

Curso: COMP 2700 – Ciberseguridad

Laboratorio: Laboratorio 2 - Identificación de Riesgos y Aplicación de Controles Básicos en Debian

Sección: 92249

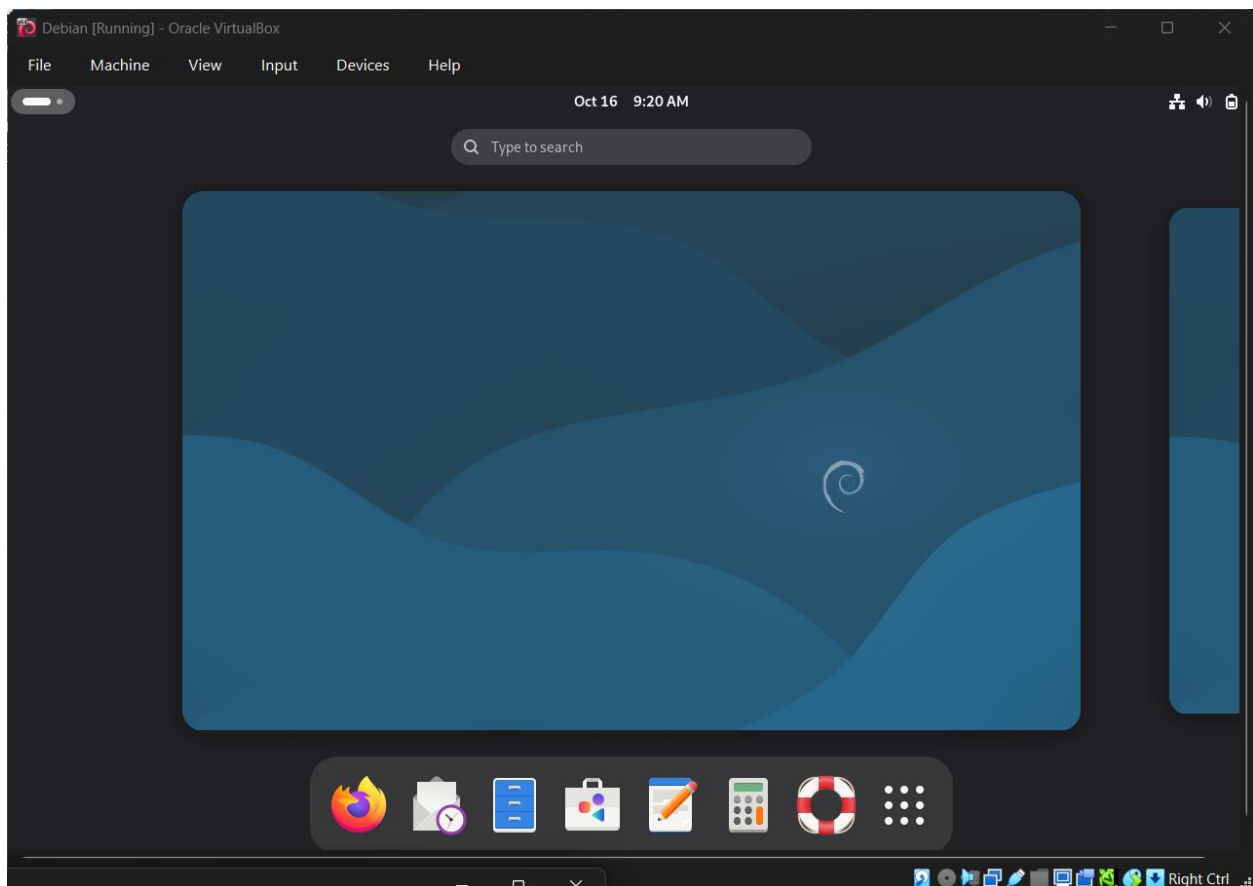
Líder del grupo: Benyahir Y. Martínez Hermina

Integrantes:

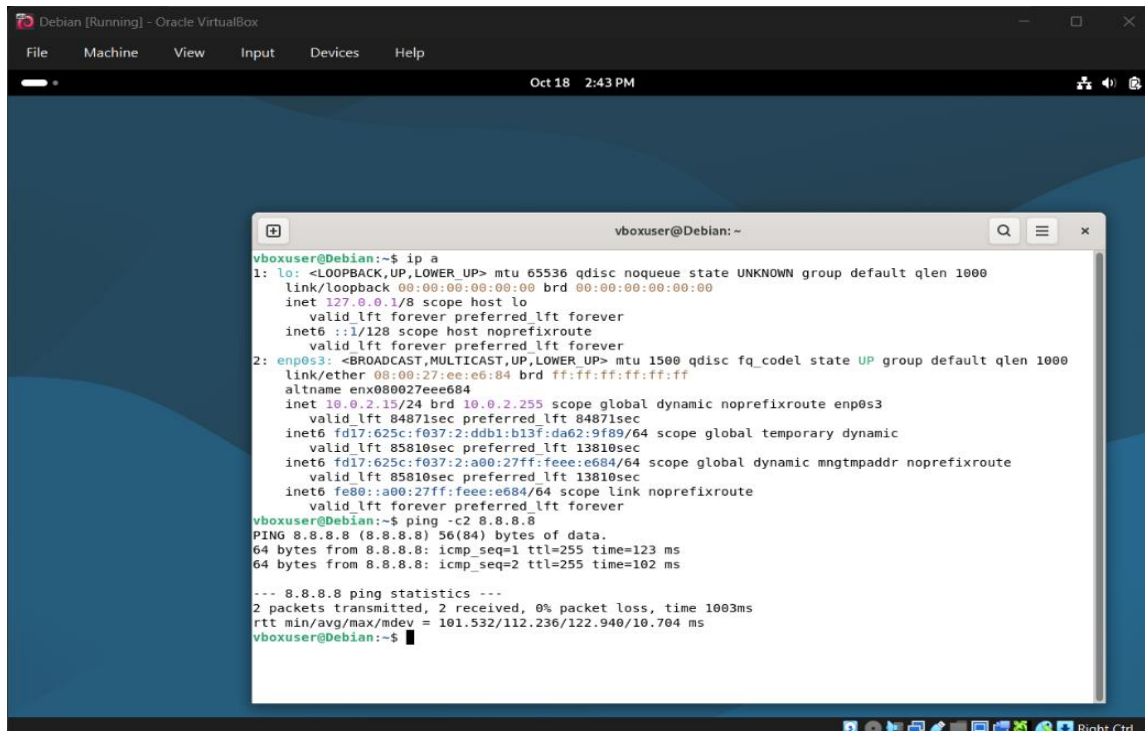
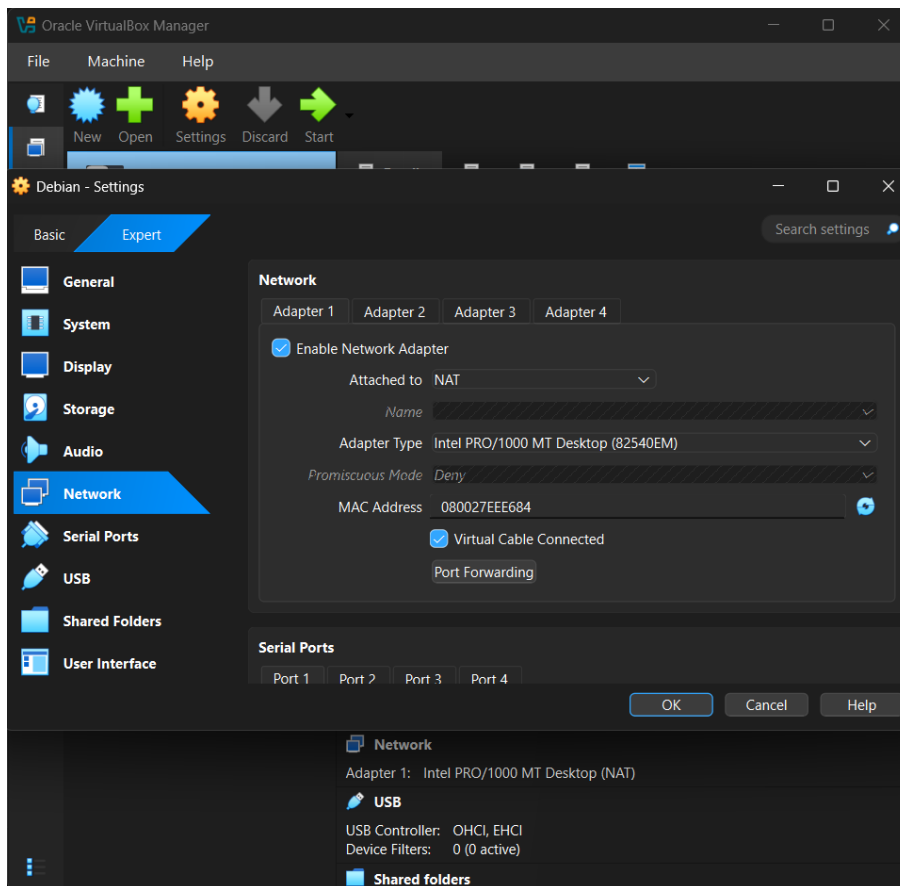
- Benyahir Y. Martínez
- Jacob J. Desuza
- Emanuel V. Rodríguez
- John A. Valentín

