

PROYECTO

Monitorización y respuesta inteligente ante ataques de fuerza bruta en servidores Linux

IES VENANCIO BLANCO



CICLO FORMATIVO DE GRADO SUPERIOR

Administración de Sistemas Informáticos en Red

I.E.S. «Venancio Blanco» SALAMANCA

AUTOR

Jorge Muñoz Ibáñez

TUTOR

Javier Ávila Miguel

1. Licencia

Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

2. Resumen

- **Empresa/organización que lo realiza** - I.E.S. Venancio Blanco
- **Necesidades que cubre** - La necesidad de mejorar la seguridad de los servidores Linux frente a ataques de fuerza bruta, protegiéndolos así de intrusos en la red.
- **Possible demanda/clientes** - Cualquier tipo de empresa (centro educativo en este caso), o administradores de sistemas.
- **Breve descripción de la solución que propone este proyecto** - El proyecto consiste en el desarrollo de un script (Bash) que analice en tiempo real los registros/logs del sistema para detectar intentos de acceso fallidos, emitir una alerta y bloquear la IP que esté tratando de acceder.

El proyecto, tiene como objetivo demostrar cómo se pueden aplicar los conocimientos adquiridos en el CFGS. De Administración de Sistemas Informáticos en Red, en un entorno de trabajo práctico y real, utilizando la ciberseguridad y la administración de sistemas para solucionar un problema muy común dentro de las empresas en general, en este caso un centro educativo.

Este proyecto, se centra en (como mencionado anteriormente) el desarrollo de un script en bash, que analice en tiempo real los registros/logs del sistema para detectar intentos de acceso fallidos, emitir una alerta y bloquear la IP que esté tratando de acceder. Estos intentos de intrusión, es uno de los problemas más comunes en la actualidad, y dentro de un sistema educativo, estos atacantes podrían conseguir información muy confidencial de cada uno de los estudiantes y/o profesorado, si no se toman la precauciones necesarias.

En este caso, los centros educativos tienen especial vulnerabilidad, por ejemplo, con los servidores de prácticas dentro de este mismo CFGS. Ya que todos los alumnos tienen servidores conectados a la red que podrían ser atacados en cualquier momento.



Por ello, se tiene la necesidad de implementar este tipo de soluciones para que ningún alumno o profesor se exponga a intentos de intrusión.

Esta solución, sería el proyecto que a continuación se explicará en qué consiste detalladamente:

- Consiste en el desarrollo de un script en bash que analice en tiempo real los logs del sistema dentro del archivo “/var/log/auth.log”, con el objetivo de, aplicando un filtro específico, “deje pasar” o bloquee la IP detectada en el servidor. Estos bloqueos de IP se realizarán utilizando reglas de “iptables”.

Además, estos bloqueos serán anunciados mostrando un mensaje en pantalla, para que el usuario pueda tener la información de lo que está ocurriendo en su sistema, y pueda prevenirse, aplicando las medidas de seguridad necesarias para que evite que vuelva a ocurrir.

El proyecto será dividido en varias fases:

Análisis: Análisis de los ataques y de los registros del sistema.

Diseño: Diseño del script del sistema de detección, especificando qué hay que identificar en los logs, y cuándo actuar.

Implementación: Desarrollo del propio script en Bash que monitorice los intentos de acceso y ejecute acciones de defensa (en caso de ser necesarias).

Pruebas: Ejecución de pruebas mediante la simulación de ataques desde una VM, hacia otra VM que actuará como servidor, verificando que el script desarrollado es capaz de analizar correctamente los intentos de acceso.

Este proyecto, además de poder ser aplicado en entornos corporativos, puede servir en el ámbito educativo, para formar a los alumnos de tal manera que puedan aplicar estos conocimientos en situaciones reales o añadirlo a su portfolio académico, a modo de experiencia.



3. Índice de contenido

1. Licencia.....	2
2. Resumen.....	2
3. Índice de contenido.....	4
4. Índice de figuras.....	5
5. Introducción.....	6
6. Necesidades Del Sector Productivo.....	8
6.1 Análisis de la situación actual.....	8
6.2 Necesidades del cliente y oportunidad de negocio.....	9
6.3 El nuevo proyecto: Monitorización y respuesta inteligente ante ataques de fuerza bruta en servidores Linux.....	10
6.3.1 Tipo de proyecto.....	10
6.3.2 Características requeridas al proyecto.....	11
6.3.3 Obligaciones fiscales, laborales y de prevención de riesgo.....	12
6.3.4 Ayudas/subvenciones.....	14
7. Diseño Del Proyecto.....	15
7.1 Análisis:.....	15
7.2 Diseño:.....	16
7.3 Implementación.....	18
8. Pruebas.....	44
9. Objetivos a conseguir.....	55
10. Previsión de los recursos materiales y humanos necesarios.....	56
11. Presupuesto económico.....	57
12. Fuentes.....	58
13. Anexos.....	59

4. Índice de figuras

1. Entorno de red pág. X

2 VM, una con el servidor y otra que simule los ataques.

2. Flujo de funcionamiento del script de detección pág. X

lectura de logs → detección de intentos fallidos → alerta → bloqueo de la IP.

3. Ejemplo de registros de intentos de acceso en auth.log pág. X

Captura de un log a tiempo real en el que se observe como hay un IP intrusa.

4. Captura de la ejecución del script pág. X

Ejemplo del script funcionando y mostrando como detecta el ataque y actúa.



5. Introducción

En los últimos años, al haber habido un incremento muy importante de equipos conectados a internet, conlleva también un aumento importante de ataques a los servidores que gestionan todos aquellos equipos. Este problema tan común actualmente, viene dado por atacantes que implementan sistemas automatizados de inicio de sesión (por ejemplo empleando millones de combinaciones de usuarios y contraseñas), para poder acceder a los sistemas sin que quede rastro de que realmente estas personas no tienen acceso al mismo servidor.

Además, gracias a los avances en este campo, se demuestra cada día, como no es necesaria acceder a ningún software de pago, o externo, sino que directamente utilizando las herramientas que tenemos y los conocimientos adquiridos en el CFGS, se puede hacer frente a un problema que puede afectar muy significativamente en distintos ámbitos del mundo empresarial.

En este entorno específicamente, siendo un centro educativo, la seguridad de los servidores cobra vital importancia al ser usados tanto para la gestión interna del centro, como para las prácticas de los alumnos en ciertos CF. Esto genera que haya decenas de equipos gestionados simultáneamente por un servidor, lo cuál puede suponer daños fatales si un intruso accede al servidor sin ser detectado.

Por ello, el proyecto busca (como comentado anteriormente), por un lado, reforzar los conocimientos aprendidos en el **CFGPS de Administración de Sistemas Informáticos en Red** y por otro, dar solución a un problema muy habitual en el entorno de la ciberseguridad en los servidores.

Esta solución vendrá dada por un sistema que analice automáticamente y a tiempo real, cada log del sistema de manera que cuando se apliquen los filtros que estarán automatizados en el desarrollo del script, pueda detectar inicios de sesión erróneos y aplicar la medidas de seguridad necesarias.

El proyecto está dividido en varias fases como comentamos en el resumen:

- **Análisis:** Análisis de los ataques y de los registros del sistema.
- **Diseño:** Definición del funcionamiento del sistema de detección, especificando qué hay que identificar en los logs, y cuándo actuar.
- **Implementación:** Desarrollo de un script en Bash que monitorice los intentos de acceso y ejecute acciones de defensa (en caso de ser necesarias).
- **Pruebas:** Ejecución de pruebas mediante la simulación de ataques desde una VM, hacia otra VM que actuará como servidor, verificando que el script desarrollado es capaz de analizar correctamente los intentos de acceso.

En definitiva, este proyecto no responde únicamente a una solución de la actualidad, sino que, desde el punto formativo, ayuda a los alumnos a comprender como funciona la ciberseguridad en el mundo real, y a actuar ante situaciones de riesgo que podrán sucederles en su futura carrera profesional, y que se conviertan en técnicos proactivos y conocedores de soluciones que por regla general, no se aprenden.



6. Necesidades Del Sector Productivo

A continuación se identifican las necesidades detectadas en el sector productivo que originan la oportunidad de negocio que se detalla en los siguientes puntos:

6.1 Análisis de la situación actual

Normalmente, la mayoría de los servidores Linux, obtienen los logs del sistema a través de servicios como rsyslog, o journalctl, que almacenan la información detallada de estos inicios de sesión, sean benignos o malignos. Igualmente, esto sería viable con un número de equipos y servidores muy reducidos, al contrario de nuestro caso.

Actualmente, cada vez aumentan más los casos de ataques dirigidos a servidores que están expuestos en internet, suelen ser realizados por atacantes que utilizan softwares automatizados para conseguir millones de combinaciones de usuarios y contraseñas, o por ejemplo, usando scripts automatizados que detecten las IP'S de servidores vulnerables, y ejecutando en ellos los mismos softwares de las credenciales.

En el ámbito educativo, como es en este caso, los servidores están siempre expuestos, debido a que se están sometiendo a continuas pruebas y prácticas experimentales, donde se realizan muchas conexiones desde distintos equipos, de manera que puede llegar a ser muy peligroso para la seguridad de la red.

Y por tanto, la revisión manual de los logs por un técnico, no sería viable, así que se necesita el sistema propuesto, que lo haga inteligentemente, sin necesidad de un técnico contratado para únicamente esa tarea.

En definitiva, la situación actual precisa de un sistema que realice estos registros de manera automatizada, ya que cada día son más los intentos de ataques a servidores, y en algún momento de manera manual, será inviable.



6.2 Necesidades del cliente y oportunidad de negocio

El cliente en cuestión en este caso, es el propio centro educativo (I.E.S Venancio Blanco). Este actuará como entorno de pruebas del proyecto.

La necesidad principal es evidente, puesto que, al haber un incremento de ataques, también debe haber un incremento de la seguridad en los servidores que alojan información muy importante del centro. Por ello, la solución es disponer de un entorno automatizado y seguro, que libere de preocupaciones y que detecte estos ataques de forma precisa e inmediata. De esta manera, se reducirá el tiempo de detección de estos mismos ataques, asegurando que la información está guardada de forma segura.

Este proyecto, resulta muy atractivo respecto a las oportunidades de negocio, debido a que es una manera sencilla y eficaz de mantener un servidor libre de ataques, sin utilizar nada más allá que herramientas que el propio S.O ofrece.

Tiene una gran oportunidad de negocio, puesto que, como en muchas otras empresas de tamaño mediano o pequeño, no tienen personal que cubra la administración de la seguridad de estos servidores, por lo que necesitarán un entorno automatizado que, además, se lleve a cabo con un coste muy bajo al lado de obtener un equipo especializado para la propia seguridad del servidor/es.

Además, el desarrollo de un sistema de seguridad propio del centro tiene varias ventajas:

- Simplicidad: Es un sistema portable, que puede adaptarse a cualquier servidor linux del centro, por lo que sería sencillo de implementar, por ejemplo, en los equipos de las aulas que utilicen de manera habitual servidores linux, como es el caso del GS. ASIR.
- Personalización: El centro podrá “elegir” cada uno de los filtros de seguridad que vea conveniente, además de ser un sistema flexible que puede adaptarse a cualquier política de seguridad, puesto que es el propio equipo el que lo diseña.
- Formación: Además de actualizar y mejorar la seguridad de los servidores, amplía los conocimientos impartidos en el grado, ya que el proyecto se convertiría en un recurso didáctico para los alumnos del centro.

Estas ventajas que conllevan la realización del sistema propio, dan una respuesta muy positiva a las necesidades del sector productivo.



6.3 El nuevo proyecto: Monitorización y respuesta inteligente ante ataques de fuerza bruta en servidores Linux

6.3.1 Tipo de proyecto

El proyecto se realizará en el centro educativo I.E.S Venancio Blanco, que actuará como la empresa/organización en la que realizaremos las pruebas necesarias.

Aunque no sea una empresa, está compuesta por decenas de servidores (sumando internos y usados en la formación), que utiliza una infraestructura de red en el que podremos simular perfectamente el desarrollo del proyecto.

El proyecto, estaría considerado como una mejora/ampliación de la misma infraestructura de red, ya que partiríamos de esa base, pero ampliando y mejorando la seguridad de la red, aplicando un nuevo sistema de seguridad para los servidores.

Este proyecto se enmarca en la categoría **B: Proyecto de innovación, investigación experimental o desarrollo**, ya que implica el **diseño, desarrollo e implementación de una herramienta técnica** destinada a mejorar la seguridad en servidores Linux, estando ambientado en el centro educativo I.E.S Venancio Blanco.

6.3.2 Características requeridas al proyecto

Para comenzar, este sistema deberá de, automáticamente y a tiempo real, analizar continuamente el fichero /var/log/auth.log , (ubicación de los logs de inicios de sesión/autenticación).

El script desarrollado, utilizará comandos como “grep” o “sed”, (son estructuras de control que utiliza bash), que detectarán, según los filtros o patrones aplicados, los inicios de sesión repetitivos y fallidos en el sistema.

Una vez se detecte que la IP atacante no cumple con los filtros de seguridad del script, el sistema la bloqueará automáticamente a través de reglas de “iptables”, y por ende se evitarán nuevos ataques de esa misma IP. Además, cada vez que se bloquee una IP sospechosa, aparecerá en pantalla para que el usuario tenga la información completa de dónde está ocurriendo el ataque.

Un elemento diferenciador de este sistema, es la simplicidad del mismo. Es muy ligero, de bajo coste y consumo, por lo que puede ejecutarse tanto en servidores internos importantes, como en cualquier servidor de prácticas de un alumno, ya que simplemente utiliza recursos del propio S.O.

Además, el sistema propuesto, será flexible y personalizable, es decir, podemos modificar, por ejemplo, cuántos inicios de sesión fallidos queremos permitir antes de categorizarlo como “sospechoso”.

El script desarrollado será seguro y robusto, de tal manera que, con pocos recursos, tengamos seguridad completa y evitemos falsos positivos o bloqueos innecesarios, quedando todo registrado en un fichero propio.

Las herramientas serán las integradas por el propio S.O, por lo que tendrá una alta compatibilidad con cualquier servidor, además de ser sencillo de utilizar, debido a su simpleza a la hora de mostrar la información al usuario.

El desarrollo del script no necesita muchos recursos humanos, más allá de un equipo capaz de realizar el mismo script, y de mantenerlo correctamente en los servidores del centro.

6.3.3 Obligaciones fiscales, laborales y de prevención de riesgo

En este proyecto, se adaptará una nueva empresa emergente especializada en ciberseguridad, que integrará como cliente al instituto I.E.S Venancio Blanco, aplicando el servicio mencionado anteriormente para ayudar con el mantenimiento y seguridad de todos los servidores linux.

Esta pequeña empresa, sería una “microempresa” en el ámbito tecnológico, y su actividad principal sería “Servicios de informática y desarrollo de software” dentro del epígrafe 765 del impuesto sobre actividades económicas o IAE.

Una de las obligaciones más importantes de esta microempresa, sería darse de alta en la Agencia Tributaria por el modelo 036, y las obligaciones serían:

- Registro de facturas (ingresos/gastos)
- Trimestral de IVA
- Impuesto de sociedades
- IRPF de trabajadores.

Y en este caso, al ofrecer el servicio al I.E.S Venancio Blanco, la facturación vendría dada por la prestación de estos “servicios tecnológico”.

Y además, al manipular información sensible del servidor, esta microempresa debe cumplir con la “Ley Orgánica 3/2018 de Protección de Datos Personales y Garantía de los Derechos Digitales”, para así confirmar que esta información no será difundida ni utilizada bajo ninguna circunstancia, y que será confidencial.



Dentro de las obligaciones laborales, se cumpliría con el Estatuto de los Trabajadores y el Convenio Colectivo Estatal de Empresas de Consultoría y Estudios de Mercado y Opinión pública.

Las principales obligaciones laborales serían:

- Medidas de igualdad y conciliación laboral.
- Alta del técnico/s en la seguridad social.
- Presentar las cotizaciones mensuales.
- Cumplir con las jornadas laborales y tener un control del horario del trabajador/es

Como se mencionaba anteriormente, este proyecto serviría como formación para alumnos, por lo que también se incluirán sesiones de formación para los profesores, de manera que estos conocimientos se puedan incluir en el temario, y que además el instituto pueda usar el sistema de forma autónoma.

Y dentro de los riesgos laborales, la empresa cumpliría con la ley 31/1995 de Prevención de Riesgos Laborales.

En este caso, los riesgos más comunes al tratarse de un trabajo de informática, serían riesgos visuales, de posturas o ergonómicos, junto con los riesgos de estrés.

Para prevenir estos riesgos, aplicamos:

- Evaluación del puesto de trabajo.
- Adaptación del puesto de trabajo con periféricos ajustables y ergonómicos.
- Descansos cada x tiempo.
- Iluminación correcta.
- Ventilación adecuada.



6.3.4 Ayudas/subvenciones

Como el proyecto se realiza dentro del ámbito de las soluciones tecnológicas innovadoras, habrían varias ayudas y subvenciones públicas para fomentar la ciberseguridad y la tecnología.

Habría varias ayudas y subvenciones disponibles:

- Subvenciones a la digitalización y modernización tecnológica:

1. Kit digital: Financia la implantación de soluciones digitales en microempresas.
2. Plan de Digitalización del Sistema Educativo: Subvenciones para proyectos que mejoren la seguridad de los entornos TIC de centros educativos.
3. Ayudas a la innovación tecnológica de la Junta de CyL: Ofrece apoyo a proyectos de desarrollo tecnológico, innovación digital y formación en ciberseguridad.

- Subvenciones para formación y sensibilización en ciberseguridad:

INCIBE: Ofrece programas de colaboración con proyectos centrados en las buenas prácticas en seguridad, y dado que este proyecto serviría como futura formación, encaja con la subvención dado que integraría más profundamente la ciberseguridad en el CFGS. ASIR.

-Posibilidad de financiación:

Podría estar combinada en:

1. Subvenciones públicas a la innovación educativa y digitalización.
2. Aportaciones del propio centro educativo.
3. Colaboración con empresas locales.

En conclusión, habría una gran cantidad de ayudas y subvenciones a las que se podría acceder con el proyecto.

Estas ayudas impulsarían notablemente la repercusión del proyecto tanto educativa como tecnológica, impulsando la microempresa hacia nuevos clientes interesados, además del centro educativo en cuestión.

7. Diseño Del Proyecto

7.1 Fases del proyecto

El desarrollo de este proyecto se llevará a cabo en cuatro fases: análisis, diseño, implementación y pruebas, que pasan a detallarse a continuación:

7.2 Análisis:

Tenemos 2 tipos de requisitos, funcionales y no funcionales:

Funcionales:

- Monitorización a tiempo real del sistema.
- Detección mediante ciertos filtros de los intentos de autenticación en el sistema.
- Bloqueo de las IP correspondientes dependiendo de estos mismos filtros.
- Quedar registrados todos estos eventos en un fichero específico.

