



# **Curso de Hacking ético Master. D**

**Ejercicio 4 - ANÁLISIS DE  
PUERTOS Y  
VULNERABILIDADES**

**Alumno: Julián Gordon**

# Índice

Introducción .....	3
Instalación de Nessus .....	4
Configuración del Escaneo.....	6
Análisis de Aplicaciones de nuestro objetivo .....	16
Plugins .....	23
Launch Scan.....	24
Análisis de Vulnerabilidades.....	25
Vulnerabilidad crítica 1 - Hydra: VNC ataque de fuerza bruta .....	28
Vulnerabilidad crítica 2 -NFS Exported Share Information Disclosure - RCP .....	29
Vulnerabilidad crítica 3 - UnrealIRCd Backdoor Detection - Backdoors .....	30
Vulnerabilidad crítica 4 - Bind Shell Backdoor Detection .....	31
Vulnerabilidad alta 1 - Rsh Service Detection.....	32
Vulnerabilidad media 1 - CGI Generic XML Injection .....	33
Vulnerabilidad baja1- X Server Detection .....	36
Vulnerabilidad de Información - CGI Generic Injectable Parameter .....	37
Conclusiones .....	38

# Introducción

En este ejercicio, lo que se plantea es utilizar Nessus, para hacer un análisis sobre las vulnerabilidades de la máquina de Metasploitable de nuestro laboratorio. Desarrollaremos sobre el proceso de instalación y la ejecución del análisis, formulando una política de escaneo. Luego con el escaneo realizado, analizaremos todas las vulnerabilidades encontradas y explicaremos en detalle las principales.

# Instalación de Nessus

Para instalar Nessus, en este caso, lo haremos sobre nuestra máquina de Kali Linux. Debemos entrar a la página de Nessus(usaremos Nessus Essentials que es gratis) <https://www.tenable.com/products/nessus/nessus-essentials> .

Nos debemos registrar y luego recibiremos un correo con un código de activación.

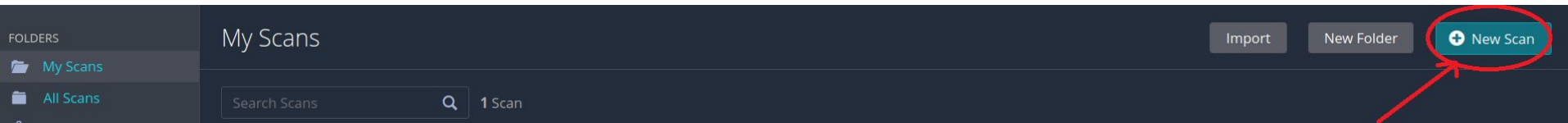
Nessus, se usa sobre un navegador y usaremos Firefox. El proceso de instalación suele tardar un poco, ya que debe instalar todos los plugins.

En la siguiente imagen podemos ver la página de inicio de Nessus, una vez ya lo tengamos instalado.

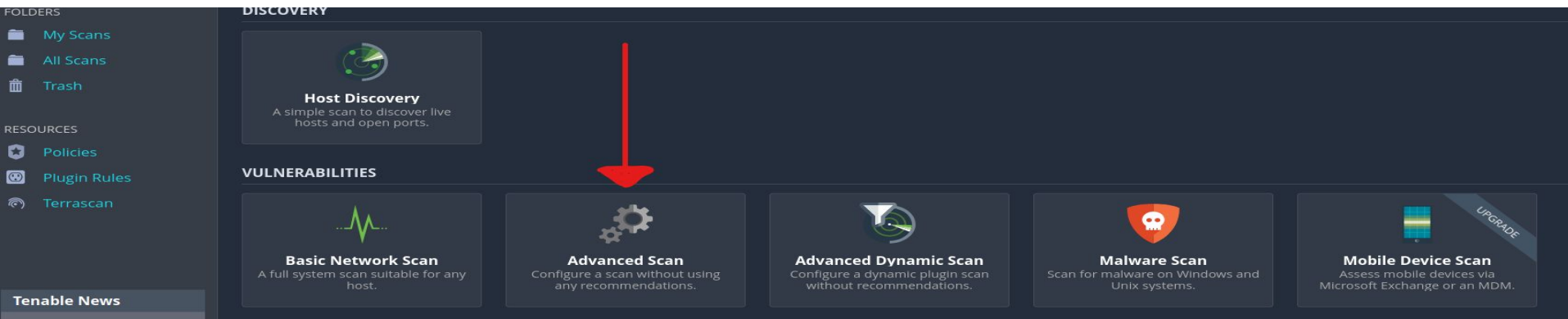
Advantech R-SeeNet  
snmpmon.ini  
Unauthenticated  
Rea...

# Configurando la ejecución del Escaneo

Ahora configuraremos nuestro primer escaneo. Para ello creamos un nuevo escaneo:



Seleccionamos Escaneo avanzado.



Ahora debemos darle un nombre al escaneo, una breve descripción y poner el IP del objetivo que queremos analizar. También se puede separar por “,” ó subir un archivo de texto, en el caso de que queramos analizar más objetivos. No utilizamos, en este caso, el apartado Schedule ni Notifications, que sirven para agendar y para recibir notificaciones por email respectivamente .

FOLDERS

My Scans

All Scans

Trash

RESOURCES

Policies

Plugin Rules

Terrascan

Tenable News

Study: Tenable Offers Fastest, Broadest Coverage o...

New Scan / Advanced Scan

[Back to Scan Templates](#)

Settings

Credentials

Plugins

BASIC

General

Schedule

Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name

Description

Folder

Targets

REQUIRED

My Scans

Example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com

REQUIRED

En Host Discovery, solo marcaremos ARP, ya que los métodos de ping TCP y ICMP pueden ser identificados por cualquier método de protección que pueda tener la maquina objetivo.

