



**Universidad Estatal de Cuenca**

FACULTAD DE INGENIERÍA

# PENTESTING

*Seguridad en Redes*

Autores:

Gomez Franklin Gomez Mauricio

Tenesaca Illares Juan Diego

Ortega Zárate Lenkn René

30 de enero de 2023

## CONTENTS

<b>I</b>	<b>Marco teórico</b>	2
I-A	Metasploit	2
I-B	Recolección de información:	2
I-C	Análisis básicos de vulnerabilidad	2
I-D	<i>Pentesting</i>	2
I-D1	Obtención de información	3
I-E	Vulnerabilidades Cibernéticas	3
I-E1	Tipos de vulnerabilidades	3
I-F	Máquina Virtual <i>Metasploitable</i>	3
I-G	Kali Linux	3
<b>II</b>	<b>Configuración y Desarrollo</b>	4
II-A	Descarga y configuración del <i>framework</i> Daniz-Red	4
II-B	Descarga y configuración de máquina virtual <i>Metasploitable</i>	5
II-C	<i>Pentesting</i> a <i>Metasploitable</i>	6
II-C1	Resultados del <i>pentesting</i> a <i>Metasploitable</i>	7
II-D	<i>Pentesting</i> a la <i>Maquina con Windows 10</i>	9
II-D1	Resultado del <i>pentesting</i> a Windows 10	12
II-E	<i>Pentesting</i> a una maquina con Windows 7 home basic	13
II-E1	Resultado del <i>pentesting</i> a Windows 7 Home Basic	16
II-E2	Resultado del <i>pentesting</i> a Windows XP	17
<b>III</b>	<b>Conclusiones</b>	19
	<b>References</b>	19

# Seguridad en Redes

## Examen Final

### Pentesting

Franklin Gómez<sup>1</sup>, Juan Tenesaca<sup>2</sup>, Lenin Ortega<sup>3</sup>

#### Abstract

Las áreas de la informática y cibernética se lleva actualizando día con día, esto conlleva a que existan diferentes tipos de personas que manipulan la información obtenida de diferentes maneras, ya sea para un bien común o intereses personales. Algo que se maneja hoy en día para hacer testeos ya sea a compañías, empresas, maquinas personales, etc es el *test de penetracion*, creado con el fin de frenar a las personas con intenciones de aprovechar de diferentes vulnerabilidades y atacarlas. Este test ayuda a que las vulnerabilidades presentes en un equipo sean explotadas para su posterior arreglo y "parchado". En este documento se realiza el testeo de penetración en distintas maquinas con distintos sistemas operativos, con el objetivo de probar su seguridad. En el documento se encuentra explicado el como se encuentran las vulnerabilidades y su posterior explotación de cada una de estas maquinas.

#### Index Terms

*Pentesting*, Kali Linux, Docker, Vulnerabilidad Cibernética, VirtualBox.

#### I. MARCO TEÓRICO

##### A. Metasploit

Es una herramienta la cual valida vulnerabilidades y explotación, tiene como principio dividir el flujo de trabajo de las pruebas de penetración en tareas mas pequeñas para manejarlos de una manera mas rápida, brinda las herramientas necesarias para realizar la fase de prueba manual de una prueba de penetración. Se puede usar Metasploit Pro para buscar puertos y servicios abiertos, explotar vulnerabilidades, avanzar más en una red, recopilar evidencia y crear un informe de los resultados de la prueba. [1] Es una maquina virtual la cual cumple la función de ser un equipo vulnerable con el fin de realizar pruebas de seguridad. Es usado para la seguridad informática para explotación de redes, desarrollo de *exploits*, pruebas software, entre otros.

##### B. Recolección de información:

El primer paso de una intrusión es siempre recoger información inicial. Esta obtención de información se denomina en el mundo anglosajón *footprinting*, y hay libros enteros sobre técnicas para realizarla. Básicamente se realiza una búsqueda de las vulnerabilidades que se tiene en una maquina y se en listan esos problemas.

##### C. Análisis básicos de vulnerabilidad

En este punto se refiere a que se va a introducir una serie de técnicas básicas para analizar vulnerabilidades, es importante tener en cuenta de vulnerabilidad depende de gran medida del uso que se supone que se quiere dar a un sistema.

##### D. Pentesting

En los últimos años el número de ciberataques han aumentado significativamente debido a la proliferación de virus, *malwares* y a la creación de nuevas técnicas cada vez más sofisticadas. Ya que, según estudios recientes [2], no todas las organizaciones son conscientes de los riesgos a los que se enfrentan y por eso las brechas de seguridad siguen creciendo. Para evitar los ciberataques y proteger los sistemas, es clave que la ciberseguridad avance a la misma velocidad que lo hacen las nuevas tecnologías y aquí es donde entra en juego el *Pentesting* y el trabajo del *Pentester*.

<sup>1,2,3</sup>Estudiantes de la Escuela de Telecomunicaciones; Facultad de Ingeniería de la Universidad de Cuenca, Av. 12 de abril, ECO 010112, Ecuador  
franklin.gomez@ucuenca.edu.ec  
@ucuenca.edu.ec

1) *Obtención de información:* El *Pentesting* o también llamado test de penetración está diseñado para determinar el alcance de los fallos de seguridad de un sistema. Asimismo, es una de las practicas más demandadas actualmente ya que gracias a estos test [3], una empresa puede llegar a saber a qué peligros está expuesta y cuál es el nivel de eficiencia de sus defensas.

#### E. Vulnerabilidades Cibernéticas

En la actualidad diferentes tipos de organizaciones utilizan Tecnologías de la Información y las Comunicaciones TIC y Sistemas de Automatización y Control Industrial SACI , como herramientas importantes para el crecimiento y evolución de las mismas, permitiendo eliminar barreras para realizar negocios, [3] aumentando las ventas, mejorando el servicio al cliente, sus operaciones y procesos. Al igual que el mundo real, el mundo cibernético también presenta amenazas, vulnerabilidades, riesgos, medidas, entre otros; donde comparten la misma lógica de operación y objetivos.

1) *Tipos de vulnerabilidades :*

- **Buffer overflow o desbordamiento de buffer**

Se da cuando las aplicaciones no controlan la cantidad de datos que copian en el buffer y que al sobrepasar el tamaño de este pueden modificar zonas de memoria contiguas afectando a los datos que albergan [1].

- **Condición de carrera**

Las aplicaciones o sistemas no implementan exclusiones mutuas en el acceso a recursos compartidos, como por ejemplo una variable, y varios procesos acceden a ella al mismo tiempo obteniendo valores no esperados.

- **Error de formato en cadenas**

Cuando las aplicaciones no validan los datos de entrada que introduce el usuario a las mismas, pudiendo ejecutar por ejemplo comandos o instrucciones que pueden permitir al atacante obtener datos confidenciales o dañar el sistema.

- **Cross Site Scripting**

Se basa en que los atacantes incrustan *scripts* en páginas web legítimas afectadas por esta vulnerabilidad y por las que navega el usuario. Este introduce datos como por ejemplo, su usuario y su contraseña, pero no en la web legítima si no en la del atacante, que roba así sus datos.

- **Inyección de SQL**

Cuando no se validan los datos de entrada a formularios que se comunican con bases de datos se podría ejecutar código SQL malicioso que por ejemplo permitiera obtener datos confidenciales o corromper los datos de las tablas.

#### F. Máquina Virtual Metasploitable

Se trata de un proyecto de código abierto orientado en la seguridad informática. Sirve para detectar vulnerabilidades y obtener información sobre cómo solucionar problemas que puedan aparecer. Este proyecto tiene ya bastantes años y con el paso del tiempo ha ido mejorando e incorporando nuevas características [2]. Esta máquina virtual preconfigurada permite depurar fallos y usar diferentes herramientas, como por ejemplo Metasploit. La opción más aconsejable es utilizar Metasploitable 3, que es la más nueva y la que va a funcionar mejor. En ella podrás probar tus habilidades en ciberseguridad y detectar vulnerabilidades.