Fecha: 23 de oct. de 25

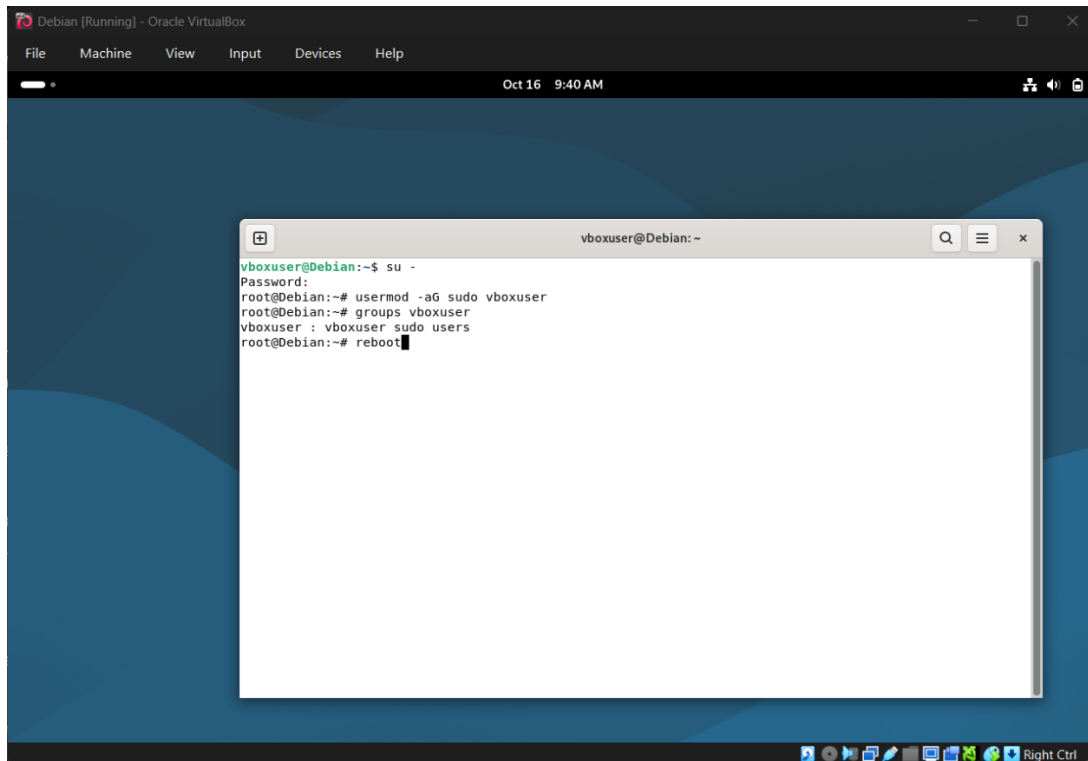
Parte 0 – Preparación del entorno Debian



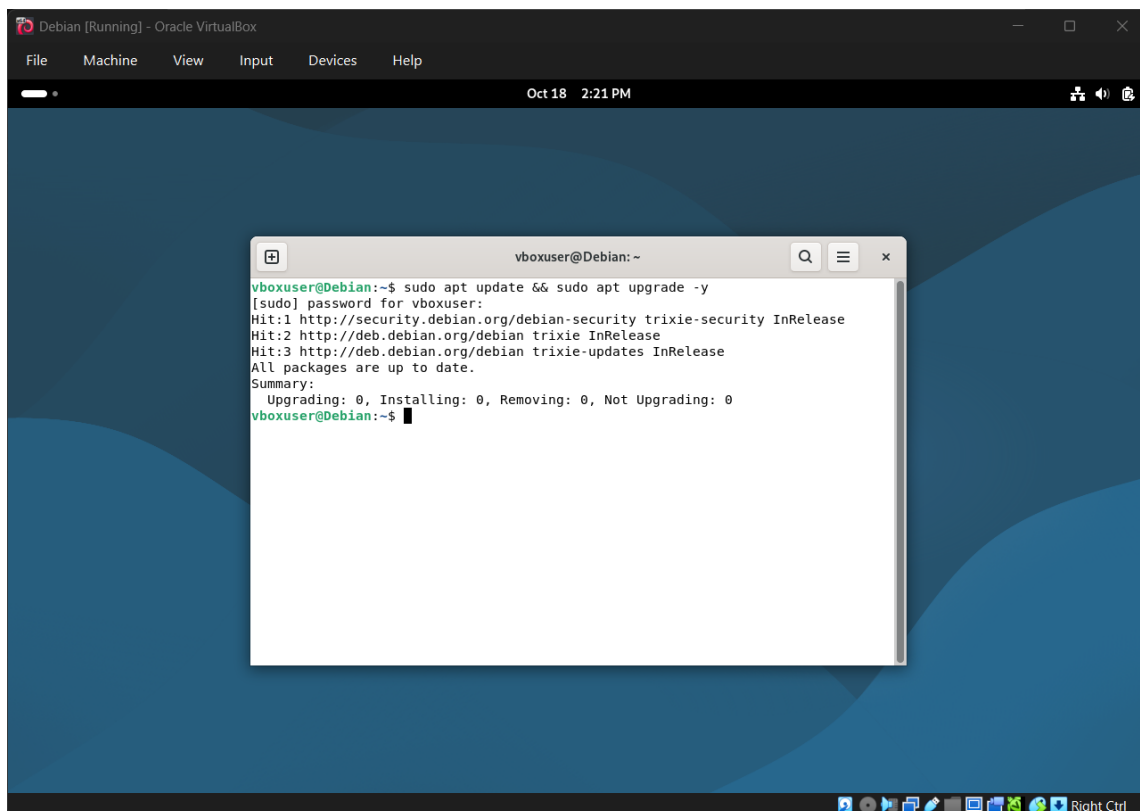
- Conexión de la computadora virtual a la red de la maquina anfitriona.



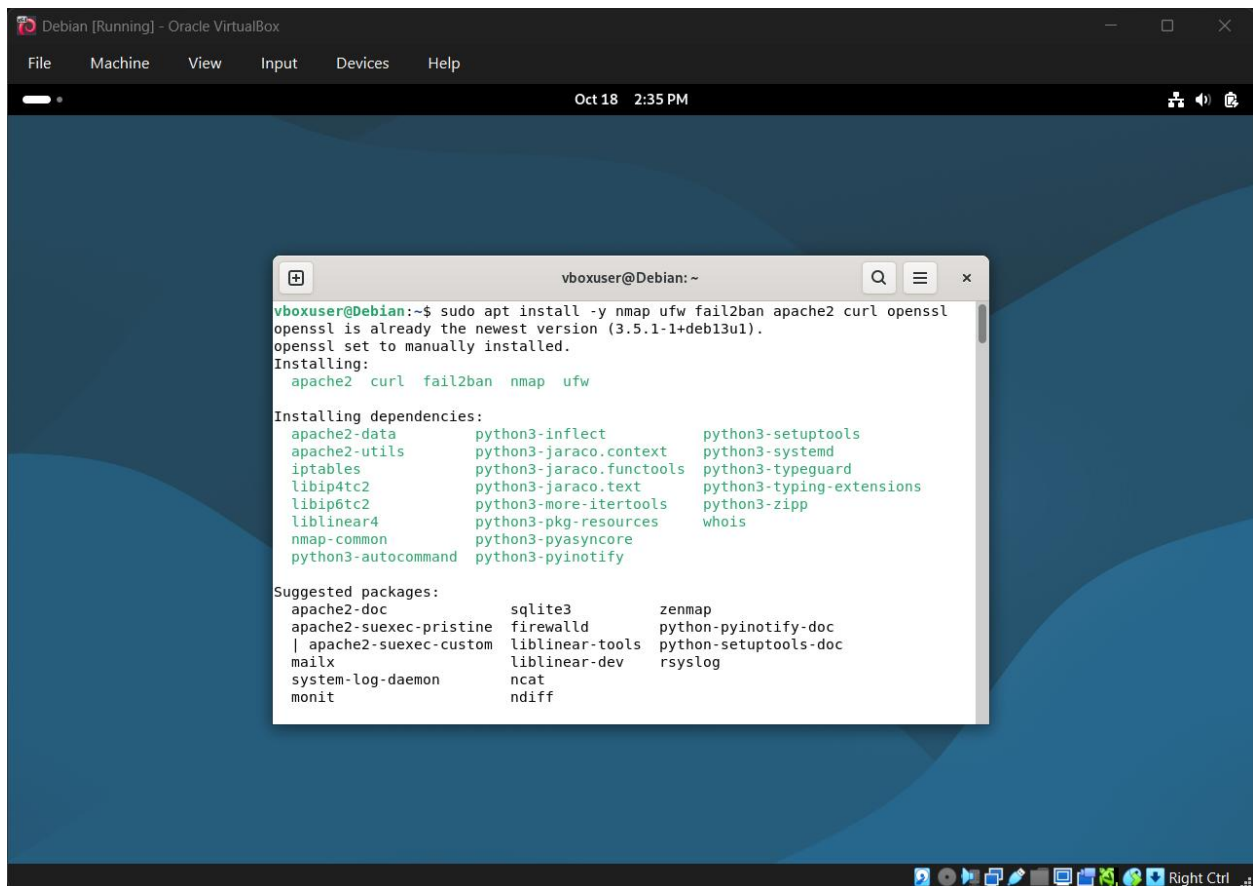
- Configurar el usuario para que sea root.



- Actualizar los paquetes.