### Ping Methods



ARP



TCP

Destination ports

built-in



ICMP



Assume ICMP unreachable from the gateway means the host is down

Maximum number of retries

2



UDP

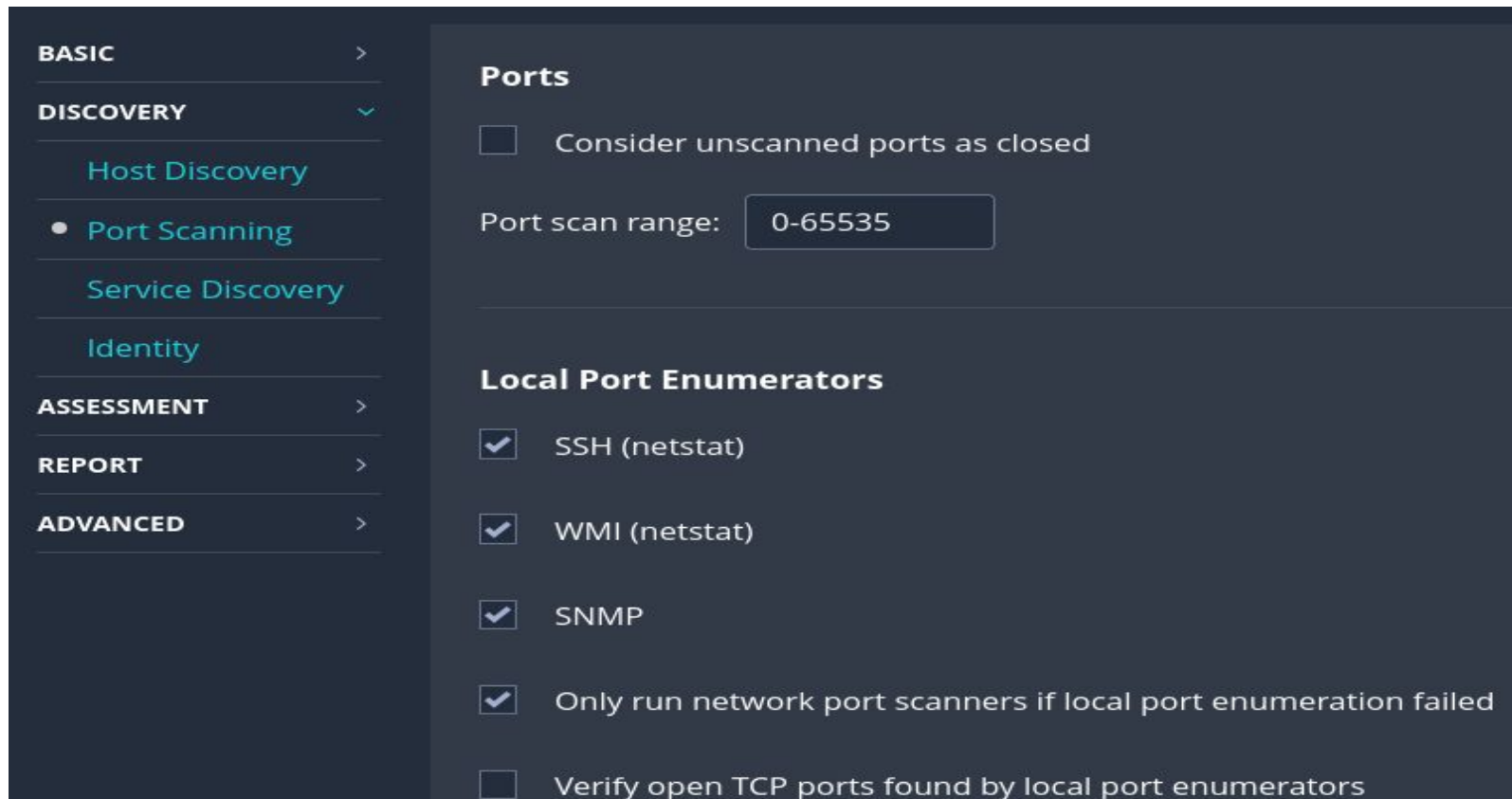


Marcamos todos los escaneos sobre impresoras, dispositivos de tecnología operativa y host Novell Netware, que es un sistema operativo de red que ofrece funciones esenciales como administración de usuarios, uso compartido de archivos y seguridad de datos.

### **Fragile Devices**

- ☒ Scan Network Printers
- ☒ Scan Novell Netware hosts
- ☒ Scan Operational Technology devices

Ahora seleccionaremos el rango de puertos que queremos escanear, que en este caso serán todos.



**BASIC** >

**DISCOVERY** v

- Host Discovery
- Port Scanning
- Service Discovery
- Identity

**ASSESSMENT** >

**REPORT** >

**ADVANCED** >

### Ports

☐ Consider unscanned ports as closed

Port scan range:

### Local Port Enumerators

- ☒ SSH (netstat)
- ☒ WMI (netstat)
- ☒ SNMP
- ☒ Only run network port scanners if local port enumeration failed
- ☐ Verify open TCP ports found by local port enumerators

Luego seleccionaremos conexión mediante el envío de un paquete de sincronización (SYN) al servidor. Y en el caso de que exista un Firewall, que intente detectarlo.

