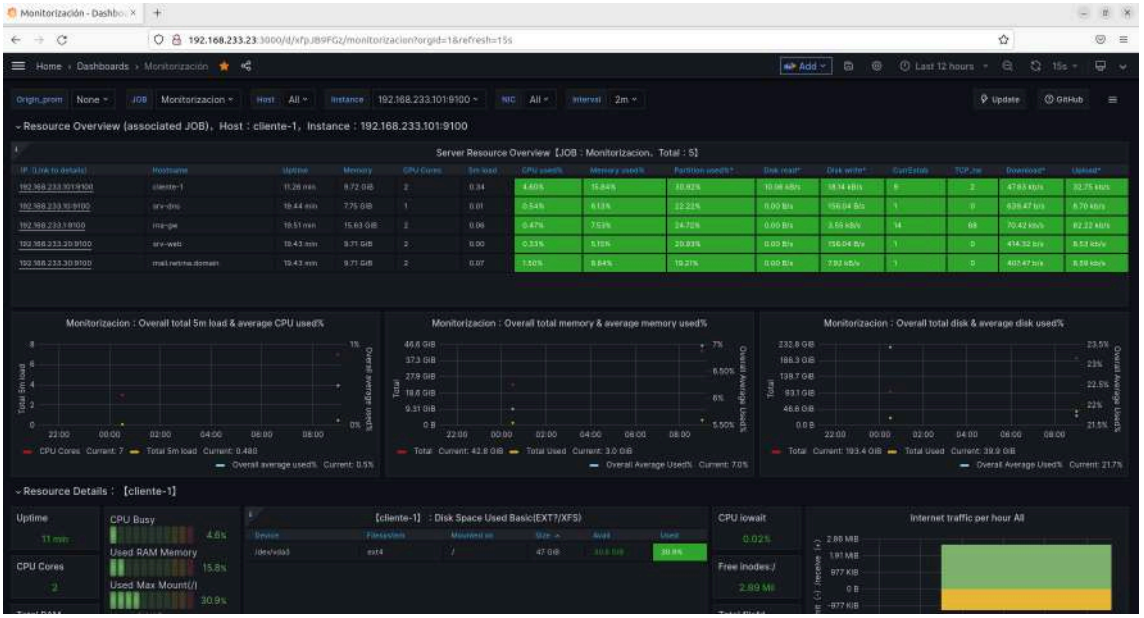


Y importamos la ID

Y así quedaría



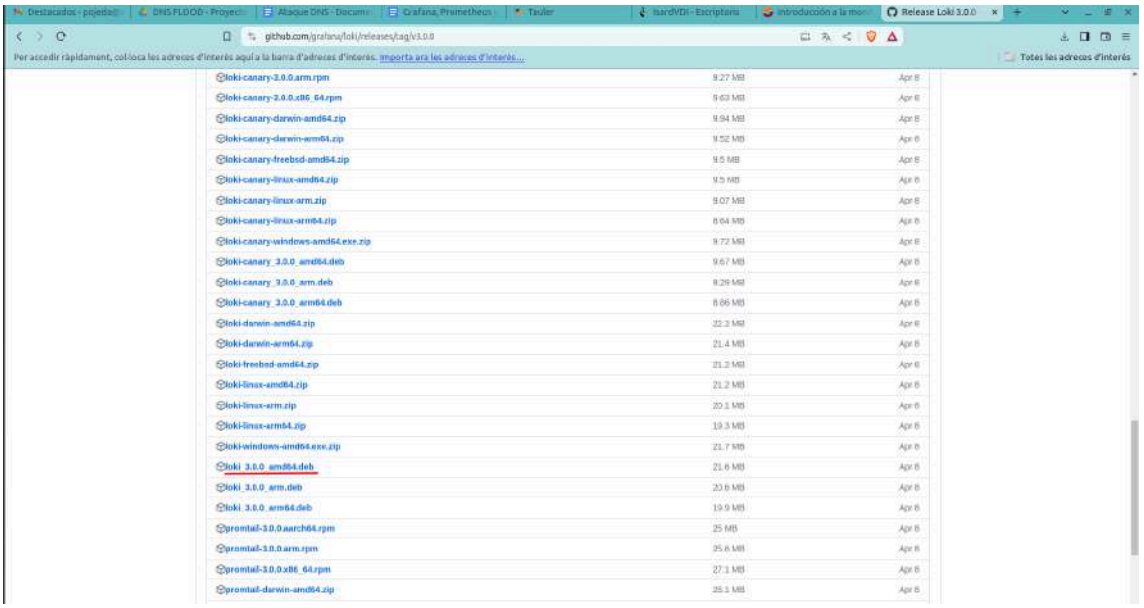
Ya deberíamos hacer un “apt install prometheus-node-exporter” en todas las máquinas y servidores que queríamos.
Y en el servidor de monitorización, en el archivo de configuración, prometheus.yml, ponemos todas las ip de todas las máquinas.
Y con todas las máquinas quedaría así:



Prometheus Finalizado 👍

Loki

Ahora vamos a descargar loki en el servidor de monitorización
https://github.com/grafana/loki/releases/download/v3.0.0/loki_3.0.0_amd64.deb



Buscamos el github de loki para poder descargarlo

```
root@srv-mon:/home/usuario# wget https://github.com/grafana/loki/releases/downlo
ad/v3.0.0/loki_3.0.0_amd64.deb
--2024-05-22 08:58:45--  https://github.com/grafana/loki/releases/download/v3.0.
0/loki_3.0.0_amd64.deb
```

Y ya lo instalamos

```
root@srv-mon:/home/usuario# apt install ./loki_3.0.0_amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Nota, seleccionando «loki» en lugar de «./loki_3.0.0_amd64.deb»
```

Ahora, entramos a la configuración de loki

```
root@srv-mon:/home/usuario# vi /etc/loki/config.yml
```

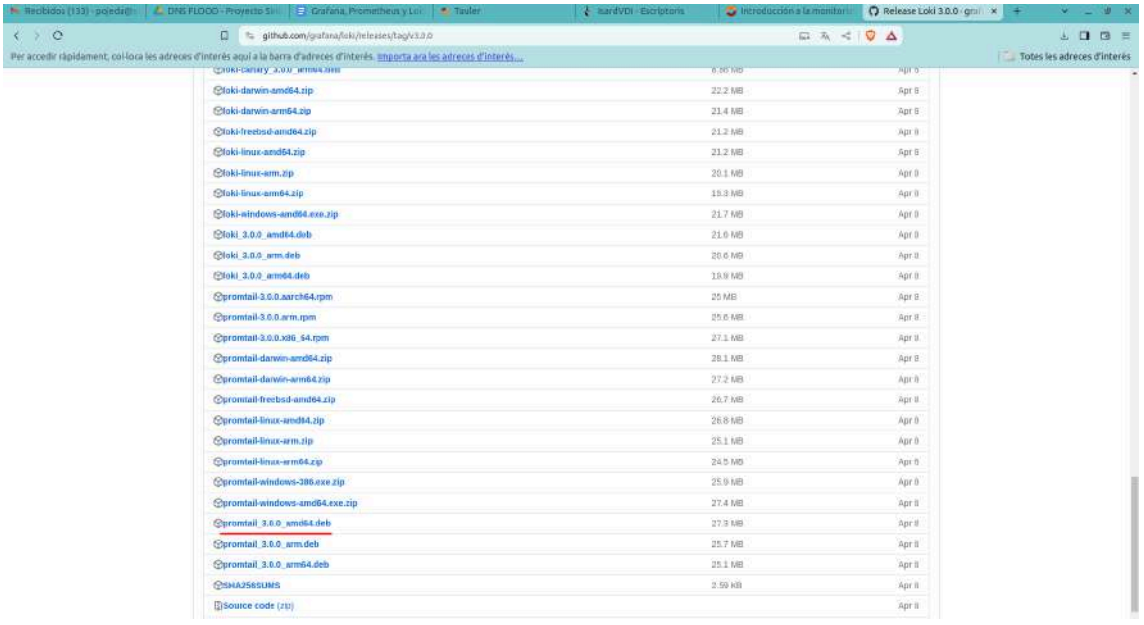
Y cambiamos los path a /var/loki

```
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

common:
  instance_addr: 127.0.0.1
  path_prefix: /var/loki
  storage:
    filesystem:
      chunks_directory: /var/loki/chunks
      rules_directory: /var/loki/rules
  replication_factor: 1
  ring:
```

Cabe aclarar que hay que poner los permisos bien en esa carpeta

```
root@srv-mon:/home/usuario# chown loki /var/loki
root@srv-mon:/home/usuario#
root@srv-mon:/home/usuario# chown loki /var/loki/chunks/
root@srv-mon:/home/usuario# chown loki /var/loki/rules/
root@srv-mon:/home/usuario# chown loki.loki /var/loki
chown: invalid user: ‘loki.loki’
root@srv-mon:/home/usuario# cd /var/loki/
root@srv-mon:/var/loki# ll
total 32
drwxr-xr-x  8 loki root    4096 may 23 09:52 ./
drwxr-xr-x 14 root root    4096 may 22 11:35 ../
drwxr-xr-x  2 loki root    4096 may 22 15:51 chunks/
drwxr-xr-x  2 loki nogroup 4096 may 23 09:52 compactor/
drwxr-xr-x  2 loki root    4096 may 22 15:51 rules/
drwxr-xr-x  7 loki nogroup 4096 may 23 09:52 tsdb-shipper-active/
drwxr-xr-x  2 loki nogroup 4096 may 23 09:52 tsdb-shipper-cache/
drwxr-xr-x  2 loki nogroup 4096 may 23 09:52 wal/
```