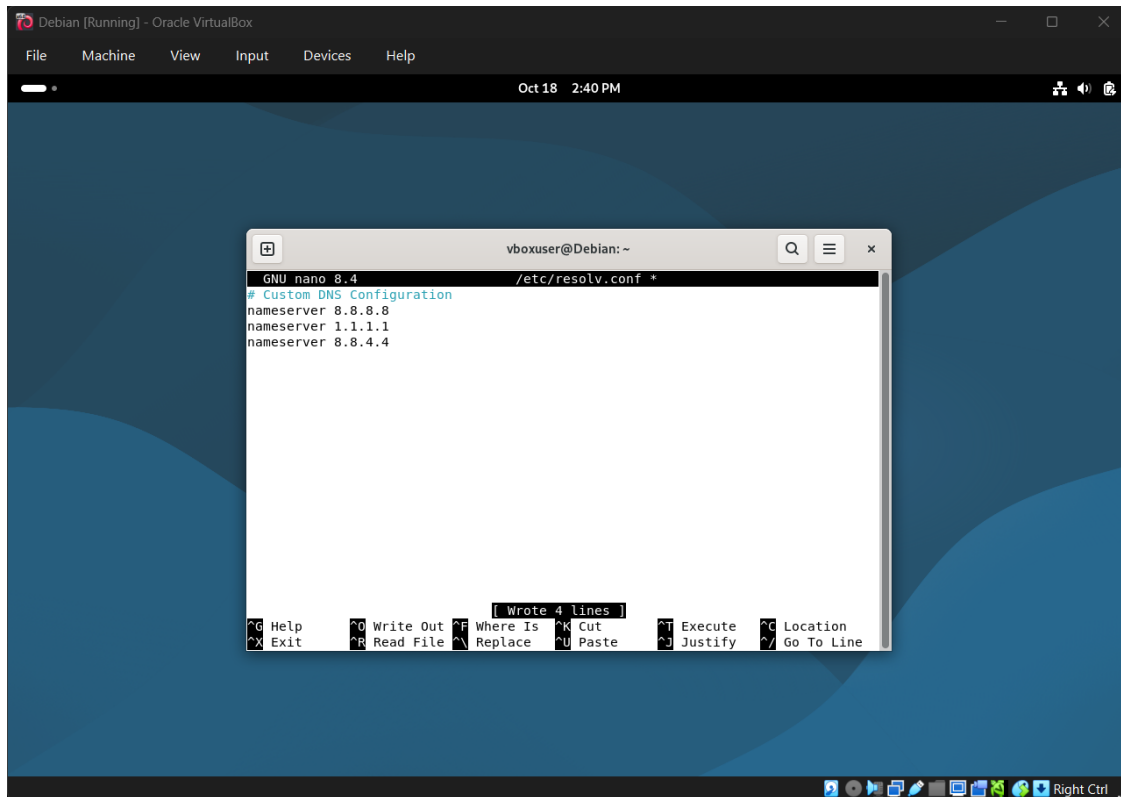
- Instalación de herramientas necesarias.



The screenshot shows a Debian virtual machine running in Oracle VM VirtualBox. The terminal window displays the command to install several packages and the resulting output, including a list of dependencies and suggested packages.

```
vboxuser@Debian: ~  
vboxuser@Debian:~$ sudo apt install -y nmap ufw fail2ban apache2 curl openssl  
openssl is already the newest version (3.5.1-1+deb13u1).  
openssl set to manually installed.  
Installing:  
  apache2  curl  fail2ban  nmap  ufw  
  
Installing dependencies:  
  apache2-data      python3-inflect      python3-setuptools  
  apache2-utils     python3-jaraco.context  python3-systemd  
  iptables          python3-jaraco.functools  python3-typeguard  
  libip4tc2         python3-jaraco.text      python3-typing-extensions  
  libip6tc2         python3-more-itertools   python3-zipp  
  liblinear4        python3-pkg-resources    whois  
  nmap-common       python3-pyasyncore  
  python3-autocommand  python3-pyinotify  
  
Suggested packages:  
  apache2-doc      sqlite3      zenmap  
  apache2-suexec-pristine  firewalld  python-pyinotify-doc  
  | apache2-suexec-custom  liblinear-tools  python-setuptools-doc  
  mailx            liblinear-dev  rsyslog  
  system-log-daemon  ncat  
  monit            ndiff
```

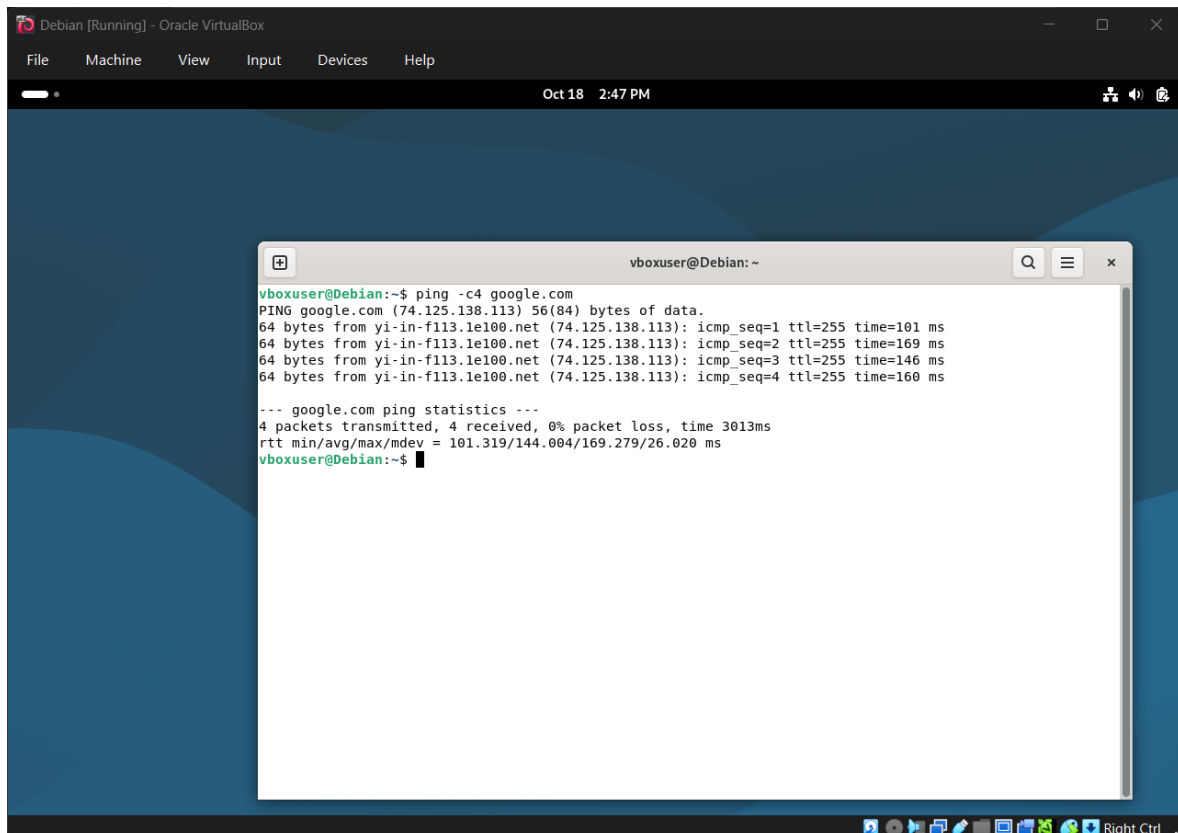
- Configuración de DNS



The screenshot shows a terminal window titled 'vboxuser@Debian: ~' with the nano 8.4 editor open to the file '/etc/resolv.conf'. The editor contains the following text:

```
# Custom DNS Configuration
nameserver 8.8.8.8
nameserver 1.1.1.1
nameserver 8.8.4.4
```

The bottom status bar of the nano editor shows: '[Wrote 4 lines]'. The terminal window is part of a Debian [Running] - Oracle VirtualBox environment, with a menu bar (File, Machine, View, Input, Devices, Help) and a system bar (Oct 18 2:40 PM).



The screenshot shows the same terminal window after running a ping command. The output is as follows:

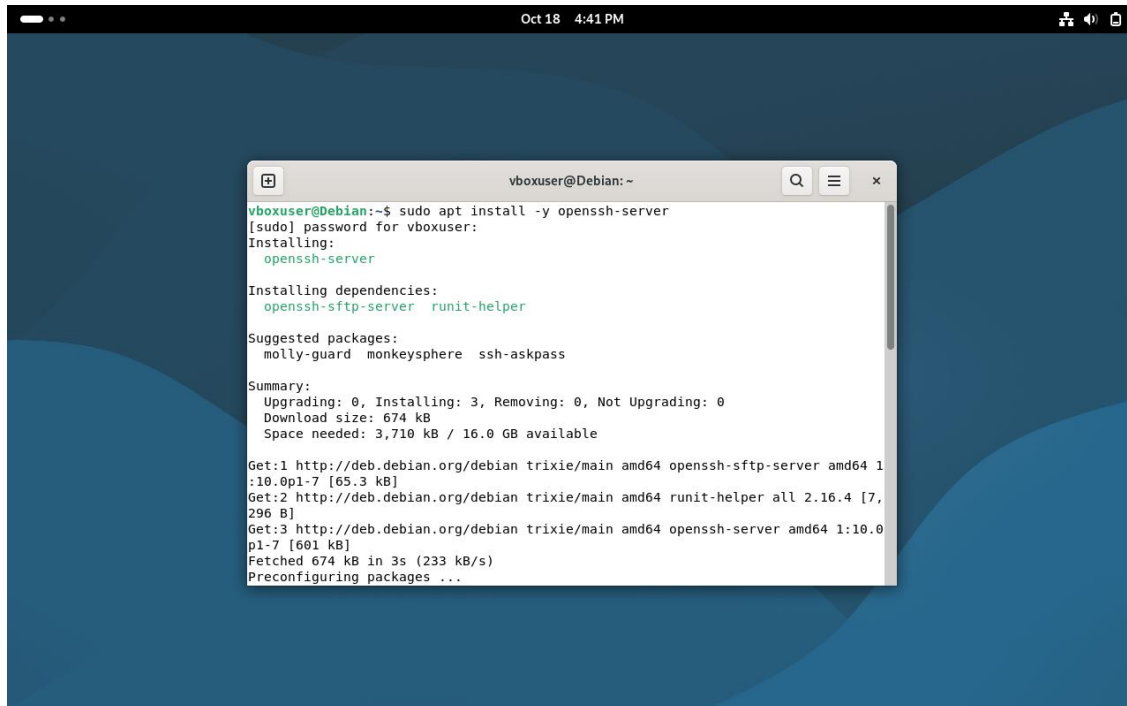
```
vboxuser@Debian:~$ ping -c4 google.com
PING google.com (74.125.138.113) 56(84) bytes of data:
64 bytes from yi-in-f113.1e100.net (74.125.138.113): icmp_seq=1 ttl=255 time=101 ms
64 bytes from yi-in-f113.1e100.net (74.125.138.113): icmp_seq=2 ttl=255 time=169 ms
64 bytes from yi-in-f113.1e100.net (74.125.138.113): icmp_seq=3 ttl=255 time=146 ms
64 bytes from yi-in-f113.1e100.net (74.125.138.113): icmp_seq=4 ttl=255 time=160 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 101.319/144.004/169.279/26.020 ms
vboxuser@Debian:~$
```

