



INSTITUTO DE EDUCACIÓN SECUNDARIA SERPIS

## Creación de un centro de ciberseguridad

Proyecto de Administración de Sistemas Informáticos en Red

**Ciclo Formativo de Grado Superior Administración de Sistemas Informáticos en Red**

**Autor:** Mollá Sanz, Eduardo

**Tutor:** Dominguis Rodrigo, Joaquín

**Curso:** 2022/2023



## **Resumen**

Este proyecto se centra en la investigación y desarrollo de un sistema de actuación frente a ataques DDoS, escaneo de redes y detección de intrusos mediante scripts diseñados específicamente para estas tareas. Estos scripts están integrados en un sistema de alertas que envía notificaciones al teléfono móvil del usuario en caso de detectar cualquier fallo de seguridad, como la detección de intrusos en la red.

El objetivo principal del proyecto es proporcionar una solución integral para proteger los sistemas informáticos contra posibles amenazas y garantizar la seguridad de la red. Para lograr esto, se ha desarrollado un conjunto de scripts personalizados que abordan diferentes aspectos de la seguridad en redes.

Una de las funcionalidades clave del sistema es la capacidad de activar un cortafuegos para mitigar los ataques DDoS y evitar la interrupción de los servicios web de los clientes. Cuando se detecta un ataque de este tipo, el sistema automáticamente habilita el cortafuegos para proteger la página web y garantizar su disponibilidad, pero también podrá ser activado manualmente desde el centro de ciberseguridad.

Además, los scripts incluyen funciones de escaneo de redes que permiten identificar y controlar los dispositivos conectados a la red. Si se detecta la conexión de un dispositivo no reconocido, se envía una notificación al usuario para que tome las medidas correspondientes. Esto asegura un control exhaustivo de los dispositivos y ayuda a prevenir intrusiones no autorizadas.

El sistema de alertas integrado es fundamental para proporcionar una respuesta rápida frente a los fallos de seguridad. Al recibir notificaciones en su teléfono móvil, el usuario puede actuar de inmediato para contrarrestar las amenazas y minimizar los riesgos.

El proyecto final será presentado en una exposición, donde se exhibirá el funcionamiento y los resultados obtenidos mediante la implementación de los scripts diseñados. Este enfoque ofrece una solución efectiva y adaptable para abordar los desafíos de seguridad en los sistemas informáticos en red, brindando una protección sólida contra ataques DDoS, escaneos de redes y actividades intrusivas.

## Resum

Aquest projecte se centra en la investigació i desenvolupament d'un sistema d'actuació davant d'atacs DDoS, escaneig de xarxes i detecció d'intrusos mitjançant scripts dissenyats específicament per a aquestes tasques. Aquests scripts estan integrats en un sistema d'alertes que envia notificacions al telèfon mòbil de l'usuari en cas de detectar qualsevol fallada de seguretat, com la detecció d'intrusos en la xarxa.

L'objectiu principal del projecte és proporcionar una solució integral per protegir els sistemes informàtics contra possibles amenaces i garantir la seguretat de la xarxa. Per aconseguir-ho, s'ha desenvolupat un conjunt de scripts personalitzats que aborden diferents aspectes de la seguretat en xarxes.

Una de les funcionalitats clau del sistema és la capacitat d'activar un tallafocs per mitigar els atacs DDoS i evitar la interrupció dels serveis web dels clients. Quan es detecta un atac d'aquest tipus, el sistema habilita automàticament el tallafocs per protegir la pàgina web i garantir la seuva disponibilitat, però també pot ser activat manualment des del centre de ciberseguretat.

A més, els scripts inclouen funcions d'escaneig de xarxes que permeten identificar i controlar els dispositius connectats a la xarxa. Si es detecta la connexió d'un dispositiu no reconegut, s'envia una notificació a l'usuari perquè prenga les mesures corresponents. Això assegura un control exhaustiu dels dispositius i ajuda a prevenir intrusions no autoritzades.

El sistema d'alertes integrat és fonamental per proporcionar una resposta ràpida davant les fallades de seguretat. En rebre notificacions en el seu telèfon mòbil, l'usuari pot actuar immediatament per contrarestar les amenaces i minimitzar els riscos.

El projecte final serà presentat en una exposició, on s'exhibirà el funcionament i els resultats obtinguts mitjançant la implementació dels scripts dissenyats. Aquest enfocament ofereix una solució efectiva i adaptable per abordar els desafiaments de seguretat en els sistemes informàtics en xarxa, proporcionant una protecció sòlida contra atacs DDoS, escanejos de xarxes i activitats intrusives.

## **Abstract**

This project focuses on the research and development of a system to counteract DDoS attacks, network scanning, and intrusion detection through scripts specifically designed for these tasks. These scripts are integrated into an alert system that sends notifications to the user's mobile phone in case of detecting any security breach, such as detecting intruders on the network.

The main objective of the project is to provide a comprehensive solution to protect computer systems against potential threats and ensure network security. To achieve this, a set of customized scripts has been developed, addressing various aspects of network security.

One of the key functionalities of the system is the ability to activate a firewall to mitigate DDoS attacks and prevent disruptions to clients' web services. When such an attack is detected, the system automatically enables the firewall to protect the website and ensure its availability. Additionally, the firewall can also be manually activated from the cybersecurity center.

Moreover, the scripts include network scanning functions that allow for the identification and control of devices connected to the network. If the connection of an unrecognized device is detected, a notification is sent to the user to take appropriate measures. This ensures comprehensive device control and helps prevent unauthorized intrusions.

The integrated alert system is crucial in providing a prompt response to security breaches. By receiving notifications on their mobile phones, users can take immediate action to counter threats and minimize risks.

The final project will be presented in an exhibition, showcasing the functionality and results achieved through the implementation of the designed scripts. This approach offers an effective and adaptable solution to address security challenges in computer networks, providing robust protection against DDoS attacks, network scanning, and intrusive activities.

# ÍNDICE

<b>1. Resumen .....</b>
- Castellano.....
-Valenciano .....
-Inglés.....
<b>2. Índice (tabla de contenidos).....</b>
<b>3. Índice de imágenes (tabla de contenidos).....</b>
<b>Introducción.....</b>
<b>4. Justificación.....</b>
<b>5. Gestión del proyecto.....</b>
<b>6. Herramientas utilizadas.....</b>
<b>7. Descripción del proyecto.....</b>
<b>7.1. Análisis.....</b>
<b>7.2. Diseño.....</b>
<b>7.3. Implementación.....</b>
<b>7.4. Pruebas.....</b>
<b>7.5. Documentación.....</b>
<b>8. Trabajos futuros.....</b>
<b>9. Conclusiones.....</b>
<b>10. Bibliografía y webgrafía.....</b>
<b>11. Anexos:.....</b>
<b>-Anexo I. Instalación y configuración del bot.....</b>
<b>-Anexo II. Partes de código/Ficheros de configuración. importantes.</b>

## **INDICE DE IMAGENES**

## **JUSTIFICACIÓN**

La justificación de este proyecto se fundamenta en la creciente importancia de garantizar la seguridad y disponibilidad de los sistemas informáticos en red en un entorno cada vez más digitalizado. Los ataques DDoS, el escaneo de redes y las intrusiones representan amenazas significativas que pueden comprometer la integridad de los sistemas y la confidencialidad de los datos.

El objetivo principal de este proyecto es proporcionar una solución integral y efectiva para enfrentar estos desafíos de seguridad. Mediante la investigación y desarrollo de scripts específicamente diseñados, se busca contar con herramientas automatizadas que detecten y mitiguen los ataques DDoS, realicen un escaneo exhaustivo de las redes en busca de vulnerabilidades, y alerten oportunamente sobre intrusiones en el sistema.

Al integrar estos scripts en un sistema de alertas que envía notificaciones al teléfono móvil del usuario, se busca asegurar una respuesta inmediata ante cualquier fallo de seguridad detectado. Esto permite al usuario tomar medidas proactivas para contrarrestar las amenazas y minimizar los riesgos antes de que se produzcan mayores daños.

La implementación de un cortafuegos automatizado es una de las funcionalidades clave de este proyecto. Esta medida de seguridad ayuda a proteger los servicios web de los clientes ante ataques DDoS, evitando interrupciones y garantizando su disponibilidad. Asimismo, la capacidad de escanear y controlar los dispositivos conectados a la red permite identificar cualquier intrusión no autorizada y tomar acciones correctivas de manera oportuna.

La presentación de este proyecto en una exposición permitirá mostrar el funcionamiento y los resultados obtenidos a través de la implementación de los scripts diseñados. Los beneficios de esta solución personalizada se reflejarán en una mayor protección contra ataques DDoS, una respuesta más rápida frente a fallos de seguridad y una mayor capacidad de detección de intrusiones en la red.

## **Gestión del proyecto**

Para la gestión del proyecto, se estableció una estimación inicial de la temporalización y la planificación con el objetivo de cumplir con los plazos establecidos. El proyecto se inició a mediados de marzo y se programó su finalización para mediados de mayo. Durante este período, se dedicó gran parte del tiempo a aprender sobre scripting y su aplicación en seguridad informática.

La temporalización y planificación inicial del proyecto se diseñaron teniendo en cuenta la complejidad de las tareas a desarrollar, la adquisición de conocimientos sobre scripting y la implementación de las funcionalidades requeridas. Se establecieron hitos y se asignaron plazos para cada etapa del proyecto, asegurando así un avance progresivo y controlado.

Sin embargo, a lo largo del desarrollo del proyecto, se identificaron desafíos y necesidades adicionales que requirieron ajustes en la planificación inicial. El proceso de aprendizaje y comprensión de los conceptos de scripting y su aplicación en seguridad informática llevó más tiempo del esperado. Se realizaron investigaciones exhaustivas y pruebas para garantizar la efectividad y la calidad de los scripts desarrollados.

En base a estos ajustes y considerando la necesidad de cumplir con los objetivos establecidos, se replanificó el proyecto. Se reevaluaron los plazos y se establecieron nuevas fechas de entrega para cada etapa. Se priorizó la calidad del trabajo y se realizó un seguimiento cercano del avance del proyecto para asegurar la finalización en el tiempo disponible.

## Herramientas utilizadas

**VirtualBox:** VirtualBox es un software de virtualización que permite crear y ejecutar múltiples máquinas virtuales en un único sistema físico, brindando flexibilidad y control sobre los entornos virtuales, por lo que ha sido utilizado para crear máquinas virtuales con las que hacer este proyecto.

**Kali Linux:** Kali Linux es una distribución de Linux especializada en seguridad informática y pruebas de penetración. Ofrece una amplia gama de herramientas y recursos específicos para evaluar y asegurar sistemas y redes. Ha sido utilizado para simular ataques en este trabajo.