#### G. Kali Linux

Kali Linux está basado en Debian GNU/Linux y fue desarrollado por la compañía de ciberseguridad Offensive Security. Kali es un sistema operativo de código abierto y se diferencia de otras distribuciones de sistemas operativos en cuanto a que reúne más de 600 programas para hacking ético, que se encuentran preinstalados en el sistema [3]. Estas herramientas, de las que hablaremos más adelante, se dividen en los siguientes trece módulos.

- Recopilación de información
- Análisis de vulnerabilidades
- Análisis de aplicaciones web
- Evaluación de bases de datos
- Ataques de contraseñas
- Ataques Wireless
- Ingeniería inversa
- Herramientas de explotación
- Sniffing and Spoofing
- Postexplotación
- Análisis forense
- Herramientas de reporte
- Herramientas de ingeniería social

## II. CONFIGURACIÓN Y DESARROLLO

### A. Descarga y configuración del framework Daniz-Red

A diferencia de las prácticas realizadas a lo largo de este ciclo, para la realización de este trabajo se empleó una máquina virtual con Ubuntu 20, esto debido a que el *framework* con el que se va a trabajar ha sido desarrollado para esta versión de Ubuntu y no se garantiza su correcto funcionamiento en otras versiones.

En esta ocasión la carpeta con el *framework* se nos fue suministrada por el docente, en dicha carpeta se encuentra todo lo necesario para su funcionamiento, incluido un archivo *docker-compose*, que es el encargado de crear el contenedor que servirá para realizar el *pentesting*.

Para construir el contenedor basta con ingresar el siguiente comando:

```
sudo docker-compose build
```

Antes de poder ejecutar el contenedor es necesario configurar ciertos parámetros, como el nombre de la interfaz de red, la dirección IP de la víctima y la dirección de correo electrónico a la que se enviará el reporte.

Los archivos a configurar son *config.txt* y *config.json*, dependiendo de cuál se vaya a usar, en este caso se empleó *config.json*. Adicionalmente, este *framework* cuenta con un modo de funcionamiento automático el cual empieza a ejecutarse al momento de levantar el contenedor. Para que este modo automático funcione correctamente se debe configurar el archivo *crontrab*, en el cual se establecerá el momento en el que se enviará el reporte.

En las figuras 1 y 2 se muestra la edición de los archivos de configuración.

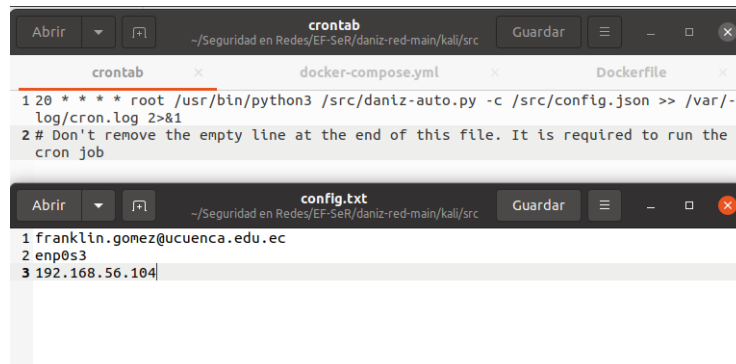


Fig. 1: Archivos de configuración

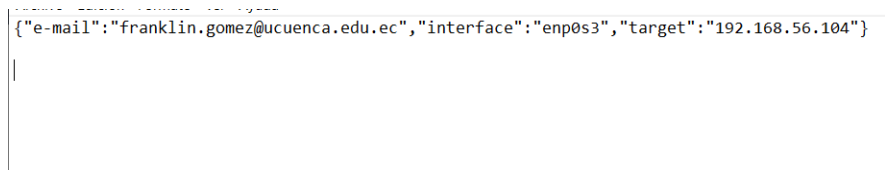


Fig. 2: Archivos de configuración

Una vez realizadas las configuraciones mencionadas anteriormente se levanta el contenedor con el siguiente comando:

```
sudo docker-compose up -d kali
```

En la figura 3 se muestra el levantamiento del *docker* y verificación de funcionamiento.

```
franklin@FG: ~/Seguridad en Redes/EF-SeR x franklin@FG: ~/Seguridad en Redes/EF-SeR/daniz-red... x
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ sudo docker-compose up -d kali
WARNING: Some networks were defined but are not used by any service: network
Starting kali ... done
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
9820b5b8191f   kali      "/bin/sh -c 'cron &&...'  About a minute ago    Up 4 seconds    Up 4 seconds    kali
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$
```

Fig. 3: Levantamiento de contenedor y verificación de ejecución

A partir de la ejecución del *docker* empieza a funcionar el modo automático, sin embargo, también existe una forma de correr el test de forma manual, para ello basta con ejecutar el comando mostrado a continuación.

```
sudo docker exec kali /usr/bin/python3 /src/daniz-auto.py -c /src/config.json
```

### B. Descarga y configuración de máquina virtual Metasploitable

Debido a que el objetivo de este trabajo es el poder detectar vulnerabilidades y encontrar errores que puedan ser un peligro para los dispositivos, se empleó *Metasploitable*, la cual es una máquina virtual que cuenta con varias vulnerabilidades, lo que permite realizar experimentos y aprender, sin correr ningún riesgo real.

El procedimiento para emplear esta máquina virtual es muy similar al de cualquier otro sistema operativo, primero se consigue la imagen ISO del SO y luego se configura los parámetros de VirtualBox para alojar dicha máquina. En la figura 4 se muestra la finalización de la instalación.

En esta máquina, tanto el usuario como la contraseña de la máquina son *msfadmin*.

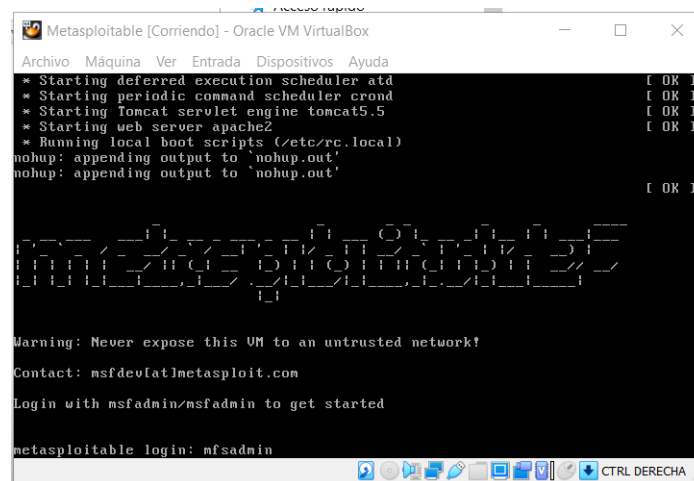


Fig. 4: Instalación de Metasploitable

Para que esta máquina virtual pueda ser alcanzable por la máquina que ejecutará el *framework* se cambió su configuración de red a **Adaptador puente**.