No funcionales:

- Facilidad de aplicación a cualquier sistema.
- Ligereza y bajo consumo usando propiedades nativas del sistema.
- Seguridad en el manejo de los ficheros.
- Compatibilidad entre distintos S.O de linux, así como Debian o Ubuntu.

Además, este sistema no almacenará contraseñas ni intentará corresponder al atacante, realizando acciones ofensivas, y se centrará en el servicio SSH.



7.3 Diseño:

-Monitorización a tiempo real del sistema: Para su monitorización a tiempo real, se utilizará “tail -F /var/log/auth.log”. La detección se ejecutará mediante un script que se realizará en BASH y este mismo script se ejecutará como servicio “systemd” para asegurar que se inicia automáticamente.

-Para la detección de los métodos de autenticación, se definirán los parámetros TRHESHOLD y WINDOW en un fichero de configuración (ej: /etc/deteccion_ip/config.cfg). Se aplicarán expresiones regulares para identificar errores importantes como “invalid user, failed password” ...

-Para el bloqueo de las direcciones IP correspondientes, se usará IPTABLES y, se realizará lo siguiente:

-La función de bloqueo, comprobará si para esa IP existe una regla con “iptables –C” y, si no existe, se insertará una regla.

-Las modificaciones de “iptables” quedarán registradas en un fichero para poder realizar las pruebas necesarias y poder visualizar lo que ocurre con las IP bloqueadas, consiguiendo de esta manera, poder tener un control de todo lo que ocurra en el sistema, de manera más visual.

-Además, como se comenta, quedarán registrados todas las acciones de “iptables” en un fichero, el cual crearemos con un formato visual y sencillo: “fecha del suceso (timestamp) --> ip --> acción realizada → motivo por el cual se ha realizado la acción”. Además añadiremos una línea para tener un control sobre cuántas veces ha intentado el atacante acceder al sistema.

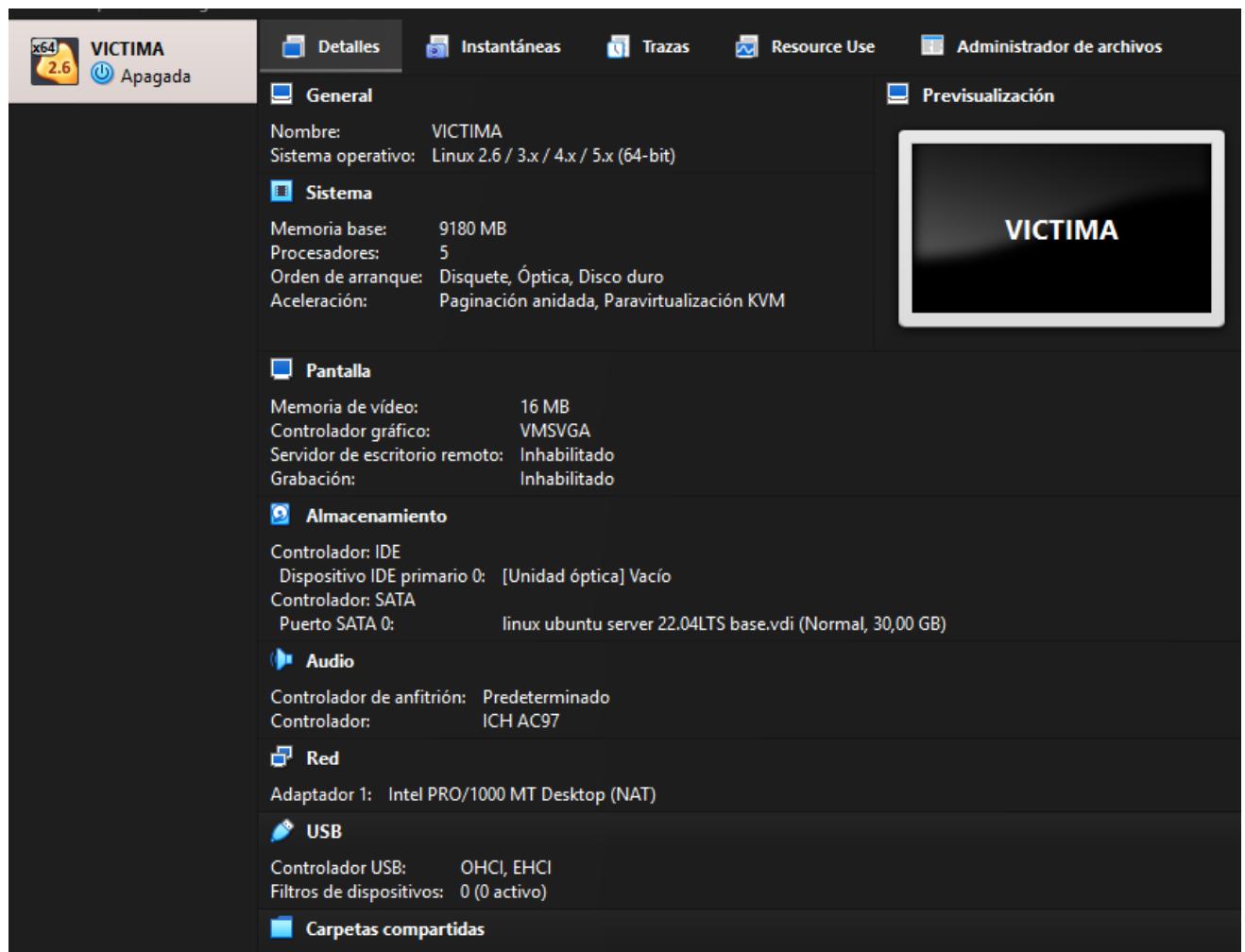
– Sobre los requisitos no funcionales:

- -Se podría realizar un script de instalación, que copie el script en el directorio correspondiente, con los permisos adecuados, instale “systemd” y habilite el servicio que crearemos.
- -Sobre la ligereza y el bajo consumo, lo realizaremos gracias a implementarlo en bash y utilizando herramientas nativas como “grep, sed” y así no depender de ninguna herramienta externa.
- -Sobre la seguridad en el manejo de los ficheros, la aseguraremos aplicando permisos restrictivos, evitando la accesibilidad por cualquier persona con acceso al sistema, y las acciones que se realicen con “iptables” se realizarán desde ROOT y quedará registrado todo lo que ocurra.

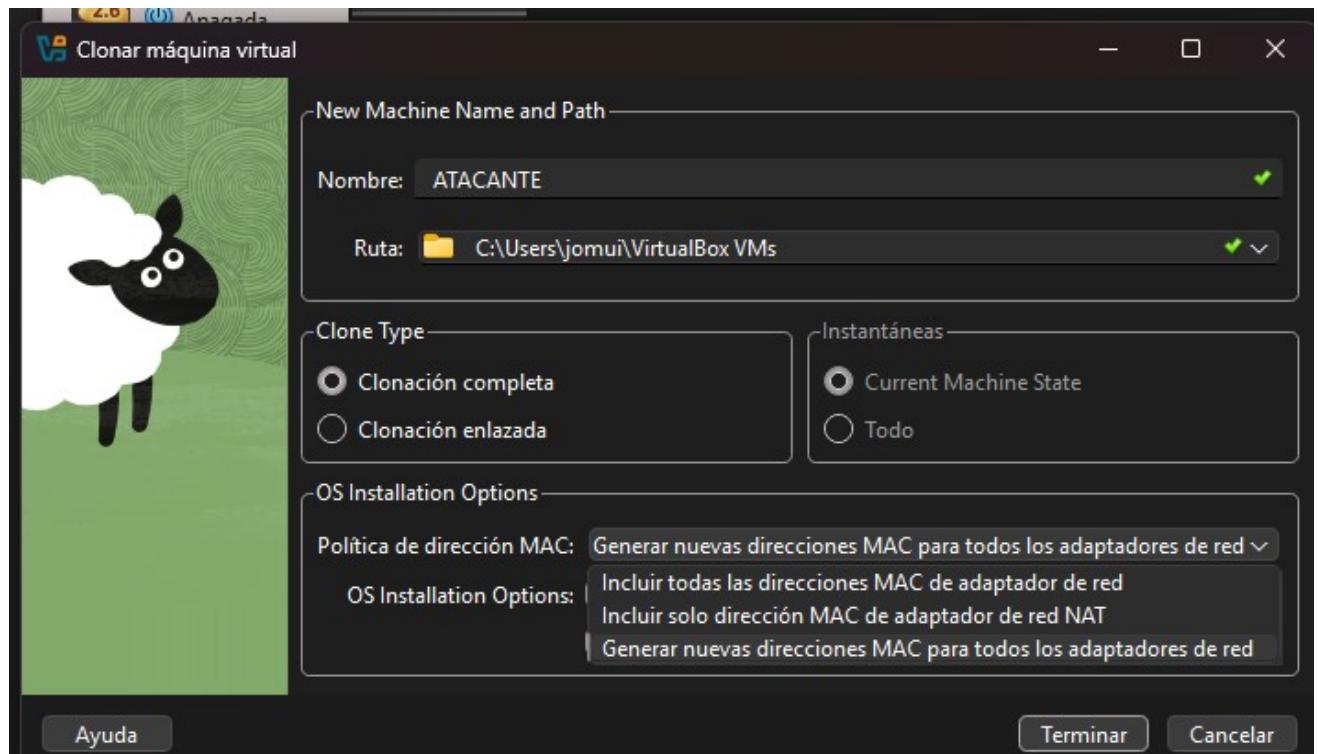


7.4 Implementación

Se empieza dentro de virtualbox. Lo que se hará ahora es clonar una máquina con UBUNTU 24.04 LTS adquirida dentro del módulo de ASO. A esta primera máquina la llamaremos VICTIMA y la clonaremos para conseguir la ATACANTE:



La clonación:



Administración de Sistemas Informáticos en Red

Ya con ambas máquinas, se debe comprobar siempre que su configuración de red sea la correcta:

The image displays two side-by-side Kali Linux desktop sessions. Both sessions show the 'Settings' window with the 'Red' (Network) option selected.

VICTIMA - Settings

- Red Configuration:**
 - Adaptador 1: Habilitar adaptador de red (checked), Conectar a: Red interna, Nombre: intnet, Tipo de Adaptador: Intel PRO/1000 MT Desktop (82540EM), Promiscuous Mode: Denegar, Dirección MAC: 0800270DC72C, Virtual Cable Connected (checked).
- Puertos serie (Serial Ports):** Puerto 1: Enable Serial Port (unchecked), Número de puerto: COM1, IRQ: 4, Puerto I/O: x3F8.

ATACANTE - Settings

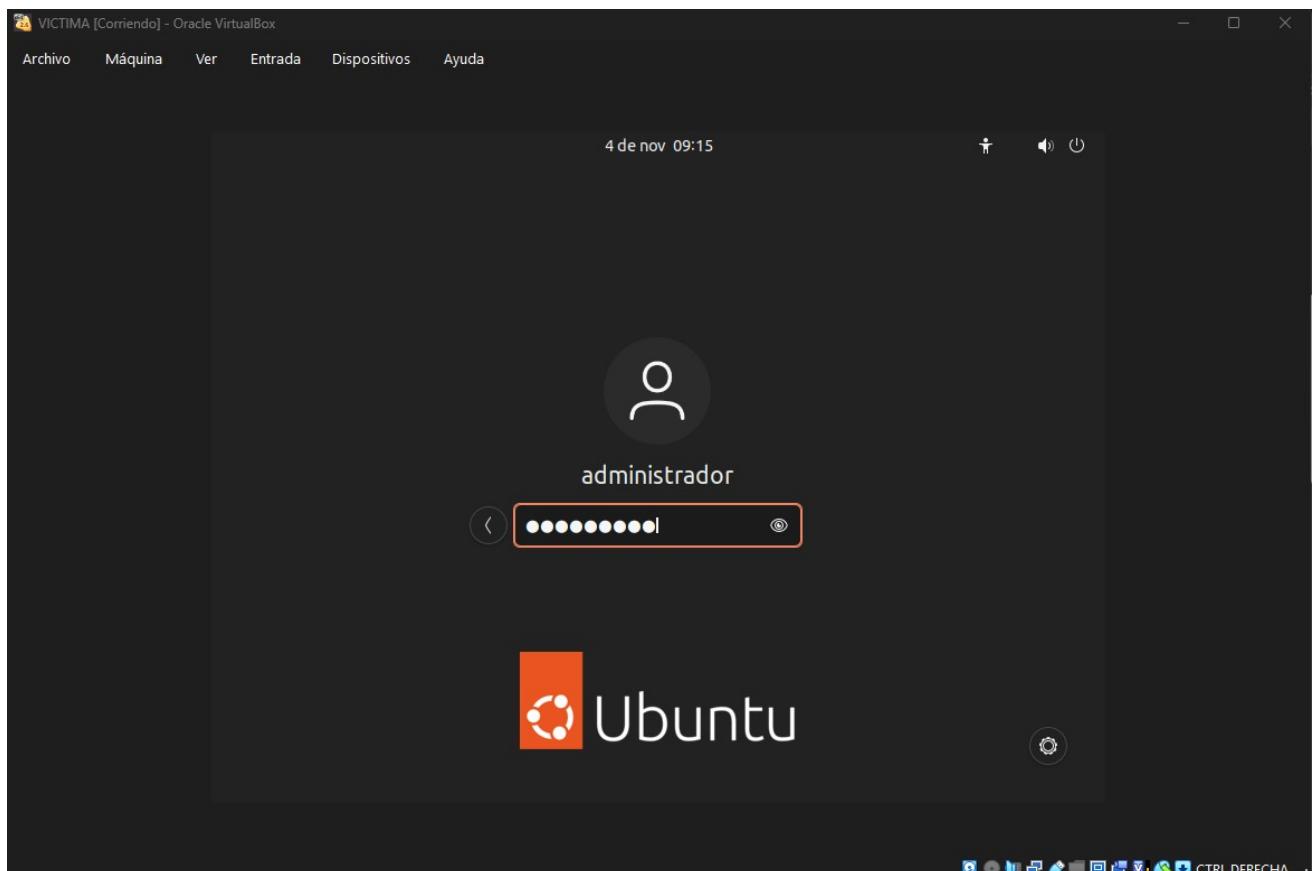
- Red Configuration:**
 - Adaptador 1: Habilitar adaptador de red (checked), Conectar a: Red interna, Nombre: intnet, Tipo de Adaptador: Intel PRO/1000 MT Desktop (82540EM), Promiscuous Mode: Denegar, Dirección MAC: 08002730EF69, Virtual Cable Connected (checked).
- Puertos serie (Serial Ports):** Puerto 1: Enable Serial Port (unchecked), Número de puerto: COM1, IRQ: 4, Puerto I/O: x3F8.

Both windows include standard buttons: Aceptar (Accept), Cancelar (Cancel), and Ayuda (Help).



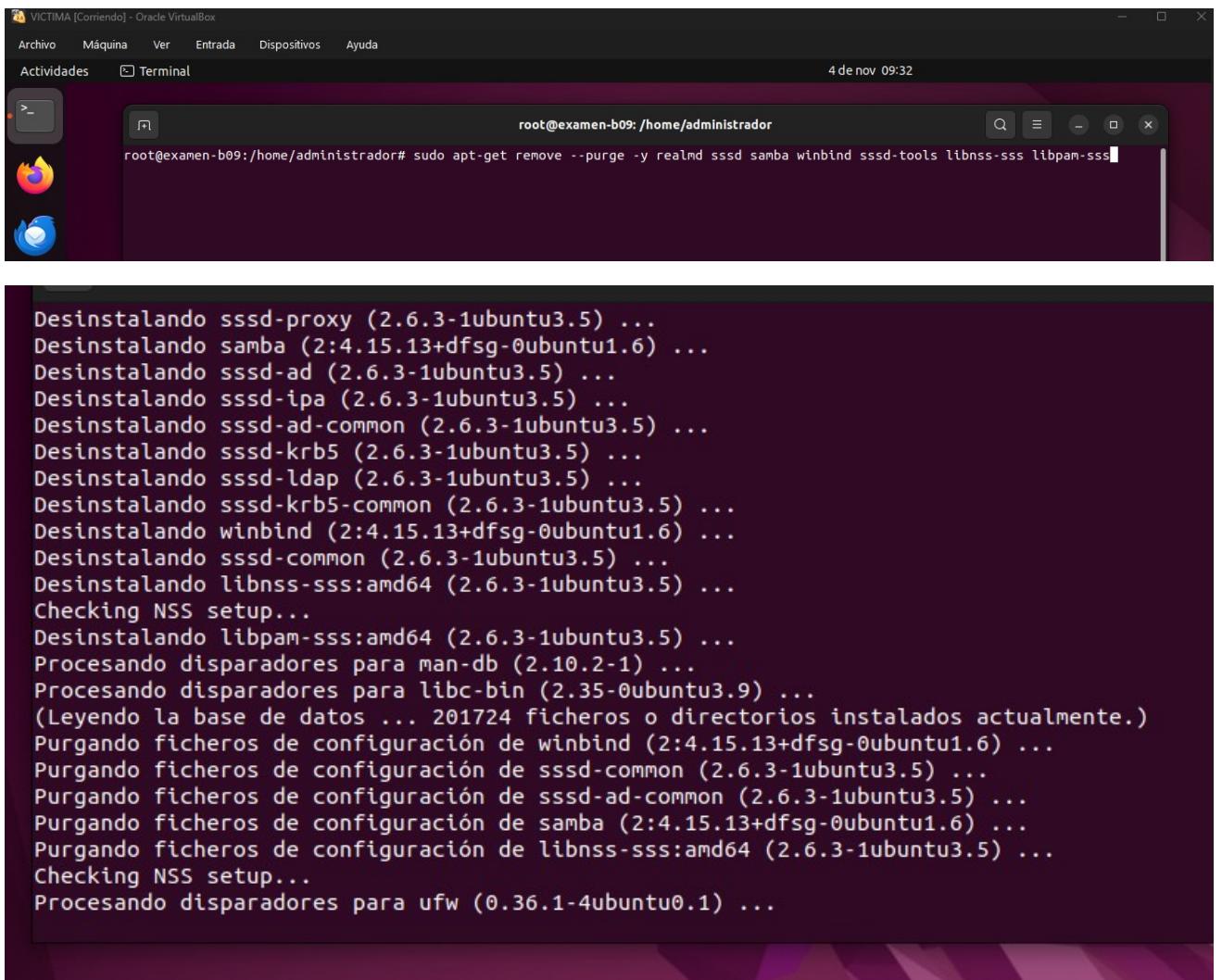
Lo primero que se hará será configurar ambas máquinas para evitar conflictos:

Primero la víctima:



Primero, se dejará la máquina totalmente limpia, de manera que se podrán evitar futuros conflictos:

Se elimina todo lo relacionado con “samba” para eliminar el dominio y empezar de 0:

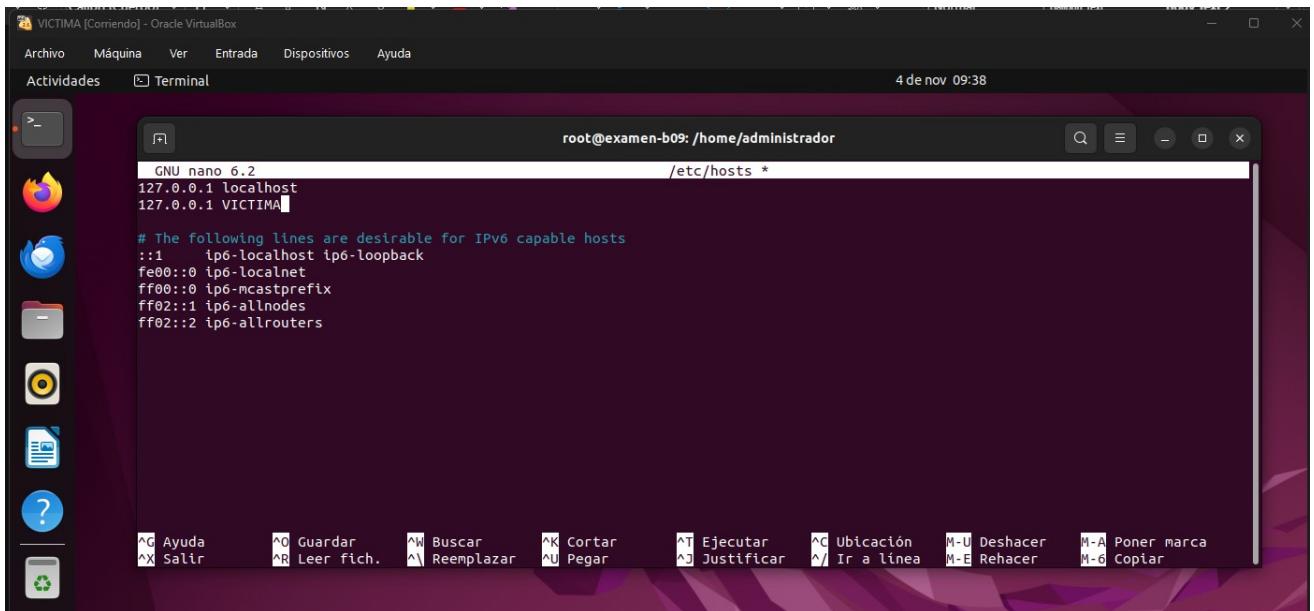


The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "root@examen-b09:/home/administrador". The command entered is "sudo apt-get remove --purge -y realmd sssd samba winbind sssd-tools libnss-sss libpam-sss". The output of the command is displayed in the terminal window, showing the removal of various Samba-related packages and their dependencies.

```
Desinstalando sssd-proxy (2.6.3-1ubuntu3.5) ...
Desinstalando samba (2:4.15.13+dfsg-0ubuntu1.6) ...
Desinstalando sssd-ad (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-ipa (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-ad-common (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-krb5 (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-ldap (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-krb5-common (2.6.3-1ubuntu3.5) ...
Desinstalando winbind (2:4.15.13+dfsg-0ubuntu1.6) ...
Desinstalando sssd-common (2.6.3-1ubuntu3.5) ...
Desinstalando libnss-sss:amd64 (2.6.3-1ubuntu3.5) ...
Checking NSS setup...
Desinstalando libpam-sss:amd64 (2.6.3-1ubuntu3.5) ...
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para libc-bin (2.35-0ubuntu3.9) ...
(Leyendo la base de datos ... 201724 ficheros o directorios instalados actualmente.)
Purgando ficheros de configuración de winbind (2:4.15.13+dfsg-0ubuntu1.6) ...
Purgando ficheros de configuración de sssd-common (2.6.3-1ubuntu3.5) ...
Purgando ficheros de configuración de sssd-ad-common (2.6.3-1ubuntu3.5) ...
Purgando ficheros de configuración de samba (2:4.15.13+dfsg-0ubuntu1.6) ...
Purgando ficheros de configuración de libnss-sss:amd64 (2.6.3-1ubuntu3.5) ...
Checking NSS setup...
Procesando disparadores para ufw (0.36.1-4ubuntu0.1) ...
```



Ahora, se queda limpio también /etc/hosts y /etc/hostname para dejar solo lo esencial y adaptarlo al proyecto:

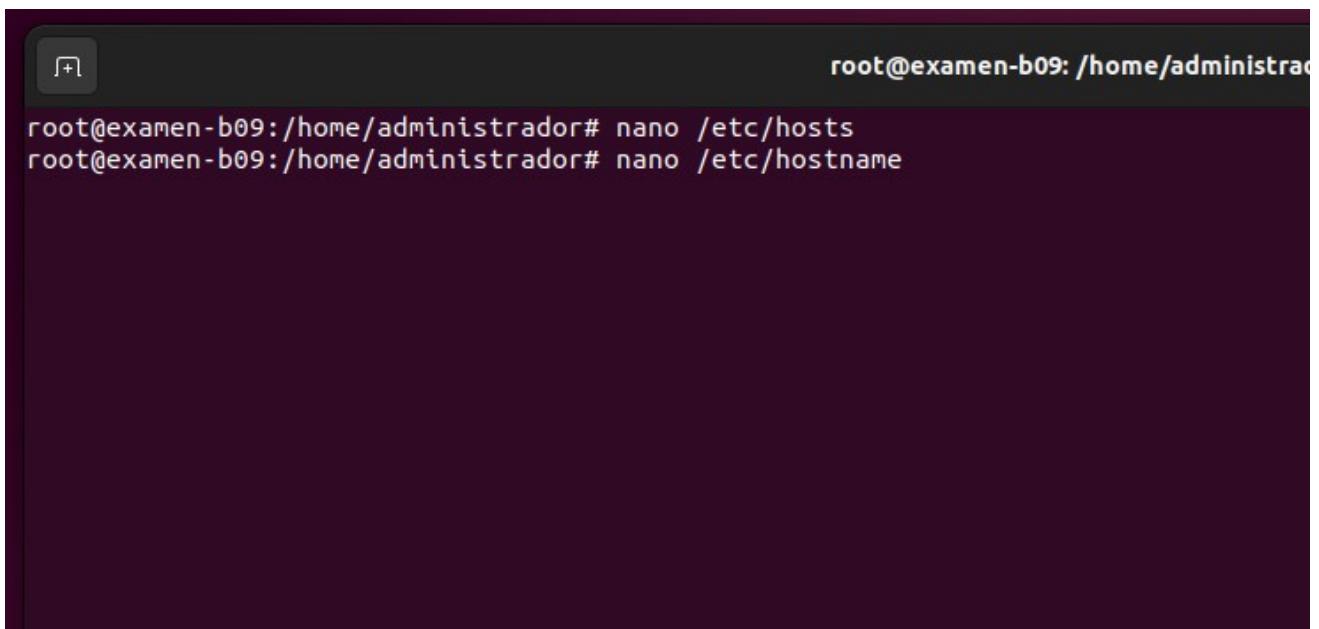


The screenshot shows a Linux desktop environment with a purple-themed window manager. A terminal window titled "root@examen-b09: /home/administrador" is open, displaying the contents of the "/etc/hosts" file. The file contains the following entries:

```
GNU nano 6.2
127.0.0.1 localhost
127.0.0.1 VICTIMA

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

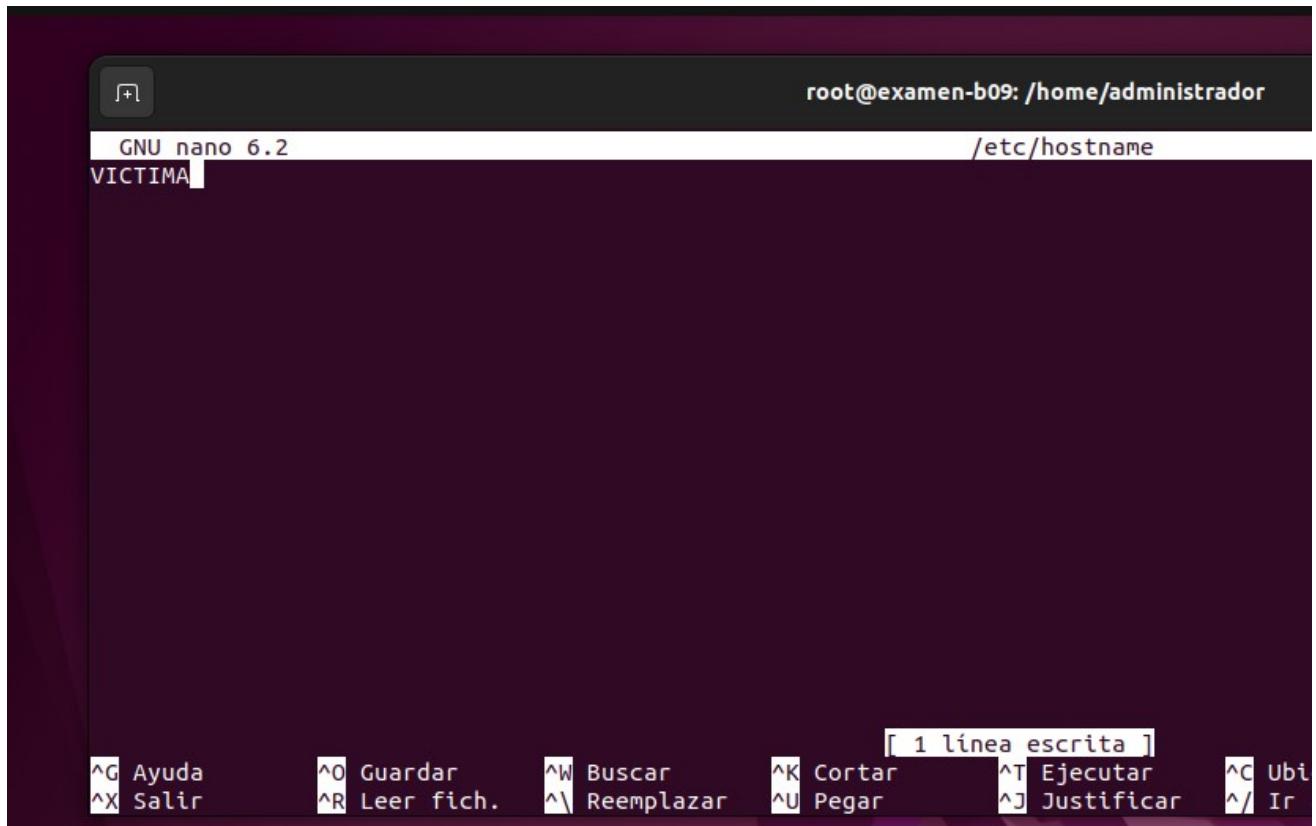
The terminal window has a standard nano editor interface with keyboard shortcuts at the bottom.



The screenshot shows a terminal window with the command "nano /etc/hosts" entered and executed. The output shows the host configuration:

```
root@examen-b09:/home/administrador# nano /etc/hosts
root@examen-b09:/home/administrador# nano /etc/hostname
```

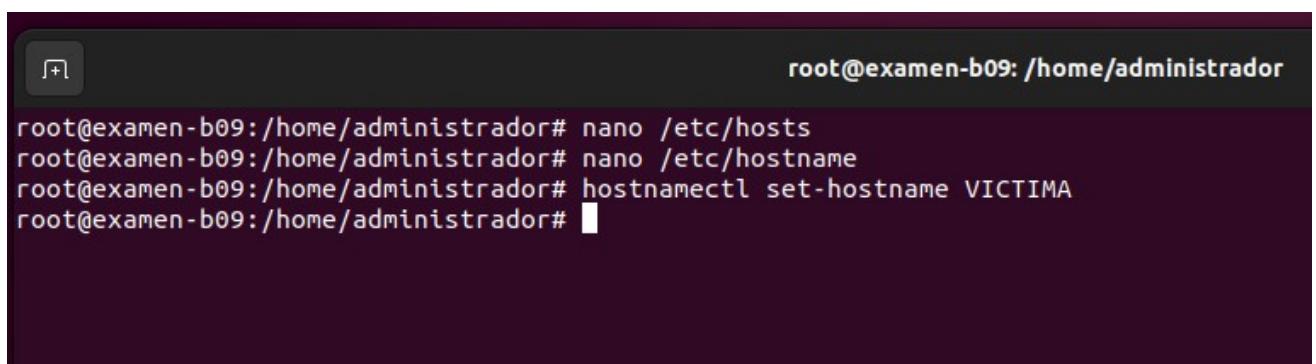




```
GNU nano 6.2
VICTIMA

[ 1 linea escrita ]
```

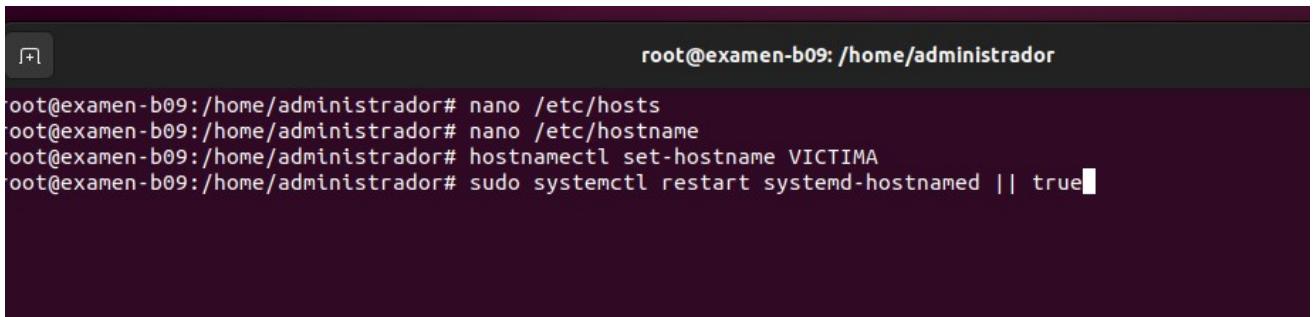
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubi
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir



```
root@examen-b09:/home/administrador# nano /etc/hosts
root@examen-b09:/home/administrador# nano /etc/hostname
root@examen-b09:/home/administrador# hostnamectl set-hostname VICTIMA
root@examen-b09:/home/administrador#
```

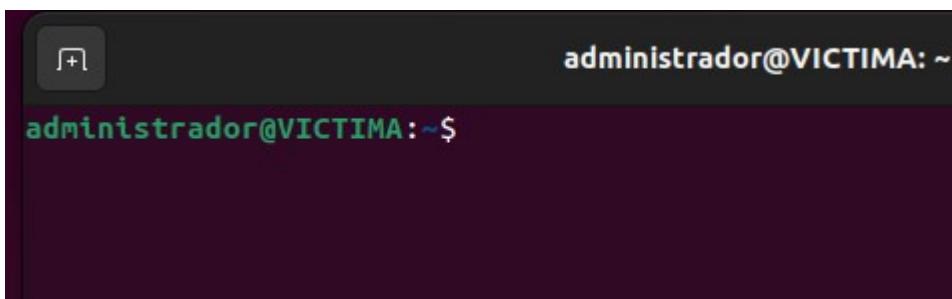


Y se reinicia el servicio de hostname para aplicar y reflejar los cambios:



```
root@examen-b09:/home/administrador# nano /etc/hosts
root@examen-b09:/home/administrador# nano /etc/hostname
root@examen-b09:/home/administrador# hostnamectl set-hostname VICTIMA
root@examen-b09:/home/administrador# sudo systemctl restart systemd-hostnamed || true
```

Se reinicia la máquina y ya están los cambios aplicados como se puede observar:



```
administrador@VICTIMA: ~
administrador@VICTIMA:~$
```



Ahora, se continuará con la máquina atacante, a la cuál, además de modificar hosts y hostname, al ser máquina clonada, también se le regenerarán el machine-id y claves SSH para evitar conflictos por duplicación:

Primero se comienza eliminando samba y todo lo relacionado con ello, al igual que hicimos en la víctima:



```
root@examen-b09:/home/administrador# sudo apt-get remove --purge -y realmd sssd samba winbind sssd-tools libnss-sss libpam-sss
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete «realm» no está instalado, no se eliminará
El paquete «sssd-tools» no está instalado, no se eliminará
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  ibverbs-providers ldap-utils libbasicobjects0 libc-ares2 libcephfs2 libcollection4 libdhash1 libgfapi0 libgfrpc0
  libgfrpc0 libglusterfs0 libibverbs1 libini-config5 libipa-hbac0 libnfsidmap1 libpam-pwquality libpath-utils1
  librados2 librdmacm1 libref-array1 libsss-certmap0 libsss-idmap0 libsss-nss-idmap0 liburing2 libwpe-1.0-1
  libwpebackend-fdo-1.0-1 python3-sss samba-vfs-modules tdb-tools
Utilice «sudo apt autoremove» para eliminarlos.
Los siguientes paquetes se ELIMINARÁN:
  libnss-sss* libpam-sss* samba* sssd* sssd-ad* sssd-ad-common* sssd-common* sssd-ipa* sssd-krb5* sssd-krb5-common*
  sssd-ldap* sssd-proxy* winbind*
0 actualizados, 0 nuevos se instalarán, 13 para eliminar y 0 no actualizados.
Se liberarán 27,0 MB después de esta operación.
(Leyendo la base de datos ... 202133 ficheros o directorios instalados actualmente.)
Desinstalando sssd (2.6.3-1ubuntu3.5) ...
Desinstalando sssd-proxy (2.6.3-1ubuntu3.5) ...
Desinstalando samba (2:4.15.13+dfsg-0ubuntu1.6) ...
```



Se cambiará el host:

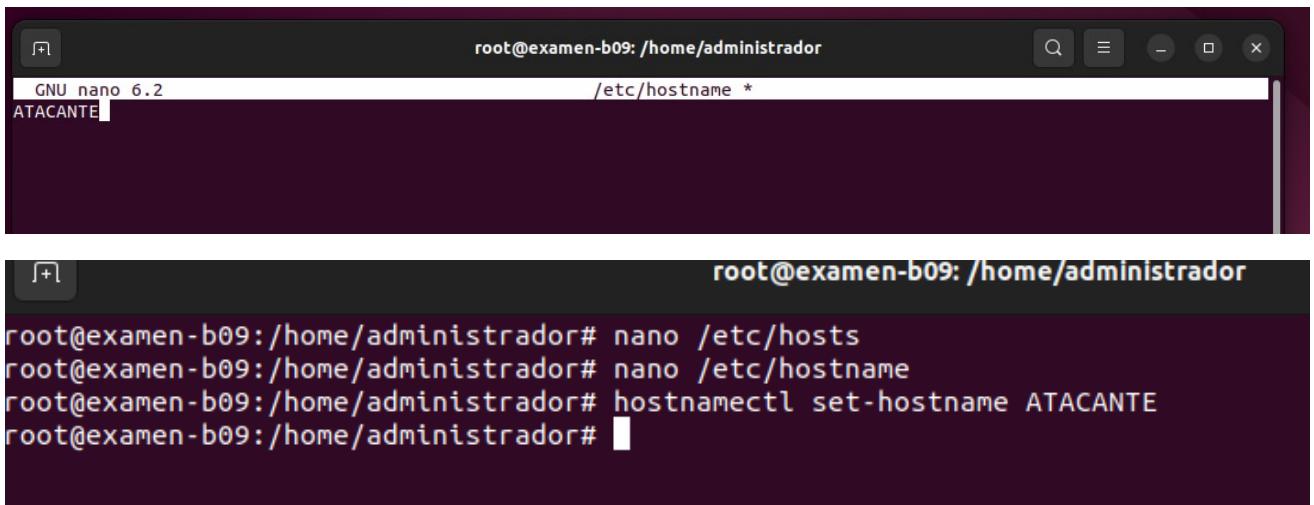
```
root@examen-b09: /h
root@examen-b09:/home/administrador# nano /etc/hosts
```