**Ubuntu:** Ubuntu es una distribución de Linux ampliamente utilizada, conocida por su facilidad de uso y estabilidad.

**Scripts:** Los scripts son secuencias de comandos escritos en un lenguaje de programación específico, como Bash, que automatizan tareas en sistemas informáticos, mejorando la eficiencia y la productividad.

**Apache2:** Apache2 es un servidor web de código abierto ampliamente utilizado en el mundo de la tecnología. Proporciona una plataforma robusta y flexible para alojar sitios web y aplicaciones web de manera segura y eficiente.

## **8. DESCRIPCIÓN DEL PROYECTO**

### **8.1 Análisis**

El proyecto se centra en desarrollar un sistema integral para proteger los sistemas informáticos contra ataques DDoS, escaneo de redes y detección de intrusos. A través de scripts personalizados, el sistema aborda diferentes aspectos de la seguridad en redes y ofrece varias funcionalidades clave:

1. Actuación frente a ataques DDoS: El sistema cuenta con la capacidad de activar automáticamente un cortafuegos cuando se detecta un ataque DDoS. Esto permite mitigar los ataques y garantizar la disponibilidad de los servicios web para los clientes. También puede activarse manualmente desde el centro de ciberseguridad, lo que brinda flexibilidad y control adicional en la protección contra ataques DDoS.
2. Escaneo de redes: Los scripts incluyen funciones de escaneo de redes para identificar y controlar los dispositivos conectados a la red. Si se detecta la conexión de un dispositivo no reconocido, se envía una notificación al usuario para que tome medidas adecuadas. Esto asegura un control exhaustivo de los dispositivos y ayuda a prevenir intrusiones no autorizadas.
3. Detección de intrusos: Los scripts diseñados específicamente para esta tarea permiten detectar intrusiones en la red. Si se identifica un intruso, el sistema envía una alerta al usuario, lo que permite una respuesta rápida y la adopción de medidas para contrarrestar la amenaza.
4. Sistema de alertas: El sistema cuenta con un sistema de alertas integrado que envía notificaciones al teléfono móvil del usuario en caso de detectar cualquier fallo de seguridad. Esto proporciona una respuesta rápida frente a los fallos de seguridad, permitiendo al usuario actuar de inmediato para minimizar los riesgos.

## 8.2 Diseño

El diseño que se describe es una simulación de un entorno de red que se utilizará para probar diversas operaciones de ciberseguridad y hacer demostraciones de cómo se pueden detectar y prevenir ciertos tipos de ataques informáticos. El diseño consta de tres partes principales:

1. **Máquina virtual con Kali Linux:** Kali Linux es una distribución de Linux que se utiliza ampliamente para pruebas de seguridad y hacking ético debido a la gran cantidad de herramientas de pruebas de penetración que incorpora. En este diseño, Kali Linux se instala en una máquina virtual (VM). Este entorno aislado permite realizar pruebas de seguridad sin afectar al sistema operativo subyacente o a las redes a las que está conectado el sistema. El objetivo de esta VM es simular un atacante en la red, y se utilizará principalmente para lanzar ataques de denegación de servicio (DoS). En un ataque de DoS, el objetivo es inundar una máquina o una red con tráfico no solicitado para sobrecargarla y provocar que se bloquee o se vuelva inutilizable.
2. **Ubuntu cliente:** Este es otro sistema que simula un nuevo dispositivo que se conecta a la red. Está basado en Ubuntu, una popular distribución de Linux conocida por su facilidad de uso y fiabilidad. En este diseño, el cliente Ubuntu se utiliza para hacer pruebas de escaneo de redes. El escaneo de redes es una técnica utilizada para identificar dispositivos activos en una red y recopilar información sobre ellos, como las puertas de enlace abiertas o los servicios en ejecución. Esto puede ser útil para identificar posibles vulnerabilidades en una red.
3. **Ubuntu servidor con sistema de alertas:** La tercera parte del diseño es otro sistema Ubuntu que actúa como servidor. Este servidor contiene un sistema de alertas y un centro de ciberseguridad. El sistema de alertas está diseñado para detectar y notificar cualquier actividad sospechosa o maliciosa en la red. Por ejemplo, podría configurarse para enviar una alerta si detecta un aumento repentino del tráfico de red, lo que podría indicar un ataque de DoS. El centro de ciberseguridad se refiere a un conjunto de herramientas y procesos utilizados para monitorizar, gestionar y proteger la red.

4. **Ubuntu cliente conocido:** Este sería un cuarto componente en el diseño. Al igual que el segundo componente, este es otro sistema Ubuntu que se utiliza como cliente en la red. Sin embargo, a diferencia del Ubuntu cliente mencionado anteriormente, este se considera un cliente "conocido" por el sistema.

En este contexto, un cliente conocido es un dispositivo que ya ha sido identificado y autorizado por el sistema de alertas y el centro de ciberseguridad en el servidor Ubuntu. Esto significa que este cliente Ubuntu ya ha sido evaluado y se ha determinado que es seguro. Por lo tanto, el sistema de alertas no generará notificaciones cuando este cliente se conecte a la red o realice actividades normales, a diferencia de lo que ocurre con los dispositivos desconocidos.

Esta inclusión proporciona un contraste útil con el cliente Ubuntu "desconocido". Mientras que la actividad del cliente desconocido puede desencadenar alertas y requerir investigaciones adicionales para determinar si es seguro, la actividad del cliente conocido puede ser más fácilmente ignorada, ya que se confía en su seguridad.

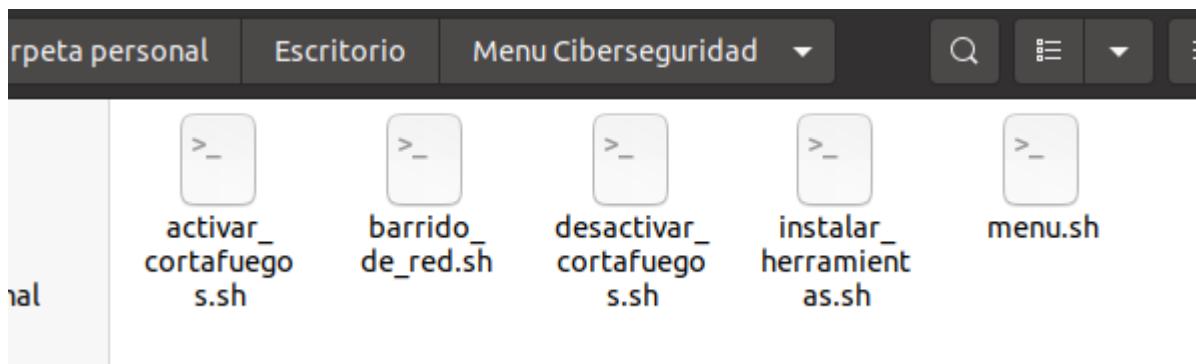
Esta configuración permite demostrar y practicar distintos aspectos de la ciberseguridad en un entorno controlado. Puedes probar diferentes tipos de ataques desde la máquina de Kali, observar cómo estos ataques afectan al cliente Ubuntu y, por último, utilizar el servidor Ubuntu para detectar y responder a estos ataques.

### 8.3 Implementación

La implementación de este proyecto se basa en implementar un conjunto de scripts en un servidor para crear un menú interactivo con múltiples opciones relacionadas con la ciberseguridad. Estos scripts serán diseñados específicamente para abordar diversas áreas de seguridad informática.

La implementación del proyecto implica los siguientes pasos:

1. Descargar los scripts: Los scripts necesarios para crear el menú de ciberseguridad se descargarán en el servidor objetivo. Estos scripts contendrán el código necesario para ejecutar las diferentes funcionalidades relacionadas con la ciberseguridad.



2. Asignar permisos de ejecución: Una vez que los scripts se encuentren en el servidor, se otorgarán los permisos adecuados para que puedan ser ejecutados. Esto permitirá que los scripts accedan a los recursos del sistema y realicen las tareas de seguridad necesarias sin restricciones.



3. Generar el menú: Los scripts se configurarán de manera que, al ejecutarlos, se

genere un menú interactivo en el servidor. Este menú presentará al usuario una lista de opciones relacionadas con la ciberseguridad que pueden seleccionarse.

4. Opciones de ciberseguridad: Cada opción en el menú corresponderá a una funcionalidad de seguridad específica.

Una vez implementado, el menú de ciberseguridad permitirá a los usuarios del servidor seleccionar y ejecutar las diferentes opciones según sus necesidades y objetivos de seguridad. Esto facilitará el acceso a herramientas y funcionalidades de ciberseguridad clave, brindando una interfaz intuitiva y simplificada para llevar a cabo tareas de seguridad informática de manera eficiente.

## 8.4 Pruebas

La implementación y uso del script "Centro de Ciberseguridad" son aspectos clave de este proyecto. El script "Centro de Ciberseguridad" es una herramienta de software diseñada para mejorar y facilitar la administración de la ciberseguridad en un servidor o red. Este script no es solo una simple lista de comandos que se ejecutan secuencialmente, sino un programa más complejo que ofrece una gama de opciones y funcionalidades interactivas para el usuario.

Para ejecutar el Centro de ciberseguridad, deberemos hacerlo de la siguiente manera, o simplemente se utilizará el autoejecutable.

```
empresa@ServidorVictima:/home/edu/Escritorio/Menu Ciberseguridad$ ./menu.sh
```

Una vez hecho esto, se desplegará un banner:

```
empresa@ServidorVictima:/home/edu/Escritorio/Menu Ciberseguridad$ ./menu.sh
```



A continuación podremos observar sus múltiples opciones:

```
root@ServidorVictima:/home/edu/Escritorio/Centro de Ciberseguridad# ./menu
```



Seleccione una opción:

- 1 Rhythmbox instalación de paquetes y herramientas necesarias
- 2. Barrido de red
- 3. Activar Cortafuegos
- 4. Desactivar Cortafuegos
- 5. Detección de intrusos
- 6. Salir

1. **Instalación de paquetes y herramientas necesarias:** Ejecuta un script llamado "instalar\_herramientas.sh" que se encarga de instalar los paquetes y herramientas necesarios para el centro de ciberseguridad.

```
root@ServidorVictima:/home/edu/Escritorio/Centro de Ciberseguridad# ./menu.sh
[CE4TRO DE CIBERSEGURIDAD]
Seleccione una opción:
1. Instalación de paquetes y herramientas necesarias
2. Barrido de red
3. Activar Cortafuegos
4. Desactivar Cortafuegos
5. Detección de intrusos
6. Salir
1
¿Desea instalar las herramientas necesarias? (s/n) [
```

Se desplegará el mensaje de confirmación para instalar los paquetes que utiliza el resto del script.

2. **Barrido de red:** Ejecuta un script llamado "barrido\_de\_red.sh" que realiza un escaneo de la red en busca de dispositivos y servicios activos.