Al hacerlo, la máquina *metasploitable* también podrá ser accesible desde cualquier navegador, como se muestra en la figura 6

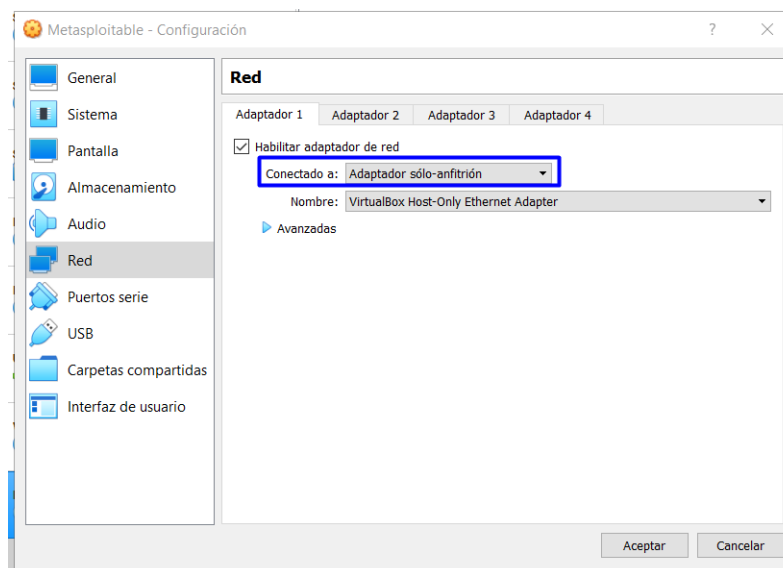


Fig. 5: Configuración de red como adaptador puente

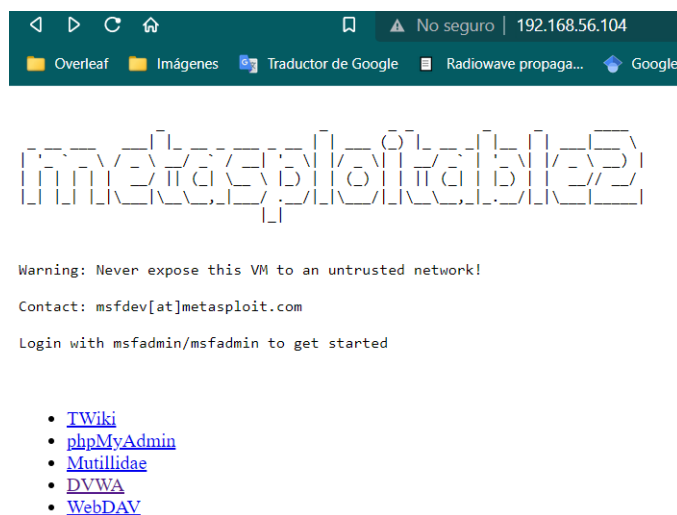


Fig. 6: Acceso a *Metasploitable* desde el navegador

### C. Pentesting a Metasploitable

La primera prueba de *pentesting* se realizó a la máquina virtual con *metasploitable*, esta máquina tiene la dirección IP *192.168.56.104*.

Para realizar el test se ingresó el siguiente comando:

```
sudo docker exec kali /usr/bin/python3 /src/daniz-auto.py -c /src/config.txt
```

En la figura 7 se muestra el inicio de la prueba en modo manual, mientras que en la figura 8 se muestra la finalización del mismo.

```

franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ sudo docker exec kali /usr/bin/python3 /src
/daniz-auto.py -c /src/config.txt
[sudo] contraseña para franklin:
mkdir: cannot create directory '/src/archivos/': File exists
mkdir: cannot create directory '/src/archivos/hosts/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_1/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_2/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_3/': File exists
mkdir: cannot create directory '/src/archivos/hosts/explotacion/': File exists
mkdir: cannot create directory '/src/Reportes/': File exists
mkdir: cannot create directory '/src/Reportes/Ataque/': File exists
mkdir: cannot create directory '/src/Reportes/Escaneo/': File exists
mkdir: cannot create directory '/src/Reporte_final/': File exists
rm: cannot remove '/src/archivos/hosts/fase_3/*': No such file or directory
rm: cannot remove '/src/archivos/hosts/explotacion/*': No such file or directory
rm: cannot remove '/src/archivos/hosts/fase_1/*': No such file or directory
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 01:38 UTC
Nmap scan report for 192.168.56.104
Host is up (0.00040s latency).
MAC Address: 08:00:27:F2:C3:90 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
cant execute
Traceback (most recent call last):
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 53, in <module>
    main()
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 46, in main
    ip_scan(parse.rango,parse.filepath) #pasamos la opcion q pase el usuario en la linea de comando
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 43, in ip_scan
    f.close()
UnboundLocalError: local variable 'f' referenced before assignment
rm: cannot remove '/src/archivos/hosts/fase_2/*': No such file or directory
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 01:38 UTC

```

Fig. 7: Inicio manual del test

```

ips selected: []
(1)--192.168.56.104
(a)--Continue with selected ips
(b)--Automatic--(All the ips)--
(q)--Go back
-----
Info:
programas selected: []
(1)--vunlnmaptcp.py
(2)--nuclei.py
(3)--vulners_nmap_serv.py
(a)--Continue with selected programas
(b)--Automatic--(All the programas)--
(q)--Go back
-----
python3 /src/modules/hosts/fase_3/vunlnmaptcp.py -o 192.168.56.104 -f /src/archivos/hosts/fase_3/19
2.168.56.104vunlnmaptcp
python3 /src/modules/hosts/fase_3/nuclei.py -o 192.168.56.104 -f /src/archivos/hosts/fase_3/192.168
.56.104nuclei
python3 /src/modules/hosts/fase_3/vulners_nmap_serv.py -o 192.168.56.104 -f /src/archivos/hosts/fas
e_3/192.168.56.104vulners_nmap_serv

At this point the information about vulnerabilities in the objectives
was obtained.

The third part of the audit are finished
192.168.56.104vunlnmaptcp.gnmap
192.168.56.104vulners_nmap_serv.nmap
192.168.56.104vulners_nmap_serv.xml
192.168.56.104vunlnmaptcp.nmap
192.168.56.104vunlnmaptcp.xml
192.168.56.104vulners_nmap_serv.gnmap
[]
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$

```

Fig. 8: Finalización del testeo

1) *Resultados del pentesting a Metasploitable*: Al terminar el testeo mostrado anteriormente, se genera un reporte en formato PDF que muestra todos los puertos abiertos y todas las vulnerabilidades encontradas en la máquina objetivo, en la figura 9 se muestra como el *framework* ha obtenido correctamente la dirección MAC de la máquina *metasploitable* a partir de la dirección IP que se proporcionó al programa.

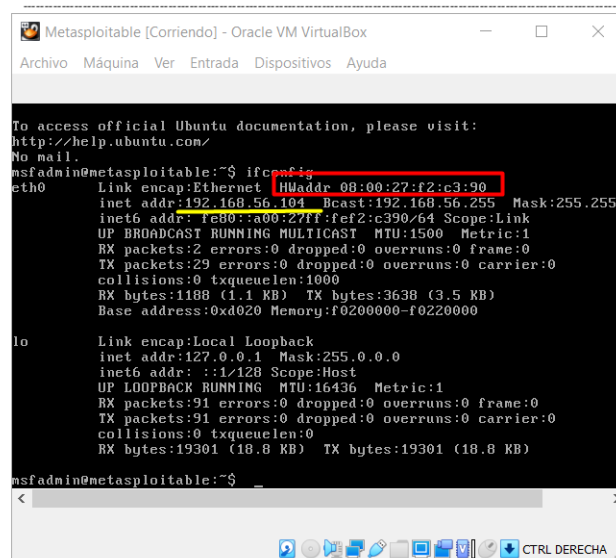
Teniendo así: **IP:** 192.168.56.104 **MAC:** 08:00:27:F2:C3:90.

Mientras que en la figura 10 se muestra dos páginas del reporte generado, donde se muestran los puertos abiertos encontrados y algunas de las vulnerabilidades encontradas, estos datos se resumen en las tablas I y II.