☒ SYN

☒ Override automatic firewall detection

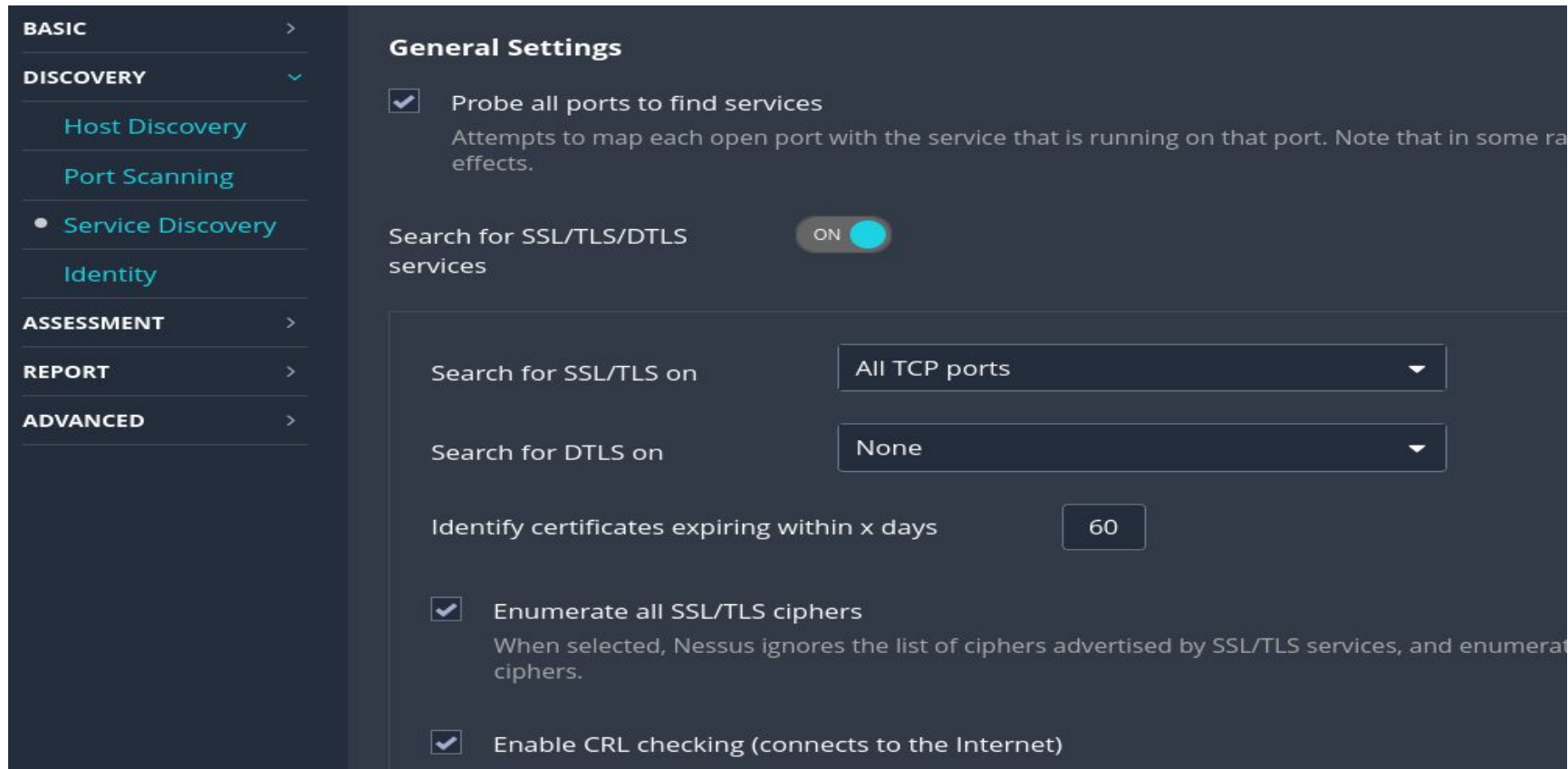
☐ Use soft detection

☒ Use aggressive detection

☐ Disable detection

☐ UDP

En el siguiente apartado “Service Discovery” , seleccionamos la opción de realizar análisis de todos los puertos TCP.



The image shows the Nessus configuration interface. On the left is a sidebar with a menu containing the following items: **BASIC**, **DISCOVERY** (which is expanded), **ASSESSMENT**, **REPORT**, and **ADVANCED**. Under the **DISCOVERY** section, the sub-items are **Host Discovery**, **Port Scanning**, **Service Discovery** (which is selected and marked with a dot), and **Identity**.

The main content area is titled **General Settings** and contains the following configuration options:

- ☒ **Probe all ports to find services**  
Attempts to map each open port with the service that is running on that port. Note that in some rare cases, this can have negative effects.
- Search for SSL/TLS/DTLS services** (toggle switch) **ON**
- Search for SSL/TLS on** (dropdown menu) **All TCP ports**
- Search for DTLS on** (dropdown menu) **None**
- Identify certificates expiring within x days** (input field) **60**
- ☒ **Enumerate all SSL/TLS ciphers**  
When selected, Nessus ignores the list of ciphers advertised by SSL/TLS services, and enumerates all supported ciphers.
- ☒ **Enable CRL checking (connects to the Internet)**

En el apartado “Assessment”, marcaremos la opción para que nos muestre falsos positivos, ya que luego los comprobaremos y confirmaremos. También marcamos la casilla de realizar pruebas exhaustivas ya que nuestra máquina tiene suficientes recursos.