El cual desplegará una opción por si queremos guardar los hosts registrados en la red para una posterior investigación.

```
1. Instalación de paquetes y herramientas necesarias
2. Barrido de red
3. Activar Cortafuegos
4. Desactivar Cortafuegos
5. Detección de intrusos
6. Salir
2
Haciendo barrido de red con nmap...
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-19 14:45 CEST
Nmap scan report for _gateway (192.168.43.1)
Host is up (0.0083s latency).
MAC Address: 62:A5:36:A8:AE:91 (Unknown)
Nmap scan report for 192.168.43.10
Host is up (0.00048s latency).
MAC Address: 08:00:27:81:87:5F (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.43.20
Host is up (0.0016s latency).
MAC Address: 08:00:27:61:08:F2 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.43.58
Host is up (0.00051s latency).
MAC Address: EC:2E:98:CC:1D:9F (Unknown)
Nmap scan report for 192.168.43.100
Host is up (0.00050s latency).
MAC Address: 08:00:27:C7:E1:36 (Oracle VirtualBox virtual NIC)
Nmap scan report for ServidorVictima (192.168.43.200)
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.43 seconds
¿Desea guardar los resultados en un archivo .txt? (s/n)
```

3.

4. **Activar Cortafuegos:** Ejecuta un script llamado "activar\_cortafuegos.sh" que activa el cortafuegos, una medida de seguridad que controla y filtra el tráfico de red.

Al activar el cortafuegos, saldrá el siguiente mensaje:

Seleccione una opción:

1. Instalación de paquetes y herramientas necesarias
  2. Barrido de red
  3. Activar Cortafuegos
  4. Desactivar Cortafuegos
  5. Detección de intrusos
  6. Salir
- 3

CORTAFUEGOS ACTIVADO

(Código en los Anexos)

Este script configura el cortafuegos utilizando el comando `iptables`, estableciendo políticas de rechazo predeterminadas y permitiendo el tráfico necesario, como el tráfico de la interfaz de loopback y las conexiones SSH entrantes en el puerto 22. Además, guarda las reglas del cortafuegos para que persistan después de un reinicio.

5. **Desactivar Cortafuegos:** Ejecuta un script llamado "desactivar\_cortafuegos.sh" que desactiva el cortafuegos, permitiendo el tráfico de red sin restricciones.

```

Seleccione una opción:
1. Instalación de paquetes y herramientas necesarias
2. Barrido de red
3. Activar Cortafuegos
4. Desactivar Cortafuegos
5. Detección de intrusos
6. Salir
4

[ CORTAFUEGOS ] DESACTIVADO

Cortafuego desactivado y reglas de iptables configuradas a los valores predeterminados.
Presione ENTER para continuar... ]

```

(Código en los Anexos)

6. **Detección de intrusos:** Ejecuta un script llamado "DetectorIntrusos.sh" que realiza una detección de posibles intrusiones o actividad maliciosa en el sistema o la red.

Al hacer iniciar esta opción, pasará lo siguiente:

```

Seleccione una opción:
1. Instalación de paquetes y herramientas necesarias
2. Barrido de red
3. Activar Cortafuegos
4. Desactivar Cortafuegos
5. Detección de intrusos
6. Salir
5
{
  "ok": true,
  "result": [
    {
      "message_id": 194,
      "from": {
        "id": 5750097206,
        "is_bot": true,
        "first_name": "AlertaIntruso",
        "username": "intruder_alert_edu_bot"
      },
      "chat": {
        "id": 5661991162,
        "first_name": "Edu",
        "type": "private"
      },
      "date": 1684501534,
      "text": "IP no autorizada detectada: 192.168.43.20",
      "entities": [
        {
          "offset": 28,
          "length": 13,
          "type": "url"
        }
      ]
    }
  ]
}

```

Este script realiza un escaneo de la red cada 5 segundos durante un periodo de 30 segundos, buscando direcciones IP que no están en la lista permitida.

Cuando detecta una dirección IP no permitida, envía una alerta a través de un bot de Telegram, indicando que se ha detectado una IP no autorizada. Para cada ciclo de escaneo de la red, el script se pausa por 5 segundos, y una vez transcurridos los 30 segundos, se detiene.

Al finalizar su ejecución, el script invoca al menú principal.

7. **Salir:** Muestra el mensaje "Saliendo..." y finaliza la ejecución del script.

Una vez finalizada la explicación detallada de todas las opciones y funcionalidades de los scripts desarrollados, este proyecto, junto con la presentación correspondiente, incluye dos demostraciones distintas.

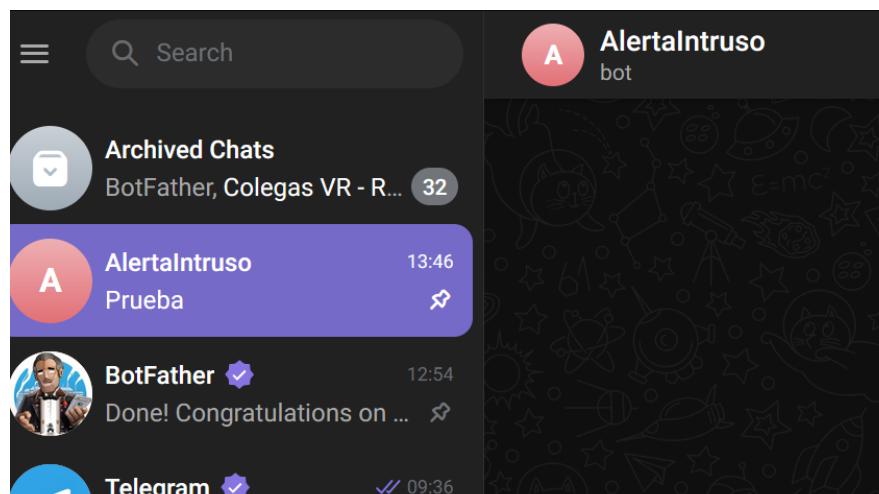
En primer lugar, se lleva a cabo una prueba exhaustiva del detector de intrusos, con el propósito de verificar su capacidad para enviar una alerta al dispositivo móvil del usuario en

el momento en que una dirección IP no confiable acceda a la red de la empresa. Esta demostración tiene como objetivo mostrar la eficacia del sistema de detección y notificación de intrusiones en tiempo real.

En segundo lugar, se realiza una simulación de un ataque DDoS utilizando una máquina Kali Linux dirigida a una máquina Ubuntu víctima. Durante esta demostración, se efectúan pruebas en las que se somete a la página web alojada en el servidor Apache2 a un alto volumen de solicitudes, con el fin de mostrar cómo el centro de ciberseguridad puede detectar y mitigar este tipo de ataques, evitando así la interrupción del servicio y protegiendo la integridad de la infraestructura.

Ambas demostraciones tienen como propósito principal destacar la efectividad y la importancia del cortafuegos implementado en el proyecto, así como resaltar las capacidades de respuesta y protección ante amenazas ciberneticas que este sistema de seguridad proporciona.

**En primer lugar, en esta imagen se puede observar el chatbot creado llamado “AlertalIntruso”, por el cual nos llegarán las notificaciones.**

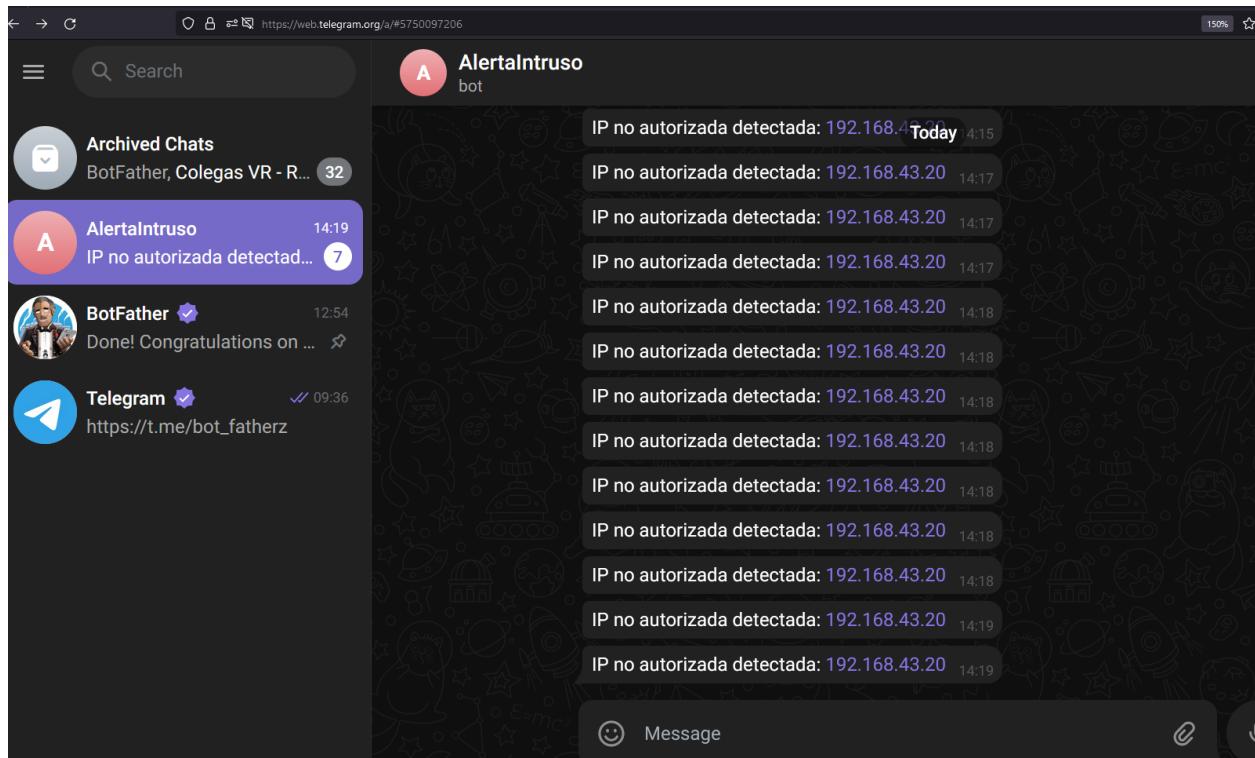


Conectaremos la nueva máquina “desconocida”:

```

edu@MaquinaDesconocida:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    default qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    UP group default qlen 1000
        link/ether 08:00:27:61:08:f2 brd ff:ff:ff:ff:ff:ff
        inet 192.168.43.20/24 brd 192.168.43.255 scope glo
    os3
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe61:8f2/64 scope link
            valid_lft forever preferred_lft forever
edu@MaquinaDesconocida:~$
```

Acto seguido, se procede a observar las consecuencias que se producen cuando un individuo externo a nuestra red accede con un dispositivo que posee una dirección IP desconocida y no ha sido previamente configurada como confiable.