#### Hosts activos

Dirección IP: 192.168.56.104 - Dirección MAC: 08:00:27:F2:C3:90



```
msfadmin@metasploitable:~$ ifconfig
eth0:
  Link encap:Ethernet  HWaddr 08:00:27:F2:C3:90
    inet addr:192.168.56.104  Bcast:192.168.56.255  Mask:255.255.255
    inet6 addr: fe80::a00:27ff:fe2c:390/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:2 errors:0 dropped:0 overruns:0 frame:0
    TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1188 (1.1 KB)  TX bytes:3638 (3.5 KB)
    Base address:0xd020  Memory:f0200000-f0220000

lo:
  Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:16436  Metric:1
    RX packets:91 errors:0 dropped:0 overruns:0 frame:0
    TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

Fig. 9: Obtención de la dirección MAC de la máquina víctima

#### 1.2\_Escaneo-de-puertos-sin-utilizar-ICMP

Dirección IP:  
### 192.168.56.104

Port	State	Service
1099/tcp	open	rmiregistry
111/tcp	open	rpcbind
139/tcp	open	netbios-ssn
1524/tcp	open	ingreslock
2049/tcp	open	nfs
21/tcp	open	ftp
2121/tcp	open	ccproxy-ftp
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp
3306/tcp	open	mysql
445/tcp	open	microsoft-ds
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
53/tcp	open	domain
5432/tcp	open	postgresql
5900/tcp	open	vnc
6000/tcp	open	X11
6667/tcp	open	irc
80/tcp	open	http
8009/tcp	open	altp13
8180/tcp	open	unknown

#### 1.3\_Vulnerabilidades-mediante-puertos-TCP

Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 02:00:16 2023 as: nmap -v -sS --script=vuln
--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl -oA /src/archivos/hosts/fase_3/192.168.56.104vulnmaptcp
192.168.56.104
Nmap scan report for 192.168.56.104
Host is up (0.00029s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vstpd-backdoor:
| VULNERABLE:
| vsFTPD version 2.3.4 backdoor
| State: VULNERABLE (Exploitable)
| IDs: CVE:CVE-2011-2523 BID:48539
| vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
| Disclosure date: 2011-07-03
| Exploit results:
| Shell command: id
| Results: uid=0(root) gid=0(root)
| References:
| https://www.securityfocus.com/bid/48539
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
| https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vstpd_234_backdoor.rb
| http://scarybeastsecurity.blogspot.com/2011/07/alert-vstpd-download-backdoored.html
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
| ssl-poodle:
| VULNERABLE:
| SSL POODLE information leak
| State: VULNERABLE
| IDs: CVE:CVE-2014-3566 BID:70574
| The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
| products, uses nondeterministic CBC padding, which makes it easier
| for man-in-the-middle attackers to obtain cleartext data via a
| padding-oracle attack, aka the "POODLE" issue.
| Disclosure date: 2014-10-14
| Check results:
| TLS_RSA_WITH_AES_128_CBC_SHA
```

Fig. 10: Puertos abiertos y vulnerabilidades encontradas

En la tabla II se muestran el identificador de la vulnerabilidad, su grado de peligro medido con el estándar CVSS ver3 y el grado de impacto sobre la máquina que tienen las vulnerabilidades. Adicionalmente, en la figura 11 se ilustra el impacto de las vulnerabilidades mediante un diagrama pastel.

TABLE I: Puertos y servicios encontrados

Puerto	Estado	Servicio
1099/tcp	abierto	rmiregistry
111/tcp	abierto	rpcbind
139/tcp	abierto	netbios-ssn
1524/tcp	abierto	ingreslock
1524/tcp	abierto	ingreslock
2049/tcp	abierto	nfs
21/tcp	abierto	ftp
2121/tcp	abierto	ccproxy-ftp
22/tcp	abierto	ssh
23/tcp	abierto	telnet
25/tcp	abierto	smtp
3306/tcp	abierto	mysql
445/tcp	abierto	microsoft-ds
512/tcp	abierto	exec
513/tcp	abierto	login
514/tcp	abierto	shell
53/tcp	abierto	domain
5432/tcp	abierto	postgresql
5432/tcp	abierto	vnc
5432/tcp	abierto	X11
5432/tcp	abierto	irc
5432/tcp	abierto	http
5432/tcp	abierto	ajp13
5432/tcp	abierto	unknown

TABLE II: Vulnerabilidad e impacto

Vulnerabilidad	CVss 3	Impacto
CVE-2011-2523	7.5	alto
CVE-2014-3566	3.4	bajo
CVE-2015-4000	3.7	bajo
CVE-2007-6750	NVD	NVD

Vulnerabilidades y su impacto

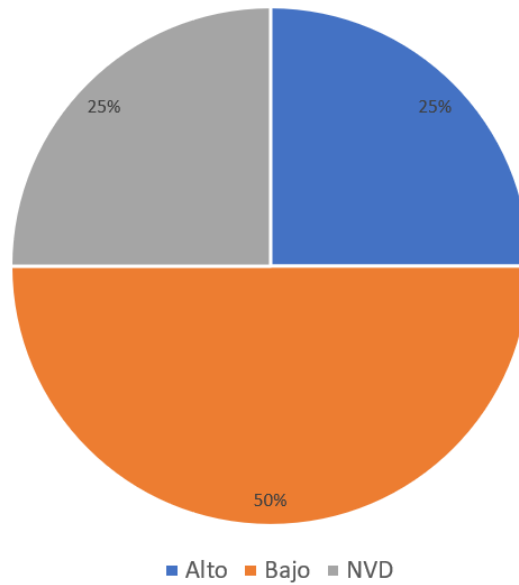


Fig. 11: Vulnerabilidades encontradas y su impacto

#### D. Pentesting a la Maquina con Windows 10

Para esto se usara la dirección IP de la PC.

```

Adaptador de Ethernet VirtualBox Host-Only Network:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . . : fe80::c4f0:7b20:63bf:f96e%17
Dirección IPv4. . . . . : 192.168.56.1
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . :

```

Fig. 12: Ip de la maquina con windows 10

Ahora se produce a realizar los cambios en los archivos *config.txt* y *config.json*, donde se coloca la dirección a la cual se va a realizar el test.



```

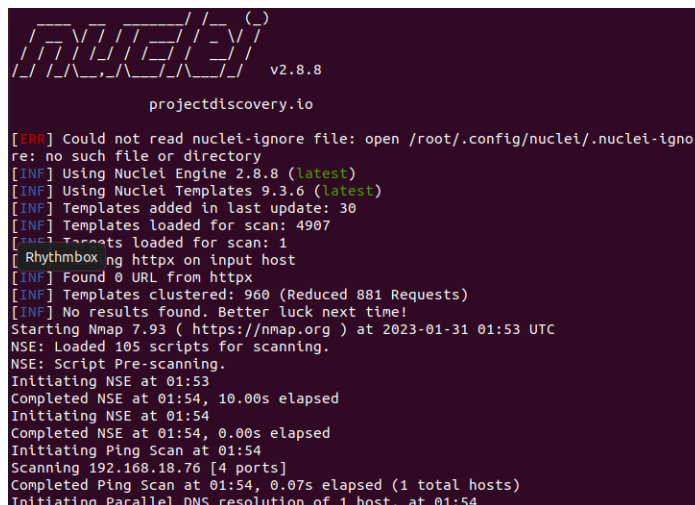
1- 1- "juan.tenesaca@ucuenca.edu.ec", "interface": "enp0s3", "target": "192.168.56.1"
2-