Una vez con esto, descargamos promtail https://github.com/grafana/loki/releases/download/v3.0.0/promtail_3.0.0_amd64.deb




```
root@srv-mon:/home/usuario# wget https://github.com/grafana/loki/releases/downlo
ad/v3.0.0/promtail_3.0.0_amd64.deb
--2024-05-22 09:54:27--  https://github.com/grafana/loki/releases/download/v3.0.
0/promtail_3.0.0_amd64.deb
```

Y lo instalamos

```
root@srv-mon:/home/usuario# apt install ./promtail_3.0.0_amd64.deb
```

Ahora después de descargarlo, tenemos que añadir el usuario promtail al grupo adm en /etc/group.

```
root@srv-mon:/home/usuario# vi /etc/group
root@srv-mon:/home/usuario# cat /etc/group | grep adm
adm:x:4:syslog,usuario,promtail
root@srv-mon:/home/usuario#
```

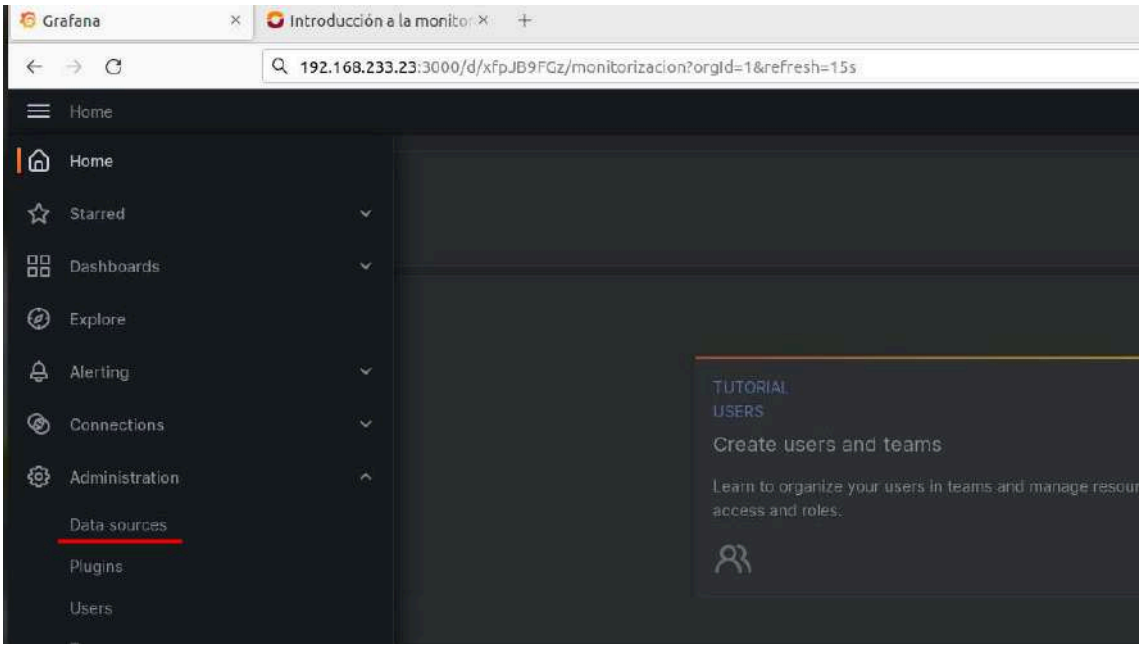
Entramos al archivo de configuración de promtail en /etc/promtail y lo configuramos a nuestra manera

```
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
- url: http://192.168.233.23:3100/loki/api/v1/push

scrape_configs:
- job_name: system
  static_configs:
  - targets:
      - localhost
    labels:
      job: varlogs
      #NOTE: Need to be modified to scrape any additional logs of the system.
      __path__: /var/log/*log
```