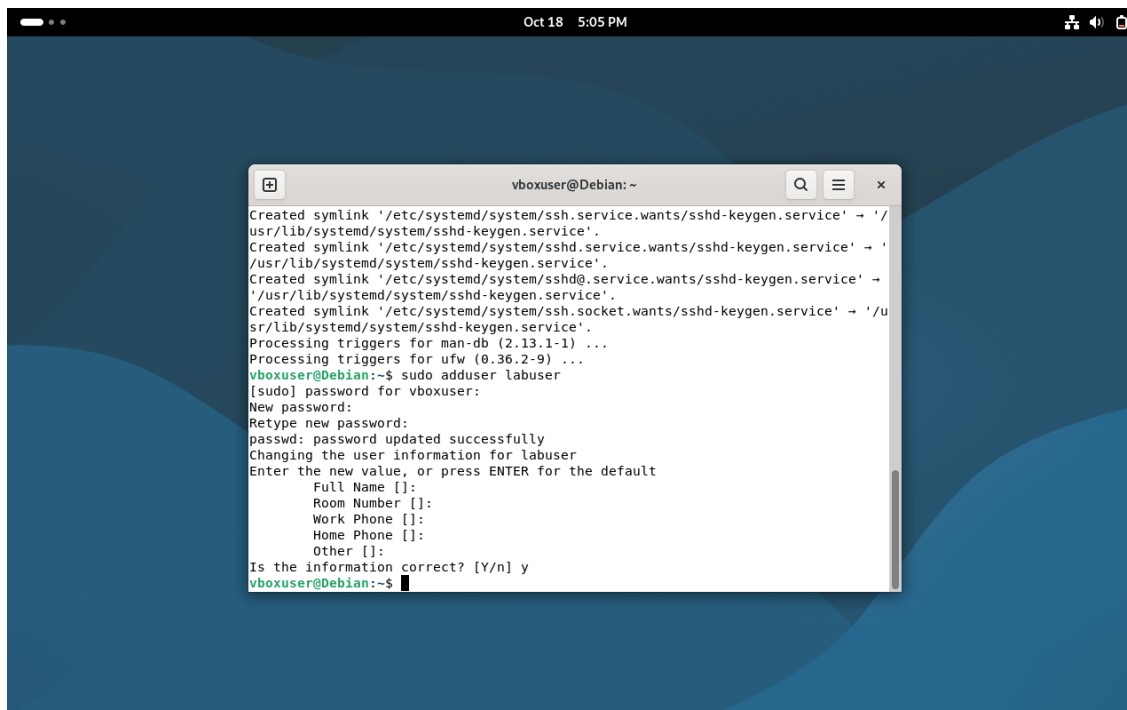
The terminal window remains in the Debian [Running] - Oracle VirtualBox environment, with the system bar now showing 'Oct 18 2:47 PM'.

Paso 1 – Creación de Vulnerabilidades

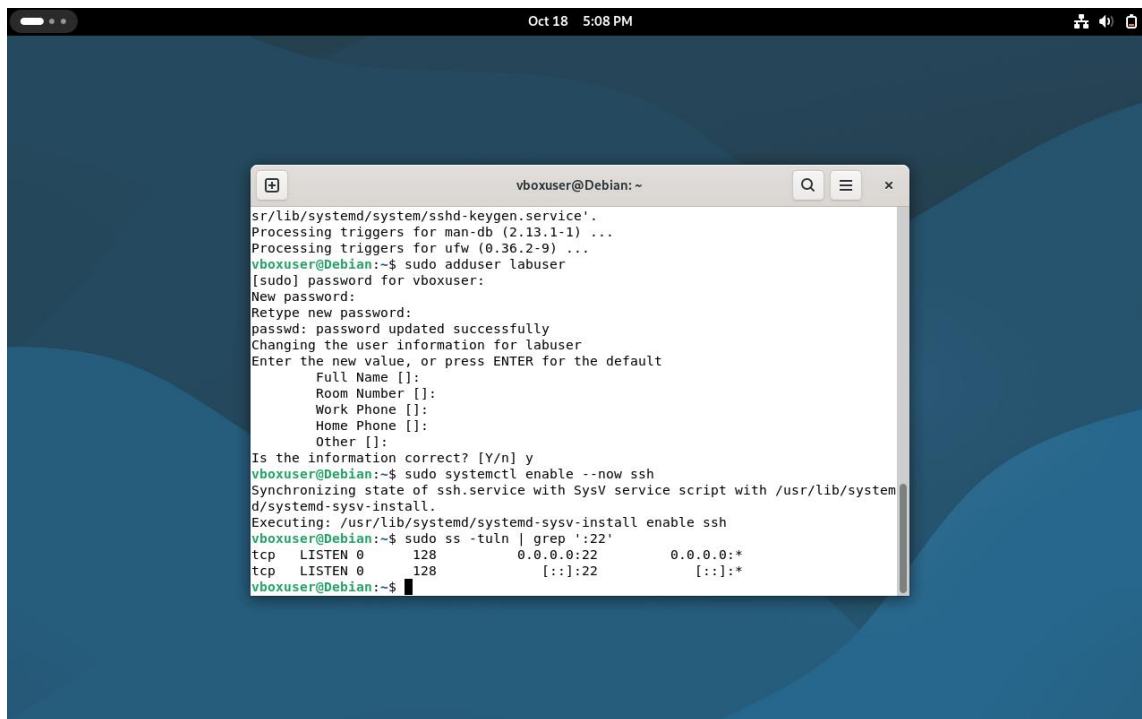
Vulnerabilidad 1: SSH expuesto con la contraseña del usuario débil.



```
vboxuser@Debian: ~  
vboxuser@Debian:~$ sudo apt install -y openssh-server  
[sudo] password for vboxuser:  
Installing:  
  openssh-server  
  
Installing dependencies:  
  openssh-sftp-server  runit-helper  
  
Suggested packages:  
  molly-guard  monkeysphere  ssh-askpass  
  
Summary:  
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 0  
  Download size: 674 kB  
  Space needed: 3,710 kB / 16.0 GB available  
  
Get:1 http://deb.debian.org/debian trixie/main amd64 openssh-sftp-server amd64 1:10.0p1-7 [65.3 kB]  
Get:2 http://deb.debian.org/debian trixie/main amd64 runit-helper all 2.16.4 [7,296 B]  
Get:3 http://deb.debian.org/debian trixie/main amd64 openssh-server amd64 1:10.0p1-7 [601 kB]  
Fetched 674 kB in 3s (233 kB/s)  
Preconfiguring packages ...
```



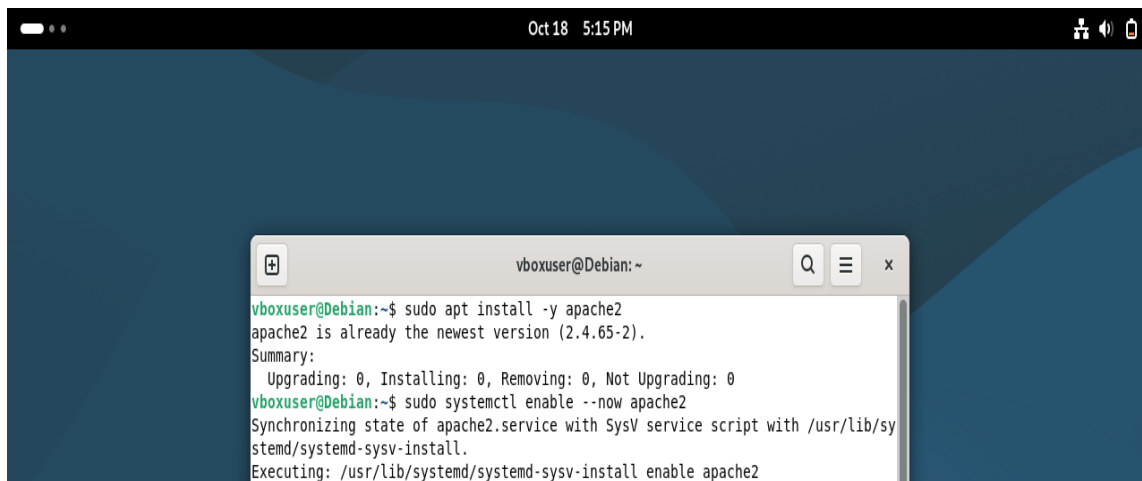
```
vboxuser@Debian: ~  
Created symlink '/etc/systemd/system/ssh.service.wants/ssh-keygen.service' -> '/usr/lib/systemd/system/ssh-keygen.service'.  
Created symlink '/etc/systemd/system/ssh.service.wants/ssh-keygen.service' -> '/usr/lib/systemd/system/ssh-keygen.service'.  
Created symlink '/etc/systemd/system/ssh.socket.wants/ssh-keygen.service' -> '/usr/lib/systemd/system/ssh-keygen.service'.  
Created symlink '/etc/systemd/system/ssh.socket.wants/ssh-keygen.service' -> '/usr/lib/systemd/system/ssh-keygen.service'.  
Processing triggers for man-db (2.13.1-1) ...  
Processing triggers for ufw (0.36.2-9) ...  
vboxuser@Debian:~$ sudo adduser labuser  
[sudo] password for vboxuser:  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for labuser  
Enter the new value, or press ENTER for the default  
  Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Is the information correct? [Y/n] y  
vboxuser@Debian:~$
```