```
root@examen-
GNU nano 6.2
127.0.0.1 localhost
127.0.0.1 ATACANTE

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```



Seguidamente el hostname:



The screenshot shows a terminal window with a dark background. At the top, it says "root@examen-b09: /home/administrador". Below that, there's a status bar with "GNU nano 6.2" on the left and "/etc/hostname *". The main area of the terminal contains the word "ATACANTE" in white text. This indicates that the user has just edited the "/etc/hostname" file to change the host name.

```
root@examen-b09: /home/administrador
GNU nano 6.2
/etc/hostname *
ATACANTE
```

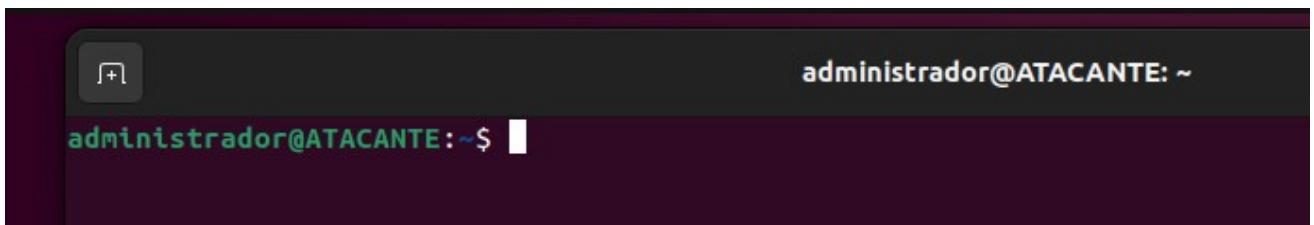


```
root@examen-b09: /home/administrador
root@examen-b09:/home/administrador# nano /etc/hosts
root@examen-b09:/home/administrador# nano /etc/hostname
root@examen-b09:/home/administrador# hostnamectl set-hostname ATACANTE
root@examen-b09:/home/administrador#
```

Se reinicia el servicio de hostname y posteriormente, la máquina:

```
root@examen-b09:/home/administrador# systemctl restart systemd-hostnamed
root@examen-b09:/home/administrador#
```

De esta manera, ya estará también el hostname del atacante bien configurado:



The screenshot shows a terminal window with a dark background. At the top, it says "administrador@ATACANTE: ~". Below that, there's a status bar with "administrador@ATACANTE: ~\$". The main area of the terminal is empty, showing a single prompt character "\$". This indicates that the host name has been successfully changed to "ATACANTE".

```
administrador@ATACANTE: ~
administrador@ATACANTE: ~$
```



En este paso, se regenerarán el machine-id y claves SSH para evitar conflictos al ser un clon, como se indica anteriormente:

Primero, el machine-id:

```
administrador@ATACANTE:~$ sudo su
[sudo] contraseña para administrador:
root@ATACANTE:/home/administrador# rm -f /etc/machine-id
root@ATACANTE:/home/administrador# systemd-machine-id-setup
Initializing machine ID from random generator.
root@ATACANTE:/home/administrador#
```

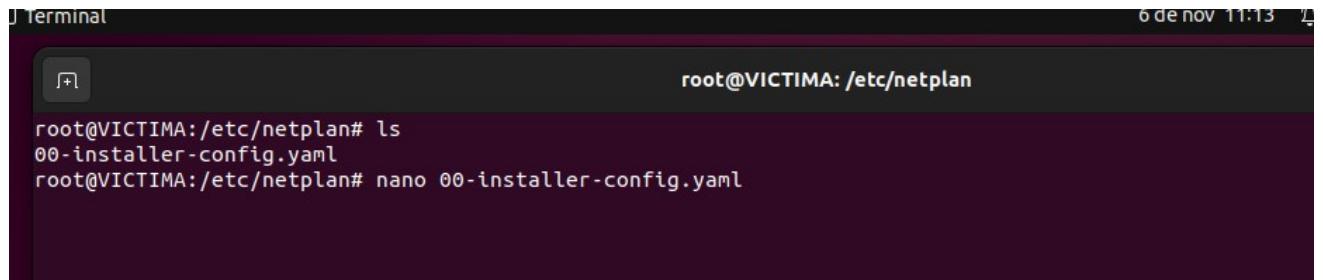
Y ahora, las claves SSH:

```
root@ATACANTE:/home/administrador# rm -f /etc/ssh/ssh_host_*
root@ATACANTE:/home/administrador# ssh-keygen -A
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
root@ATACANTE:/home/administrador# systemctl restart ssh
root@ATACANTE:/home/administrador#
```

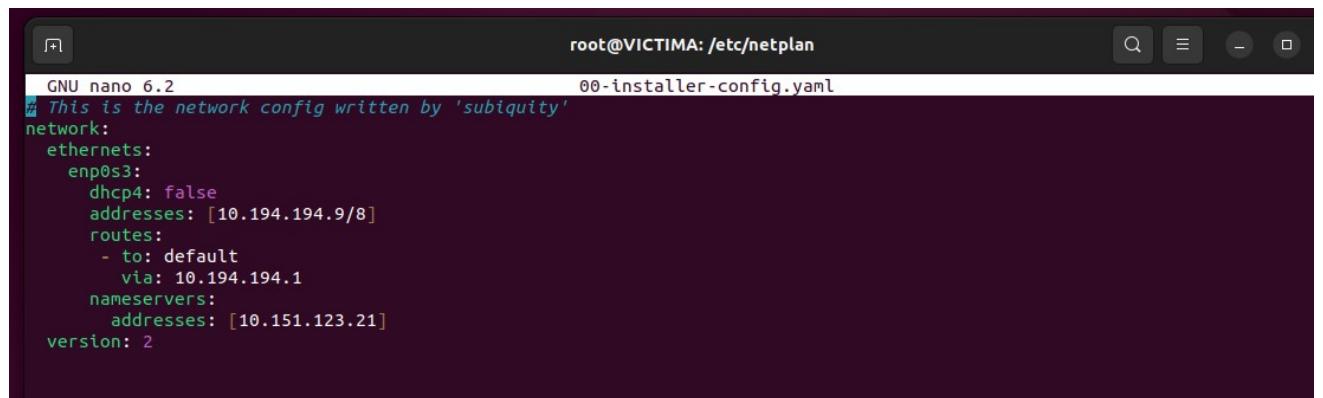


Ahora si, con las máquinas limpias y configuradas, estarán las máquinas totalmente listas para comenzar con el proyecto.

Lo primero que se hará será configurar el directorio NETPLAN para que se pueda trabajar correctamente:



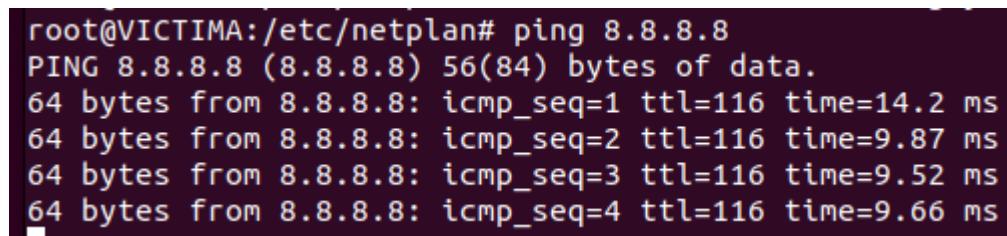
```
Terminal 6 de nov 11:13
root@VICTIMA: /etc/netplan
root@VICTIMA:/etc/netplan# ls
00-installer-config.yaml
root@VICTIMA:/etc/netplan# nano 00-installer-config.yaml
```



```
root@VICTIMA: /etc/netplan
root@VICTIMA:/etc/netplan
GNU nano 6.2
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [10.194.194.9/8]
      routes:
        - to: default
          via: 10.194.194.1
      nameservers:
        addresses: [10.151.123.21]
  version: 2
```

Se comprueba el ping, por ejemplo a 8.8.8.8, para comprobar si está funcionando correctamente:

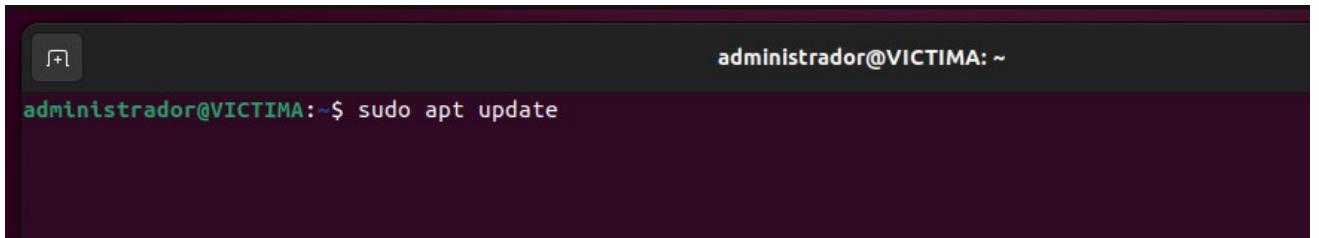
Y se observa que no hay problema:



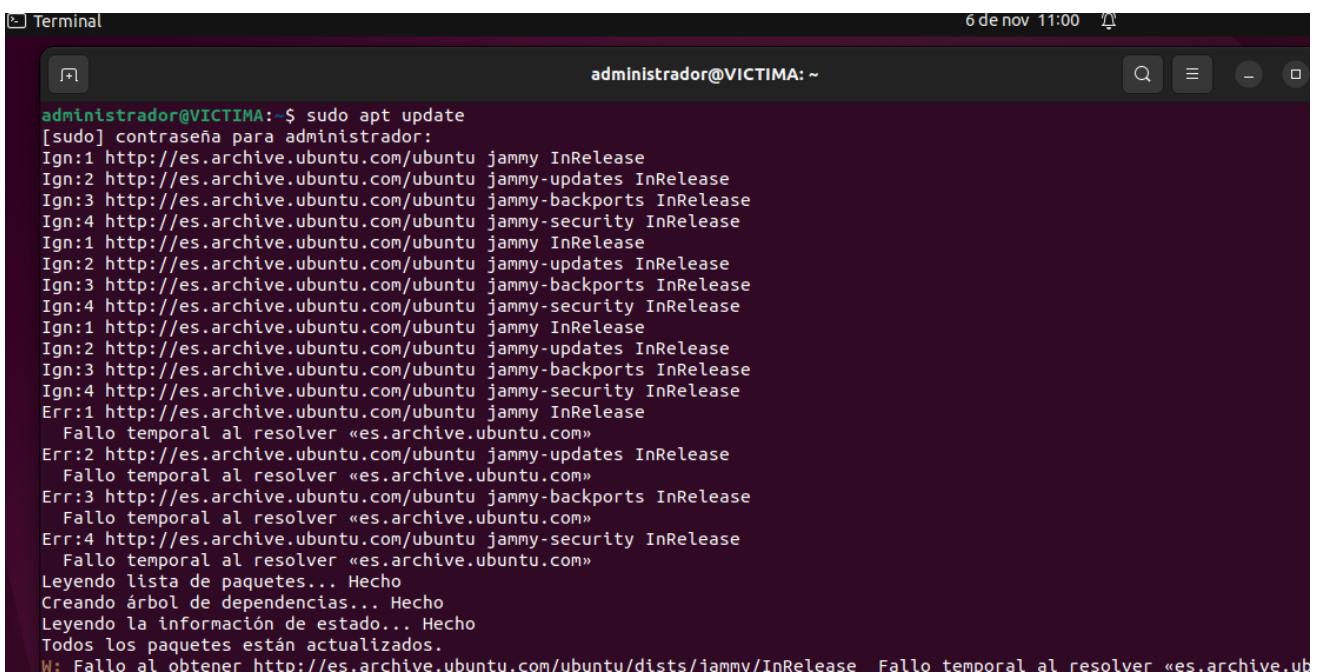
```
root@VICTIMA:/etc/netplan# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=14.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=9.87 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=9.52 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=9.66 ms
```



Para comenzar, se empezaría en la máquina VÍCTIMA y se verificaría que está lo básico instalado, que sería “openssh” e “iptables”:



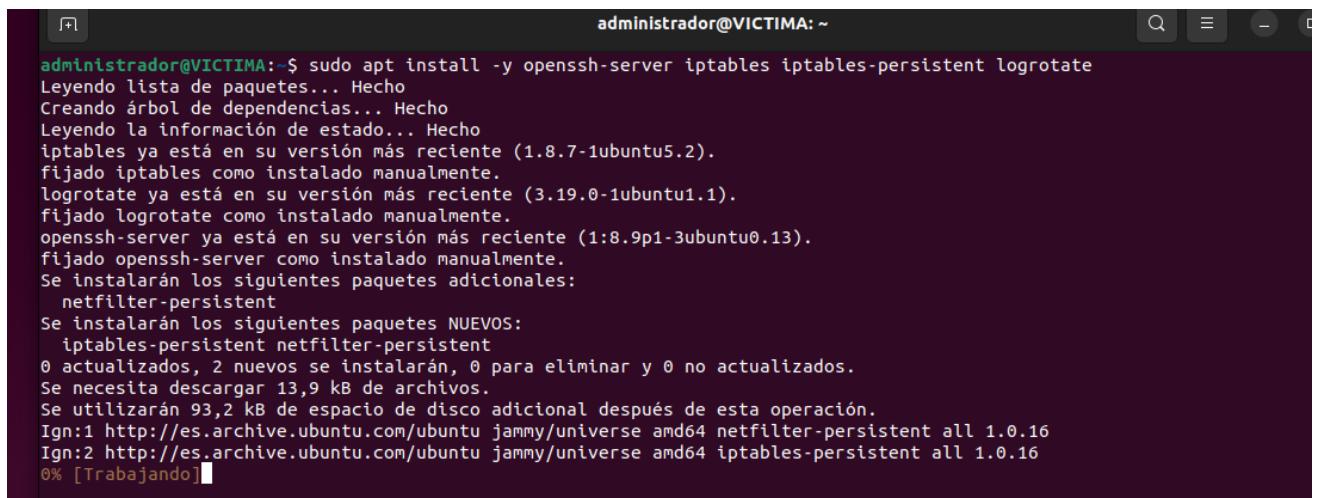
```
administrador@VICTIMA:~$ sudo apt update
```



```
6 de nov 11:00 administrador@VICTIMA: ~
administrador@VICTIMA:~$ sudo apt update
[sudo] contraseña para administrador:
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Err:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
  Fallo temporal al resolver «es.archive.ubuntu.com»
Err:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
  Fallo temporal al resolver «es.archive.ubuntu.com»
Err:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
  Fallo temporal al resolver «es.archive.ubuntu.com»
Err:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
  Fallo temporal al resolver «es.archive.ubuntu.com»
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
W: Fallo al obtener http://es.archive.ubuntu.com/ubuntu/dists/jammy/InRelease. Fallo temporal al resolver «es.archive.ub
```

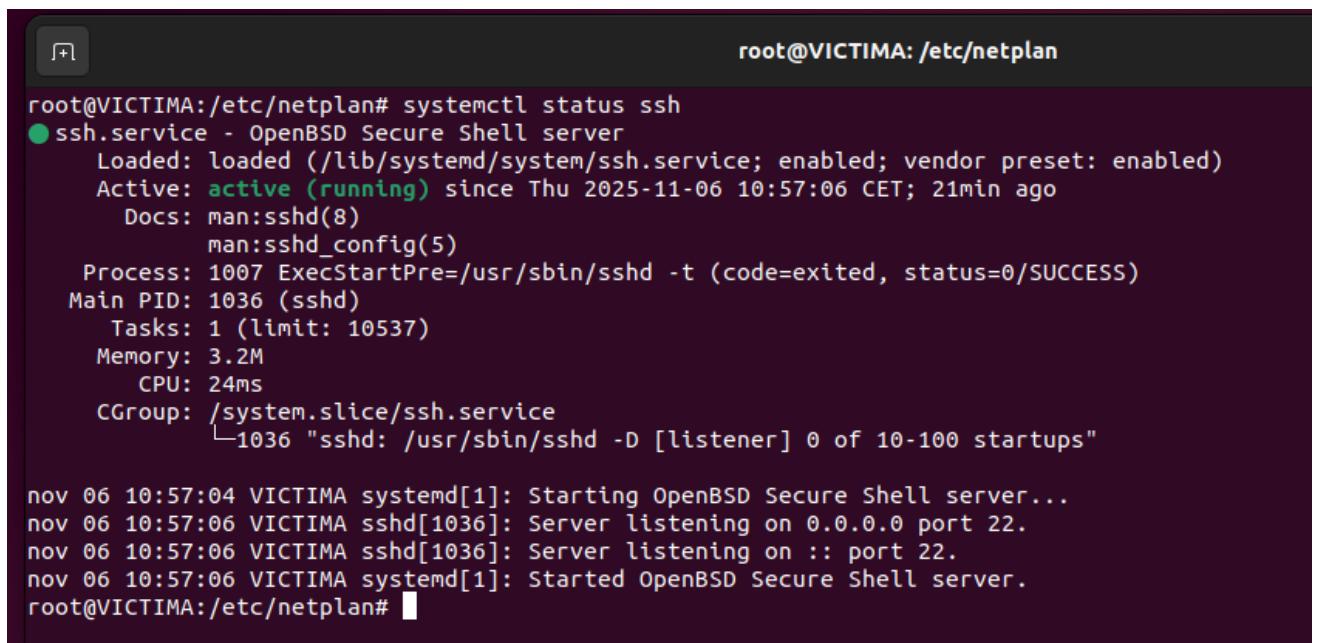


Tras actualizar, se instalará openssh e iptables, por si no estuviera ya instalado, y que no haya problemas a futuro:



```
administrador@VICTIMA:~$ sudo apt install -y openssh-server iptables iptables-persistent logrotate
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
iptables ya está en su versión más reciente (1.8.7-1ubuntu5.2).
fijado iptables como instalado manualmente.
logrotate ya está en su versión más reciente (3.19.0-1ubuntu1.1).
fijado logrotate como instalado manualmente.
openssh-server ya está en su versión más reciente (1:8.9p1-3ubuntu0.13).
fijado openssh-server como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
 netfilter-persistent
Se instalarán los siguientes paquetes NUEVOS:
 iptables-persistent netfilter-persistent
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 13,9 kB de archivos.
Se utilizarán 93,2 kB de espacio de disco adicional después de esta operación.
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 netfilter-persistent all 1.0.16
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 iptables-persistent all 1.0.16
0% [Trabajando]
```