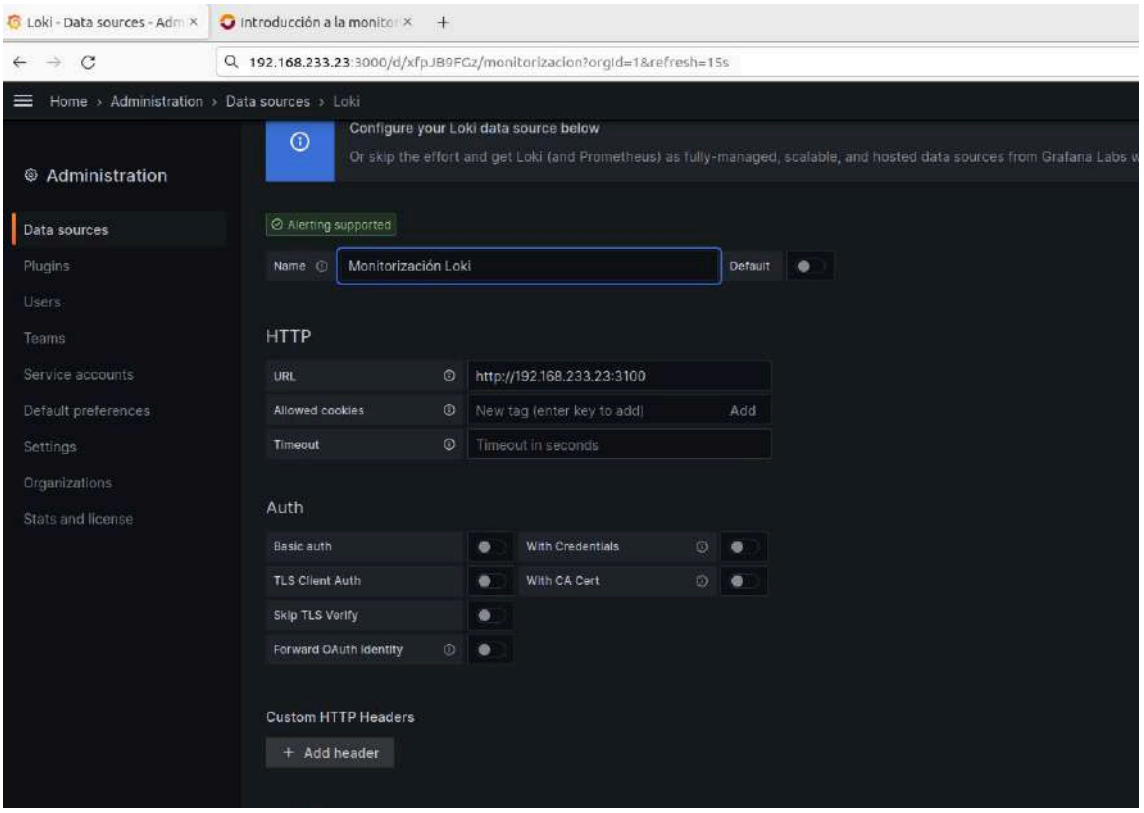
Y también añadimos un par de líneas nuevas para añadir también el journal junto a los logs

```
- job_name: journal
  journal:
    max_age: 12h
    path: /var/log/journal
    labels:
      job: systemd-journal
  relabel_configs:
    - source_labels: ['__journal__systemd_unit']
      target_label: 'unit'
    - source_labels:
      - __journal__hostname
      target_label: nodename
```

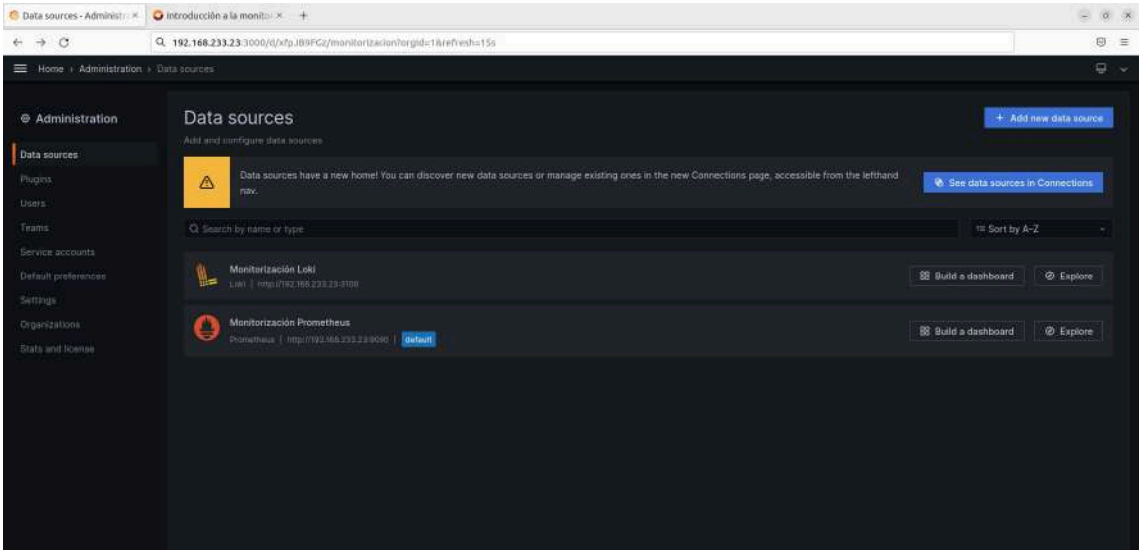
Ahora, vamos a configurar el Loki en Grafana:



Añadimos una nueva database (Loki)