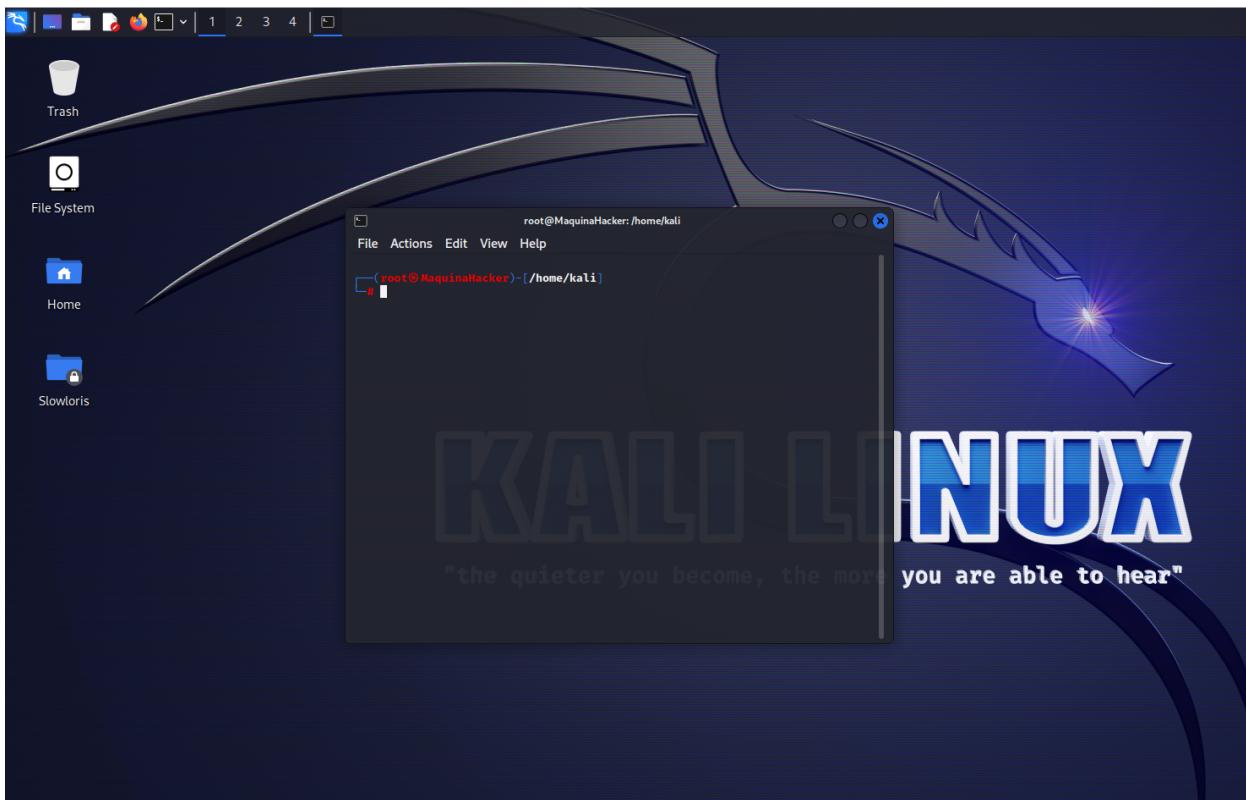


(Código del script en el anexo)

Gracias a la implementación de un script que está vinculado a un token y una identificación

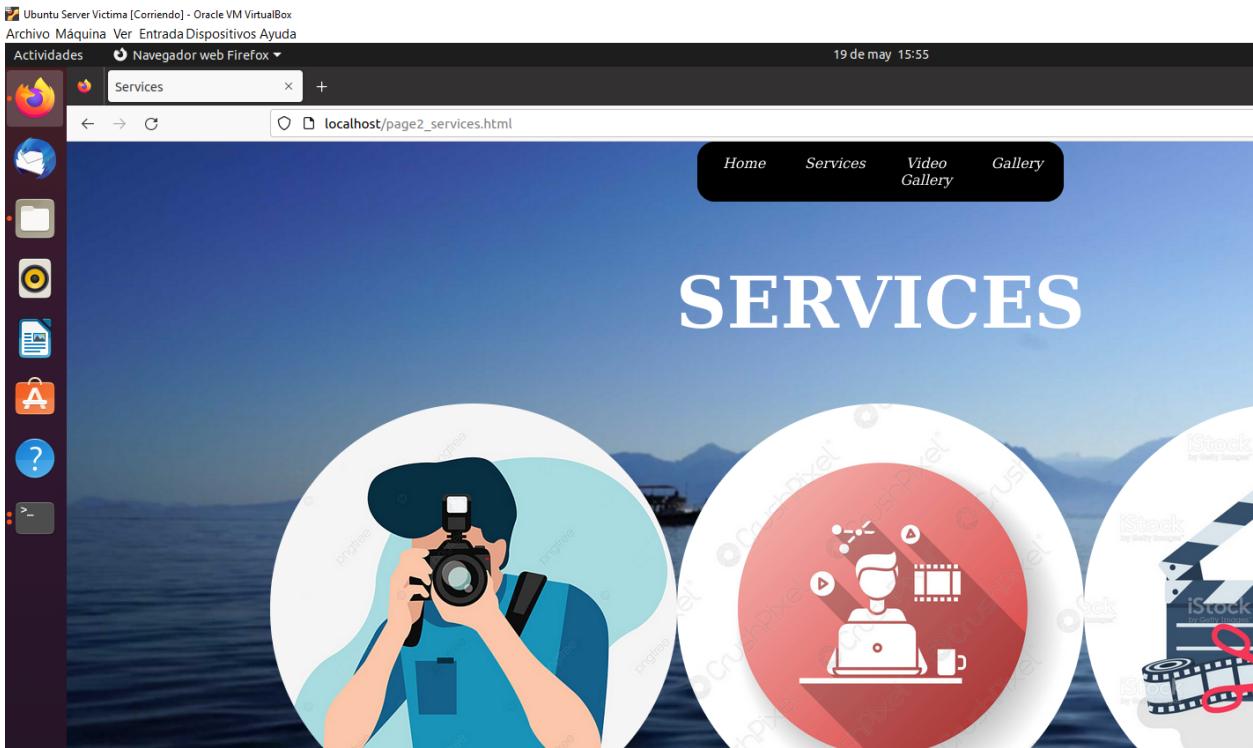
de Telegram, se logra la capacidad de enviar notificaciones desde un servidor Ubuntu a un dispositivo móvil. Esta integración permite establecer una comunicación bidireccional y enviar mensajes de alerta, avisos o cualquier otra información relevante desde el sistema Ubuntu al usuario a través de la plataforma de mensajería Telegram. De esta manera, se brinda una forma eficiente y conveniente de mantener informado al usuario sobre eventos importantes o situaciones críticas que ocurran en el entorno del servidor Ubuntu.

**Pasemos ahora a la prueba de ataque de DDos, para ello usamos el kali linux:**



Y utilizaremos la máquina virtual **Server Victima**.

En este caso, tenemos una página web subida como localhost, gracias al apache2 para hacer las pruebas, ya que va a ser en local.



Con Kali linux podemos hacer ataques DDos de diferentes maneras, la manera mas conocida es el ping de la muerte, con la que puede bastar segun la potencia del sistema victimas.

Utilizaremos el comando “htop” para monitorear el rendimiento del sistema en tiempo real. Al ejecutar el comando htop, se muestra una interfaz interactiva que presenta información detallada y en tiempo real sobre los procesos en ejecución, el uso de la CPU, la memoria, los

recursos del sistema y otras métricas importantes.

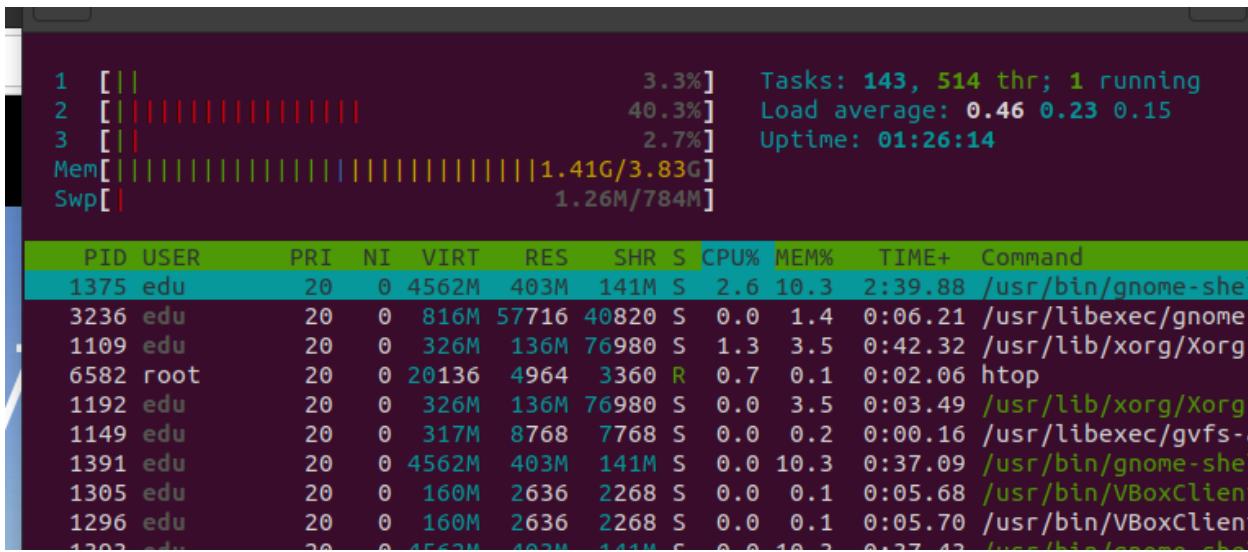
Estos son los valores habituales de la máquina antes del ataque.