```

Fig. 13: Ip de la maquina con windows 10

Seguidamente se realiza la ejecución manual del test con el siguiente comando :

```
sudo docker exec kali /usr/bin/python3 /src/daniz-auto.py -c /src/config.json
```



```

projectdiscovery.io
v2.8.8

[ERR] Could not read nuclei-ignore file: open /root/.config/nuclei/.nuclei-ignore: no such file or directory
[INF] Using Nuclei Engine 2.8.8 (latest)
[INF] Using Nuclei Templates 9.3.6 (latest)
[INF] Templates added in last update: 30
[INF] Templates loaded for scan: 4907
[INF] Targets loaded for scan: 1
[INF] Rhythmbbox httpx on input host
[INF] Found 0 URL from httpx
[INF] Templates clustered: 960 (Reduced 881 Requests)
[INF] No results found. Better luck next time!
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 01:53 UTC
NSE: Loaded 105 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 01:53
Completed NSE at 01:54, 10.00s elapsed
Initiating NSE at 01:54
Completed NSE at 01:54, 0.00s elapsed
Initiating Ping Scan at 01:54
Scanning 192.168.18.76 [4 ports]
Completed Ping Scan at 01:54, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:54

```

Fig. 14: Testeo manual

```

Completed Ping Scan at 01:54, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:54
Completed Parallel DNS resolution of 1 host. at 01:54, 0.01s elapsed
Initiating SYN Stealth Scan at 01:54
Scanning 192.168.18.76 [1000 ports]
Completed SYN Stealth Scan at 01:54, 4.45s elapsed (1000 total ports)
NSE: Script scanning 192.168.18.76.
Initiating NSE at 01:54
Completed NSE at 01:54, 1.00s elapsed
Initiating NSE at 01:54
Completed NSE at 01:54, 0.00s elapsed
Nmap scan report for 192.168.18.76
Host is up (0.0012s latency).
All 1000 scanned ports on 192.168.18.76 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

NSE: Script Post-scanning.
Initiating NSE at 01:54
Completed NSE at 01:54, 0.00s elapsed
Initiating NSE at 01:54
Completed NSE at 01:54, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 16.33 seconds
Raw packets sent: 2007 (88.272KB) | Rcvd: 4 (160B)
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 01:54 UTC
Nmap scan report for 192.168.18.76
Host is up (0.045s latency).
All 1000 scanned ports on 192.168.18.76 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

```

Fig. 15: Testeo manual

```

***** Starting analysis *****
This framework can perform the following audits:

{1}--Hosts
{2}--Web pages
{3}--Network
{99}--Go back

Info:
programs selected: []
{1}--scanner_de_red.py
{2}--theharve.py
{3}--upnmap.py
{a}--Continue with selected programs
{b}--Automatic--(All the programs)--
{q}--Go back
python3 /src/modules/hosts/fase_1/scanner_de_red.py -o 192.168.56.1 -f /src/archivos/hosts/fase_1/scanner_de_red
python3 /src/modules/hosts/fase_1/theharve.py -o 192.168.56.1 -f /src/archivos/hosts/fase_1/theharve
python3 /src/modules/hosts/fase_1/upnmap.py -o 192.168.56.1 -f /src/archivos/hosts/fase_1/upnmap
['upnmap.xml']
Info:

```

Fig. 16: Testeo manual

```

Using existing ports in the ip addresses the better objectives can be de
termined.
Cliente de correo Thunderbird
Info:
Ip: 192.168.56.1 Mac: No se pudo obtener puertos =[' ']
Ip: 192.168.18.76 Mac: No se pudo obtener puertos =[' ']

ips selected: []
{1}--192.168.56.1
{2}--192.168.18.76
{a}--Continue with selected ips
{b}--Automatic--(All the ips)--
{q}--Go back
Info:
programas selected: []
{1}--nuclei.py
{2}--vunlnmptcp.py
{3}--vulners_nmap_serv.py
{a}--Continue with selected programas
{b}--Automatic--(All the programas)--
{q}--Go back
python3 /src/modules/hosts/fase_3/nuclei.py -o 192.168.56.1 -f /src/archivos/hosts/fase_3/192.168.56.1nuclei
python3 /src/modules/hosts/fase_3/vunlnmptcp.py -o 192.168.56.1 -f /src/archivos/hosts/fase_3/192.168.56.1vunlnmptcp
python3 /src/modules/hosts/fase_3/vulners_nmap_serv.py -o 192.168.56.1 -f /src/archivos/hosts/fase_3/192.168.56.1vulners_nmap_serv

```

Fig. 17: Testeo manual

```

python3 /src/modules/hosts/fase_3/vunlnmptcp.py -o 192.168.18.76 -f /src/archivos/hosts/fase_3/192.168.18.76vunlnmptcp
P Cliente de correo Thunderbird s/fase_3/vulners_nmap_serv.py -o 192.168.18.76 -f /src/archivos/hosts/fase_3/192.168.18.76vulners_nmap_serv

At this point the information about vulnerabilities in the objectives
was obtained.

The third part of the audit are finished
192.168.56.1vulners_nmap_serv.gnmap
192.168.56.1vulners_nmap_serv.xml
192.168.56.1vunlnmptcp.nmap
192.168.18.76vunlnmptcp.gnmap
192.168.18.76vunlnmptcp.nmap
192.168.56.1vulners_nmap_serv.nmap
192.168.18.76vunlnmptcp.xml
192.168.56.1vunlnmptcp.xml
192.168.18.76vulners_nmap_serv.gnmap
192.168.18.76vulners_nmap_serv.xml
192.168.56.1vunlnmptcp.gnmap
192.168.18.76vulners_nmap_serv.nmap
[]
seguridadredes@se:~/Escritorio/Examen/daniz-red-main-old/daniz-red-main$ cd kali/src/Reporte_final/
seguridadredes@se:~/Escritorio/Examen/daniz-red-main-old/daniz-red-main/kali/src/Reporte_final$ ls
reporte-192.168.0.0:24.pdf reporte-taringa.net.pdf
reporte-192.168.0.104.pdf reporte-vuln-calculadora.pdf

```

Fig. 18: Testeo manual

Ahora se muestran los pdfs, generados por el pentesting:

### 1.3\_Vulnerabilidades-mediante-puertos-TCP

#### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 01:53:53 2023 as: nmap -v -sS --script=vuln
--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl -oA /src/archivos/hosts/fase_3/192.168.18.76vuninmactcp
192.168.18.76
Nmap scan report for 192.168.18.76
Host is up (0.0012s latency).
All 1000 scanned ports on 192.168.18.76 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
```

Read data files from: /usr/bin/.share/nmap

# Nmap done at Tue Jan 31 01:54:09 2023 -- 1 IP address (1 host up) scanned in 16.33 seconds

(end of excerpt)

### 1.2\_Obtención-de-puertos-TCP-abiertos

#### Direcciones:

IP: 192.168.56.1 - ipv4

Física: no se pudo obtener - no se pudo obtener

#### Puertos:

Los 1000 puertos escaneados pero no presentados, están en estado: filtered

Los 1000 puertos respondieron con: no-response

### 1.3\_Escaneo-de-vulnerabilidades-utilizando-servicios-obtenidos-con-NMAP

#### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 02:57:49 2023 as: nmap -sV --script=vulners --script-args mincvss=7.5
--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl -oA /src/archivos/hosts/fase_3/192.168.56.1vulners_nmap_serv
192.168.56.1
```

# Nmap done at Tue Jan 31 02:57:53 2023 -- 1 IP address (0 hosts up) scanned in 3.72 seconds

(end of excerpt)