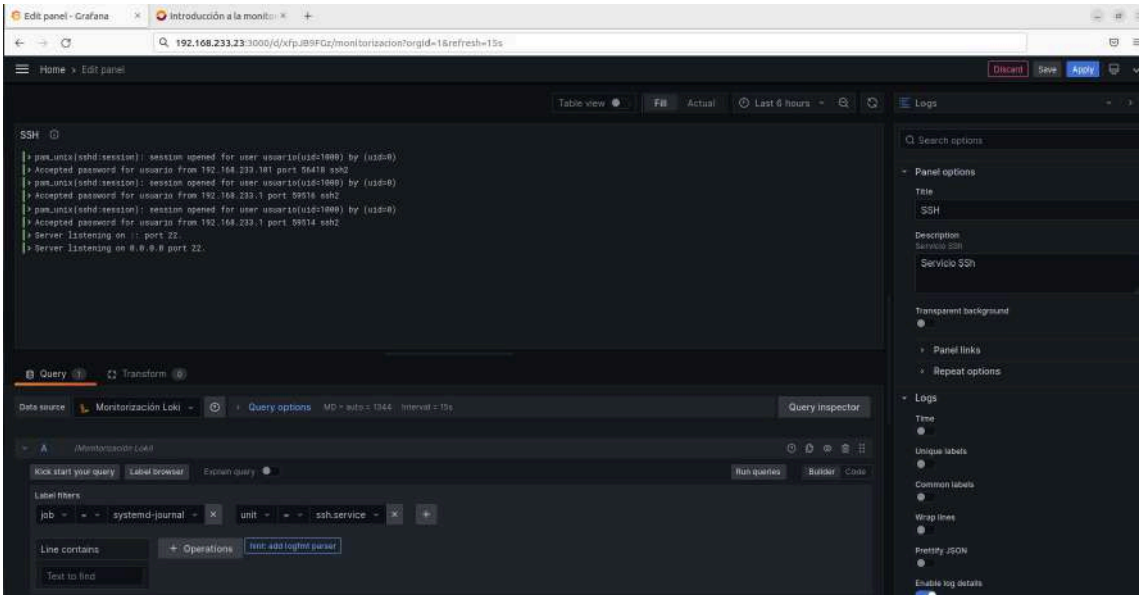


Y ya tendremos las dos (Prometheus y Loki)