1	[ ]	0.6%	Tasks: 147, 502 thr; 1 running								
2	[ ]	1.2%	Load average: 0.18 0.13 0.14								
3	[ ]	0.6%	Uptime: 01:17:39								
Mem	[██████████]	1.39G/3.83G									
Swp	[ ]	1.26M/784M									
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
5865	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:34.72	/usr/lib/firefox/firefox -new-window
1375	edu	20	0	4554M	394M	146M	S	1.9	10.1	2:18.10	/usr/bin/gnome-shell
1109	edu	20	0	330M	140M	81576	S	0.6	3.6	0:35.20	/usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background
612	root	20	0	264M	19340	16816	S	0.0	0.5	0:00.59	/usr/sbin/NetworkManager --no-daemon
3236	edu	20	0	812M	55160	40292	S	0.0	1.4	0:03.45	/usr/libexec/gnome-terminal-server
6051	edu	20	0	2613M	235M	142M	S	0.0	6.0	0:13.29	/usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 28765 -p
5885	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:00.88	/usr/lib/firefox/firefox -new-window
6534	root	20	0	20456	5044	3292	R	1.2	0.1	0:04.03	htop
1391	edu	20	0	4554M	394M	146M	S	0.6	10.1	0:32.28	/usr/bin/gnome-shell
1392	edu	20	0	4554M	394M	146M	S	0.0	10.1	0:32.10	/usr/bin/gnome-shell
5951	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:01.45	/usr/lib/firefox/firefox -new-window
1192	edu	20	0	330M	140M	81576	S	0.0	3.6	0:02.77	/usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background
6058	edu	20	0	2613M	235M	142M	S	0.0	6.0	0:00.73	/usr/lib/firefox/firefox -contentproc -childID 4 -isForBrowser -prefsLen 28765 -p
5912	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:00.96	/usr/lib/firefox/firefox -new-window
1296	edu	20	0	160M	2636	2268	S	0.0	0.1	0:05.01	/usr/bin/VBoxClient --draganddrop
1393	edu	20	0	4554M	394M	146M	S	0.0	10.1	0:32.61	/usr/bin/gnome-shell
1305	edu	20	0	160M	2636	2268	S	0.0	0.1	0:05.00	/usr/bin/VBoxClient --draganddrop
5880	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:00.86	/usr/lib/firefox/firefox -new-window
5905	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:06.95	/usr/lib/firefox/firefox -new-window
5948	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:02.79	/usr/lib/firefox/firefox -new-window
5949	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:00.97	/usr/lib/firefox/firefox -new-window
5903	edu	20	0	3464M	441M	255M	S	0.0	11.2	0:00.52	/usr/lib/firefox/firefox -new-window
610	messagebu	20	0	9724	5764	3520	S	0.0	0.1	0:01.07	/usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-a
1149	edu	20	0	317M	8768	7768	S	0.0	0.2	0:00.15	/usr/libexec/gvfs-afc-volume-monitor
1150	edu	20	0	317M	8768	7768	S	0.0	0.2	0:00.13	/usr/libexec/gvfs-afc-volume-monitor
826	kernoops	20	0	11264	444	0	S	0.0	0.0	0:00.04	/usr/sbin/kerneloops --test
1081	root	20	0	296M	2668	2292	S	0.0	0.1	0:00.06	/usr/sbin/VBoxService --pidfile /var/run/vboxadd-service.sh
606	avahi	20	0	8532	3252	2924	S	0.0	0.1	0:00.59	avahi-daemon: running [edu.local]
5041	edu	20	0	229M	114M	87052	S	0.0	0.2	0:00.05	/usr/libexec/gvfs-afc-volume-monitor

Iniciado el ataque de Kali Linux:

```
File Actions Edit View Help
root@MaquinaHacker: /home/kali
15008 bytes from 192.168.43.200: icmp_seq=19586 ttl=64 time=0.245 ms
15008 bytes from 192.168.43.200: icmp_seq=19587 ttl=64 time=0.527 ms
15008 bytes from 192.168.43.200: icmp_seq=19588 ttl=64 time=0.243 ms
15008 bytes from 192.168.43.200: icmp_seq=19589 ttl=64 time=0.272 ms
15008 bytes from 192.168.43.200: icmp_seq=19590 ttl=64 time=0.311 ms
15008 bytes from 192.168.43.200: icmp_seq=19591 ttl=64 time=0.248 ms
15008 bytes from 192.168.43.200: icmp_seq=19592 ttl=64 time=0.332 ms
15008 bytes from 192.168.43.200: icmp_seq=19593 ttl=64 time=0.259 ms
15008 bytes from 192.168.43.200: icmp_seq=19594 ttl=64 time=0.875 ms
15008 bytes from 192.168.43.200: icmp_seq=19595 ttl=64 time=0.256 ms
15008 bytes from 192.168.43.200: icmp_seq=19596 ttl=64 time=0.214 ms
15008 bytes from 192.168.43.200: icmp_seq=19597 ttl=64 time=0.167 ms
15008 bytes from 192.168.43.200: icmp_seq=19598 ttl=64 time=0.178 ms
15008 bytes from 192.168.43.200: icmp_seq=19599 ttl=64 time=0.220 ms
15008 bytes from 192.168.43.200: icmp_seq=19600 ttl=64 time=0.191 ms
15008 bytes from 192.168.43.200: icmp_seq=19601 ttl=64 time=0.190 ms
15008 bytes from 192.168.43.200: icmp_seq=19602 ttl=64 time=1.19 ms
15008 bytes from 192.168.43.200: icmp_seq=19603 ttl=64 time=0.302 ms
```

Al ser atacado, pasará lo siguiente dependiendo de la potencia de la máquina:

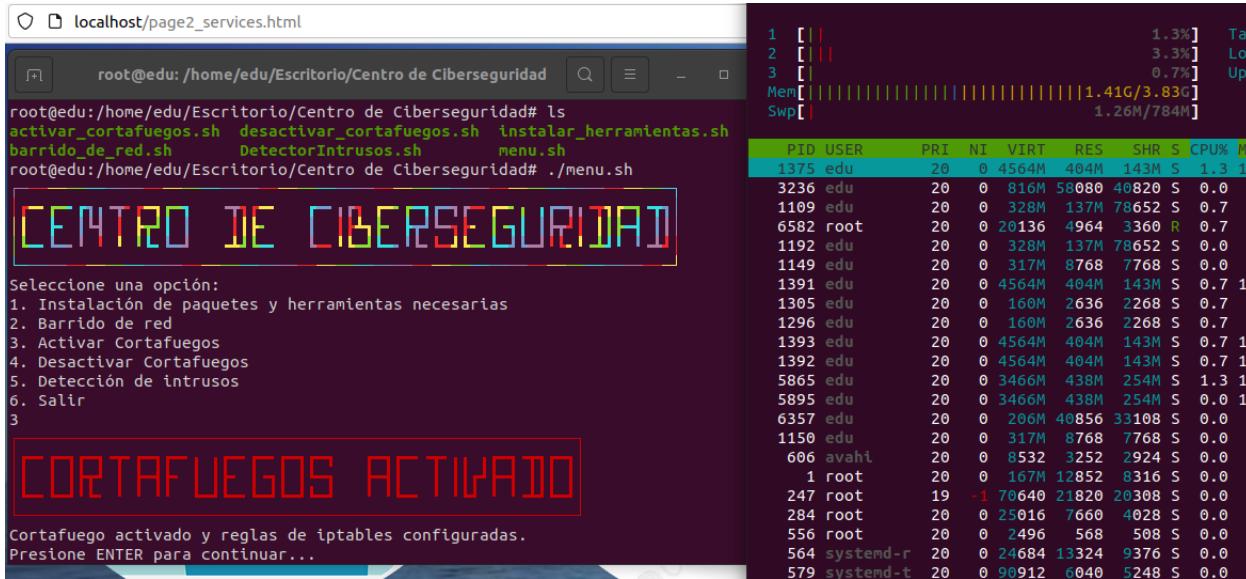


Algunos posibles cambios en los valores de htop durante un ataque DDoS pueden incluir:

1. Uso de la CPU: Debido a la gran cantidad de solicitudes que llegan al sistema, la CPU puede verse altamente utilizada al intentar procesar y responder a todas ellas. El uso de la CPU puede aumentar significativamente, lo que indica una carga excesiva y un posible impacto en el rendimiento general.
2. Uso de la memoria: El ataque DDoS puede agotar los recursos de memoria de la máquina al generar múltiples solicitudes simultáneas. Esto puede llevar a un aumento en el uso de la memoria, especialmente si el sistema intenta almacenar y procesar información sobre todas las solicitudes entrantes.
3. Uso de la red: Durante un ataque DDoS, la máquina objetivo puede estar recibiendo una gran cantidad de tráfico de red malicioso. Esto puede causar un aumento en la carga de la red, lo que se reflejará en los valores observados en htop, como un aumento en la tasa de transferencia de red o el número de paquetes recibidos.
4. Procesos en ejecución: Dependiendo del tipo de ataque DDoS, es posible que se generen múltiples instancias de procesos maliciosos en la máquina objetivo. Estos procesos pueden aparecer en la lista de procesos mostrada por htop y consumir recursos adicionales, como CPU y memoria.

En este caso, el uso a aumentado al 40%, y la página tarda bastante en refrescar.

Al activar el cortafuegos del Centro de Ciberseguridad, volverá a reducirse la cantidad de carga de trabajo de los recursos de la máquina, con lo cual querrá decir que se ha mitigado gran parte de las peticiones gracias al script cortafuegos



La activación del cortafuegos del Centro de Ciberseguridad resultará en una disminución de la carga de trabajo de los recursos de la máquina objetivo. Esto implica que se ha logrado mitigar una gran parte de las solicitudes maliciosas gracias al script del cortafuegos.

El cortafuegos es una medida de seguridad que se implementa para proteger una red o sistema informático de amenazas externas. Actúa como una barrera que controla y filtra el tráfico de red, permitiendo únicamente el paso de conexiones legítimas y bloqueando o restringiendo las conexiones no autorizadas o maliciosas.

Cuando se activa el cortafuegos del Centro de Ciberseguridad, se configura para analizar y evaluar el tráfico entrante y saliente de la máquina objetivo. Utilizando reglas predefinidas y personalizadas, el cortafuegos identifica y bloquea las solicitudes que se consideran sospechosas o maliciosas, impidiendo que lleguen al sistema y generen una carga adicional en los recursos.

Como resultado, la activación del cortafuegos reduce significativamente la cantidad de solicitudes dañinas o no deseadas que llegan a la máquina objetivo. Esto alivia la carga de trabajo de los recursos de la máquina, como la CPU, la memoria y el ancho de banda de red, ya que no se ven obligados a procesar y responder a estas solicitudes no deseadas.

En consecuencia, al mitigar gran parte de las peticiones maliciosas mediante el script del cortafuegos, se logra una protección efectiva contra los ataques y se mejora el rendimiento general del sistema. La activación del cortafuegos del Centro de Ciberseguridad permite mantener la disponibilidad y la integridad de los recursos de la máquina al limitar el acceso no autorizado y asegurar que solo se permita el tráfico legítimo.

En cuanto se elimine la amenaza, se podrá volver a desactivar el cortafuegos para que el tráfico de red de la máquina vuelva a sus valores por defecto.



La **implementación** de este script implica varias etapas. Primero, se necesita desarrollar el script. Este proceso puede implicar escribir código en un lenguaje de scripting, como Python, Bash o PowerShell, para crear las funcionalidades deseadas. El script deberá incluir código para generar un menú de opciones cuando se ejecute. Cada opción en este menú corresponderá a una tarea de ciberseguridad diferente que el script puede realizar.