### 1.3\_Vulnerabilidades-mediante-puertos-TCP

#### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 02:57:11 2023 as: nmap -v -sS --script=vuln
--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xsl -oA /src/archivos/hosts/fase_3/192.168.56.1vuninmactcp
192.168.56.1
```

Pre-scan script results:

| broadcast-avahi-dos:

| Discovered hosts:

| 224.0.0.251

| After NULL UDP avahi packet DoS (CVE-2011-1002).

| Hosts are all up (not vulnerable).

Nmap scan report for 192.168.56.1 [host down]

Read data files from: /usr/bin/.share/nmap

# Nmap done at Tue Jan 31 02:57:49 2023 -- 1 IP address (0 hosts up) scanned in 38.34 seconds

(end of excerpt)

### 1.2\_Obtención-de-puertos-UDP-abiertos

#### Direcciones:

IP: 192.168.56.1 - ipv4

Física: no se pudo obtener - no se pudo obtener

#### Puertos:

Los 1000 puertos escaneados pero no presentados, están en estado: open|filtered

Los 1000 puertos respondieron con: no-response

### 1.2\_Escaneo-de-puertos-sin-utilizar-ICMP

#### Dirección IP:

(end of excerpt)

Fig. 19: Resultados del test en Windows 10

1) *Resultado del pentesting a Windows 10:* Al analizar el reporte generado de la prueba de penetración de red interna se encontraron que no existen vulnerabilidades, se observó que se intentaron analizar puertos TCP, los cuales no dan respuesta y se encuentran en estado *filtered*. Tampoco se logró encontrar puertos UDP abiertos y cuando se realizó el escaneo de vulnerabilidades utilizando servicios obtenidos con NMAP tampoco se logró encontrar alguna y finalmente no existen puertos sin utilizar ICMP.

## Vulnerabilidad y su impacto

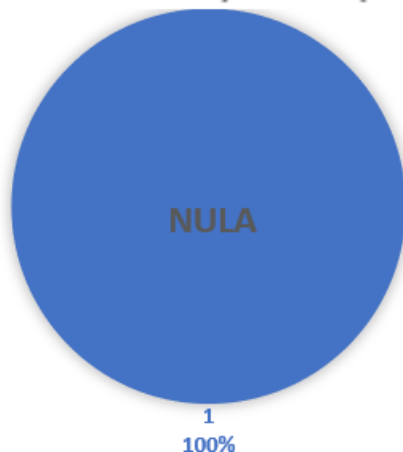


Fig. 20: Gráfica de pastel sobre la severidad de las vulnerabilidades encontradas

### E. Pentesting a una maquina con Windows 7 home basic

Se usara la dirección IP de la maquina:

```
Adaptador de Ethernet Conexión de Área local:  
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . : fe80::4c37:2267:6077:1d11  
Dirección IPv4. . . . . : 192.168.18.57  
Máscara de subred. . . . . : 255.255.255.0  
Puerta de enlace predeterminada. . . . . : 192.168.18.1  
  
Adaptador de túnel isatap.{7B22828E-2642-4AA1-AF86-67CD1E933E46}:  
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :
```

Fig. 21: IP de la maquina con windows 7

Ahora se produce a realizar los cambios en los archivos `config.txt` y `config.json`, donde se coloca la dirección a la cual se va a realizar el test

```
config.json  
~/Escritorio/Examen/daniz-red-main-old/daniz-red-main/kali/src  
1 {"e-mail": "juan.tenesaca@ucuenca.edu.ec", "interface": "enp0s3", "target": "192.168.18.57"}  
2
```

Fig. 22: Configuración del archivo `config.json`

Seguidamente se realiza la ejecución manual del test con el siguiente comando :

```
sudo docker exec kali /usr/bin/python3 /src/daniz-auto.py -c /src/config.json
```

```
projectdiscovery.io v2.8.8  
[ERR] Could not read nuclei-ignore file: open /root/.config/nuclei/.nuclei-ignore: no such file or directory  
[INF] Using Nuclei Engine 2.8.8 (latest)  
[Terminal ng Nuclei Templates 9.3.6 (latest)  
[INF] Templates added in last update: 30  
[INF] Templates loaded for scan: 4907  
[INF] Targets loaded for scan: 1  
[INF] Running httpx on input host  
[INF] Found 0 URL from httpx  
[INF] Templates clustered: 958 (Reduced 880 Requests)  
[INF] No results found. Better luck next time!  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 03:47 UTC  
NSE: Loaded 105 scripts for scanning.  
NSE: Script Pre-scanning.  
Initiating NSE at 03:47  
NSE Timing: About 40.00% done; ETC: 03:48 (0:00:48 remaining)  
Completed NSE at 03:47, 34.11s elapsed  
Initiating NSE at 03:47  
Completed NSE at 03:47, 0.00s elapsed  
Pre-scan script results:  
| broadcast-avahi-dos:  
| Discovered hosts:  
| 224.0.0.251  
| After NULL UDP avahi packet DoS (CVE-2011-1002).  
| Hosts are all up (not vulnerable).
```

Fig. 23: Testeo manual

```

| After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Initiating ARP Ping Scan at 03:47
Scanning 192.168.18.57 [1 port]
Completed ARP Ping Scan at 03:47, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:47
Completed Parallel DNS resolution of 1 host. at 03:47, 0.01s elapsed
Initiating SYN Stealth Scan at 03:47
Scanning 192.168.18.57 [1000 ports]
Discovered open port 5357/tcp on 192.168.18.57
Completed SYN Stealth Scan at 03:48, 11.79s elapsed (1000 total ports)
NSE: Script scanning 192.168.18.57.
Initiating NSE at 03:48
Completed NSE at 03:48, 1.02s elapsed
Initiating NSE at 03:48
Completed NSE at 03:48, 0.00s elapsed
Nmap scan report for 192.168.18.57
Host is up (0.00047s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
5357/tcp  open  wsddapi
MAC Address: 08:00:27:01:68:B4 (Oracle VirtualBox virtual NIC)

NSE: Script Post-scanning.
Initiating NSE at 03:48
Completed NSE at 03:48, 0.00s elapsed
Initiating NSE at 03:48
Completed NSE at 03:48, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 47.73 seconds
Raw packets sent: 2001 (88.028KB) | Rcvd: 3 (116B)

```

Fig. 24: Testeo manual

```

This framework can perform the following audits:

    {1}--Hosts
    {2}--Web pages
    {3}--Network
    {99}--Go back

Info:

programs selected: []
{1}--scanner_de_red.py
{2}--theharve.py
{3}--upnmap.py
{a}--Continue with selected programs
{b}--Automatic--(All the programs)--
{q}--Go back
python3 /src/modules/hosts/fase_1/scaner_de_red.py -o 192.168.18.57 -f /src/archivos/hosts/fase_1/scaner_de_red
python3 /src/modules/hosts/fase_1/theharve.py -o 192.168.18.57 -f /src/archivos/hosts/fase_1/theharve
python3 /src/modules/hosts/fase_1/upnmap.py -o 192.168.18.57 -f /src/archivos/hosts/fase_1/upnmap
['upnmap.xml']
Info:
unooo
ips selected: []
{1}--192.168.18.57
{a}--Continue with selected ips
{b}--Automatic--(All the ips)--
{q}--Go back
Info:

```

Fig. 25: Testeo manual