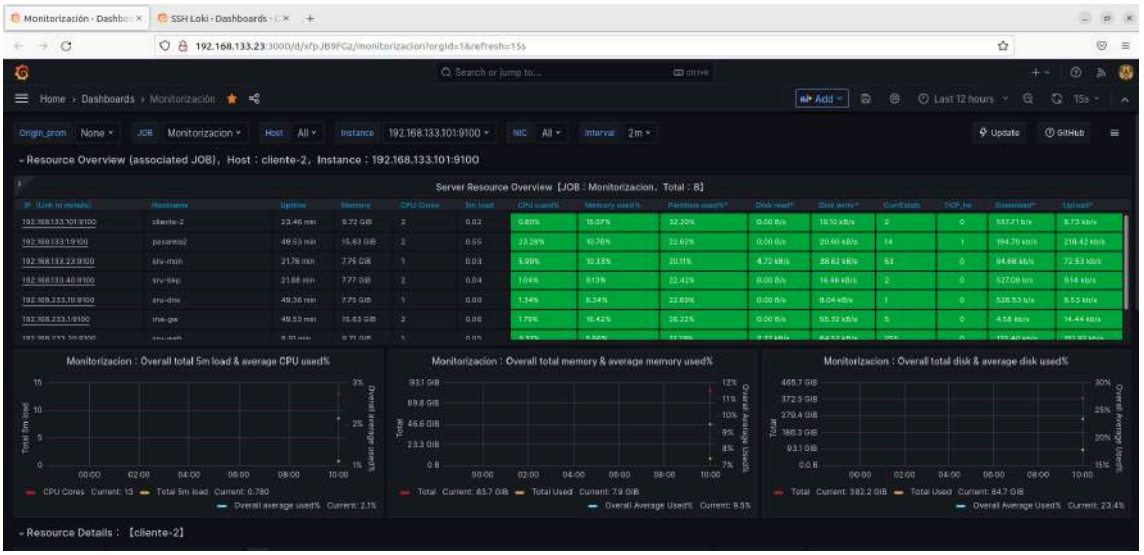
Y ya podemos hacer el grafana con loki

En este caso, vamos a ponerlo un panel sobre ssh



Dashboards

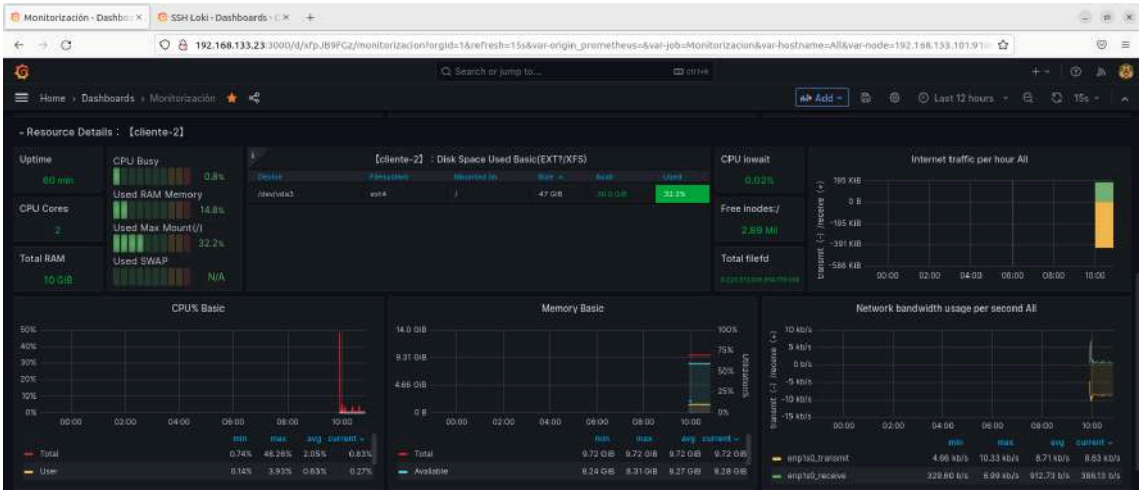
Prometheus



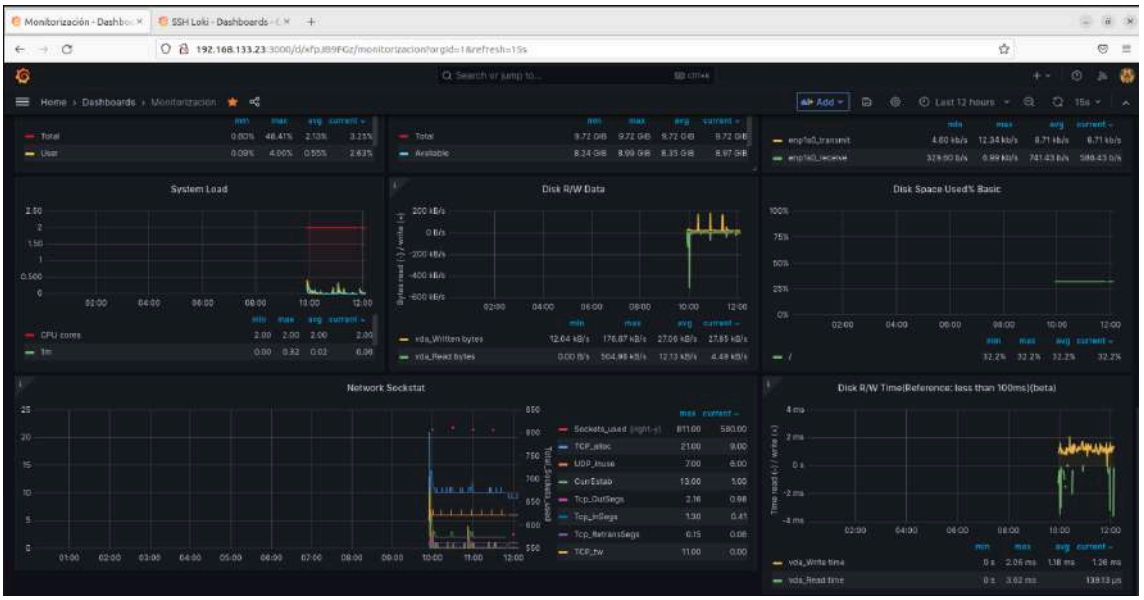
En la primera tabla explica todo sobre una máquina en concreto, la cual pone el nombre y la ip a la izquierda, para que puedas identificarla, y explica los siguiente:

1. CPU used%: El porcentaje de uso de la CPU.
2. Memory used%: El porcentaje de uso de la memoria RAM.
3. Partition used%: El porcentaje de uso de las particiones de disco.
4. Disk read/write: La velocidad de lectura/escritura en el disco.
5. CurrEstab: El número de conexiones TCP actualmente establecidas.
6. TCP_tw: El número de conexiones TCP en estado TIME_WAIT.
7. Download/Upload: La velocidad de descarga y subida de datos.

Ya abajo hay 3 tablas que explican el porcentaje de cpu usado y total, el porcentaje de memoria usada y total, y el porcentaje de disco usado y total.