A terminal window titled 'vboxuser@Debian: ~' showing the following commands and output:

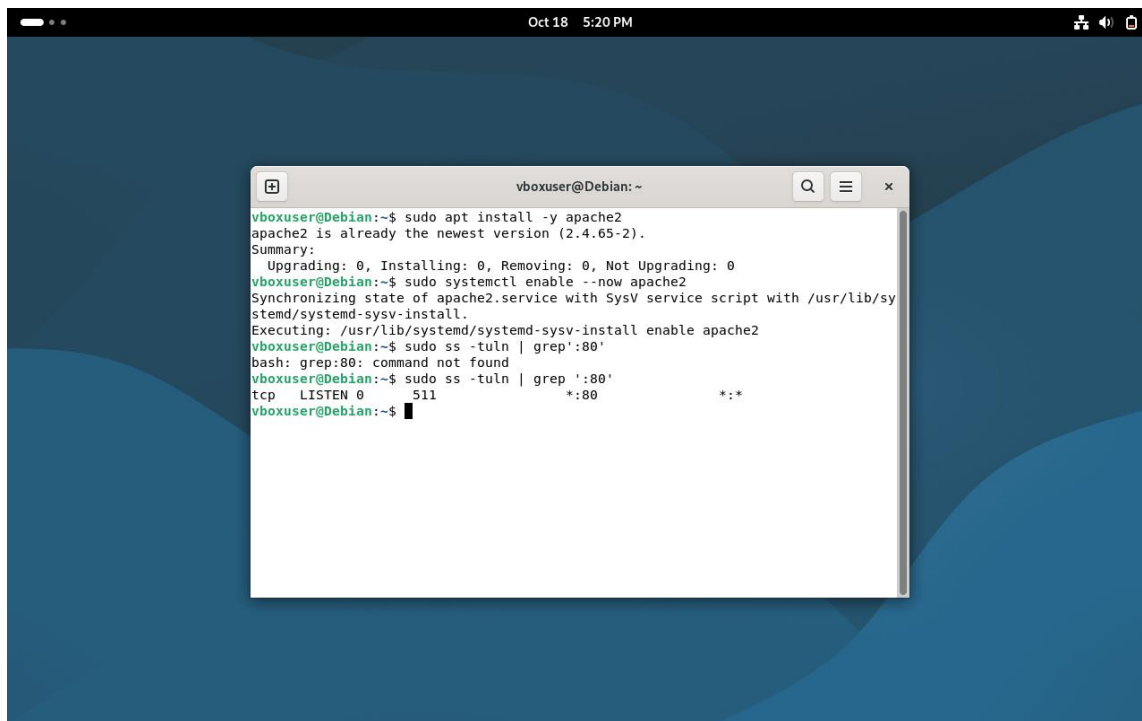
```
sr/lib/systemd/system/ssh-keygen.service'.
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for ufw (0.36.2-9) ...
vboxuser@Debian:~$ sudo adduser labuser
[sudo] password for vboxuser:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for labuser
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
vboxuser@Debian:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
vboxuser@Debian:~$ sudo ss -tln | grep ':22'
tcp    LISTEN 0      128          0.0.0.0:22      0.0.0.0:*
tcp    LISTEN 0      128          ::::22          ::::*
```

Vulnerabilidad 2: Apache solo en HTTP (sin HTTPS)



A terminal window titled 'vboxuser@Debian: ~' showing the following commands and output:

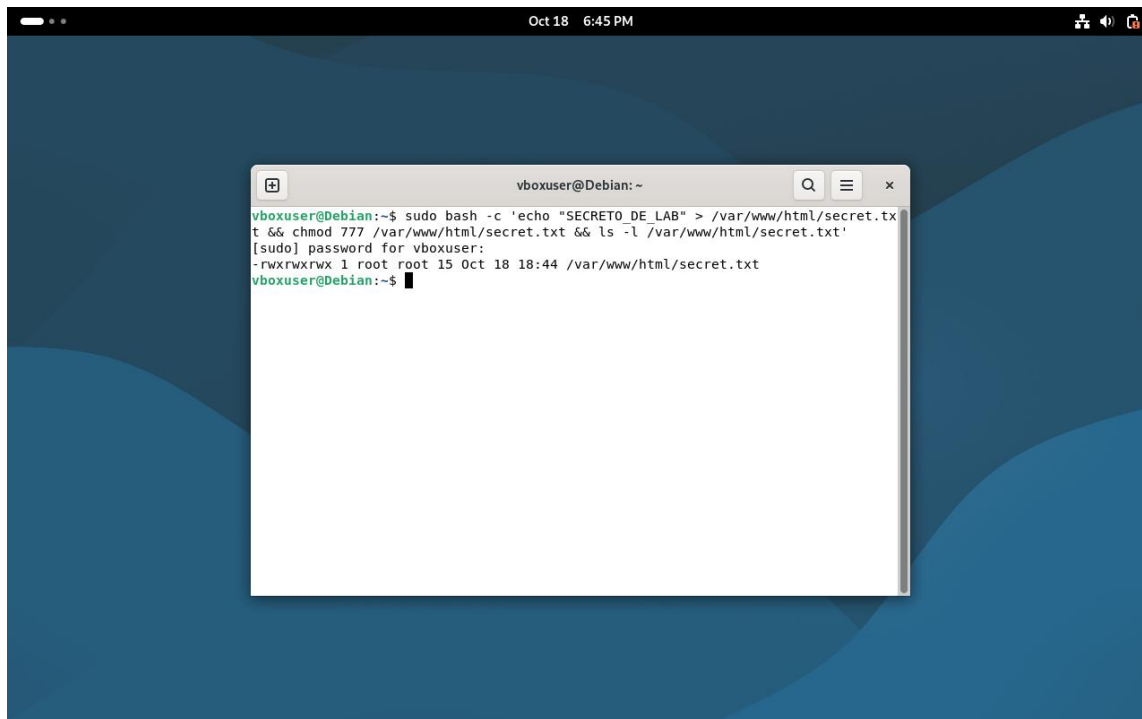
```
vboxuser@Debian:~$ sudo apt install -y apache2
apache2 is already the newest version (2.4.65-2).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
vboxuser@Debian:~$ sudo systemctl enable --now apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
```



A terminal window titled 'vboxuser@Debian: ~' showing the installation and configuration of Apache2. The user runs 'sudo apt install -y apache2', which shows that Apache2 is already the newest version (2.4.65-2). Then, the user runs 'sudo systemctl enable --now apache2', which synchronizes the state of the service with the SysV script and executes the installation. Finally, the user runs 'sudo ss -tln | grep ':80'', which shows that the Apache2 service is listening on port 80.

```
vboxuser@Debian:~$ sudo apt install -y apache2
apache2 is already the newest version (2.4.65-2).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
vboxuser@Debian:~$ sudo systemctl enable --now apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/sy
stemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
vboxuser@Debian:~$ sudo ss -tln | grep ':80'
bash: grep:80: command not found
vboxuser@Debian:~$ sudo ss -tln | grep ':80'
tcp        LISTEN    0          511             *:80             **
vboxuser@Debian:~$
```

Vulnerabilidad 3: Archivo con permisos inseguros (777)



A terminal window titled 'vboxuser@Debian: ~' showing the creation of a file with insecure permissions (777) in the /var/www/html directory. The user runs 'sudo bash -c 'echo "SECRETO_DE_LAB" > /var/www/html/secret.txt && chmod 777 /var/www/html/secret.txt && ls -l /var/www/html/secret.txt'', which creates the file and sets the permissions to 777. The output shows the file permissions as '-rwxrwxrwx 1 root root 15 Oct 18 18:44 /var/www/html/secret.txt'.

```
vboxuser@Debian:~$ sudo bash -c 'echo "SECRETO_DE_LAB" > /var/www/html/secret.tx
t && chmod 777 /var/www/html/secret.txt && ls -l /var/www/html/secret.txt'
[sudo] password for vboxuser:
-rwxrwxrwx 1 root root 15 Oct 18 18:44 /var/www/html/secret.txt
vboxuser@Debian:~$
```