Se puede ver que SSH está activo:



```
root@VICTIMA:/etc/netplan# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-11-06 10:57:06 CET; 21min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Process: 1007 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 1036 (sshd)
      Tasks: 1 (limit: 10537)
     Memory: 3.2M
        CPU: 24ms
       CGroup: /system.slice/ssh.service
                 └─1036 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

nov 06 10:57:04 VICTIMA systemd[1]: Starting OpenBSD Secure Shell server...
nov 06 10:57:06 VICTIMA sshd[1036]: Server listening on 0.0.0.0 port 22.
nov 06 10:57:06 VICTIMA sshd[1036]: Server listening on :: port 22.
nov 06 10:57:06 VICTIMA systemd[1]: Started OpenBSD Secure Shell server.
root@VICTIMA:/etc/netplan#
```



Y ahora se observa como IPTABLES está correctamente instalado:

```
root@VICTIMA:/etc/netplan# dpkg -l | grep iptables
ii  iptables                         1.8.7-1ubuntu5.2          amd64      administration tools for
  packet filtering and NAT
root@VICTIMA:/etc/netplan#
```

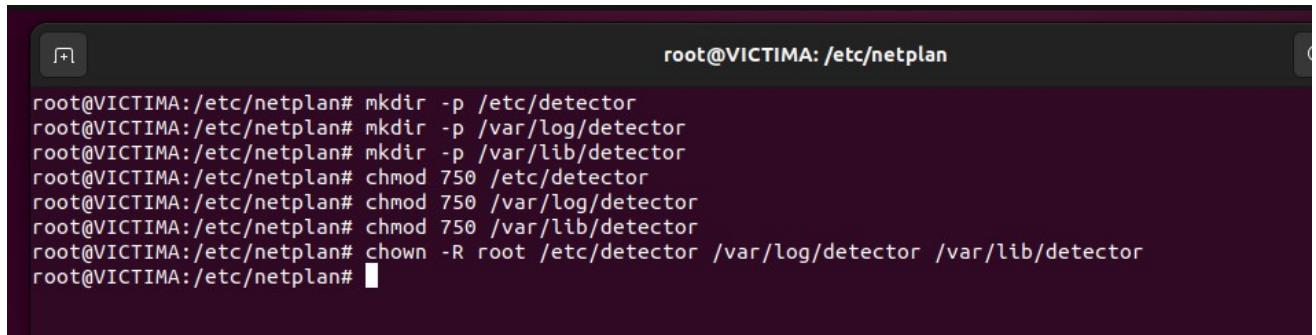
Ya con estas 2 cosas, y la configuración de red correcta, se pueden crear los directorios necesarios para el proyecto, que serían:

- El directorio para el propio detector.
- El directorio de /var/log
- El directorio de /var/lib

```
Terminal 6 de nov 11:23
root@VICTIMA: /etc/netplan
root@VICTIMA:/etc/netplan# mkdir -p /etc/detector
root@VICTIMA:/etc/netplan# mkdir -p /var/log/detector
root@VICTIMA:/etc/netplan# mkdir -p /var/lib/detector
root@VICTIMA:/etc/netplan#
```

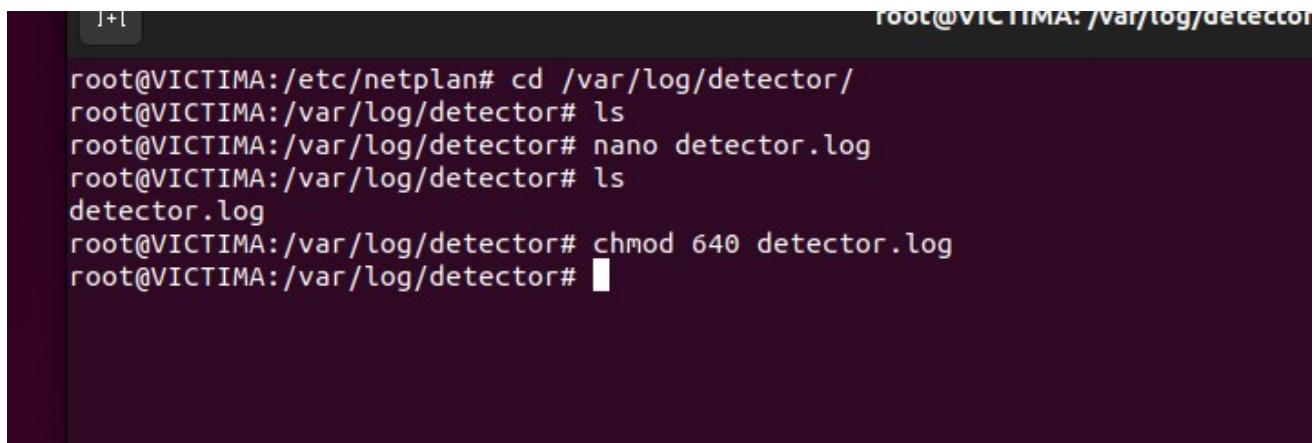


Para continuar, se configuran los permisos de lectura y escritura para los 3 directorios y como propietario estará root:



```
root@VICTIMA:/etc/netplan# mkdir -p /etc/detector
root@VICTIMA:/etc/netplan# mkdir -p /var/log/detector
root@VICTIMA:/etc/netplan# mkdir -p /var/lib/detector
root@VICTIMA:/etc/netplan# chmod 750 /etc/detector
root@VICTIMA:/etc/netplan# chmod 750 /var/log/detector
root@VICTIMA:/etc/netplan# chmod 750 /var/lib/detector
root@VICTIMA:/etc/netplan# chown -R root /etc/detector /var/log/detector /var/lib/detector
root@VICTIMA:/etc/netplan#
```

Ahora, se crea un fichero de auditoria (monitorización) dentro del detector, en /var/log el cual se llamará “detector.log”, a este fichero se le darán permisos de lectura y escritura para el propietario, pero las demás personas no tendrán permisos:



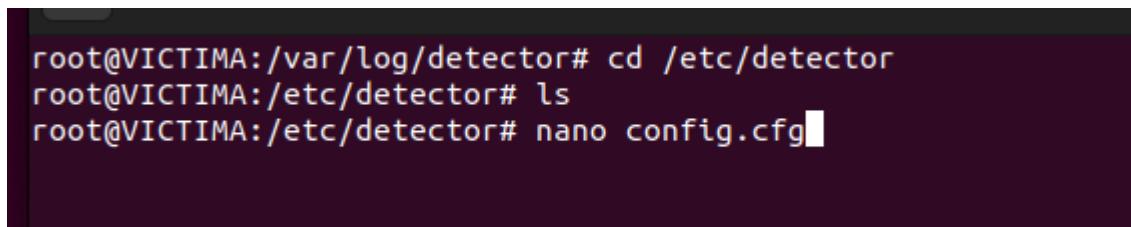
```
root@VICTIMA:/etc/netplan# cd /var/log/detector/
root@VICTIMA:/var/log/detector# ls
root@VICTIMA:/var/log/detector# nano detector.log
root@VICTIMA:/var/log/detector# ls
detector.log
root@VICTIMA:/var/log/detector# chmod 640 detector.log
root@VICTIMA:/var/log/detector#
```



Posteriormente se creará el archivo de configuración, dentro del directorio previamente creado “/etc/detector”.

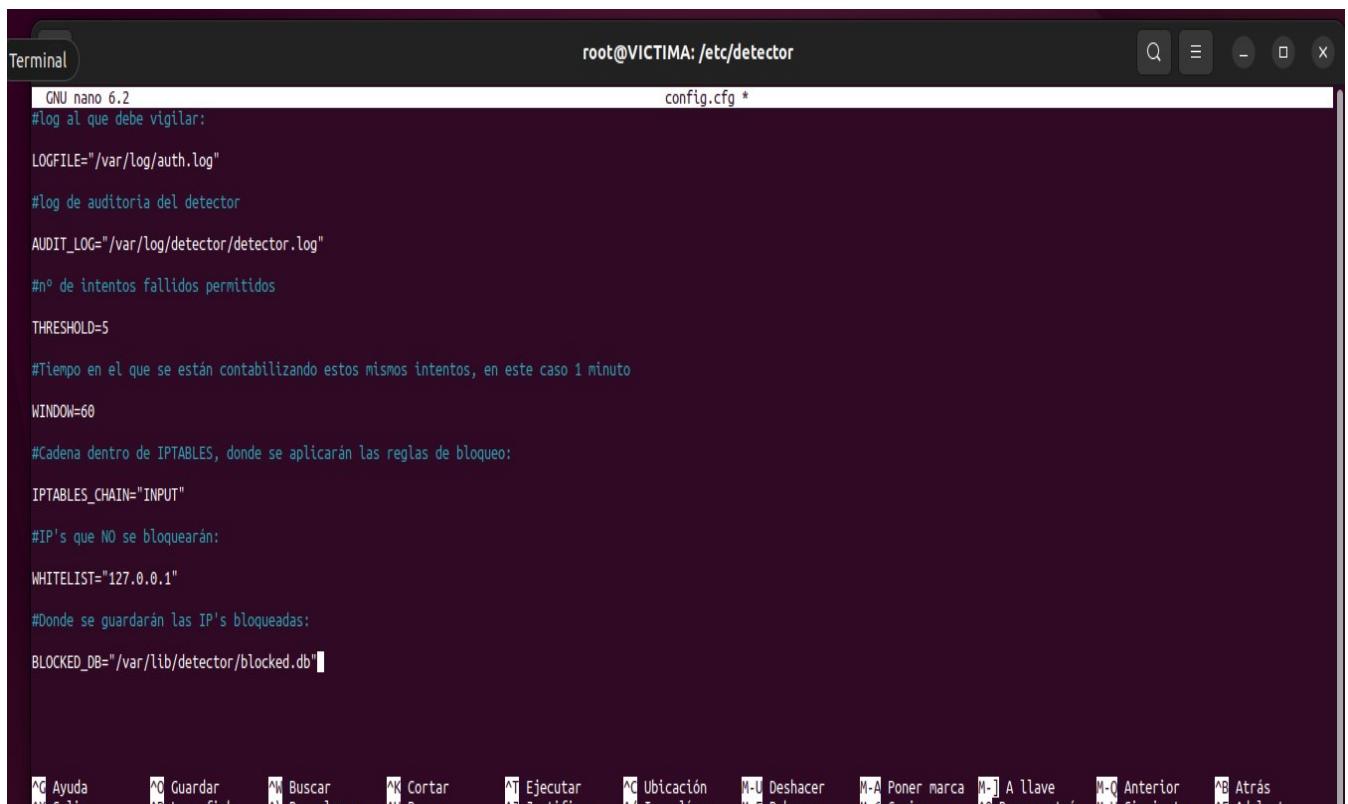
Este archivo de configuración es, básicamente, donde se guardarán todos los ajustes y el script principal, es decir:

- Qué log tiene que mirar dentro de /var/log
- Dónde guardar los registros del detector
- Nº de intentos fallidos, IP's que deben bloquearse, IP's que NO deben bloquearse, etc...



```
root@VICTIMA:/var/log/detector# cd /etc/detector
root@VICTIMA:/etc/detector# ls
root@VICTIMA:/etc/detector# nano config.cfg
```

Se configurarán los filtros que se mencionaban anteriormente y se comentará todo para verlo más claro:



```
GNU nano 6.2
#log al que debe vigilar:
LOGFILE="/var/log/auth.log"
#log de auditoria del detector
AUDIT_LOG="/var/log/detector/detector.log"
#nº de intentos fallidos permitidos
THRESHOLD=5
#Tiempo en el que se están contabilizando estos mismos intentos, en este caso 1 minuto
WINDOW=60
#Cadena dentro de IPTABLES, donde se aplicarán las reglas de bloqueo:
IPTABLES_CHAIN="INPUT"
#IP's que NO se bloquearán:
WHITELIST="127.0.0.1"
#Donde se guardarán las IP's bloqueadas:
BLOCKED_DB="/var/lib/detector/blocked.db"
```



Se guarda con “Ctrl + o” y ya podemos cerrar.

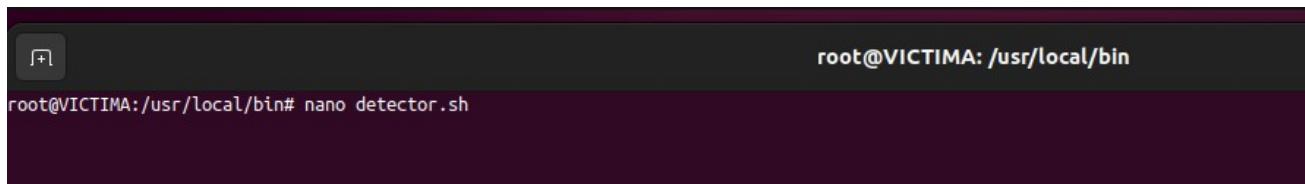
Este archivo de configuración, tendrá los permisos correspondientes para que solo root pueda escribir, pero los usuarios administradores puedan leer:

```
root@VICTIMA:/etc/detector# nano config.cfg
root@VICTIMA:/etc/detector# chown root:adm /etc/detector/config.cfg
root@VICTIMA:/etc/detector# chmod 640 /etc/detector/config.cfg
root@VICTIMA:/etc/detector#
```

Ahora, se comenzará con el script principal, el cual se explicará paso a paso y estará comentado tal y como se ha hecho con el archivo de configuración “config.cfg”:

Se creará el detector en sí, creando el fichero “detector.sh” en la ruta /usr/local/bin.

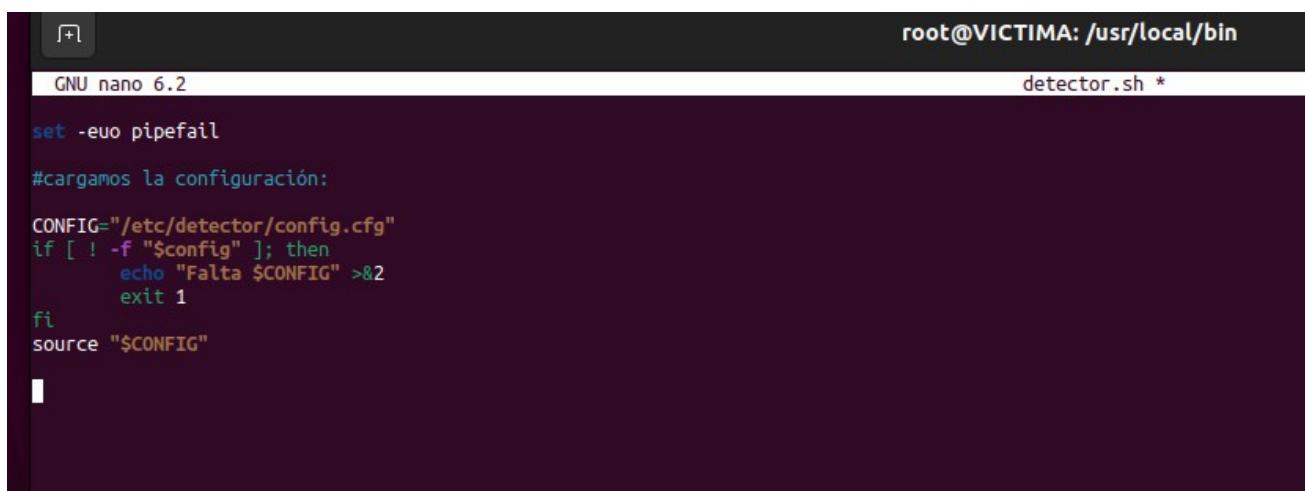
Se crea en esta ruta, porque son ficheros ejecutables normalmente instalados por el administrador de manera local, y así no se mezclan los ejecutables con los ficheros y se evitan posibles errores.



```
root@VICTIMA: /usr/local/bin
root@VICTIMA:/usr/local/bin# nano detector.sh
```

El primer paso del script es cargar la configuración, aquí lo que se hace es indicar qué configuración tiene que usar el script, y que si no existiera ese archivo de configuración, no siga ejecutado el script, porque no tendría sentido. Antes de ello, se pone “set -euo pipefail” para que si hay algún error en el script entero, no continue

ejecutándose.



```
GNU nano 6.2
root@VICTIMA: /usr/local/bin
detector.sh *
set -euo pipefail
#cargamos la configuración:
CONFIG="/etc/detector/config.cfg"
if [ ! -f "$CONFIG" ]; then
    echo "Falta $CONFIG" >&2
    exit 1
fi
source "$CONFIG"
```



Lo siguiente que se hará es crear los directorios y ficheros, en este caso se han creado anteriormente, pero está preparado para que cuando un usuario necesite el detector, se facilite su instalación creando estos automáticamente y no tenga que crear nada ni tener conocimientos sobre ello:

```
#Creación de directorios y ficheros

mkdir -p "$(dirname "$AUDIT_LOG")" "$(dirname "$BLOCKED_DB")" 2>/dev/null || true
touch "$AUDIT_LOG" "$BLOCKED_DB"
chown root:adm "$AUDIT_LOG" 2>/dev/null || true
chmod 640 "$AUDIT_LOG" 2>/dev/null || true

timestamp(){ date +"%Y-%m-%d %H:%M:%s"; }
log(){ echo "[${timestamp}] $*" >> "$AUDIT_LOG"; }
```

El siguiente paso, será convertir la variable WHITELIST (que utilizamos para tener una serie de IP'S que NUNCA deben ser bloqueadas, como la del host, por ejemplo) a un array llamado WL para comprobar si la IP está en esa lista.

Se observa que actúa de tal manera que al estar el “return 0” es verdadero, es decir que la IP está en la whitelist, y “return 1” significa que es falso y que no lo está.

```
#Convertir la variable WHITELIST a un array llamado WL

read -ra WL <<< "$WHITELIST"

is_whitelisted(){
    local ip="$1"
    for w in "${WL[@]}"; do
        [ "$ip" = "$w" ] && return 0
    done
    return 1
}
```



Lo siguiente que se necesita es un conteo de intentos de inicio de sesión, lo cual se hará con ATT, que es un array que guarda para cada IP una cadena con su “timestamp”, es decir, lo cuenta por tiempo.