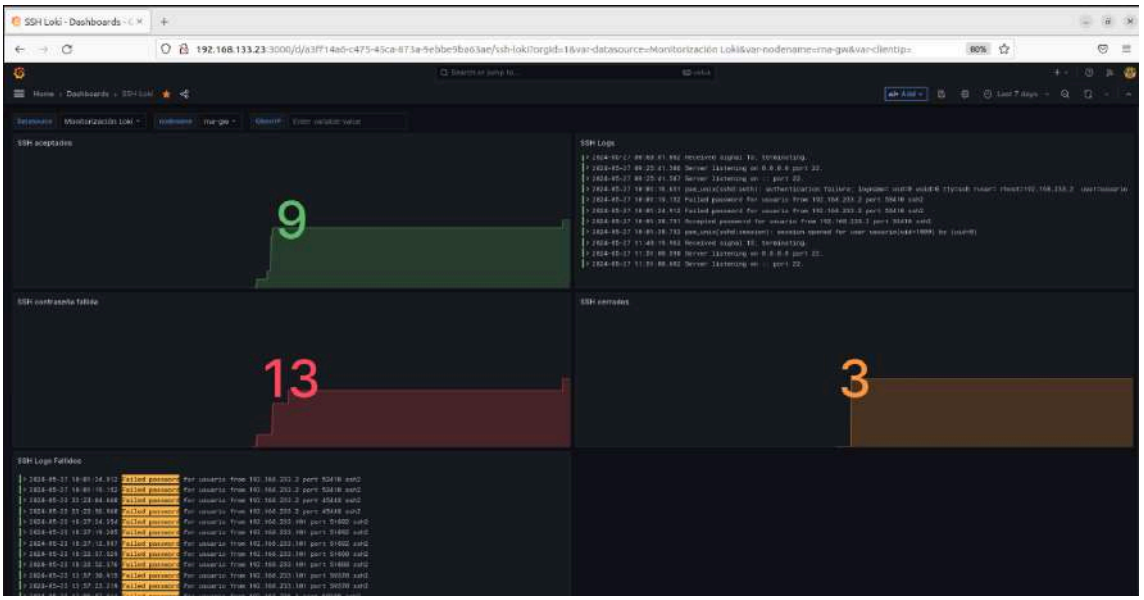
Aquí explica todo respecto el cliente 2 (cpu, memoria, discos)
Ya a la derecha pone el tráfico en internet por hora en megas
Ya abajo pone el uso de cpu en porcentaje, la memoria total, usada y disponible y por último el ancho de banda usado por segundo.



Arriba a la izquierda sale la carga del sistema en cpu, después en el medio sale una gráfica de lectura y escritura del disco en bytes, y a la derecha sale cuanto porcentaje del disco se ha utilizado, y cuánto hay libre.

Abajo salen todas las estadísticas de los sockets de red como los números de sockets usados, socket TCP asignados y UDP en uso, segmentos TCP enviados y recibidos. Y a la derecha sale la lectura y escritura del disco en tiempo.

Loki



En este dashboard sale toda la información sobre los ssh.
Arriba puedes elegir la máquina a la que mirar los ssh.
El número verde son los ssh que le hacen a esa máquina en concreto que aceptan.
El número rojo son los ssh que le hacen a esa máquina en concreto que falla la contraseña.
El número naranja son los ssh que le hacen a esa máquina en concreto que se cierran.
Arriba a la derecha te salen todos los logs del ssh.
Y abajo a la izquierda te sale todo el log de los ssh con contraseña fallida.

4. Dificultades que nos hemos encontrado a la hora de hacer proyecto

En nuestro proyecto, como en cualquier otro, hemos enfrentado varios desafíos técnicos y logísticos que hemos tenido que superar. A continuación, detallo algunos de los principales problemas y cómo los resolvemos:

1. **Configuración del servidor proxy:**

   Problema: No sabíamos en qué puerto debía funcionar el servidor proxy, lo que causaba que no se conectará correctamente.

   Solución: Tuvimos que investigar a fondo sobre la configuración de puertos para proxies. Consultamos documentación técnica y foros especializados para identificar el puerto correcto. Finalmente, configuramos el servidor en el puerto adecuado y verificamos su funcionamiento.

2. **Bloqueo de puertos por iptables:**

   Problema: Las iptables de nuestro sistema estaban bloqueando ciertos puertos que necesitábamos abiertos para la comunicación del servidor proxy y otros servicios.

   Solución: Revisamos y ajustamos las reglas de iptables para permitir el tráfico a través de los puertos necesarios. Esto implicó aprender sobre la configuración de iptables y aplicar las reglas adecuadas para nuestro entorno.

3. **Desarrollo de un bot de Telegram:**

   Problema: No tuvimos suficiente tiempo para desarrollar un bot de Telegram que actuara como asistente, una funcionalidad que considerábamos valiosa para el proyecto.

   Solución: Priorizamos otras tareas críticas para el proyecto, con la intención de retomar el desarrollo del bot en una fase posterior. Esto nos enseñó a gestionar mejor nuestro tiempo y a ser realistas con los plazos.

4. **Espacio insuficiente en máquinas virtuales:**

   Problema: Nos encontramos con la limitación de espacio en las máquinas virtuales, lo que impedía la instalación y ejecución de algunas aplicaciones necesarias.

   Solución: Optimizamos el uso del espacio existente eliminando archivos innecesarios y configuraciones redundantes. Además, solicitamos ampliaciones de espacio al administrador del sistema cuando fue necesario.

5. **Configuración del DNS:**

   Problema: Tuvimos problemas con la configuración del DNS, lo que afectaba la resolución de nombres y, por ende, la comunicación entre diferentes componentes del proyecto.

   Solución: Realizamos una revisión exhaustiva de la configuración del DNS. Consultamos documentación y ejemplos de configuración para asegurarnos de que todos los registros y parámetros fueran correctos. Ajustamos las configuraciones hasta que logramos una resolución de nombres consistente y precisa.

A pesar de estos desafíos, logramos superarlos utilizando una combinación de recursos:

Trabajo en grupo: Colaboramos estrechamente para dividir las tareas y compartir conocimientos.