**BASIC** >

**DISCOVERY** >

**ASSESSMENT** ✓

- General
- Brute Force
- Web Applications
- Windows
- Malware
- Databases

**REPORT** >

### Accuracy

☒ Override normal accuracy

☐ Avoid potential false alarms

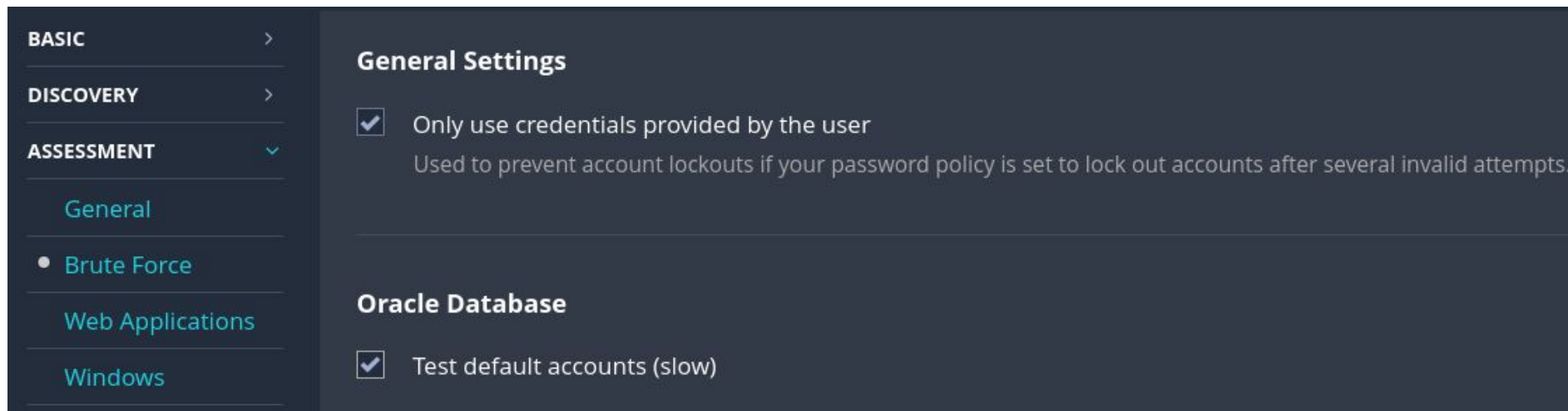
☒ Show potential false alarms

☒ Perform thorough tests (may disrupt your network or impact scan speed)

### Antivirus

Antivirus definition grace period (in days): 0 ▼

Nessus permite realizar ataques de diccionario. En el apartado “Brute Force”, desmarcamos la casilla, solo credenciales que incluimos en el escaneo y marcamos la casilla que utilizaremos credenciales por defecto para bases de datos Oracle.



The screenshot shows the Nessus configuration interface for Brute Force attacks. On the left is a sidebar with a dark blue background and white text. It contains three main sections: 'BASIC' with a right-pointing chevron, 'DISCOVERY' with a right-pointing chevron, and 'ASSESSMENT' with a down-pointing chevron. Under 'ASSESSMENT', there are four sub-items: 'General' (in teal), 'Brute Force' (in teal with a white dot), 'Web Applications' (in teal), and 'Windows' (in teal). The main content area has a dark blue background. It features a 'General Settings' section with a checked checkbox for 'Only use credentials provided by the user' and a descriptive text below it. A horizontal separator line is present. Below the line is the 'Oracle Database' section with a checked checkbox for 'Test default accounts (slow)'.

**BASIC** >

**DISCOVERY** >

**ASSESSMENT** v

- General
- Brute Force
- Web Applications
- Windows

### General Settings

☒ Only use credentials provided by the user

Used to prevent account lockouts if your password policy is set to lock out accounts after several invalid attempts.

---

### Oracle Database

☒ Test default accounts (slow)

Un poco más abajo en la misma página, se puede ver el apartado sobre Hydra. Hydra es una herramienta para descifrar contraseñas, en ataques de fuerza bruta y de diccionario contra diversos protocolos y servicios. Para ello debemos subir 2 archivos que encontraremos en /usr/share/wordlists/metasploit/ , uno contiene una lista de usuarios y el otro una lista de passwords. default\_userpass\_for\_services\_unhash.txt ,default\_pass\_for\_services\_unhash.txt

## Hydra




Always enable Hydra (slow)

Nessus uses Hydra to attempt brute force attacks when either this setting or the "Perform thorough tests" setting in

Logins file

default\_userpass\_for\_services\_unhash.txt 

Passwords file

default\_pass\_for\_services\_unhash-1.txt 

# **Análisis de Aplicaciones de nuestro objetivo**

Ahora analizaremos las aplicaciones web de nuestro objetivo. Deshabilitamos la casilla que pararía el test en caso de que falle el login, probamos todos los métodos HTTP, habilitamos un ataque de denegación de servicios, buscamos servidores embebidos y probamos todas las combinaciones de parámetros. Que no se detenga al encontrar una vulnerabilidad y daremos 5 minutos para los distintos procedimientos(XSS - SQL injection, local file inclusion, etc ).



- ☒ Follow dynamically generated pages

## Application Test Settings

- ☒ Enable generic web application tests
  - ☐ Abort web application tests if HTTP login fails
  - ☒ Try all HTTP methods
  - ☒ Attempt HTTP Parameter Pollution
  - ☒ Test embedded web servers
  - ☒ Test more than one parameter at a time per form
    - ☐ Test random pairs of parameters
    - ☐ Test all pairs of parameters (slow)
    - ☐ Test random combinations of three or more parameters (slower)
    - ☒ Test all combinations of parameters (slowest)

- ☒ Do not stop after the first flaw is found per web page
- ☐ Stop after one flaw is found per web server (fastest)
- ☐ Stop after one flaw is found per parameter (slow)
- ☒ Look for all flaws (slowest)