Parte 2: Detección de las Vulnerabilidades

1) Detectar SSH expuesto

Ver servicio SSH escuchando

```
sudo ss -tln | grep ':22'
```

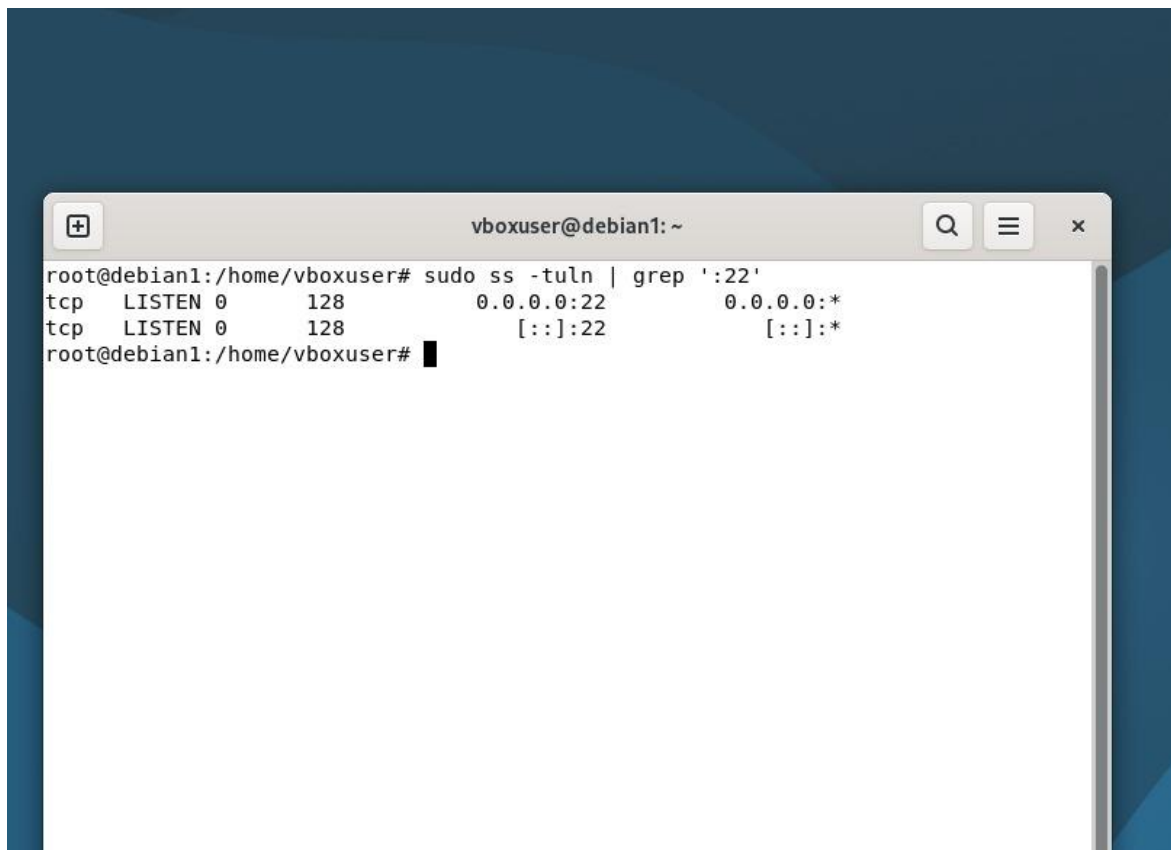
Obtener logs de sshd (usar journalctl)

```
sudo bash -c 'journalctl -u ssh -S "today" >
/home/labuser/evidencia_auth.log'
```

```
sudo chown labuser:labuser /home/labuser/evidencia_auth.log
```

```
sudo chmod 640 /home/labuser/evidencia_auth.log
```

- **Qué copiar al informe:** salidas de `ss -tln | grep ':22'` y el contenido de `~/evidencia_auth.log` (ej.: "Server listening on 0.0.0.0 port 22").



2) Detectar Apache sin HTTPS

Comandos:

Ver puerto 80

```
sudo ss -tuln | grep ':80'
```

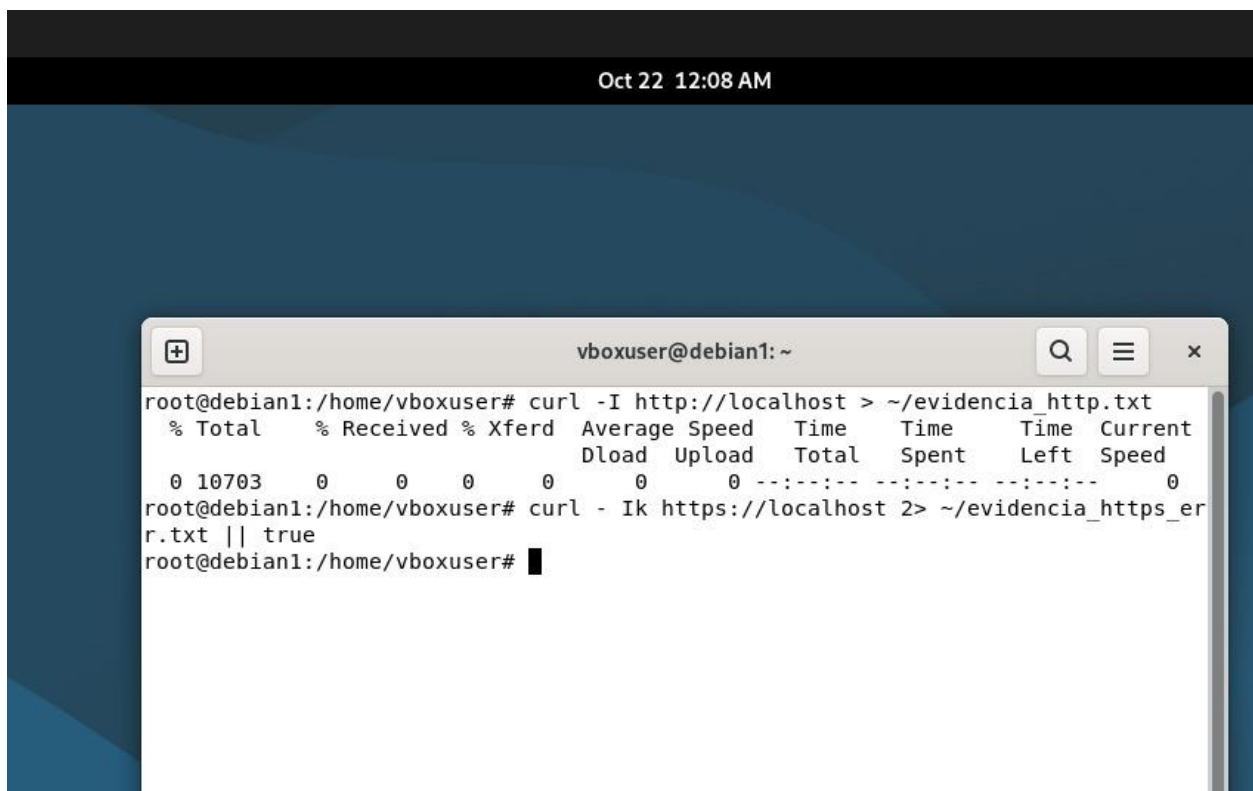
Probar HTTP (usa curl — instálalo si falta)

```
curl -I http://localhost > ~/evidencia_http.txt
```

Probar HTTPS (debe FALLAR si no hemos habilitado SSL)

```
curl -Ik https://localhost 2> ~/evidencia_https_err.txt || true
```

- **Qué copiar al informe:** Imagen Screenshot de salida de curl y error de conexión a puerto 443.



3) Detectar archivo con permisos inseguros

Comandos:

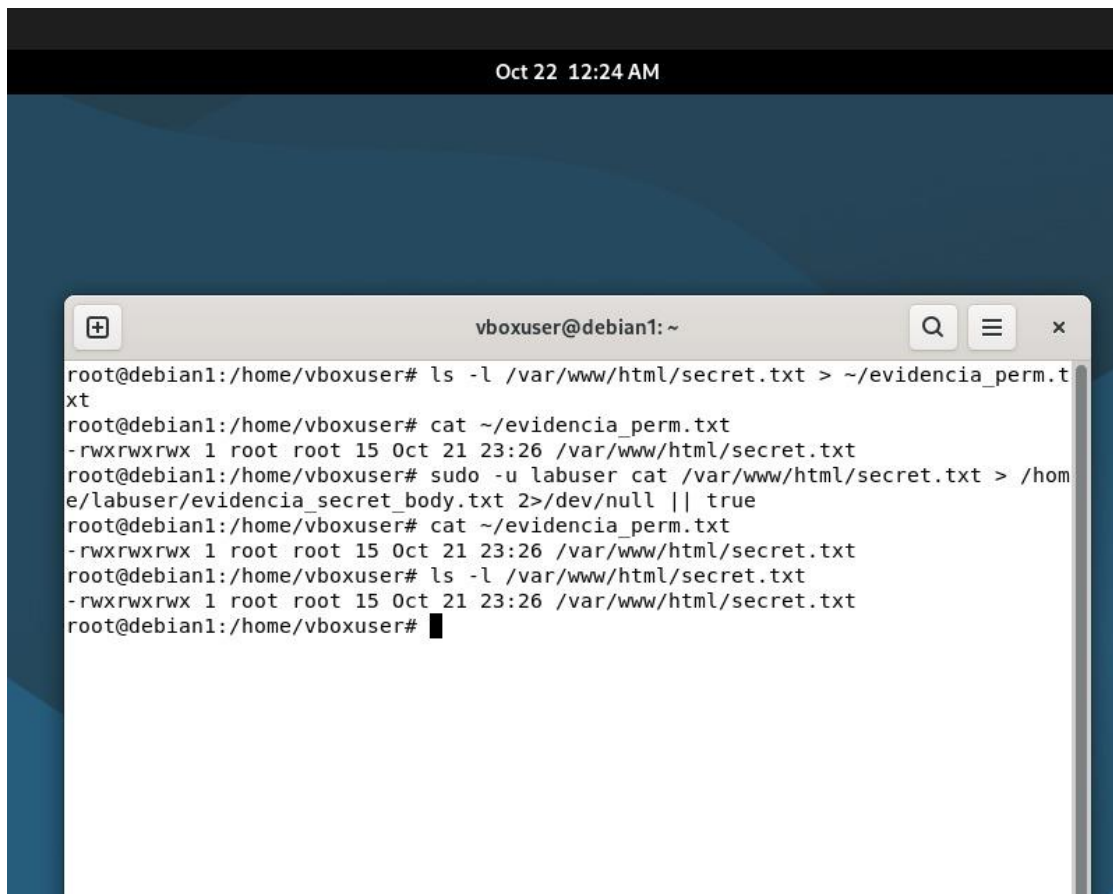
Mostrar permisos

```
ls -l /var/www/html/secret.txt > ~/evidencia_perm.txt
```

Intentar leer el archivo como usuario no privilegiado (ej. labuser)

```
sudo -u labuser cat /var/www/html/secret.txt > /home/labuser/evidencia_secret_body.txt 2>/dev/null || true
```

- **Qué copiar al informe:** salida de `ls -l` mostrando `-rwxrwxrwx` y contenido del archivo (SECRETO_DE_LAB).