Una vez que el script está desarrollado, debe ser probado para asegurarse de que funciona correctamente. Esto podría implicar ejecutar el script en un entorno de prueba y verificar que cada opción del menú realiza la acción esperada. Cualquier error o problema que surja durante las pruebas deberá ser corregido antes de que el script esté listo para ser implementado.

Cuando el script ha sido probado y se ha confirmado que funciona correctamente, se puede implementar en el servidor. Esto implica copiar o descargar el script en el servidor y otorgarle permisos de ejecución, lo que permitirá al servidor ejecutar el script.

El **uso** del script "Centro de Ciberseguridad" es relativamente sencillo gracias a su interfaz basada en menús. Cuando el script se ejecuta en el servidor, desplegará un menú con varias opciones. Cada opción en este menú corresponderá a una funcionalidad o tarea de ciberseguridad diferente que el script puede realizar. Para usar una opción, el usuario simplemente seleccionará esa opción en el menú.

## 8.5 Documentación

## 9. Trabajos futuros

En este proyecto, se ha llevado a cabo un enfoque intuitivo para abordar la ciberseguridad, con el objetivo de facilitar su implementación y uso por parte de los usuarios. Sin embargo, dada la naturaleza amplia y compleja de la programación y la utilización de scripts, es evidente que existen numerosas oportunidades de mejora y posibilidades para enriquecer aún más este proyecto mediante la adición de nuevas funciones.

1. Mejora en la detección de ataques DDoS: Implementar técnicas avanzadas de detección de ataques de denegación de servicio distribuidos (DDoS) para identificar y mitigar de manera más eficiente los ataques que intentan saturar los recursos del sistema.
2. Análisis de comportamiento de usuarios: Realizar un seguimiento de las actividades de los usuarios en la red para detectar comportamientos sospechosos, como accesos no autorizados o transferencias de datos inusuales, lo que permitirá una respuesta temprana ante posibles amenazas.
3. Integración con sistemas de gestión de eventos e información de seguridad (SIEM): Conectar el sistema de seguridad con un SIEM permitirá una correlación y análisis más amplio de eventos de seguridad, lo que ayudará a identificar amenazas de manera más efectiva y a responder rápidamente a los incidentes.
4. Análisis de vulnerabilidades y parcheo automático: Realizar un análisis automatizado de las vulnerabilidades en los sistemas y aplicar correcciones o parches de seguridad de forma automática para mantener los sistemas protegidos y actualizados.
5. Integración con soluciones de autenticación multifactor (MFA): Reforzar la seguridad de la autenticación de usuarios mediante la implementación de métodos de autenticación multifactor, como el uso de códigos de verificación o tokens, junto con las contraseñas tradicionales.
6. Capacidades de análisis forense: Incorporar herramientas y técnicas de análisis forense para investigar y recopilar pruebas después de un incidente de seguridad, lo que ayudará a identificar las causas y tomar medidas para prevenir futuros ataques.
7. Integración con servicios de inteligencia de amenazas: Conectar el sistema de seguridad con servicios de inteligencia de amenazas permitirá recibir información actualizada sobre las últimas amenazas y patrones de ataque, mejorando así la detección y respuesta ante nuevas y emergentes amenazas.
8. Análisis de comportamiento de red: Analizar el tráfico de red en busca de patrones de comportamiento anómalos o indicadores de ataques, lo que permitirá una detección temprana de actividades maliciosas y una respuesta rápida para mitigar las amenazas.

9. Capacidades de respuesta automatizada: Implementar la capacidad de tomar acciones de manera automática ante amenazas detectadas, como bloquear direcciones IP sospechosas o deshabilitar cuentas de usuario comprometidas, lo que mejorará la respuesta y minimizará el impacto de los ataques.
10. Gestión de contraseñas: Establecer un sistema seguro de gestión de contraseñas que promueva el uso de contraseñas robustas y la rotación periódica de las mismas, reduciendo así los riesgos de brechas de seguridad causadas por contraseñas débiles o comprometidas.
11. Capacidades de análisis de registros: Analizar los registros de eventos y registros del sistema en busca de actividades sospechosas o indicadores de compromiso, permitiendo una detección temprana de incidentes de seguridad y una respuesta oportuna para minimizar el impacto.
12. Entrenamiento y concientización en seguridad: Proporcionar programas de capacitación y concientización en seguridad informática para los usuarios finales, educándolos sobre las mejores prácticas de seguridad y

## **10. Conclusiones**

En conclusión, el proyecto se ha centrado en la investigación y desarrollo de un sistema integral de seguridad informática que aborda la protección contra ataques DDoS, escaneo de redes y detección de intrusos. Se han diseñado scripts personalizados que permiten detectar y mitigar amenazas, y se ha integrado un sistema de alertas para una respuesta rápida frente a posibles fallos de seguridad.

Sin embargo, durante la realización de la simulación y demostración del proyecto, se encontraron dificultades relacionadas con los ataques DDoS. El uso de la red WiFi del móvil como punto de acceso para simular estos ataques resultó menos confiable que un router, lo que provocó desconexiones y dificultó la demostración efectiva de los ataques y su defensa.

Además, aunque inicialmente se tenía la intención de simular otros tipos de ataques y mostrar cómo mitigarlos, la investigación realizada concluyó que era más efectivo prevenirlos en lugar de enfrentarse a ellos posteriormente. Por lo tanto, no fue posible incluir otros tipos de ataques en la demostración.

A pesar de estas dificultades, el proyecto ha logrado desarrollar un sistema sólido de protección contra ataques DDoS, escaneo de redes y actividades intrusivas. La implementación de los scripts diseñados proporciona una solución integral para abordar los desafíos de seguridad en los sistemas informáticos en red, brindando una protección efectiva y adaptable. La exhibición de los resultados obtenidos en la exposición permitirá mostrar el funcionamiento y la eficacia de la solución desarrollada.

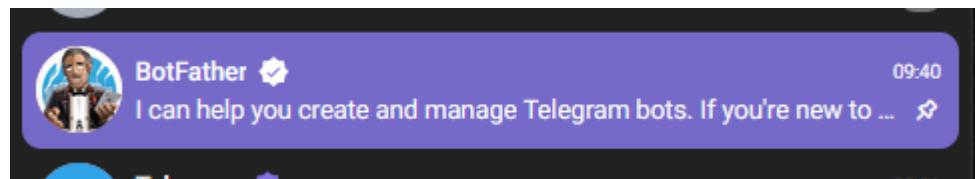
## 11. Bibliografía y webgrafía

## 12. Anexos

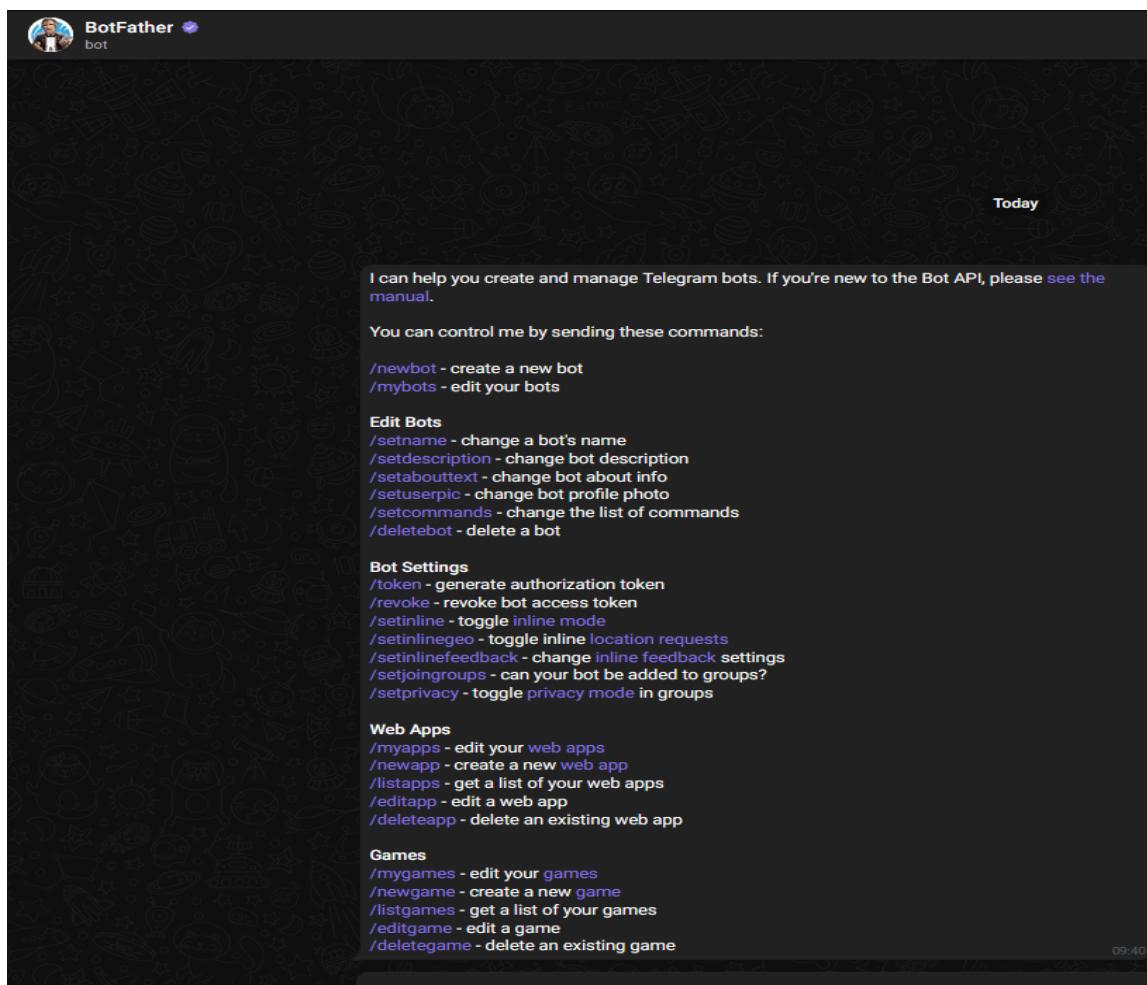
- ANEXO I. Manual Instalación/Configuración

Para la configuración de las alertas de telegram, se ha requerido los siguientes pasos.