URL for Remote File Inclusion

`http://rfi.nessus.org/rfi.txt`

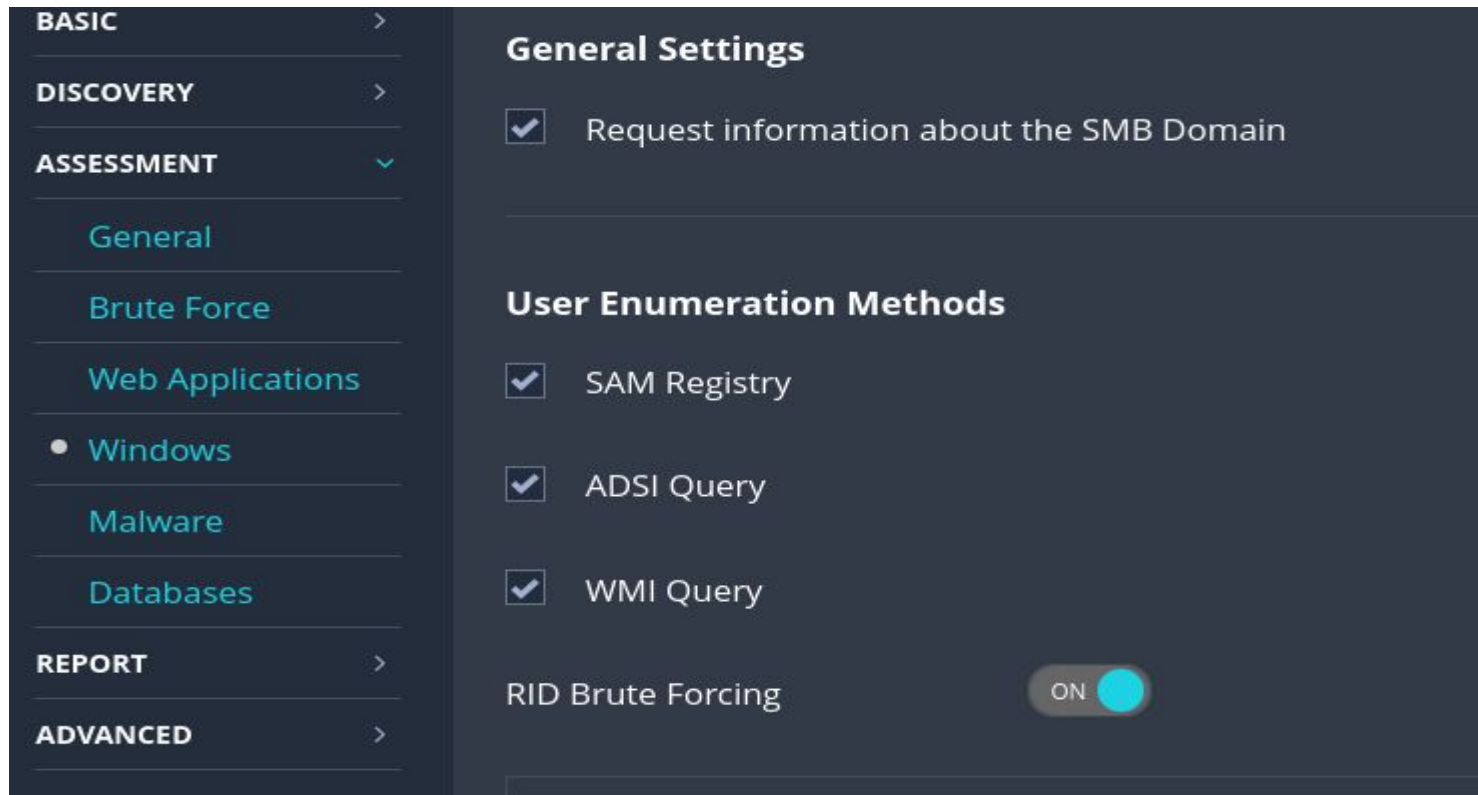
If the target(s) being scanned cannot reach the Internet, the default URL internally hosted file. The file must contain PHP source code that displays when executed.

Maximum run time (minutes)

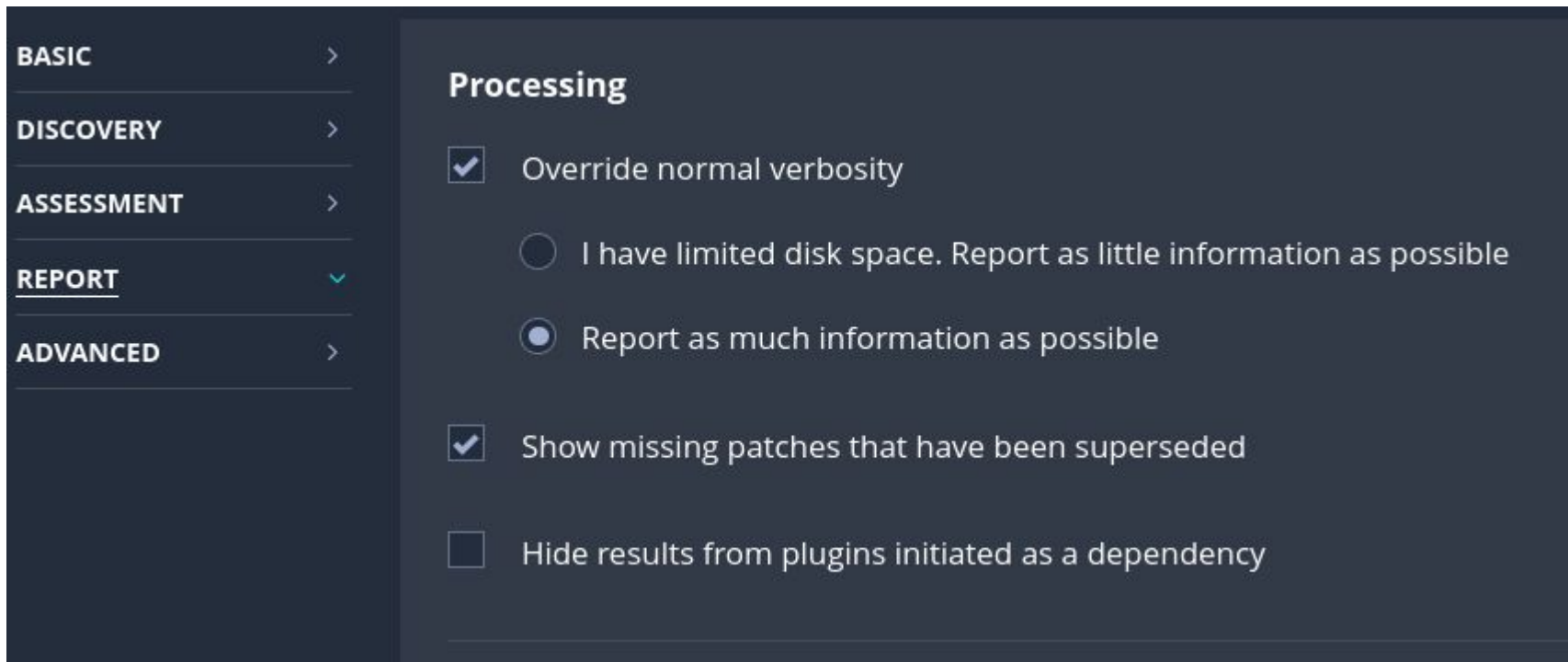
5

This limit refers to the maximum amount of time spent attempting each type (e.g., XSS, SQL injection).