Lo que se ve es que primero se declaran las variables locales a la función con “cutoff”, luego con la variable “new” se tiene la nueva lista según el tiempo (timestamp) para ver los intentos de sesión de cada IP, y se irán recorriendo las IP'S de esa variable para ver los intentos de sesión correspondientes.

```
# CONTEO

declare -A ATT

cleanup(){

    local now cutoff ip new
    now=$(date + %s)
    cutoff=$((now - WINDOW))
    for ip in "${!ATT[@]}"; do
        new=""
        IFS=',' read -ra T <<< "${ATT[$ip]}"
        for t in "${T[@]}"; do
            [ -n "$t" ] && [ "$t" -ge "$cutoff" ] && new+="$t,"
        done
        ATT[$ip]="${new%,}"
        [ -z "${ATT[$ip]}" ] && unset ATT[$ip]
    done
}

[

^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación
^X Salir      ^R Leer fich.   ^\ Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a carpeta
```



El siguiente paso será una de las cosas más importantes, configurar el bloqueo de las IP's siguiendo los filtros.

Para esto se definirá la función `block_ip` que la bloqueará utilizando `IPTABLES` como se mencionaba anteriormente.

Se llamará a la función “`is_whitelisted`” y, si está en la whitelist, se ignorará, pero si no, se bloqueará automáticamente siguiendo de manera precisa las reglas de `iptables` que anteriormente se configuraron:

Primero se comprobará si ya estaba bloqueada, para evitar duplicados, y si no, la bloqueará guardándola en el fichero “`blocked_db`”.

```
# bloqueo de las IP's

block_ip(){
    local ip="$1"

    is_whitelisted "$ip" && {log "IGNORADA $ip"; return; }

    if iptables -C "$IPTABLES_CHAIN" -s "$ip" -j DROP &>/dev/null; then
        log "YA_BLOQUEADA $ip"
        return
    fi

    if iptables -I "$IPTABLES_CHAIN" -s "$ip" -j DROP &>/dev/null; then
        log "BLOQUEADA $ip"
        echo "$(date +%s),$ip" >> "$BLOCKED_DB"
    else
        log "ERROR_BLOQUEO $ip"
    fi
}
```

[73 líneas escritas]



Lo siguiente será extraer la IP para poder tenerla registrada cuando salte cualquier mensaje.

```
extract_ip(){
    local line="$1"
    echo "$line" | grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" | tail -n1 || true
}
```

Lo siguiente será la estructura principal, es decir, el sistema que comprueba los patrones de inicio de sesión, extrae la IP, comprueba si está en la whitelist o no, utiliza el contador, y siguiendo los filtros, bloquea la IP en caso de que sea necesario y vuelve a borrar el contador, para seguir monitorizando el servidor a tiempo real:

```
# Estructura principal
process_line(){
    local L="$1"
    if echo "$L" | egrep -i "Failed password|Invalid user|authentication failure" >/dev/null 2>&1; then
        ip=$(extract_ip "$L" || true)
        [ -z "$ip" ] && { log "PARSE_NO_IP" | $L"; return; }
        is_whitelisted "$ip" && { log "IGNORADA_WHITELIST" $ip | $L"; return; }
        now=$(date +%s)
        if [ -z "${ATT[$ip]+_}" ]; then
            ATT[$ip]="$now"
        else
            ATT[$ip]="${ATT[$ip]},$now"
        fi
        cleanup
        IFS=' '
        read -ra A <<< "${ATT[$ip]}"
        c=${#A[@]}
        log "IP $ip intentos=$c"
        [ "$c" -ge "$THRESHOLD" ] && { log "UMBRAL_SUPERADO $ip"; block_ip "$ip"; unset ATT[$ip]; }
    fi
}
log "INICIANDO detector (LOG=$LOGFILE TH=$THRESHOLD W=$WINDOW)"
[ -f "$LOGFILE" ] || { log "ERROR: LOGFILE no existe: $LOGFILE"; exit 1; }
```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer M-A Poner marca M-



Por último, el siguiente paso será hacer que este script funcione a tiempo real.

Para esto, se realizará un bucle por el cual se llamará continuamente a la función “process_line” que es la estructura principal del paso anterior, a través de “tail” que es un comando que se utiliza para leer los ficheros, y sobre todo la última línea.

Por tanto, cada vez que tail lee una línea nueva, se llama a la función “process_line” y el script empieza a funcionar:

```
# Script a tiempo real

tail -n0 -F "$LOGFILE" 2>/dev/null | while IFS= read -r line; do
    process_line "$line" || true
done

[ 111 líneas escritas ]

^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^] Justificar   ^/ Ir a línea
```



Así pues, el código quedaría de esta manera:

```
root@VICTIMA:/usr/local/bin# nano 6.2
GNU nano 6.2
set -euo pipefail
#cargamos la configuración:
CONFIG="/etc/detector/config.cfg"
if [ ! -f "$config" ]; then
    echo "Falta $CONFIG" >&2
    exit 1
fi
source "$CONFIG"
#Creación de directorios y ficheros
mkdir -p "$(dirname "$AUDIT_LOG")" "$dirname "$BLOCKED_DB"" 2>/dev/null || true
touch "$AUDIT_LOG" "$BLOCKED_DB"
chown root:adm "$AUDIT_LOG" 2>/dev/null || true
chmod 640 "$AUDIT_LOG" 2>/dev/null || true
timestamp(){ date +"%Y-%m-%d %H:%M:%S"; }
log(){ echo "[${timestamp}] ${*}" >> "$AUDIT_LOG"; }

#Convertir la variable WHITELIST a un array llamado WL
read -ra WL <<< "$WHITELIST"
is_whitelisted(){
    local ip=$1
    for w in "${WL[@]}"; do
        [ "$ip" = "$w" ] && return 0
    done
    return 1
}
# CONTEO
declare -A ATT
cleanup(){
    local now cutoff ip new
    now=$(date + %s)
    cutoff=$((now - WINDOW))
    for ip in "${!ATT[@]}"; do
        new=""
        IFS=',' read -ra T <<< "${ATT[$ip]}"
        for t in "${T[@]}"; do
            [ -n "$t" ] && [ "$t" -ge "$cutoff" ] && new+="$t,"
        done
        ATT[$ip]="${new%,}"
        [ -z "${ATT[$ip]}" ] && unset ATT[$ip]
    done
}

root@VICTIMA:/usr/local/bin#
```



Administración de Sistemas Informáticos en Red

```
# bloqueo de las IP's
block_ip(){
    local ip="$1"
    if_is_whitelisted "$ip" && { log "IGNORADA $ip"; return; }
    if iptables -C "$IPTABLES_CHAIN" -s "$ip" -j DROP &>/dev/null; then
        log "YA_BLOQUEADA $ip"
        return
    fi
    if iptables -I "$IPTABLES_CHAIN" -s "$ip" -j DROP &>/dev/null; then
        log "BLOQUEADA $ip"
        echo "$(date +%s),$ip" >> "$BLOCKED_DB"
    else
        log "ERROR_BLOQUEO $ip"
    fi
}

# Extracción de IP para que aparezca por pantalla.
extract_ip(){
    local line="$1"
    echo "$line" | grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" | tail -n1 || true
}

# Estructura principal
process_line(){
    local L="$1"
    if echo "$L" | egrep -i "Failed password|Invalid user|authentication failure" >/dev/null 2>&1; then
        ip=$(extract_ip "$L" || true)
        [ -z "$ip" ] && { log "PARSE_NO_IP | $L"; return; }
        is_whitelisted "$ip" && { log "IGNORADA_WHITELIST $ip | $L"; return; }
        now=$(date +%s)
        if [ -z "${ATT[$ip]+_}" ]; then
            ATT[$ip]="$now"
        else
            ATT[$ip]="${ATT[$ip]},$now"
        fi
        cleanup
        IFS=','
        read -ra A <<< "${ATT[$ip]}"
        c=${#A[@]}
        log "IP $ip intentos=$c"
        [ "$c" -ge "$THRESHOLD" ] && { log "UMbral_SUPERADO $ip"; block_ip "$ip"; unset ATT[$ip]; }
    fi
}
log "INICIANDO detector (LOG=$LOGFILE TH=$THRESHOLD W=$WINDOW)"
[ -f "$LOGFILE" ] || { log "ERROR: LOGFILE no existe: $LOGFILE"; exit 1; }

# Script a tiempo real
tail -n0 -F "$LOGFILE" 2>/dev/null | while IFS= read -r line; do
    process_line "$line" || true
done
```



8. Pruebas

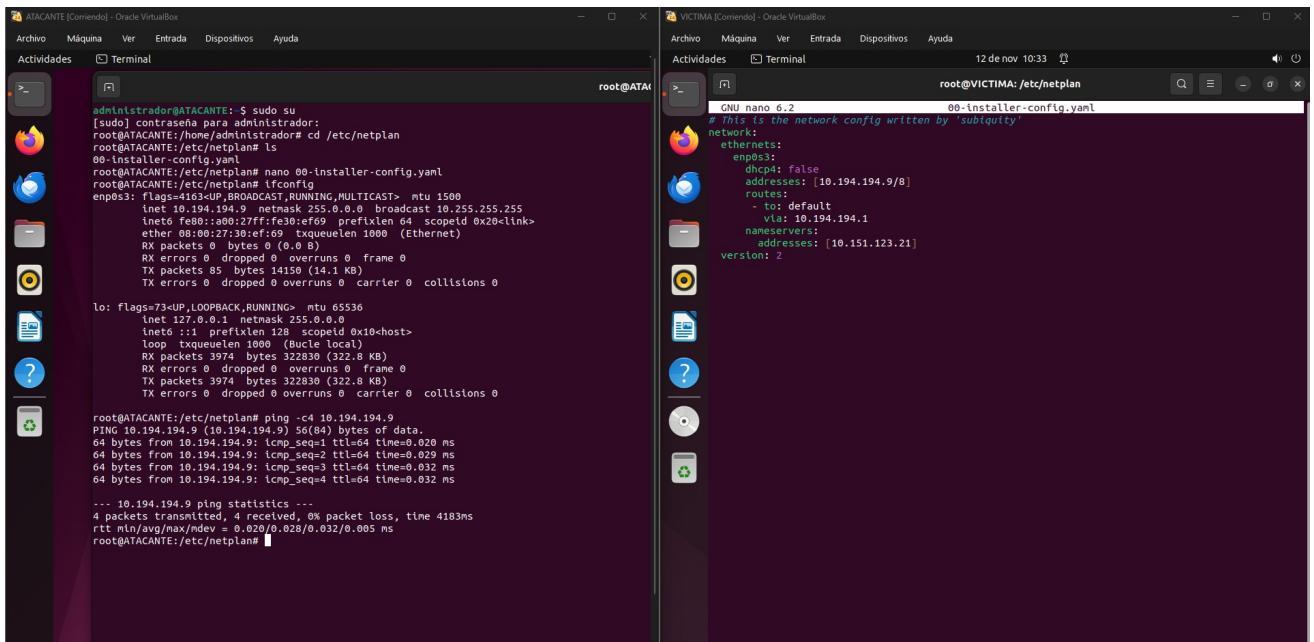
Para el primer caso de pruebas, primero se configurará la máquina atacante, de tal manera que ambas máquinas se detecten entre si:

```
root@ATACANTE: /etc/netplan
root@ATACANTE: /etc/netplan
administrador@ATACANTE: $ sudo su
[sudo] contraseña para administrador:
root@ATACANTE:/home/administrador# cd /etc/netplan
root@ATACANTE:/etc/netplan# ls
00-installer-config.yaml
root@ATACANTE:/etc/netplan# nano 00-installer-config.yaml
```

```
root@ATACANTE: /etc/netplan
root@ATACANTE: /etc/netplan
GNU nano 6.2
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [10.194.194.10/8]
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [10.151.123.21]
version: 2
```



Se puede ver como ya la atacante, es capaz de detectar a la víctima en la red:



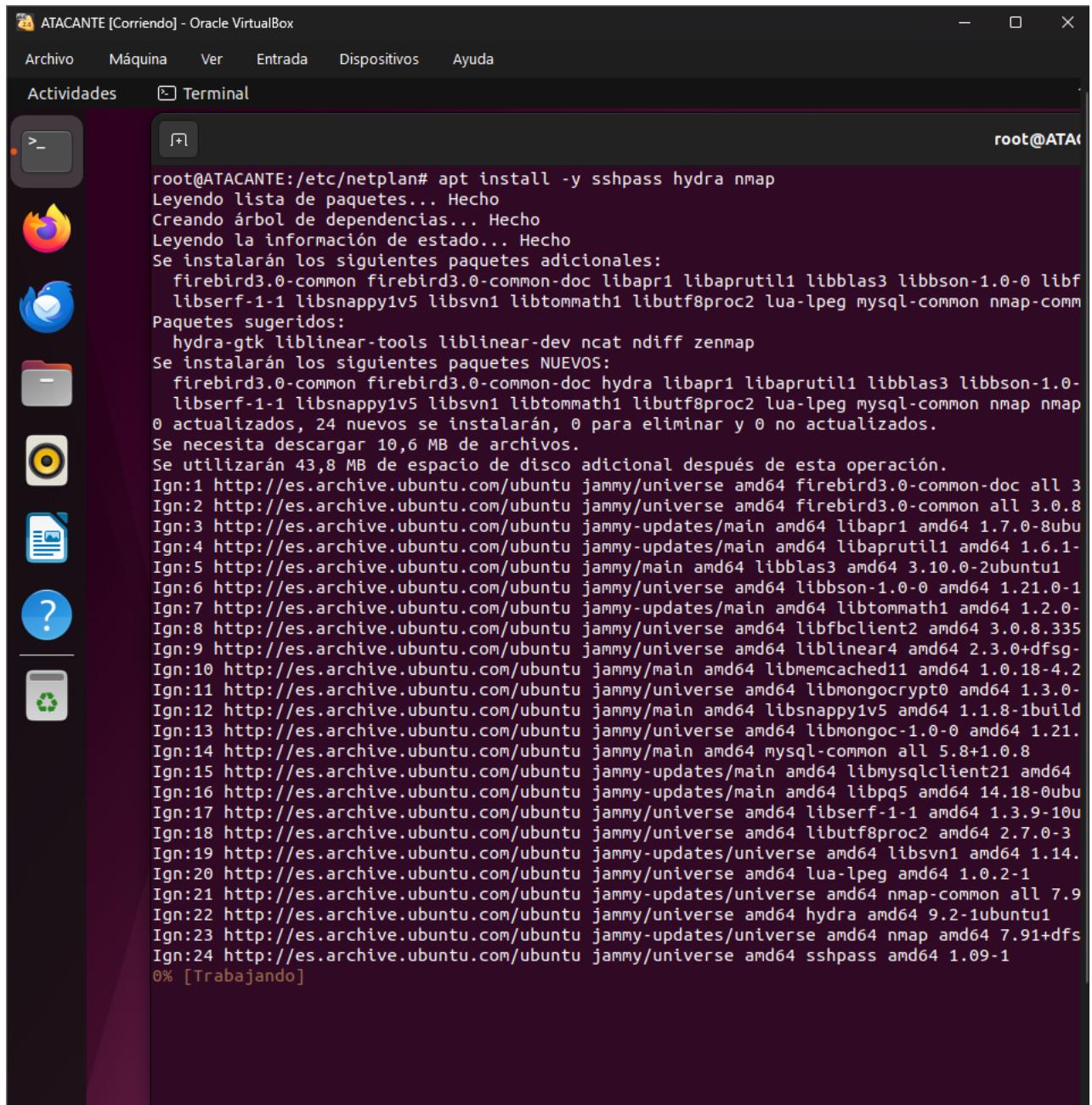
Ya configurado, se instalan en la máquina atacante 3 herramientas para las pruebas. Estas serán:

-Sshpass: sirve para hacer búcles con contraseñas erróneas.

-Hydra: para intentos de inicios de sesión.

-CrackMapExec: Framework de pentesting que también se usa en SSH.





The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "root@ATACANTE [Corriendo] - Oracle VirtualBox". The terminal content is as follows:

```
root@ATACANTE:/etc/netplan# apt install -y sshpass hydra nmap
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  firebird3.0-common firebird3.0-common-doc libapr1 libaprutil1 libblas3 libbson-1.0-0 libfb
  libserf-1-1 libsnappy1v5 libsvn1 libtommath1 libutf8proc2 lua-lpeg mysql-common nmap-comm
Paquetes sugeridos:
  hydra-gtk liblinear-tools liblinear-dev ncat ndiff zenmap
Se instalarán los siguientes paquetes NUEVOS:
  firebird3.0-common firebird3.0-common-doc hydra libapr1 libaprutil1 libblas3 libbson-1.0-
  libserf-1-1 libsnappy1v5 libsvn1 libtommath1 libutf8proc2 lua-lpeg mysql-common nmap nmap
0 actualizados, 24 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 10,6 MB de archivos.
Se utilizarán 43,8 MB de espacio de disco adicional después de esta operación.
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common-doc all 3
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common all 3.0.8
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubu
Ign:4 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-
Ign:5 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 libblas3 amd64 3.10.0-2ubuntu1
Ign:6 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libbson-1.0-0 amd64 1.21.0-1
Ign:7 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtommath1 amd64 1.2.0-
Ign:8 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libfbclient2 amd64 3.0.8.335
Ign:9 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 liblinear4 amd64 2.3.0+dfsg-
Ign:10 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 libmemcached11 amd64 1.0.18-4.2
Ign:11 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libmongocrypt0 amd64 1.3.0-
Ign:12 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 libsnappy1v5 amd64 1.1.8-1build
Ign:13 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libmongoc-1.0-0 amd64 1.21.
Ign:14 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8
Ign:15 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libmysqlclient21 amd64
Ign:16 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpq5 amd64 14.18-0ubu
Ign:17 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libserf-1-1 amd64 1.3.9-10u
Ign:18 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 libutf8proc2 amd64 2.7.0-3
Ign:19 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libsvn1 amd64 1.14.
Ign:20 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 lua-lpeg amd64 1.0.2-1
Ign:21 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap-common all 7.9
Ign:22 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 hydra amd64 9.2-1ubuntu1
Ign:23 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap amd64 7.91+dfs
Ign:24 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 sshpass amd64 1.09-1
0% [Trabajando]
```

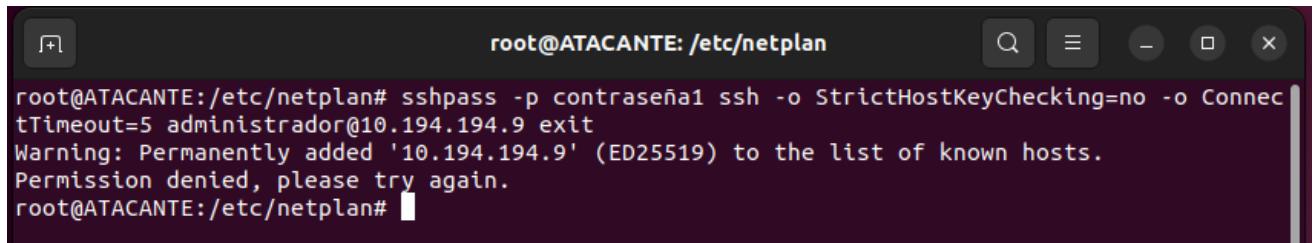
8.1 Prueba con SSHPASS

Se empieza con la primera prueba, en la cual se utilizará SSHPASS con el siguiente comando:

```
"sshpss -p contraseña1 ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 administrador@10.194.194.9  
exit"
```

Esto lo que hace es un primer inicio de sesión, para comprobar si el detector funciona. Se establece como contraseña en este caso “contraseña1” pero puede ser cualquiera, mientras que no sea la correcta.