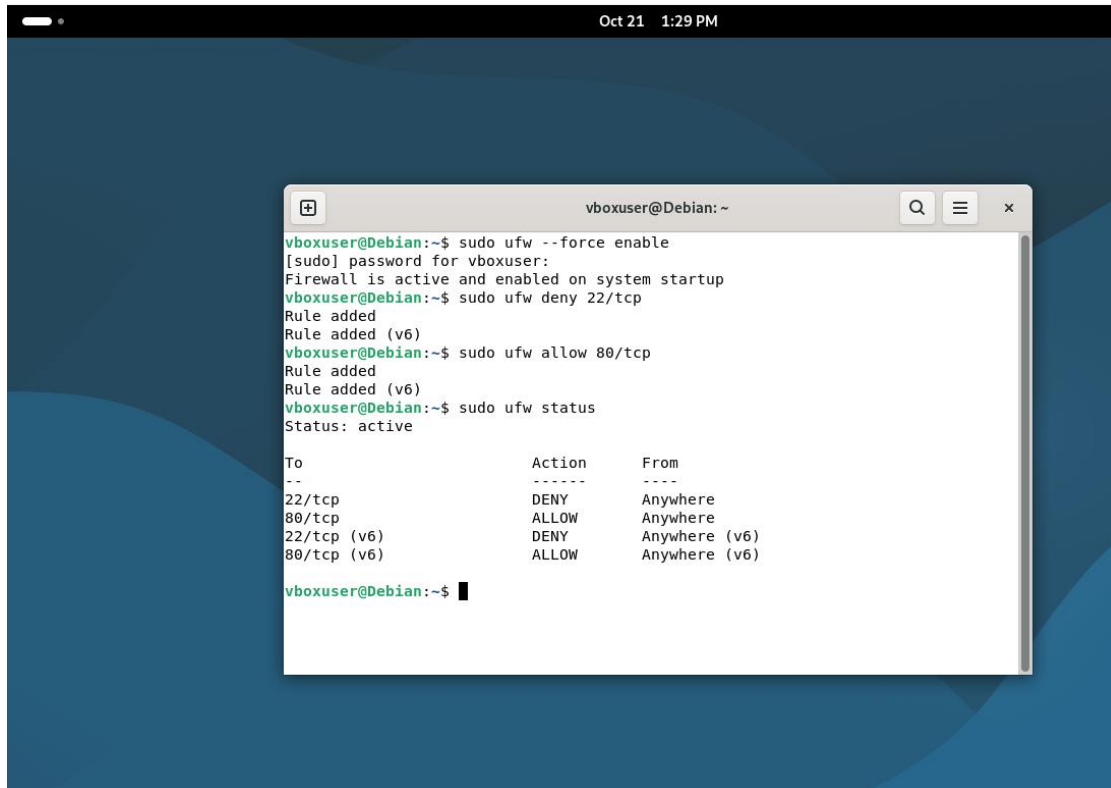


The screenshot shows a terminal window titled "vboxuser@debian1: ~" with a timestamp "Oct 22 12:24 AM". The terminal output is as follows:

```
root@debian1:/home/vboxuser# ls -l /var/www/html/secret.txt > ~/evidencia_perm.txt
root@debian1:/home/vboxuser# cat ~/evidencia_perm.txt
-rwxrwxrwx 1 root root 15 Oct 21 23:26 /var/www/html/secret.txt
root@debian1:/home/vboxuser# sudo -u labuser cat /var/www/html/secret.txt > /home/labuser/evidencia_secret_body.txt 2>/dev/null || true
root@debian1:/home/vboxuser# cat ~/evidencia_perm.txt
-rwxrwxrwx 1 root root 15 Oct 21 23:26 /var/www/html/secret.txt
root@debian1:/home/vboxuser# ls -l /var/www/html/secret.txt
-rwxrwxrwx 1 root root 15 Oct 21 23:26 /var/www/html/secret.txt
root@debian1:/home/vboxuser#
```

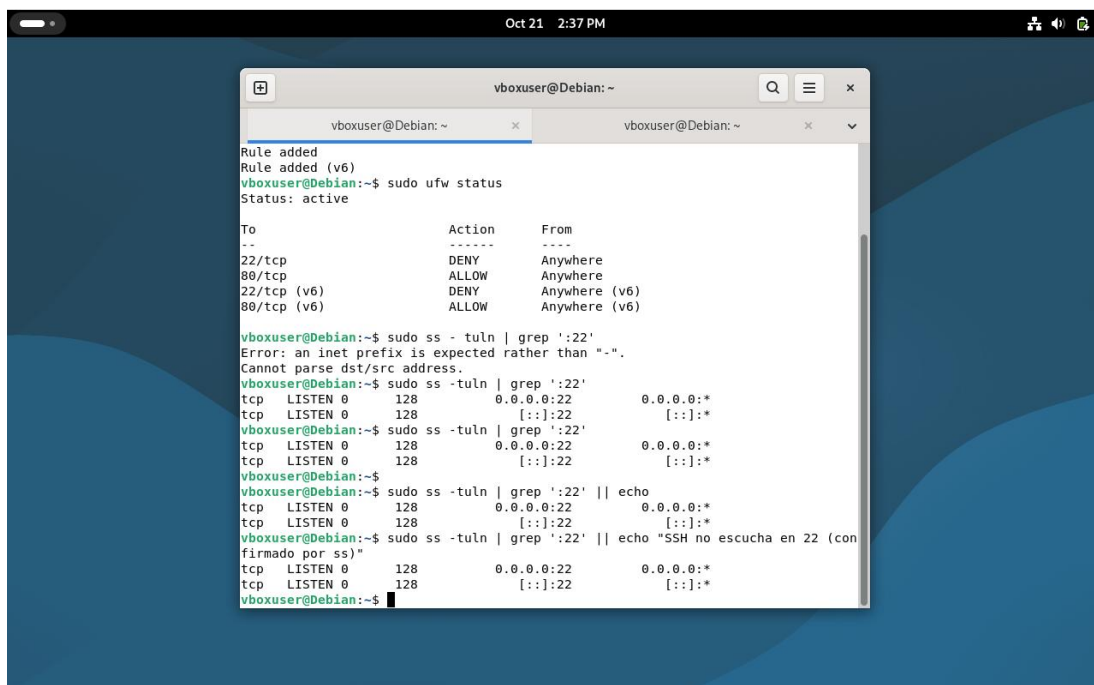
Parte 3: Mitigación de las Vulnerabilidades

Mitigación 1: SSH: denegar puerto 22 y pedir contraseña fuerte o usar claves



Oct 21 1:29 PM

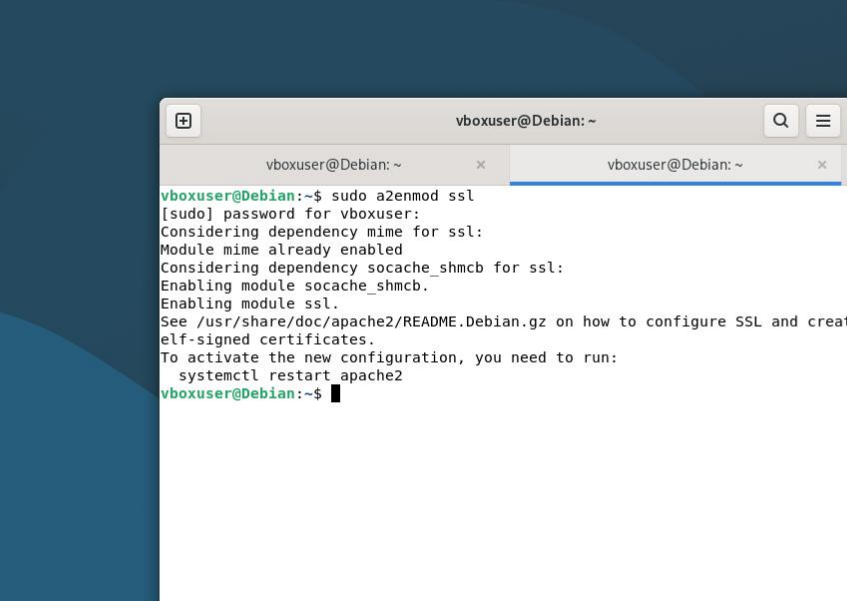
```
vboxuser@Debian: ~  
vboxuser@Debian:~$ sudo ufw --force enable  
[sudo] password for vboxuser:  
Firewall is active and enabled on system startup  
vboxuser@Debian:~$ sudo ufw deny 22/tcp  
Rule added  
Rule added (v6)  
vboxuser@Debian:~$ sudo ufw allow 80/tcp  
Rule added  
Rule added (v6)  
vboxuser@Debian:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp DENY Anywhere  
80/tcp ALLOW Anywhere  
22/tcp (v6) DENY Anywhere (v6)  
80/tcp (v6) ALLOW Anywhere (v6)  
  
vboxuser@Debian:~$
```



Oct 21 2:37 PM

```
vboxuser@Debian: ~  
vboxuser@Debian:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp DENY Anywhere  
80/tcp ALLOW Anywhere  
22/tcp (v6) DENY Anywhere (v6)  
80/tcp (v6) ALLOW Anywhere (v6)  
  
vboxuser@Debian:~$ sudo ss -tln | grep ':22'  
Error: an inet prefix is expected rather than ":".  
Cannot parse dst/src address.  
vboxuser@Debian:~$ sudo ss -tln | grep ':22'  
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*  
tcp LISTEN 0 128 [::]:22 [::]:*  
vboxuser@Debian:~$ sudo ss -tln | grep ':22'  
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*  
tcp LISTEN 0 128 [::]:22 [::]:*  
vboxuser@Debian:~$ sudo ss -tln | grep ':22' || echo  
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*  
tcp LISTEN 0 128 [::]:22 [::]:*  
vboxuser@Debian:~$ sudo ss -tln | grep ':22' || echo "SSH no escucha en 22 (con  
firmado por ss)"  
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*  
tcp LISTEN 0 128 [::]:22 [::]:*  
vboxuser@Debian:~$
```