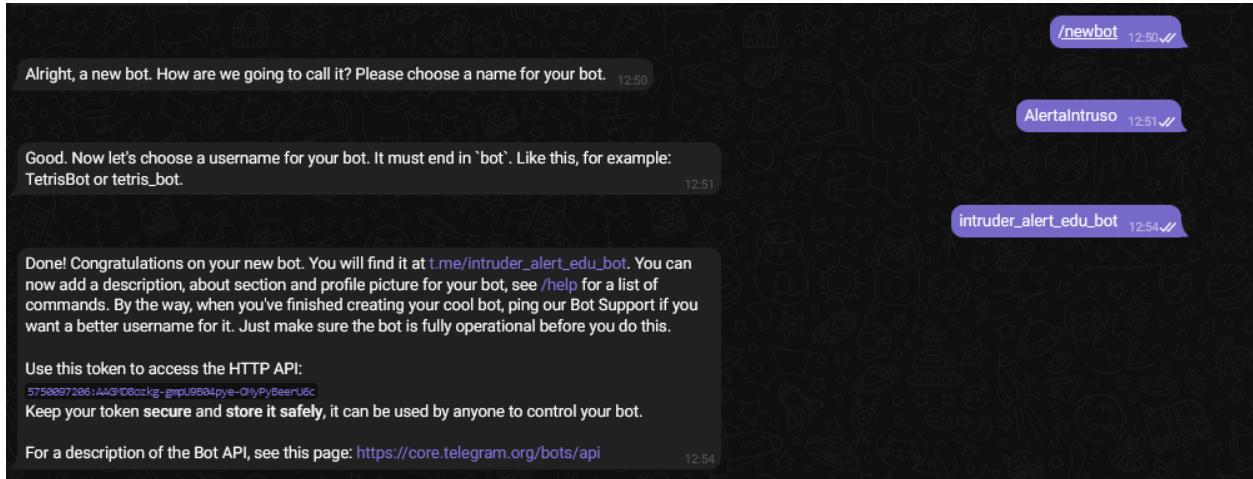
En primer lugar, hablar al chatbot “BotFather”



Contestará lo siguiente:



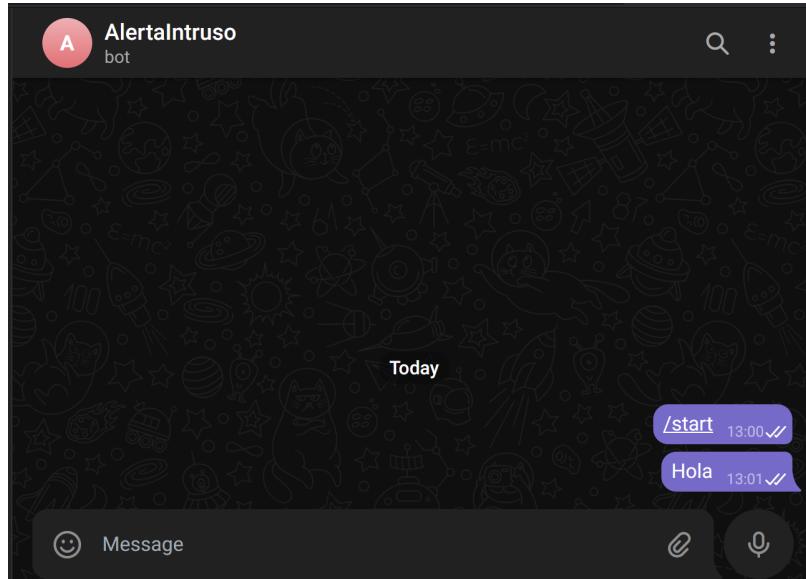
Se le solicitará un nuevo bot, se elegirá nombre de bot y usuario



En este caso nos ha dado la siguiente información.

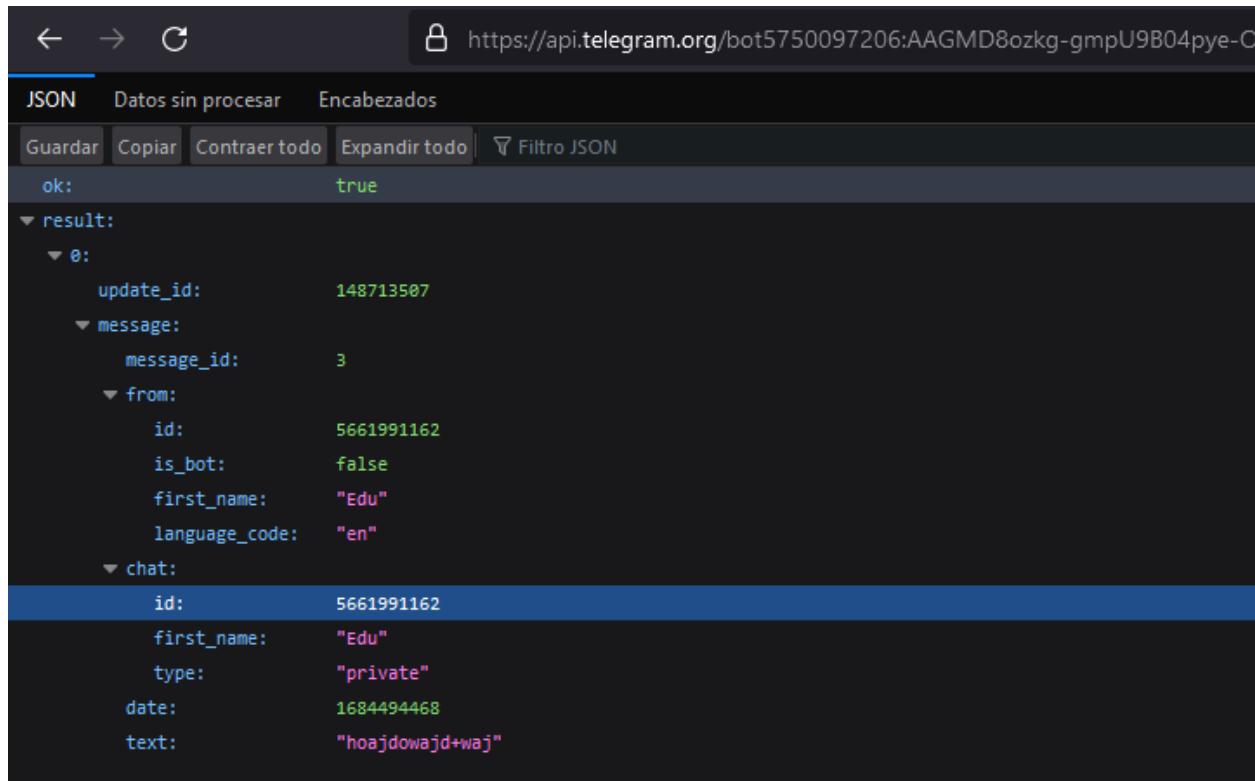
**Token:5750097206:AAGMD8ozkg-gmpU9B04pye-OMyPyBeerU6c**

Le hablaremos al bot con un mensaje de prueba



Y luego se necesita averiguar la ID, para ello:

Ver de nuevo a la URL <https://api.telegram.org/bot<your-bot-token>/getUpdates> en tu navegador, reemplazando <your-bot-token> con el token de tu bot. Ahora deberías ver un mensaje en la respuesta.

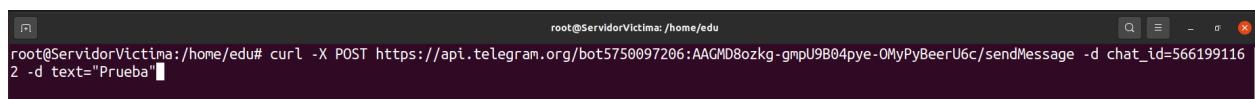


The screenshot shows a JSON response from the Telegram API. The response is a single object with the key 'ok' set to 'true'. The 'result' key contains an array of one element, which is another object representing an update. This update has an 'update\_id' of 148713507. It contains a 'message' object, which has its own 'message\_id' (3), 'from' object (with fields id, is\_bot, first\_name, language\_code), and 'chat' object (with fields id, first\_name, type, date, text). The 'text' field of the message is "hoajdowajd+waj".

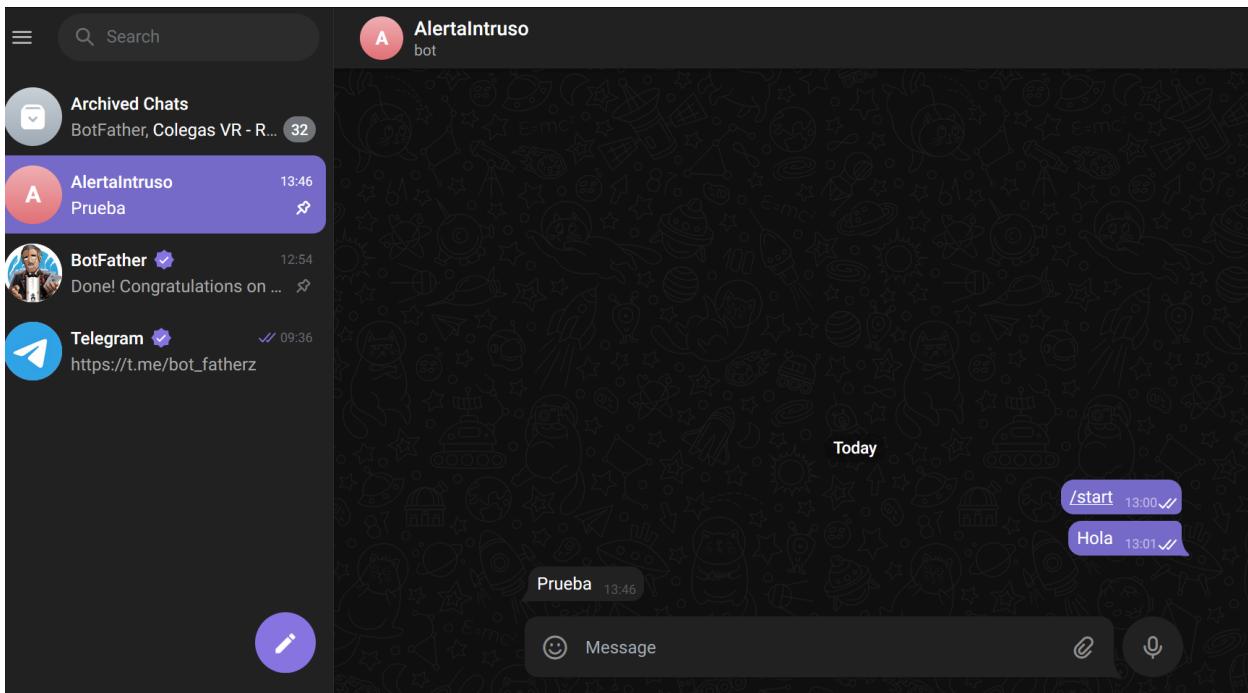
```
ok: true
{
  result: [
    {
      update_id: 148713507,
      message: {
        message_id: 3,
        from: {
          id: 5661991162,
          is_bot: false,
          first_name: "Edu",
          language_code: "en"
        },
        chat: {
          id: 5661991162,
          first_name: "Edu",
          type: "private",
          date: 1684494468,
          text: "hoajdowajd+waj"
        }
      }
    }
  ]
}
```

Para comprobar que ya se envian mensajes de linux a la terminal:

```
root@ServidorVictima:/home/edu# curl -X POST
https://api.telegram.org/bot5750097206:AAGMD8ozkg-gmpU9B04pye-OMyPyBeerU6c/send
Message -d chat_id=5661991162 -d text="Prueba"
```



The terminal window shows the command being run: 'curl -X POST https://api.telegram.org/bot5750097206:AAGMD8ozkg-gmpU9B04pye-OMyPyBeerU6c/sendMessage -d chat\_id=5661991162 -d text="Prueba"'.