Aparecería algo así:

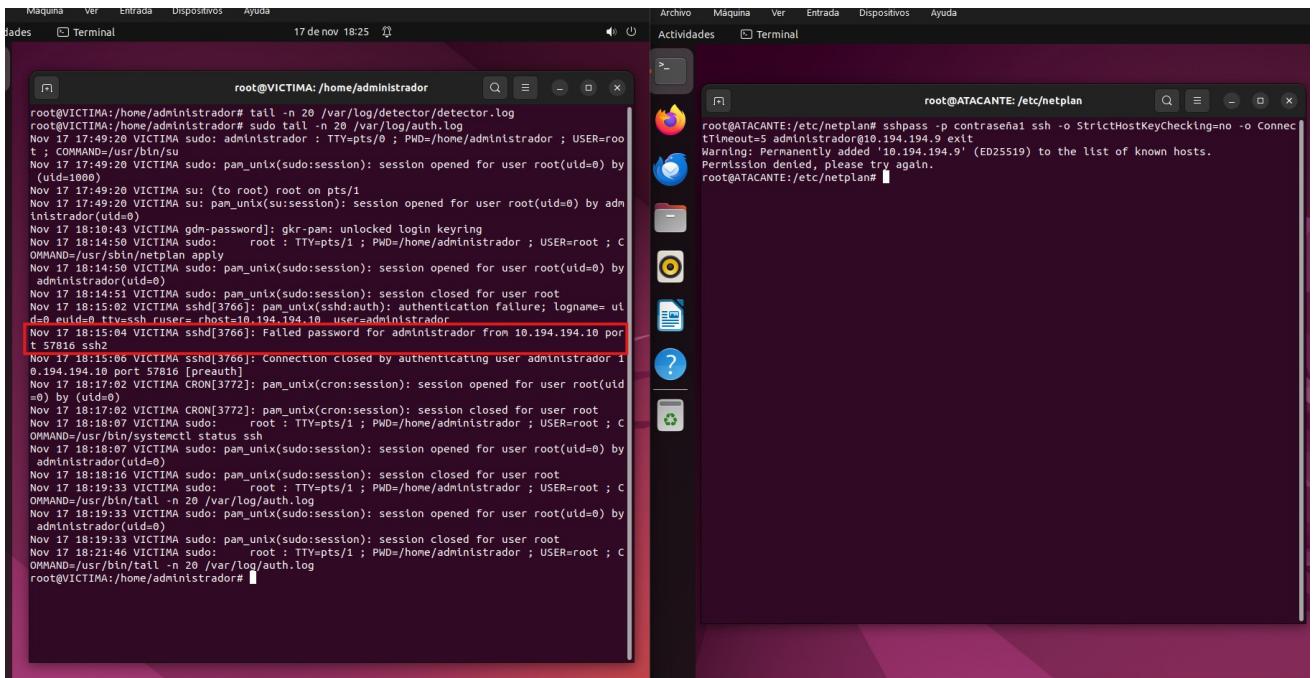


The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "root@ATACANTE: /etc/netplan". The command entered was "sshpss -p contraseña1 ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 administrador@10.194.194.9 exit". The output shows a warning about adding the host to the known hosts list and a permission denied message. The terminal prompt "root@ATACANTE:/etc/netplan#" is visible at the bottom.

```
root@ATACANTE:/etc/netplan# sshpass -p contraseña1 ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 administrador@10.194.194.9 exit  
Warning: Permanently added '10.194.194.9' (ED25519) to the list of known hosts.  
Permission denied, please try again.  
root@ATACANTE:/etc/netplan#
```

Administración de Sistemas Informáticos en Red

Así que ahora, en la víctima, se comprobará si el detector que hemos creado, está funcionando correctamente. Esto se hará con “tail -n 20 /var/log/auth.log” para comprobar si está llegando el ataque a la máquina:



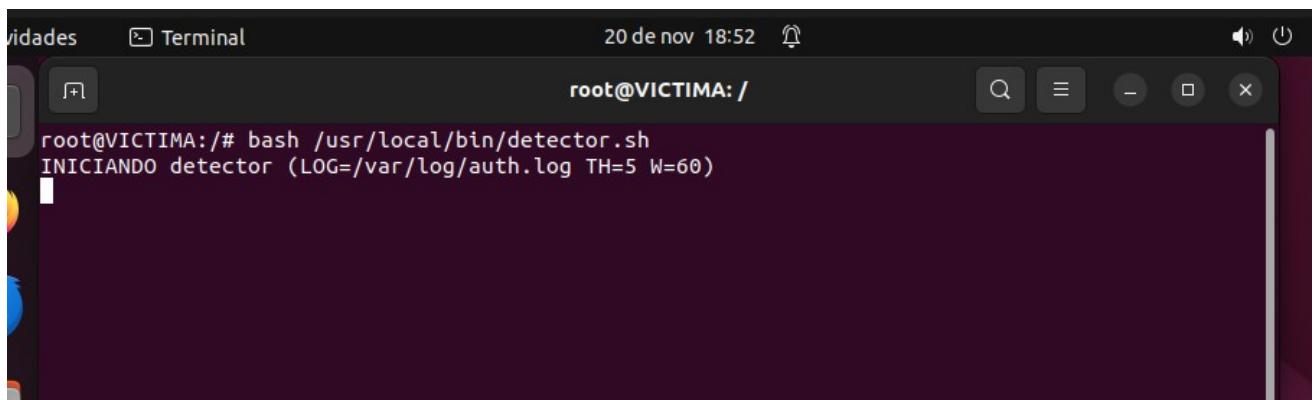
The screenshot shows two terminal windows side-by-side. The left window is titled 'root@VICTIMA:/home/administrador' and displays the contents of the '/var/log/auth.log' file. The right window is titled 'root@ATACANTE:/etc/netplan' and also displays log entries. Both logs show multiple failed password attempts from the IP address 10.194.194.10, indicating a brute-force attack.

```
root@VICTIMA:/home/administrador# tail -n 20 /var/log/auth.log
root@VICTIMA:/home/administrador# sudo tail -n 20 /var/log/auth.log
Nov 17 17:49:20 VICTIMA sudo: administrador : TTY=pts/0 ; PWD=/home/administrador ; USER=root ; COMMAND=/usr/bin/su
Nov 17 17:49:20 VICTIMA sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Nov 17 17:49:20 VICTIMA su: (to root) root on pts/1
Nov 17 17:49:20 VICTIMA su: pam_unix(su:session): session opened for user root(uid=0) by administrador(uid=0)
Nov 17 18:10:43 VICTIMA gdm-password: gkr-pam: unlocked login keyring
Nov 17 18:14:50 VICTIMA sudo:      root : TTY=pts/1 ; PWD=/home/administrador ; USER=root ; COMMAND=/usr/bin/netplan apply
Nov 17 18:14:50 VICTIMA sudo: pam_unix(sudo:session): session opened for user root(uid=0) by administrador(uid=0)
Nov 17 18:14:51 VICTIMA sudo: pam_unix(sudo:session): session closed for user root
Nov 17 18:15:02 VICTIMA sudo: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=sshd user= rhost=10.194.194.10 user=administrador
Nov 17 18:15:04 VICTIMA sshd[3766]: Failed password for administrador from 10.194.194.10 port 57816
Nov 17 18:15:06 VICTIMA sshd[3766]: Connection closed by authenticating user administrador from 10.194.194.10 port 57816 [preauth]
Nov 17 18:17:07 VICTIMA CRON[3772]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Nov 17 18:17:02 VICTIMA CRON[3772]: pam_unix(cron:session): session closed for user root
Nov 17 18:18:07 VICTIMA sudo:      root : TTY=pts/1 ; PWD=/home/administrador ; USER=root ; COMMAND=/usr/bin/systemctl status ssh
Nov 17 18:18:07 VICTIMA sudo: pam_unix(sudo:session): session opened for user root(uid=0) by administrador(uid=0)
Nov 17 18:18:16 VICTIMA sudo: pam_unix(sudo:session): session closed for user root
Nov 17 18:18:33 VICTIMA sudo:      root : TTY=pts/1 ; PWD=/home/administrador ; USER=root ; COMMAND=/usr/bin/tail -n 20 /var/log/auth.log
Nov 17 18:18:33 VICTIMA sudo: pam_unix(sudo:session): session opened for user root(uid=0) by administrador(uid=0)
Nov 17 18:19:33 VICTIMA sudo: pam_unix(sudo:session): session closed for user root
Nov 17 18:21:46 VICTIMA sudo:      root : TTY=pts/1 ; PWD=/home/administrador ; USER=root ; COMMAND=/usr/bin/tail -n 20 /var/log/auth.log
root@VICTIMA:/home/administrador# 
```

```
root@ATACANTE:/etc/netplan
root@ATACANTE:/etc/netplan# sshpass -p contraseña1 ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 administrador@10.194.194.9 exit
Warning: Permanently added '10.194.194.9' (ED25519) to the list of known hosts.
Permission denied, please try again.
root@ATACANTE:/etc/netplan# 
```

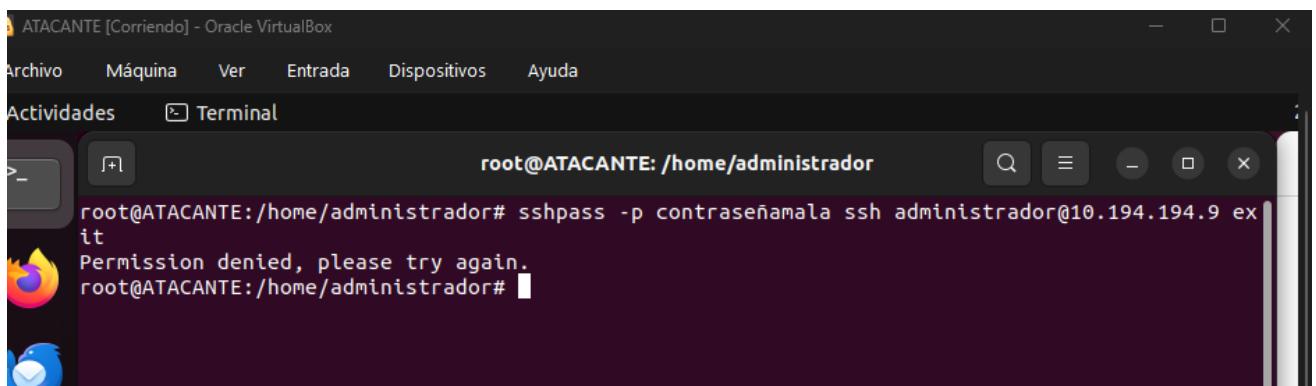


Primero, se iniciará el DETECTOR ejecutándolo con el comando “BASH”:



A screenshot of a terminal window titled "Terminal". The window shows the command "root@VICTIMA:/# bash /usr/local/bin/detector.sh" being run. The output of the command is "INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)". The terminal has a dark background and standard Linux-style icons.

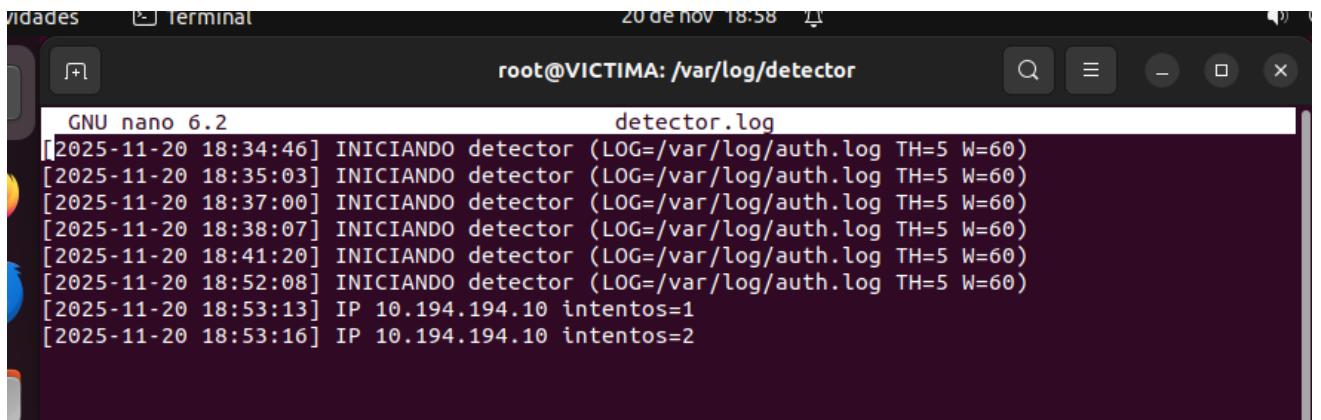
Y se comprobará que funciona correctamente, lanzando un inicio de sesión desde la máquina atacante:



A screenshot of a terminal window titled "Terminal" running inside a virtual machine named "ATACANTE [Corriendo] - Oracle VirtualBox". The window shows the command "root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit". The output shows an "Permission denied, please try again." message. The terminal has a dark background and standard Linux-style icons.



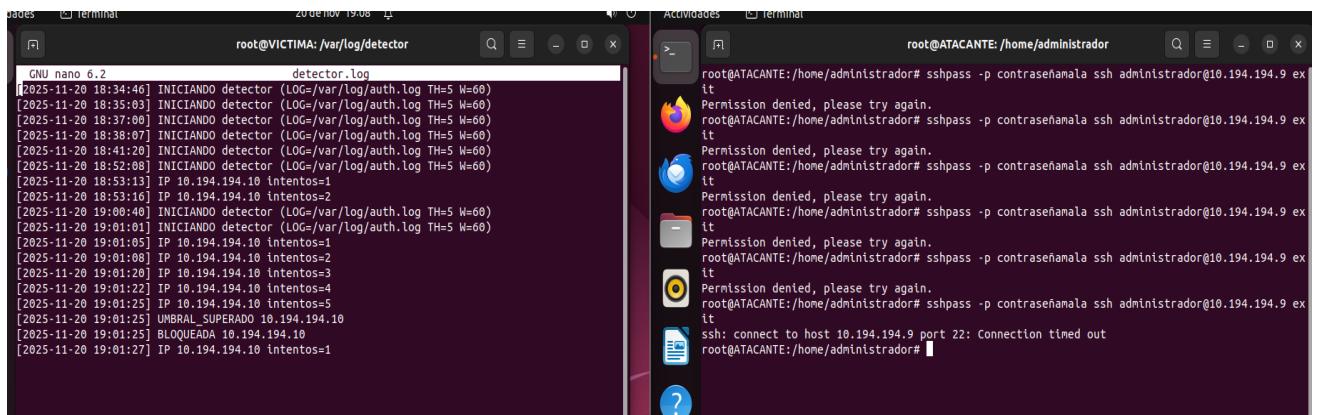
Ahora, se accede al directorio en el cual se guardan los logs a tiempo real que se han creado, y deberían de salir 2 intentos, primero el que se probó anteriormente y ahora este:



```
GNU nano 6.2          detector.log
[2025-11-20 18:34:46] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:35:03] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:37:00] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:38:07] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:41:20] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:52:08] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:53:13] IP 10.194.194.10 intentos=1
[2025-11-20 18:53:16] IP 10.194.194.10 intentos=2
```

Con esto se comprueba que el detector está funcionando a la perfección detectando los intentos a tiempo real.