Mitigación 2 — Apache: habilitar HTTPS (certificado autofirmado)



The screenshot shows a terminal window titled "vboxuser@Debian: ~". The user has executed the command `sudo a2enmod ssl`. The terminal output shows the following steps:

- `[sudo] password for vboxuser:` (password prompt)
- `Considering dependency mime for ssl:`
- `Module mime already enabled`
- `Considering dependency socache_shmcb for ssl:`
- `Enabling module socache_shmcb.`
- `Enabling module ssl.`

Below the output, there is a reference to the documentation: `See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.`

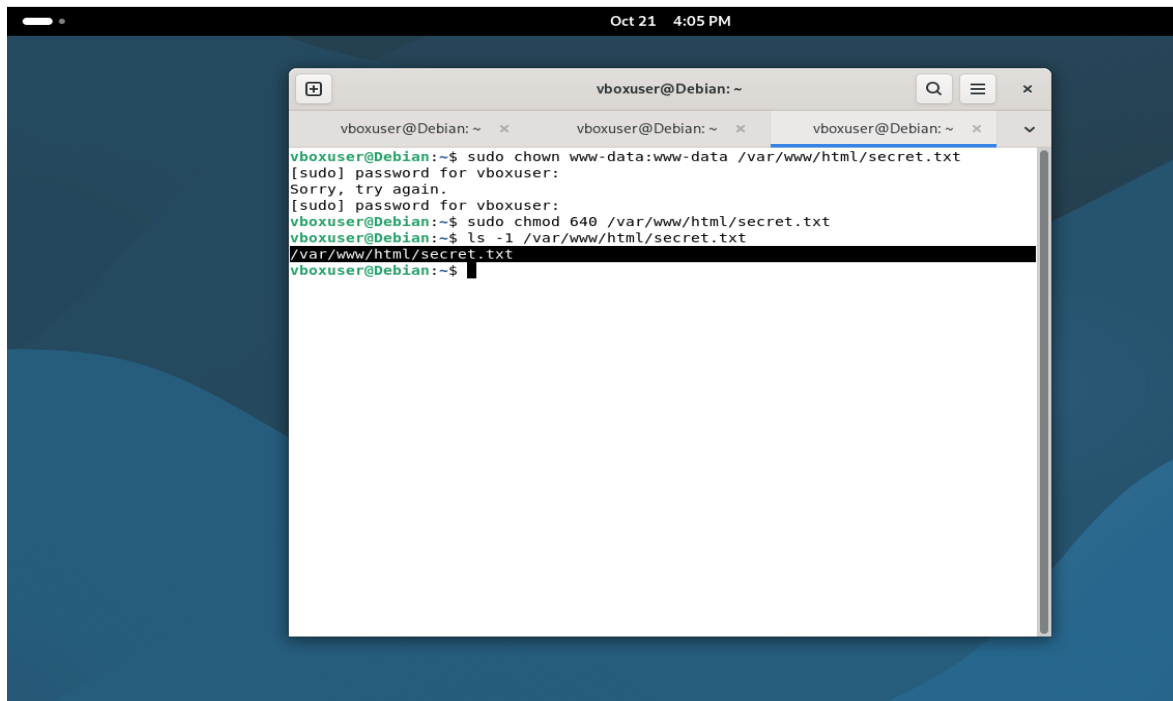
Then, the user is instructed to activate the configuration by running:

```
systemctl restart apache2
```

The terminal shows the prompt `vboxuser@Debian:~$` with a cursor, indicating the command has been executed.

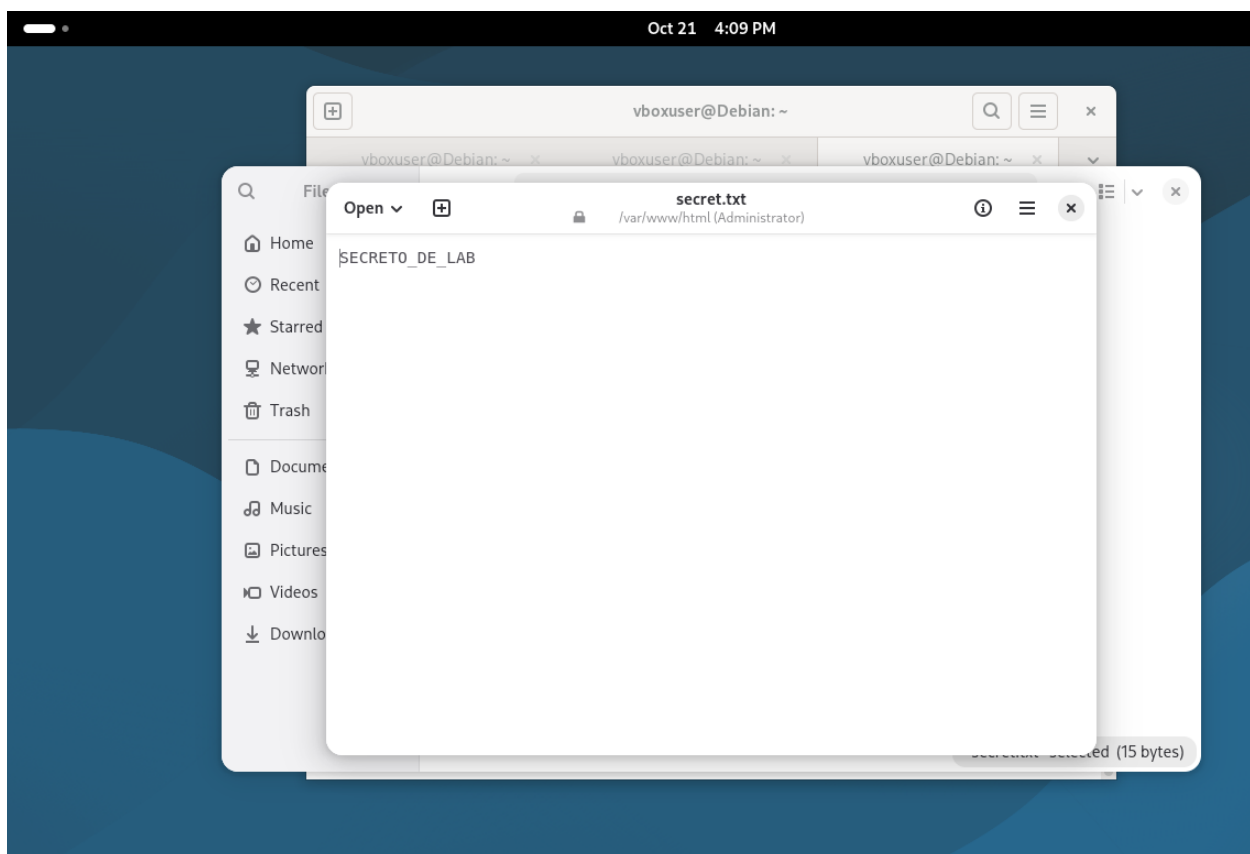
[illegible]

Mitigacion 3: Archivo con permisos inseguros: aplicar mínimo privilegio (640) y propiedad www-data



A terminal window titled 'vboxuser@Debian: ~' showing the following commands and output:

```
vboxuser@Debian:~$ sudo chown www-data:www-data /var/www/html/secret.txt
[sudo] password for vboxuser:
Sorry, try again.
[sudo] password for vboxuser:
vboxuser@Debian:~$ sudo chmod 640 /var/www/html/secret.txt
vboxuser@Debian:~$ ls -l /var/www/html/secret.txt
-rw-r--r-- 1 www-data www-data 15 Oct 21 16:05 /var/www/html/secret.txt
vboxuser@Debian:~$
```



Parte 4: Tabla de datos encontrados y análisis

Vulnerabilidad	Amenaza	Riesgo (Alto/Medio/Bajo)	Control Usado	Modelo de Ataque (los 4 modelos)
SSH expuesto con la contraseña débil del usuario.	Acceso inicial con usuario débil puede permitir al atacante obtener permisos de root o administrador.	Alto	denegar puerto 22 y pedir contraseña fuerte o usar claves	Intercepción
Apache solo en HTTP (sin HTTPS)	Sin HTTPS, es más fácil que un atacante suplante el sitio web legítimo para engañar a los usuarios.	Alto	Apache: habilitar HTTPS (certificado auto firmado)	Fabricación
Archivo con permisos inseguros (777)	Los usuarios pueden acceder a todos los archivos ya que los permisos no los están restringiendo correctamente.	Medio	Aplicar mínimo privilegio (640) y propiedad www-data	Modificación

Conclusión: ¿Cuál fue la amenaza más probable en su sistema y qué control resultó más efectivo?

- Consideramos que la amenaza más probable que ocurra en nuestro sistema es la exposición de SSH con el uso de contraseña débil. Debido a que la contraseña es fácil, un atacante pueda adivinar la contraseña y entrar al sistema del usuario. Esta situación abre la puerta a una gran cantidad de diferentes explotaciones de vulnerabilidades una vez que el atacante ingrese al sistema provocando ataques de interceptación, modificación, ect. Consideramos que, dentro de estos controles, los más efectivos fueron interceptación y modificación debido a que vimos la importancia de buscar e identificar problemas dentro del sistema y hacer los arreglos correspondientes para resolver el problema. Sin duda esta ha sido una actividad en la que hemos logrado comprender de manera extensiva el uso tanto de herramientas para maquinas virtuales, como la identificación de amenazas y vulnerabilidades y como mitigarlas.

Nombre del integrante	¿Qué realizo?	% del trabajo
Benyahir Y. Martínez	Parte 0, Parte 4, Conclusión	25%
Emanuel V. Rodríguez	Parte 1, Parte 4, Conclusión	25%
John A. Valentín	Parte 2, Parte 4, Conclusión	25%
Jacob J. Desuza	Parte 3, Parte 4, Conclusión	25%