Y ahora haremos el siguiente script

```

GNU nano 4.8                                         DetectorIntrusos.sh
#!/bin/bash

# Detalles del bot de Telegram
TELEGRAM_BOT_TOKEN="tu-token"
TELEGRAM_CHAT_ID="tu-id-de-chat"

# Función para enviar mensajes a Telegram
function sendMessageToTelegram () {
  curl -s -X POST "https://api.telegram.org/bot$TELEGRAM_BOT_TOKEN/sendMessage" -d chat_id=$TELEGRAM_CHAT_ID -d text="$1"
}

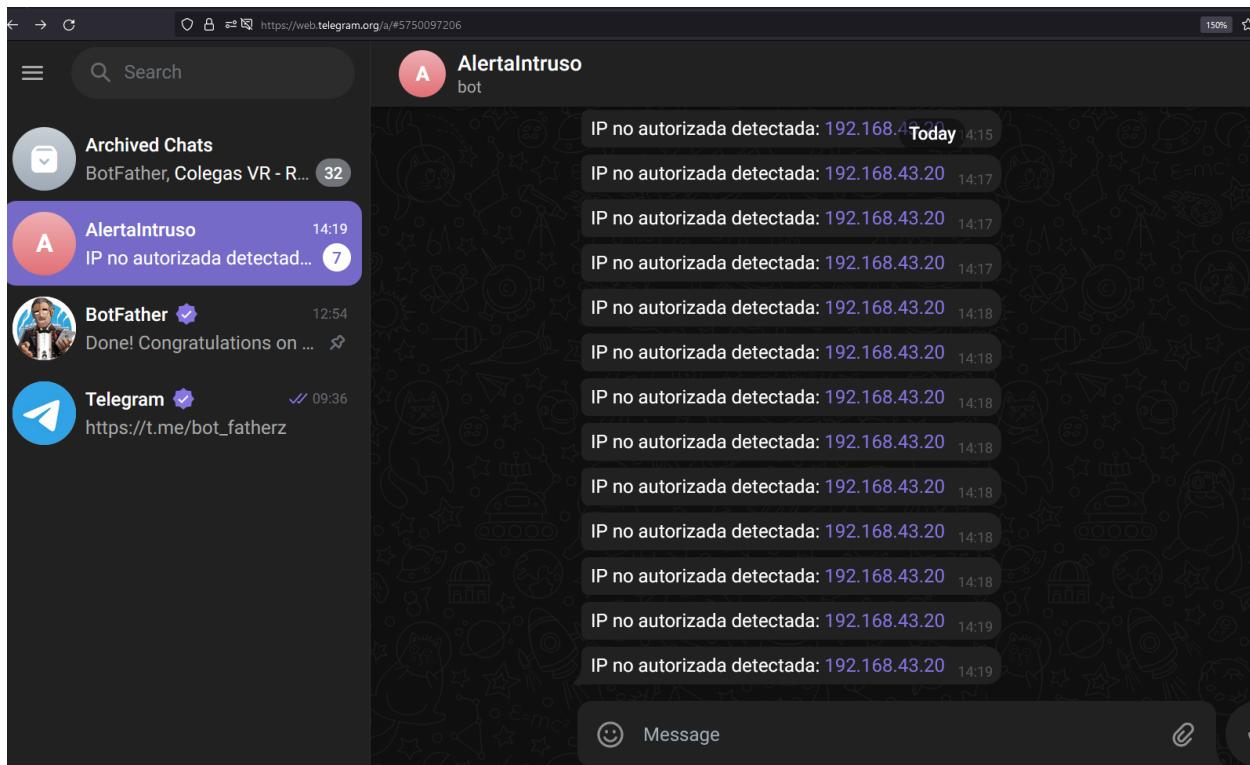
# Array con las IPs permitidas
allowed_ips=("192.168.43.100" "192.168.43.200" "192.168.43.10" "192.168.43.1" "192.168.43.58")

while true; do
  # Obtiene las IPs activas en la red
  active_ips=$(nmap -sn 192.168.43.1-255 -oG - | awk '/Up$/{print $2}')

  for ip in $active_ips; do
    # Comprueba si la IP está en la lista de IPs permitidas
    if ! printf '%s\n' "${allowed_ips[@]}" | grep -q -P "^$ip$"; then
      # Si no está en la lista, envía un mensaje a Telegram
      sendMessageToTelegram "IP no autorizada detectada: $ip"
    fi
  done

  # Espera 5 segundos antes de escanear de nuevo
  sleep 5
done

```



- ANEXO II. Partes de código/Ficheros de configuración importantes.

Aquí está todo el código de los scripts utilizados para el proyecto.

#### **Script “menu.sh” (Este script es el menú que ofrece las opciones)**

```
#!/bin/bash

# texto del banner
mensaje="Centro de Ciberseguridad"

# crear el banner con toilet
banner=$(echo -e "$mensaje" | toilet -f future -F border -F crop -F gay)

# establecer el color verde
banner="\e[32m$banner\e[0m"

# mostrar el banner
echo -e "$banner"

# menu de opciones
while :
do
    echo "Seleccione una opción:"
    echo "1. Instalación de paquetes y herramientas necesarias"
    echo "2. Barrido de red"
    echo "3. Activar Cortafuegos"
    echo "4. Desactivar Cortafuegos"
    echo "5. Detección de intrusos"
    echo "6. Salir"

    read -r opcion

    case $opcion in
        1)
            ./instalar_herramientas.sh
            ;;
        2)
            ./barrido_de_red.sh
            ;;
        3)
            ./activar_cortafuegos.sh
            ;;
    esac
done
```

```

4)
./desactivar_cortafuegos.sh
;;
5)
./DetectorIntrusos.sh
;;
6)
echo "Saliendo..."
exit 0
;;
*)
echo "Opción desconocida: $opcion"
;;
esac
done

```

#### **Script “instalar\_herramientas.sh”**

```

#!/bin/bash

read -p "¿Desea instalar las herramientas necesarias? (s/n) " respuesta

if [[ $respuesta == 's' || $respuesta == 'S' ]]; then
    sudo apt update
    sudo apt install -y nmap net-tools toilet
fi

echo "Volviendo al menú principal..."
exec ./menu.sh

```

### **Script “barrido\_de\_red.sh”**

```
#!/bin/bash

echo "Haciendo barrido de red con nmap..."

# Mi red es 192.168.1.0/24
result=$(nmap -sn 192.168.43.255/24)

echo "$result"

read -p "¿Desea guardar los resultados en un archivo .txt? (s/n) " guardar
if [[ $guardar == 's' || $guardar == 'S' ]]; then
    echo "$result" > barrido.txt
    echo "Resultados guardados en barrido.txt."
fi

echo "Volviendo al menú principal..."
exec ./menu.sh
```

### **Script “activar\_cortafuegos.sh”**

```
#!/bin/bash

# Crear el mensaje con toilet
mensaje="Cortafuegos Activado"
banner=$(echo -e "$mensaje" | toilet -f future -F border -F crop)"

# Establecer el color del texto a rojo
tput setaf 1

# mostrar el mensaje
echo -e "$banner"

# Resetear el color del texto
tput sgr0
```

```
# Limpiar todas las reglas existentes
sudo iptables -F

# Configurar políticas por defecto para rechazar todo el tráfico entrante y saliente
sudo iptables -P INPUT DROP
sudo iptables -P OUTPUT DROP
sudo iptables -P FORWARD DROP

# Permitir todo el tráfico en la interfaz de loopback
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT

# Permitir conexiones existentes y conexiones relacionadas
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Permitir todo el tráfico saliente
sudo iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Permitir tráfico SSH entrante
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT

# Guardar las reglas para que persistan después de un reinicio
sudo sh -c 'iptables-save > /etc/iptables/rules.v4'

echo "Cortafuego activado y reglas de iptables configuradas."

read -p "Presione ENTER para continuar..."
```

#### Script “desactivar\_cortafuegos.sh”

```
#!/bin/bash
```

```

# Crear el mensaje con toilet
mensaje="Cortafuegos Desactivado"
banner=$(echo -e "$mensaje" | toilet -f future -F border -F crop)"

# Establecer el color del texto a verde
tput setaf 2

# mostrar el mensaje
echo -e "$banner"

# Resetear el color del texto
tput sgr0

# Limpiar todas las reglas existentes
sudo iptables -F

# Establecer políticas por defecto para aceptar todo el tráfico entrante, saliente y forward
sudo iptables -P INPUT ACCEPT
sudo iptables -P OUTPUT ACCEPT
sudo iptables -P FORWARD ACCEPT

# Guardar las reglas para que persistan después de un reinicio
sudo sh -c 'iptables-save > /etc/iptables/rules.v4'

echo "Cortafuego desactivado y reglas de iptables configuradas a los valores predeterminados."

read -p "Presione ENTER para continuar..."

```

#### **Script “DetectorIntrusos.sh”**

```

#!/bin/bash

# Detalles del bot de Telegram
TELEGRAM_BOT_TOKEN="tu-token"

```

```

TELEGRAM_CHAT_ID="tu-id-de-chat"

# Función para enviar mensajes a Telegram
function sendMessageToTelegram () {
    curl -s -X POST
    "https://api.telegram.org/bot5750097206:AAGMD8ozkg-gmpU9B04pye-OMyPyBeerU6c/
    sendMessage" -d chat_id=5661991162 -d text="$1"
}

# Array con las IPs permitidas
allowed_ips=("192.168.43.100" "192.168.43.200" "192.168.43.10" "192.168.43.1"
"192.168.43.58")

# Iniciar un contador
counter=0

while ((counter < 30)); do
    # Obtiene las IPs activas en la red
    active_ips=$(nmap -sn 192.168.43.1-255 -oG - | awk '/Up$/{print $2}')

    for ip in $active_ips; do
        # Comprueba si la IP está en la lista de IPs permitidas
        if ! printf '%s\n' "${allowed_ips[@]}" | grep -q -P "^$ip$"; then
            # Si no está en la lista, envía un mensaje a Telegram
            sendMessageToTelegram "IP no autorizada detectada: $ip"
        fi
    done

    # Espera 5 segundos antes de escanear de nuevo
    sleep 5

    # Incrementa el contador en 5 (ya que estamos durmiendo durante 5 segundos)
    ((counter+=5))
done

# Ejecuta otro script
./menu.sh

```