Y para finalizar esta primera prueba con SSHPASS, si se realizan 6 intentos, se podrá comprobar como en el intento 5, se bloquea automáticamente la IP atacante y aparece de manera clara en el log dentro del directorio:



```
root@VICTIMA:/var/log/detector
GNU nano 6.2          detector.log
[2025-11-20 18:34:46] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:35:03] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:37:00] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:38:07] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:41:20] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:52:08] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:53:13] IP 10.194.194.10 Intentos=1
[2025-11-20 18:53:16] IP 10.194.194.10 intentos=2
[2025-11-20 19:00:40] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 19:01:01] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 19:01:05] IP 10.194.194.10 intentos=1
[2025-11-20 19:01:08] IP 10.194.194.10 intentos=2
[2025-11-20 19:01:20] IP 10.194.194.10 intentos=3
[2025-11-20 19:01:22] IP 10.194.194.10 intentos=4
[2025-11-20 19:01:25] IP 10.194.194.10 intentos=5
[2025-11-20 19:01:25] UMBRAL SUPERADO 10.194.194.10
[2025-11-20 19:01:25] BLOQUEADA 10.194.194.10
[2025-11-20 19:01:27] IP 10.194.194.10 intentos=1

root@ATACANTE:/home/administrador
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
Permission denied, please try again.
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
Permission denied, please try again.
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
Permission denied, please try again.
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
Permission denied, please try again.
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
Permission denied, please try again.
root@ATACANTE:/home/administrador# sshpass -p contraseñamala ssh administrador@10.194.194.9 exit
ssh: connect to host 10.194.194.9 port 22: Connection timed out
root@ATACANTE:/home/administrador#
```



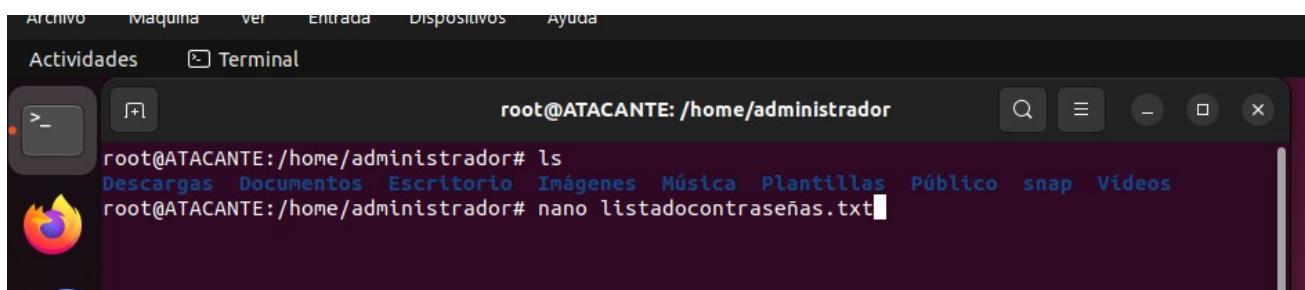
8.2 Segunda prueba con Hydra

La segunda prueba, se realizará con la herramienta HYDRA, para realizar muchos intentos de inicios de sesión en muy poco tiempo:

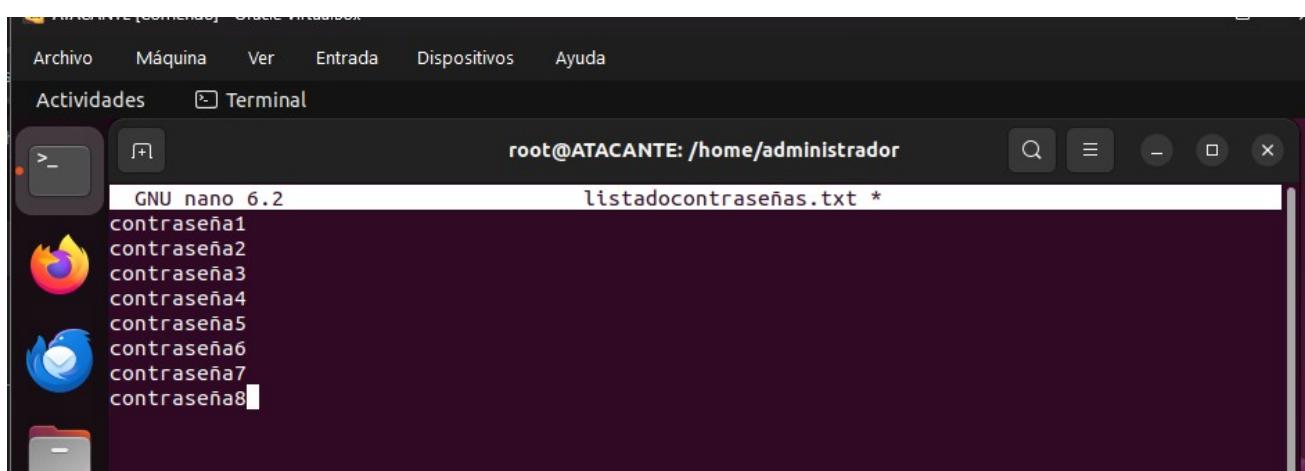
Hydra funciona de la siguiente manera:

Sirve para hacer muchos intentos de sesión prácticamente a la vez, cogiendo las contraseñas desde un fichero de texto que podremos crear nosotros. Se podrían hacer tantos intentos como se quisiera.

En este caso se llamará al fichero por ejemplo, “listadocontraseñas.txt” , y estará compuesto por unas 10 contraseñas, para comprobar que a la sexta vez, se bloquea la IP gracias al detector.



```
root@ATACANTE:/home/administrador# ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público snap Vídeos
root@ATACANTE:/home/administrador# nano listadocontraseñas.txt
```



```
GNU nano 6.2                               listadocontraseñas.txt *
contraseña1
contraseña2
contraseña3
contraseña4
contraseña5
contraseña6
contraseña7
contraseña8
```

Administración de Sistemas Informáticos en Red

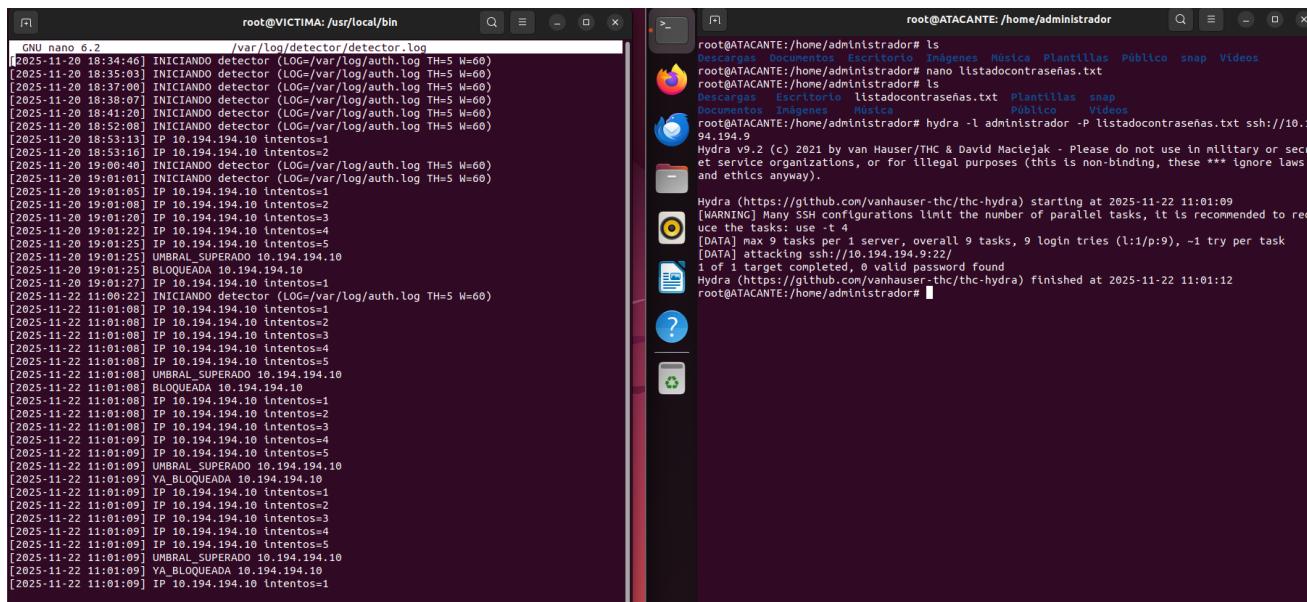
Ahora, ya creado el archivo, tras guardarlo con Ctrl + o como siempre, se ejecutará el comando para ejecutar HYDRA contra la máquina víctima.

```
DOCUMENTOS IMÁGENES MÚSICA PÚBLICO VÍDEOS
root@ATACANTE:/home/administrador# hydra -l administrador -P listadocontraseñas.txt ssh://10.194.194.9
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-22 11:01:09
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:1/p:9), ~1 try per task
[DATA] attacking ssh://10.194.194.9:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-22 11:01:12
root@ATACANTE:/home/administrador#
```

Se puede ver que ha finalizado el ataque correctamente, pero que no se ha encontrado ninguna contraseña válida.

Ahora se accede al log del detector y, efectivamente, cada 5 intentos, al siguiente se supera el umbral y automáticamente se bloquea, asegurando que el detector funciona perfectamente.



```
root@VICTIMA:/usr/local/bin
GNU nano 6.2          /var/log/detector/detector.log
[2025-11-20 18:34:46] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:35:03] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:37:00] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:38:07] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:41:20] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:52:08] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 18:53:13] IP 10.194.194.10 Intentos=1
[2025-11-20 18:53:16] IP 10.194.194.10 Intentos=2
[2025-11-20 18:53:20] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 19:01:01] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-20 19:01:05] IP 10.194.194.10 Intentos=1
[2025-11-20 19:01:08] IP 10.194.194.10 Intentos=2
[2025-11-20 19:01:20] IP 10.194.194.10 Intentos=3
[2025-11-20 19:01:22] IP 10.194.194.10 Intentos=4
[2025-11-20 19:01:25] IP 10.194.194.10 Intentos=5
[2025-11-20 19:01:25] UMBRAL SUPERADO 10.194.194.10
[2025-11-20 19:01:25] BLOQUEADA 10.194.194.10
[2025-11-20 19:01:27] IP 10.194.194.10 Intentos=1
[2025-11-22 11:00:22] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=1
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=2
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=3
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=4
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=5
[2025-11-22 11:01:08] UMBRAL SUPERADO 10.194.194.10
[2025-11-22 11:01:08] BLOQUEADA 10.194.194.10
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=1
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=2
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=3
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=4
[2025-11-22 11:01:08] IP 10.194.194.10 Intentos=5
[2025-11-22 11:01:09] YA BLOQUEADA 10.194.194.10
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=1
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=2
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=3
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=4
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=5
[2025-11-22 11:01:09] UMBRAL SUPERADO 10.194.194.10
[2025-11-22 11:01:09] YA BLOQUEADA 10.194.194.10
[2025-11-22 11:01:09] IP 10.194.194.10 Intentos=1
```

```
root@ATACANTE:/home/administrador#
root@ATACANTE:/home/administrador# ls
Descargas Documentos Escritorio Imágenes Plantillas Públco snap Videos
root@ATACANTE:/home/administrador# nano listadocontraseñas.txt
root@ATACANTE:/home/administrador# ls
Descargas Escritorio listadocontraseñas.txt Plantillas snap
Documentos Imágenes Música Públco Videos
root@ATACANTE:/home/administrador# hydra -l administrador -P listadocontraseñas.txt ssh://10.194.194.9
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

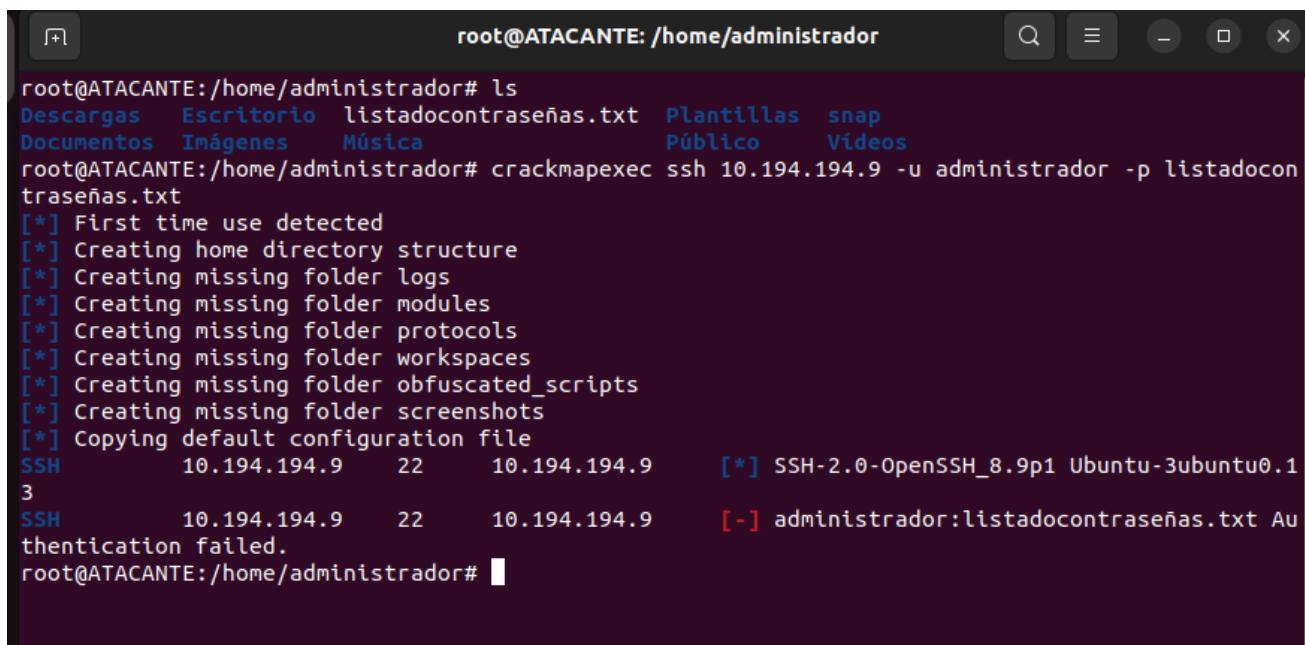
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-22 11:01:09
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:1/p:9), ~1 try per task
[DATA] attacking ssh://10.194.194.9:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-22 11:01:12
root@ATACANTE:/home/administrador#
```



8.3 Tercera prueba con CrackMapExec

Para finalizar las pruebas, se hará una más con CrackMapExec, haciendo un ataque por fuerza bruta.

Para comenzar, se crearía un fichero con contraseñas, pero en este caso se puede reutilizar el de la prueba anterior “listadocontraseñas.txt”.



```
root@ATACANTE:/home/administrador# ls
Descargas Escritorio listadocontraseñas.txt Plantillas snap
Documentos Imágenes Música Público Videos
root@ATACANTE:/home/administrador# crackmapexec ssh 10.194.194.9 -u administrador -p listadocontraseñas.txt
[*] First time use detected
[*] Creating home directory structure
[*] Creating missing folder logs
[*] Creating missing folder modules
[*] Creating missing folder protocols
[*] Creating missing folder workspaces
[*] Creating missing folder obfuscated_scripts
[*] Creating missing folder screenshots
[*] Copying default configuration file
SSH      10.194.194.9    22    10.194.194.9      [*] SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.1
3
SSH      10.194.194.9    22    10.194.194.9      [-] administrador:listadocontraseñas.txt Authentication failed.
root@ATACANTE:/home/administrador#
```



Se pone como método de autenticación el usuario administrador y el listado de contraseñas, por lo que se observará el log:

```
[2025-11-22 14:43:16] INICIANDO detector (LOG=/var/log/auth.log TH=5 W=60)
[2025-11-22 14:44:04] IP 10.194.194.10 intentos=1
[2025-11-22 14:44:06] IP 10.194.194.10 intentos=2
[2025-11-22 14:44:09] IP 10.194.194.10 intentos=3
[2025-11-22 14:44:11] IP 10.194.194.10 intentos=4
[2025-11-22 14:44:13] IP 10.194.194.10 intentos=5
[2025-11-22 14:44:13] UMBRAL_SUPERADO 10.194.194.10
[2025-11-22 14:44:13] BLOQUEADA 10.194.194.10
[2025-11-22 14:44:15] IP 10.194.194.10 intentos=1

^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar  ^/ Ir a línea
```

Y efectivamente, se observa que, indiferentemente de la herramienta, el script no rompe y bloquea las IP's continuamente cuando es necesario.



9. Objetivos a conseguir

Objetivos más técnicos:

- Reducir lo máximo posible los intentos de ataques reales a los servidores.
- Asegurarnos de que el sistema no de errores ni rompa cuando haya mucho volumen.
- Integrar el sistema dentro de los centros educativos de manera que los alumnos puedan tener una mayor formación de cara a su carrera profesional.

Objetivos medibles:

- Conseguir integrar el proyecto en los 18 centros educativos en Salamanca en el primer año.
- Ampliar a toda Castilla y León, integrando el proyecto en los 42 institutos que imparten F.P en toda Castilla y León, en los siguientes 4 años.
- Realizar cursos de formación a 100 estudiantes durante el primer año que el proyecto salga a la luz.
- Elaborar guías y procedimientos para que, tras haber dado el curso, los centros que integren el sistema puedan formar a los alumnos de manera más profesional.



10. Previsión de los recursos materiales y humanos necesarios

-Recursos materiales:

- Infraestructura hardware:

- Servidor Linux dedicado
- Firewall corporativo
- Sistema de copias de seguridad

- Infraestructura software:

- S.O Linux (Ubuntu Server/Debian/Kali)
- Sistema de repositorios internos (GitHub)
- Red corporativa.

-Recursos humanos:

- Administrador de sistemas (configuración de servidor Linux, SSH y logs).
- Analista de ciberseguridad (Analizar los ataques, configurar umbrales de bloqueo y comprobar si es un falso positivo o se trata de un ataque real).
- Técnico de soporte (Copias de seguridad, documentación...)
- Técnico de comunicaciones (Gestionar la red corporativa)



11. Presupuesto económico.

Se divide como en el anterior apartado, este sería su presupuesto anual:

- Infraestructura hardware:
 - Servidor Linux dedicado - 1200€
 - Firewall corporativo - 900€
 - Sistema de copias de seguridad (NAS) - 500€
- Infraestructura software:
 - S.O Linux - 0€ (software libre)
 - Sistema de repositorios internos - 300€ (GitHub enterprise)
 - Red corporativa. - 1000€
- Recursos humanos:
 - Administrador de sistemas - 30000 €
 - Analista de ciberseguridad - 35000 €
 - Técnico de soporte - 23000 €
 - Técnico de comunicaciones - 28000€



12. Fuentes

A continuación, se indican los sitios consultados (webs, youtube) para la realización del Proyecto presentado:

<https://www.youtube.com/watch?v=aXON7IC7wxw>

PDF – Unidad 7. Bash (Dado en ISO)

PDF – Prácticas de SAD + temario

<https://datascientest.com/es/fuerza-bruta-hydra>

<https://karpoke.ignaciocano.com/2013/06/09/conectarse-por-ssh-utilizando-sshpass/>

<https://thehackerway.es/2023/11/29/pentesting-sobre-active-directory-con-crackmapexec/>

https://www.linkedin.com/posts/alejandro-fern%C3%A1ndez-gonz%C3%A1lez-752406232_lab-crackmapexec-activity-7273064080972673024-wYZE/?originalSubdomain=es

<https://www.proofpoint.com/es/threat-reference/brute-force-attack>

<https://www.youtube.com/watch?v=6jM00NIRI9c>

<https://directorio.educa.jcyl.es/es/directorios/centros-infoeduca-salamanca/listado-10>

<https://ubuntu.com/blog/tag/ubuntu-24-04-lts>

<https://netfilter.org/projects/iptables/>

<https://www.rocajunyent.com/es/blog/post/principales-obligaciones-fiscales-de-las-pequenas-y-medianas-empresas-pymes>

<https://sede.agenciatributaria.gob.es/Sede/ayuda/manuales-videos-folletos/manuales-practicos/folleto-actividades-economicas.html>

<https://empresas.jcyl.es/web/es/plataforma-financiera/ayudas.html>

<https://www.cdti.es/ayudas-y-servicios>

13. Anexos

13.1 Conclusión final

La realización de este proyecto, me ha servido mucho tanto a título personal, siendo un apasionado de la ciberseguridad, como a título “profesional” demostrándome que realmente es el sector al que quiero dedicar mi vida.

Este trabajo final, ha sido una experiencia que me ha enriquecido desde el primer momento. Me ha obligado a continuar intentando sacar el proyecto, haciendo que me enfrente a problemas técnicos reales que en un entorno laboral podrían sucederme, desde la configuración de las máquinas, pasando por la realización del propio script, hasta llegar al final, con el desarrollo de las pruebas, que sin duda ha sido el apartado que más me ha costado, ya que cualquier error significaba reescribir el código de una manera o de otra hasta dar con la solución final.

Por lo tanto, estos 3 meses de aprendizaje continuo día tras día, me han permitido aumentar mi confianza a nivel técnico, me han permitido afrontar los problemas con una mentalidad más profesional, y confirman mi “teoría” de que quiero orientar mi carrera a la ciberseguridad y de que estoy en el camino correcto, entendiendo cómo funciona cada uno de los ámbitos de este sector.