En el apartado “Windows” podemos solicitar que nos dé información sobre el dominio de SMB y podemos hacer un ataque de fuerza bruta contra los IDs de usuarios.



Apartado “Report”, se refiere a cómo vamos a querer el informe. En este caso vamos a marcar la casilla que nos dé la máxima información posible y desmarcar la casilla de esconder resultados de plugins iniciados como dependencia.



**BASIC** >

**DISCOVERY** >

**ASSESSMENT** >

**REPORT** ✓

**ADVANCED** >

### Processing

- ☒ Override normal verbosity
- ☐ I have limited disk space. Report as little information as possible
- ☒ Report as much information as possible
- ☒ Show missing patches that have been superseded
- ☐ Hide results from plugins initiated as a dependency

En el último apartado, encontramos las configuraciones avanzadas. Marcamos las casillas para habilitar controles seguros, la segunda para que pare el proceso en caso de que la máquina objetivo esté apagada y que seleccione aleatoriamente el rango de IPs que le hayamos pasado(en este caso solo la de Metasploitable). Escaneamos objetivos con múltiples nombres de dominio en paralelo.

BASIC

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

General Settings

☒ Enable safe checks

☐ Stop scanning hosts that become unresponsive during the scan

☒ Scan IP addresses in a random order

☐ Automatically accept detected SSH disclaimer prompts

This will automatically attempt to agree to prompts in SSH connections that Tenable products are configured to recognize.

☒ Scan targets with multiple domain names in parallel

Este sería el último apartado dentro de Settings, más abajo encontramos la parte de Debug Settings, que en la misma marcaremos las casillas para que incluya toda la información de depuración.

### Debug Settings

- ☒ **Log scan details**  
Logs the start and finish time for each plugin used during a scan to nessusd.messages.
- ☒ **Always report SSH commands**  
Attaches all SSH commands run on target hosts irrespective of debug settings.
- ☒ **Enable plugin debugging**  
Attaches available debug logs from plugins to the vulnerability output of this scan.

Debug Log Level

Level 1: Basic Debugging

- ☒ **Enumerate launched plugins**  
Adds a list of plugins that were launched during the scan.

Audit Trail Verbosity

All audit trail data

Include the KB

Include KB

# Plugins

Nessus es un escáner de vulnerabilidades que utiliza un motor basado en plugins. Los plugins son pequeños programas que se utilizan para evaluar los objetivos en busca de vulnerabilidades. Están escritos en un lenguaje de scripting denominado Nessus Attack Scripting Language (NASL) que le indican al motor que es lo que debe evaluar en el objetivo, con el fin de determinar las fallas del mismo. Ahora seleccionaremos los plugins que queremos utilizar en nuestro escaneo. Como ya sabemos que nuestra máquina objetivo es una máquina Linux, podemos deshabilitar todos los plugins relacionados a Windows y MacOS.

# Launch Scan

Ahora ya tenemos el escaneo preparado, solo nos resta lanzarlo. Este proceso puede llevar bastante tiempo.

LDERS

My Scans

All Scans

Trash

SOURCES

Policies

Plugin Rules

Terrascan

My Scans

More

Import

New Folder

New Scan

Search Scans

2 Scans (1 Selected) Clear Selected Item

<input type="checkbox"/>	Name	Schedule		Last Scanned		
<input type="checkbox"/>	Escaneo Linux Syn	On Demand	✓	Today at 10:18 AM	▶	✗
<input checked="" type="checkbox"/>	Escaneo Metasploitable	On Demand	📅	N/A	▶	✗



# Análisis de vulnerabilidades

Ahora que ya tenemos el resultado del escaneo que hicimos sobre la máquina de Metasploitable, analizamos las vulnerabilidades que ha detectado.

Las vulnerabilidades se clasifican en función de su gravedad.

**Críticas:** Representan las amenazas más graves y peligrosas. Pueden permitirnos tomar el control total del sistema o acceder a datos altamente confidenciales sin autenticación.

**Altas:** Son vulnerabilidades de alta gravedad, son serias pero no tan catastróficas como las críticas. Pueden permitir el acceso no autorizado o la explotación significativa.

**Medias:** Representan un riesgo moderado y generalmente requieren condiciones específicas para la explotación.

**Bajas:** Generalmente tienen un impacto mínimo en la seguridad y son difíciles de explotar o tienen un alcance muy limitado.

**Informativas o de información:** No representan un riesgo real para la seguridad, pero proporcionan información sobre la configuración del sistema o los servicios que podrían ser útiles.

En esta imagen vemos parte del informe de vulnerabilidades que encontró Nessus.

Hosts 1

Vulnerabilities 112

Remediations 13

History 2

Filter

Search Vulnerabilities

112 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
CRITICAL	10.0 *		Hydra: VNC	Brute force attacks	1
CRITICAL	10.0 *	5.9	NFS Exported Share Information Disclosure	RPC	1
CRITICAL	10.0		Unix Operating System Unsupported Version Dete...	General	1
CRITICAL	10.0 *	7.4	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	10.0 *		VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors	1
MIXED	...	...	ISC Bind (Multiple Issues)	DNS	33
MIXED	...	...	Apache HTTP Server (Multiple Issues)	Web Servers	25
MIXED	...	...	PHP (Multiple Issues)	CGI abuses	23

Scan Details

Policy: Advanced Scan

Status: Completed

Severity Base: CVSS v3.0


Scanner: Local Scanner

Start: Today at 10:53 AM

End: Today at 1:50 PM

Elapsed: 3 hours

Vulnerabilities



Critical

High

Medium

Low

Info

Podemos verificar que el escaneo de Nessus ha detectado 112 vulnerabilidades, en la máquina de Metasploitable. Entre ellas, Nessus cataloga cada vulnerabilidad dentro de una familia de vulnerabilidades. Podemos encontrar ataques de fuerza bruta, divulgación de información en compartición NFS exportada, backdoors, rsh service detection, ISC bind, diversos problemas en servidor HTTP de Apache, problemas en MySql, etc. Analizaremos algunas de ellas más en detalle.