```
ip: 192.168.18.57 Mac: 08:00:27:01:68:B4 puertos =[' ']\n\nips selected: []\n[1]--192.168.18.57\n[a]--Continue with selected ips\n[b]--Automatic--(All the ips)--\n[q]--Go back\nInfo:\n\nTerminal selected: []\n[1]--nuclei.py\n[2]--vunlnmaptcp.py\n[3]--vulners_nmap_serv.py\n[a]--Continue with selected programas\n[b]--Automatic--(All the programas)--\n[q]--Go back\npython3 /src/modules/hosts/fase_3/nuclei.py -o 192.168.18.57 -f /src/archivos/hosts/fase_3/192.168.18.57\npython3 /src/modules/hosts/fase_3/vunlnmaptcp.py -o 192.168.18.57 -f /src/archivos/hosts/fase_3/192.168.18.57\npython3 /src/modules/hosts/fase_3/vulners_nmap_serv.py -o 192.168.18.57 -f /src/archivos/hosts/fase_3/192.168.18.57\n\nAt this point the information about vulnerabilities in the objectives\nwas obtained.\n\nThe third part of the audit are finished\n192.168.18.57vunlnmaptcp.xml\n192.168.18.57vulners_nmap_serv.nmap\n192.168.18.57vulners_nmap_serv.gnmap\n192.168.18.57vunlnmaptcp.nmap\n192.168.18.57vulners_nmap_serv.xml\n192.168.18.57vunlnmaptcp.gnmap
```

Fig. 26: Testeo manual

Ahora se muestran los pdfs, generados por el pentesting:

## 1.2\_Escaneo-de-puertos-sin-utilizar-ICMP

### Dirección IP:

### 192.168.18.57

Port	State	Service
5357/tcp	open	wsdapi

(end of excerpt)

## 1.2\_Obtención-de-puertos-UDP-abiertos

### Direcciones:

IP: 192.168.18.57 - ipv4  
Física: mac - 08:00:27:01:68:B4

### Puertos:

Los 999 puertos escaneados pero no presentados, están en estado: open|filtered  
Los 999 puertos respondieron con: no-response

### 192.168.18.57

Port	State	Service
5357/tcp	open	wsdapi

## 1.3\_Vulnerabilidades-mediante-puertos-TCP

### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 03:47:23 2023 as: nmap -v -sS --script=vuln\n--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xml -oA /src/archivos/hosts/fase_3/192.168.18.57vunlnmaptcp\n192.168.18.57
```

Pre-scan script results:

| broadcast-avahi-dos:  
| Discovered hosts:  
| 224.0.0.251  
| After NULL UDP avahi packet DoS (CVE-2011-1002).  
| Hosts are all up (not vulnerable).

Nmap scan report for 192.168.18.57

Host is up (0.00047s latency).

Not shown: 999 filtered tcp ports (no-response)

PORT STATE SERVICE

5357/tcp open wsdapi

MAC Address: 08:00:27:01:68:B4 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap

# Nmap done at Tue Jan 31 03:48:11 2023 -- 1 IP address (1 host up) scanned in 47.73 seconds

## 1.2\_Obtención-de-puertos-TCP-abiertos

### Direcciones:

IP: 192.168.18.57 - ipv4  
Física: mac - 08:00:27:01:68:B4

### Puertos:

Los 999 puertos escaneados pero no presentados, están en estado: filtered  
Los 999 puertos respondieron con: no-response

### 192.168.18.57

Port	State	Service
5357/tcp	open	wsdapi

## 1.3\_Escaneo-de-vulnerabilidades-utilizando-servicios-obtenidos-con-NMAP

### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 03:48:11 2023 as: nmap -sV --script=vulners --script-args mincvss=7.1\n--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xml -oA /src/archivos/hosts/fase_3/192.168.18.57vulners_nmap_serv\n192.168.18.57
```

Nmap scan report for 192.168.18.57

Host is up (0.00047s latency).

Not shown: 999 filtered tcp ports (no-response)

PORT STATE SERVICE

5357/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

\_http-server-header: Microsoft-HTTPAPI/2.0

MAC Address: 08:00:27:01:68:B4 (Oracle VirtualBox virtual NIC)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

† Nmap done at Tue Jan 31 03:48:44 2023 -- 1 IP address (1 host up) scanned in 32.94 seconds

end of excerpt)

Fig. 27: Resultados del test en Windonws 7



1) *Resultado del pentesting a Windows 7 Home Basic:* Al analizar el reporte generado de la prueba de penetración de red interna se encontraron que existen una única vulnerabilidad, en la dirección IP se observó el puerto 5357/tcp el cual está abierto y tiene el servicio de wsddapi, los 999 puertos restantes están en estado open/filtered, por otro lado en el escaneo de vulnerabilidad utilizando servicios obtenidos con NMAP se observa que la versión de servicio del puerto de estado es 5357/tcp open http Microsoft HTTPAPI. Finalmente el identificador de la vulnerabilidad es *CVE* – 2011 – 1002, este es de tipo error en la gestión de recursos, con gravedad media, permite a atacantes remotos provocar una denegación de servicio (bucle infinito) a través de un paquete UDP (1) IPv4 o (2) IPv6 vacíos al puerto 5353. NOTA: esta vulnerabilidad existe debido a una corrección incorrecta del CVE-2010-2244. [4]

TABLE III: Puertos y servicios encontrados

Puerto	Estado	Servicio
5357/tcp	abierto	wsddapi

### Vulnerabilidad y su impacto

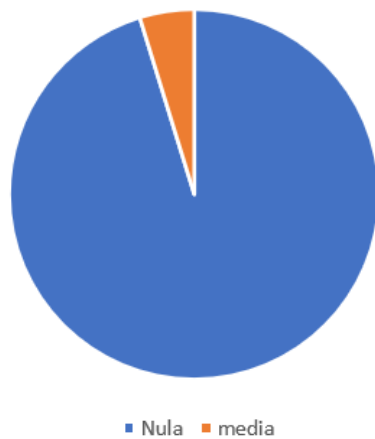


Fig. 28: Gráfico de pastel sobre la severidad de las vulnerabilidades encontradas

### *Pentesting a una máquina con Windows XP*

Como último escenario se atacó a una máquina virtual con Windows XP, esta máquina tiene la dirección IP: 192.168.56.106, y para generar mejores resultados en el test se desactivó el *firewall* de la máquina, en la figura a continuación se muestra el inicio del *pentesting*.

```

franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ # Pentesting a Windows XP
franklin@FG:~/Seguridad en Redes/EF-SeR/daniz-red-main$ sudo docker exec kali /usr/bin/python3 /src/daniz-auto.py -c /src/config.json
[sudo] contraseña para franklin:
mkdir: cannot create directory '/src/archivos/': File exists
mkdir: cannot create directory '/src/archivos/hosts/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_1/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_2/': File exists
mkdir: cannot create directory '/src/archivos/hosts/fase_3/': File exists
mkdir: cannot create directory '/src/archivos/hosts/explotacion/': File exists
mkdir: cannot create directory '/src/Reportes/': File exists
mkdir: cannot create directory '/src/Reportes/Ataque/': File exists
mkdir: cannot create directory '/src/Reportes/Escaneo/': File exists
mkdir: cannot create directory '/src/Reporte_final': File exists
rm: cannot remove '/src/archivos/hosts/fase_3/*': No such file or directory
rm: cannot remove '/src/archivos/hosts/explotacion/*': No such file or directory
rm: cannot remove '/src/archivos/hosts/fase_1/*': No such file or directory
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 08:23 UTC
Nmap scan report for 192.168.56.106
Host is up (0.00056s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
cant execute
Traceback (most recent call last):
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 53, in <module>
    main()
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 46, in main
    ip_scan(parse.rango,parse.filepath) #pasamos la opcion q pase el usuario en la linea de comando
  File "/src/modules/hosts/fase_1/scaner_de_red.py", line 43, in ip_scan
    f.close()
UnboundLocalError: local variable 'f' referenced before assignment
rm: cannot remove '/src/archivos/hosts/fase_2/*': No such file or directory
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-31 08:23 UTC

```

Fig. 29: Inicio de *pentesting* hacia la máquina con Windows XP

2) *Resultado del pentesting a Windows XP*: Al terminar el testeo mostrado anteriormente, se genera un reporte en formato PDF en el que se muestran todos los puertos abiertos y todas las vulnerabilidades encontradas en la máquina objetivo, para el caso de esta máquina, solo se encontró tres puertos abiertos, los cuales coinciden con los puertos encontrados en la realización del *pentesting* de la máquina *metasploitable*, en la figura 30 se muestra el resumen de los puertos abiertos y las vulnerabilidades encontradas.

Las tablas IV y V muestran un resumen de lo encontrado en el reporte, adicionalmente, la tabla V muestra el nivel de peligro de la vulnerabilidad basándose en el estándar CVSS ver3, la cual indica que se trata de una vulnerabilidad peligrosa con un impacto alto en la seguridad del equipo.

Y en la figura 31 se muestra una representación en un diagrama de la única vulnerabilizada.

## 1.2\_Escaneo-de-puertos-sin-utilizar-ICMP

### Dirección IP:

```
### 192.168.56.106
```

```
| Port | State | Service |
|-----|-----|-----|
| 135/tcp | open | msrpc |
| 139/tcp | open | netbios-ssn |
| 445/tcp | open | microsoft-ds |
```

(end of excerpt)

### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 08:52:44 2023 as: nmap -sV --script=vulners --script-args mincvss=7.5
--stylesheet=https://svn.nmap.org/nmap/docs/nmap.xml -oA /src/archivos/hosts/fase_3/192.168.56.106vulners_nmap_serv 192.168.56.106
Nmap scan report for 192.168.56.106
Host is up (0.0015s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
135/tcp   open  msrpc    Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jan 31 08:52:55 2023 -- 1 IP address (1 host up) scanned in 11.74 seconds

(end of excerpt)
```

## 1.3\_Vulnerabilidades-mediante-puertos-TCP

### Dirección IP:

```
# Nmap 7.93 scan initiated Tue Jan 31 08:51:17 2023 as: nmap -v -sS --stylesheet=https://svn.nmap.org/nmap/docs/nmap.xml -oA /src/archivos/hosts/fase_3/192.168.56.106vulners_nmap_serv 192.168.56.106
Nmap scan report for 192.168.56.106
Host is up (0.0016s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Host script results:
_smb-vuln-ms10-054: false
smb-vuln-ms17-010:
VULNERABLE:
Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
State: VULNERABLE
IDs: CVE:CVE-2017-0143
Risk factor: HIGH
A critical remote code execution vulnerability exists in Microsoft SMBv1 servers (ms17-010).

Disclosure date: 2017-03-14
References:
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)

Read data files from: /usr/bin/./share/nmap
# Nmap done at Tue Jan 31 08:52:02 2023 -- 1 IP address (1 host up) scanned in 44.45 seconds

(end of excerpt)
```

Fig. 30: Puertos abiertos y vulnerabilidades encontradas

TABLE IV: Puertos y servicios encontrados

Puerto	Estado	Servicio
135/tcp	abierto	msrpc
139/tcp	abierto	netbios-ssn
445/tcp	abierto	microsoft-ds

TABLE V: Vulnerabilidad e impacto

Vulnerabilidad	CVss 3	Impacto
CVE-2017-0143	8.1	alto

### Vulnerabilidad y su Impacto

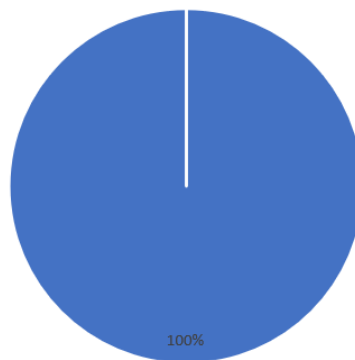


Fig. 31: Gráfico de pastel sobre la severidad de las vulnerabilidades encontradas

### III. CONCLUSIONES

Las amenazas de ciberseguridad en la actualidad requieren que sean analizadas de manera automatizada e inteligente para agilizar el proceso de testeo. En este documento se implementó las pruebas de penetración, las cuales fueron dirigidas hacia máquinas víctimas con diferentes sistemas operativos. Esto con la finalidad de comprobar la fiabilidad que tienen estos productos con respecto a sus posibles vulnerabilidades. Se realizaron cuatro testeos con máquinas con Windows 7, Windows 10, Windows XP y a una con Metasploitable OS. En los tres casos se realizó la recolección de información de cada uno de estos y según el análisis obtenido se enlistaron las vulnerabilidades encontradas, las cuales fueron categorizadas según su grado de impacto. Los reportes obtenidos con el testeo dio como resultado que actualmente en Windows 10 no se encontraron vulnerabilidades, con respecto a Windows 7 se encontró únicamente una vulnerabilidad de un puerto abierto con mediano riesgo, en cambio en Windows XP se encontraron tres puertos abiertos con diferentes servicios, su estado de vulnerabilidad es alto debido al tipo de vulnerabilidad que se encontró y finalmente en la máquina virtual metasploitable se encontraron varias vulnerabilidades como se pudo observar en el documento, esto dependiendo de como se configure la máquina para el testeo. Por otro lado, el *pentesting* ayuda a las empresas a evaluar si se están siguiendo las políticas y procedimientos por parte de sus empleados, dando una medida para reforzar la capacitación de estos en seguridad. Ver como los empleados responden a estas situaciones de amenaza, en donde ellos creen que están seguros hace que las empresas estimulen el cumplimiento de los programas de concentración y hacen que estos programas estén adaptados a cada una de las necesidades. En concreto, esta herramienta denominada *hacking* ético ayuda a detallar riesgos y a explorar los impactos que una posible intrusión que podría llegar a tener una empresa, con el *pentesting* se puede verificar si los tiempos de respuesta del personal que se encuentra disponible es el adecuado, adicional se valida si el tiempo promedio para la restauración de los sistemas es el óptimo.

Al usar como escenarios de ataque a equipos con diferentes sistemas operativos se observó la gran variación de vulnerabilidades de un equipo con respecto a otro, en primer lugar se empleó una máquina virtual que está dedicada al estudio y práctica de *hacking* ético, por lo que este sistema viene preparado con varias vulnerabilidades y puertos abiertos, contrastando completamente con los resultados obtenidos al analizar un equipo con Windows 10 que no presentaba puertos abiertos ni vulnerabilidades. Para encontrar vulnerabilidades se tuvo que recurrir a versiones más antiguas, como Windows 7 en donde se encontró un puerto abierto pero ninguna vulnerabilidad. También se atacó a una máquina con Windows XP, en donde se encontró cuatro puertos abiertos y una vulnerabilidad de alto impacto.

En este trabajo de fin de ciclo se empleó el *framework* Daniz-Red, que ya está preparado para realizar una auditoría completa hacia un objetivo. Lo cual facilita la búsqueda de vulnerabilidades en los equipos, ya que devuelve como resultado todos los puertos abiertos y las vulnerabilidades que ya hayan sido encontradas previamente, por lo que este *framework* no es capaz de encontrar vulnerabilidades del tipo *zero day*.

### REFERENCES

- [1] G. Weidman, *Penetration testing: a hands-on introduction to hacking*. No starch press, 2014.
- [2] D. Santo Orcero and D. Santo Orcero, "Pentesting con kali," *BLURB Incorporated*, 2017.
- [3] J. Arnez Jimenez, "Pentesting," Ph.D. dissertation, 2019.
- [4] "CVE-2011-1002," Feb. 2011. [Online]. Available: <https://www.incibe-cert.es/alerta-temprana/vulnerabilidades/cve-2011-1002>
- [5] [Online]. Available: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143>
- [6] [Online]. Available: <https://nvd.nist.gov/vuln/detail/CVE-2017-0143>