Investigación en Internet: Utilizamos foros, tutoriales, y documentación técnica disponible en línea.

Ayuda de los profesores: Aprovechamos la experiencia y orientación de nuestros profesores para resolver problemas complejos y obtener nuevas perspectivas.

Este proceso de resolución de problemas nos ha permitido aprender y crecer tanto individualmente como en equipo, mejorando nuestras habilidades técnicas y de colaboración.

5. Pàgina web

Hemos desarrollado una página web para presentar nuestro proyecto de manera clara y accesible. En esta página, compartimos detalles sobre quiénes somos como equipo, los servicios especializados que ofrecemos en ciberseguridad, así como información sobre nuestros productos destacados. Además, hemos incluido una sección de preguntas frecuentes para abordar cualquier duda que puedan tener nuestros clientes. Esta plataforma digital tiene como objetivo ofrecer una visión completa de nuestra iniciativa, permitiendo a los usuarios conocer más sobre nuestro enfoque, valores y cómo podemos contribuir a fortalecer la seguridad digital a las empresas.

ACCEDER A LA PÁGINA WEB

6. Conclusiones

Conclusiones generales del proyecto

Este proyecto ha sido una experiencia increíble para nosotros, estudiantes apasionados por la ciberseguridad. A través de JPA Cybersecurity Coop, liderado por Pau, Anass y Justin, hemos aprendido mucho sobre la importancia de proteger la información en el mundo digital de hoy. Nos emociona ver cómo nuestros esfuerzos pueden hacer que las empresas sean más seguras y protegidas contra posibles ataques cibernéticos.

Consecución de los objetivos

Estamos orgullosos de decir que hemos logrado alcanzar los objetivos que nos propusimos al inicio del proyecto. Desde el principio, nuestro objetivo fue establecer una empresa especializada en Pentesting y ofrecer servicios de seguridad de calidad. A través de nuestra dedicación y trabajo en equipo, hemos cumplido este objetivo y

hemos superado nuestras propias expectativas al identificar y solucionar vulnerabilidades en sistemas y redes.

Valoración de la metodología y planificación

A lo largo del proyecto, hemos aprendido la importancia de una buena planificación y una metodología sólida. Trabajar juntos como equipo ha sido fundamental para nuestro éxito. Mantenernos disciplinados y flexibles nos ha permitido adaptarnos a los desafíos que surgieron en el camino y cumplir con los plazos establecidos.

Visión de futuro

Mirando hacia adelante, estamos emocionados por las oportunidades que se presentan para nosotros como emprendedores en el campo de la ciberseguridad. Hemos adquirido habilidades valiosas y estamos listos para enfrentar los desafíos futuros con confianza. Nos comprometemos a seguir desarrollando soluciones innovadoras para proteger a nuestros clientes en un mundo digital cada vez más complejo y cambiante.

7. Bibliografia

https://elpuig.xeill.net/Members/vcarceler

https://www.youtube.com/watch?v=ER9S6sI-QLI&list=PLUs9Ztsn4LSqJOx3XzSsVG8flUCKLF97b&index=2

https://github.com/DNS-OARC/sample-query-data

https://grafana.com/grafana/dashboards/11074-node-exporter-for-prometheus-dashboard-en-v20201010/

https://github.com/grafana/loki/releases/download/v3.0.0/promtail_3.0.0_amd64.deb

https://bytelearning.blogspot.com/2016/11/como-configurar-una-dmz-con-linux.html

https://netcloudengineering.com/realizar-mitm-ettercap/

https://elpuig.xeill.net/Members/jordifarrero/2014-15-seguretat-en-xarxes-sm2ab-diurn/uf2-scripts-demo/thor-hammer-python

https://elpuig.xeill.net/Members/vcarceler/articulos/pruebas-de-rendimiento-de-un-servidor-dns-con-dnsperf-y-resperf

https://www.youtube.com/watch?v=43wbfCsFefg

https://www.youtube.com/watch?v=ER9S6sI-QLI

https://elpuig.xeill.net/Members/vcarceler/articulos/dhcp-con-kea

https://elpuig.xeill.net/Members/vcarceler/c1/didactica/apuntes/ud4/na8

https://elpuig.xeill.net/Members/vcarceler/articulos/squid/index_html

https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-a-apache-http-server

https://elpuig.xeill.net/Members/vcarceler/articulos/correo-electronico-con-postfix-dovecot-y-thunderbird-en-ubuntu-20.04

https://elpuig.xeill.net/Members/vcarceler/articulos/un-mua-web-roundcube

8. Annexos

Proyecto Síntesis (Parte EIE):

https://docs.google.com/document/d/1dBp593Rraaf2jJagH1QemaFlbkoGhWsJdEKvaFYkX5o/edit?usp=drive_link

Estatutos:

https://drive.google.com/file/d/1KpXlXZL-gRaqMfNejYf5TRpG9585A4op/view?usp=sharing

Canal de Youtube con los videos explicativos:

https://www.youtube.com/@JPACybersecurity