Además de las vulnerabilidades detectadas, Nessus nos sugiere cómo podemos arreglarlas.

## **Vulnerabilidad crítica 1 - Hydra: VNC ataque de fuerza bruta**

Como hemos comentado anteriormente, Hydra se utiliza para realizar ataques de fuerza bruta en sistemas y servicios para adivinar contraseñas, en este caso un servicio VNC (Virtual Network Computing). VNC es un protocolo que permite a los usuarios, controlar de forma remota un sistema o computadora. Este protocolo hace uso principalmente del puerto 5900, necesario para poder establecer la conexión. El informe que nos da Nessus, dice que se pudo iniciar sesión, utilizando la autenticación VNC, con la contraseña que encontró, que es "password". Esta vulnerabilidad es crítica y tiene un puntaje CVSS v2.0 de 10.0 porque le puede dar al atacante el control total de la máquina. Nessus nos sugiere que, para solucionar esta vulnerabilidad, cambiemos dicho password.

## **Vulnerabilidad crítica 2 - NFS Exported Share Information Disclosure - RCP**

Esta vulnerabilidad es una "NFS Exported Share Information Disclosure" (Divulgación de Información en Compartición NFS Exportada). NFS (Network File System) es un protocolo que permite compartir archivos y directorios entre sistemas en una red. Los servidores NFS exportan ó comparten ciertos directorios ó recursos con otros sistemas para su acceso.

Esta vulnerabilidad indica que, al menos una de las particiones NFS exportadas por el servidor remoto, puede ser accedida. Podríamos aprovechar esta situación para acceder a los directorios o recursos compartidos por NFS. Esto nos puede permitir leer y posiblemente escribir, archivos en el servidor. Esta vulnerabilidad es crítica y tiene un puntaje CVSS v2.0 de 10.0

La solución que nos propone Nessus es configurar el sistema NFS en el servidor remoto, de manera que sólo los hosts autorizados puedan montar sus recursos compartidos.

## Vulnerabilidad crítica 3 - UnrealIRCd Backdoor Detection - Backdoors

Esta vulnerabilidad "UnrealIRCd Backdoor Detection", hace referencia a la detección de un backdoor en una versión específica del servidor de IRC llamado UnrealIRCd. Un backdoor es una puerta trasera intencionada o una funcionalidad oculta en el software, que permite a un atacante ejecutar código arbitrario en el sistema afectado sin que el usuario o el administrador lo sepan. UnrealIRCd es un servidor de IRC (Internet Relay Chat) que permite a los usuarios comunicarse en tiempo real a través de salas de chat en línea. Esto quiere decir que podemos utilizarlo para ejecutar cualquier código que queramos en la máquina, como la ejecución de comandos maliciosos, el robo de información confidencial, el daño al sistema o incluso la toma de control completo del servidor. Esta vulnerabilidad es crítica y tiene un puntaje CVSS v2.0 de 10.0.

La solución que nos propone Nessus es volver a descargar el software UnrealIRCd, desde el sitio web oficial del proyecto. Verificar la integridad de la descarga, utilizando los valores de resumen de control de integridad, como los checksums MD5 o SHA1, que son proporcionados por el proyecto en su sitio web. Luego, reinstalar el software utilizando la copia descargada y verificada.

## Vulnerabilidad crítica 4 - Bind Shell Backdoor Detection

La vulnerabilidad "Bind Shell Backdoor Detection" se refiere a la detección de un backdoor. El escaneo ha detectado un servicio en el puerto remoto que actúa como un "bind shell" ó "shell enlazada". Esto significa que hay una shell escuchando en el puerto 1524 sin requerir autenticación. Nos podemos aprovechar de esto conectándonos al puerto remoto y enviar comandos directamente a la máquina. Nessus nos muestra que ejecutó el comando "id" en la máquina y recibió una respuesta que indica que se ejecutó como el usuario "root". Esto significa que podemos obtener acceso al sistema y ejecutar comandos con privilegios de administrador. Esto puede llevar al compromiso total del sistema y al robo de datos, la modificación de configuraciones o la toma de control de la máquina. La solución recomendada es verificar si el sistema remoto ha sido comprometido y, si es necesario, reinstalar el sistema desde cero. El motivo de la reinstalación es garantizar que el backdoor no vuelva a estar presente y que el sistema sea seguro.

# Vulnerabilidad alta 1 - rsh Service Detection

Esta vulnerabilidad "rsh Service Detection" se refiere a la detección de un servicio rsh (Remote Shell) en un host remoto. El servicio rsh es un antiguo protocolo de acceso remoto que permite a los usuarios ejecutar comandos en un sistema remoto de forma similar al comando ssh. Sin embargo, rsh es notoriamente inseguro, ya que transmite datos, incluidas las credenciales de inicio de sesión, en texto sin formato, lo que hace que sea vulnerable a la interceptación por parte de atacantes. El servicio rsh puede permitir accesos con autenticación débil o nula. Debido a esto, podemos capturar credenciales de inicio de sesión, como nombres de usuario y contraseñas. Si el host es vulnerable a adivinanza de números de secuencia TCP o suplantación de IP, podríamos burlar la autenticación y obtener acceso. Además, el servicio rsh puede facilitar el acceso no autorizado mediante la configuración de archivos *.rhosts* ó *rhosts.equiv*, lo que permite a los usuarios especificar quién puede acceder sin necesidad de contraseña. Esta vulnerabilidad es alta y tiene un puntaje CVSS v2.0 de 7.5.

La solución que nos sugiere Nessus, es desactivar el servicio rsh en el host remoto y utilizar en su lugar un protocolo más seguro como SSH (Secure Shell). SSH cifra las comunicaciones y proporciona un acceso seguro a través de la red. Para desactivar el servicio rsh, se puede comentar la línea que hace referencia a "rsh" en el archivo de configuración */etc/inetd.conf* y luego reiniciar el proceso *inetd*. La desactivación de rsh es fundamental para mejorar la seguridad del sistema y evitar problemas de seguridad asociados con la transmisión de datos sin cifrar y la autenticación débil.



# Vulnerabilidad media 1 - CGI Generic XML Injection

La vulnerabilidad "CGI Generic XML Injection" se refiere a una amenaza en la que un atacante puede enviar parámetros especialmente diseñados a uno o más scripts CGI alojados en un servidor web remoto. Cuando esto ocurre, Nessus recibe una respuesta inusual o inesperada, lo que sugiere que el atacante podría haber modificado el comportamiento de la aplicación y tener acceso directo a un servicio SOAP(Simple Object Access Protocol) en el servidor.

Los scripts CGI (Common Gateway Interface) son programas o secuencias de comandos utilizados para generar contenido dinámico en un servidor web. Estos scripts pueden procesar datos XML o interactuar con servicios web basados en SOAP. Esta vulnerabilidad es peligrosa, ya que podríamos aprovecharla para eludir la autenticación, acceder a datos confidenciales, modificar la base de datos remota y hasta tomar el control del sistema operativo remoto. Esta vulnerabilidad tiene un puntaje CVSS v2.0 de 6.8.

La solución propuesta por Nessus, implica en modificar los scripts CGI afectados para que escapen adecuadamente los argumentos, especialmente las etiquetas XML y los caracteres especiales, como los corchetes angulares y las barras diagonales. La validación y la sanitización adecuada de la entrada del usuario en los scripts CGI son pasos fundamentales para prevenir inyecciones de XML y otros tipos de ataques.

A continuación se muestra una imagen que indica que se ha encontrado un recurso o parámetro en la aplicación web que podría ser vulnerable a una inyección de XML. En este caso, se menciona un parámetro llamado 'nocache' en la URL /phpMyAdmin/phpmyadmin.css.php.

## Output

Using the GET HTTP method, Nessus found that :

+ The following resources may be vulnerable to XML injection :

+ The 'nocache' parameter of the /phpMyAdmin/phpmyadmin.css.php CGI :

```
/phpMyAdmin/phpmyadmin.css.php?token=c9b16c3351aef465b24596c4cbceae05&js  
_frame=right&nocache=2457687151</%20foo>
```

----- output -----

```
/* general tags */  
html {  
    font-size: 84%;  
}
```

----- vs -----

```
/* general tags */  
html {  
    font-size: 82%;  
}
```

-----

# Vulnerabilidad baja 1 - X Server Detection

Esta vulnerabilidad se refiere a la detección de un servidor X (X11) en el sistema y señala un problema de seguridad potencial asociado con la comunicación no cifrada a través del protocolo X11. X11 es un sistema de ventanas, utilizado en sistemas operativos tipo Unix y Linux. Es un protocolo cliente-servidor, que permite que las aplicaciones gráficas se ejecuten en un servidor y se muestren en un cliente (como una computadora local). El servidor X11 controla cómo se muestra la interfaz gráfica de las aplicaciones en una pantalla. La vulnerabilidad se refiere al hecho de que la comunicación entre el cliente y el servidor X11 no está cifrada de forma predeterminada. Esto significa que si tenemos acceso a la red entre el cliente y el servidor, podríamos interceptar la información transmitida. Podríamos obtener información sensible, como datos confidenciales o contraseñas que se muestran en ventanas de aplicaciones. La solución que nos sugiere Nessus es restringir el acceso al servidor X11 y, si no se necesita la funcionalidad de cliente/servidor X11, deshabilitar el soporte TCP en X11 por completo. Esto se puede hacer agregando la opción `-nolisten tcp` al servidor X11 ó configurando adecuadamente las reglas de firewall para limitar quién puede acceder al servidor X11.

# Vulnerabilidad baja 1 - CGI Generic Injectable Parameter

La vulnerabilidad "CGI Generic Injectable Parameter", se refiere a una técnica utilizada por Nessus para inyectar cadenas inofensivas en los parámetros CGI de una aplicación web y luego verificar si esas cadenas se reflejan en la respuesta HTTP del servidor. En aplicaciones web, los parámetros CGI (Common Gateway Interface) son datos que se envían al servidor web a través de la URL ó mediante formularios web. Estos parámetros son utilizados por la aplicación web para procesar las solicitudes del usuario. Nessus, inyecta cadenas inofensivas en estos parámetros CGI para ver si la aplicación web los refleja en su respuesta. La inyección de cadenas inofensivas no debe causar daño a la aplicación ni a los datos del usuario, pero se utiliza para verificar si la aplicación es vulnerable a ataques de inyección más graves, como los que vimos anteriormente. Esta técnica no representa una debilidad o vulnerabilidad real por sí sola, su propósito principal es acelerar otras pruebas de seguridad.

# Conclusiones

Una vez terminamos de analizar todas las vulnerabilidades, podemos obtener muchas conclusiones al respecto. Nessus es una herramienta muy útil en el análisis de vulnerabilidades y su función principal es identificar y evaluar vulnerabilidades en sistemas informáticos y redes, mediante la realización de escaneos automatizados que buscan debilidades conocidas, como fallos de seguridad, configuraciones incorrectas o problemas que podrían ser explotados por atacantes. Nos permite también, con los informes generados, corregir las vulnerabilidades identificadas y fortalecer la seguridad de los sistemas.

En este ejercicio hicimos un análisis, en profundidad, sobre 8 vulnerabilidades (ataques de fuerza bruta, NFS exported share information disclosure, backdoors, rsh service detection, etc). Algunas de ellas nos pueden dar el control total de la máquina ó nos pueden permitir acceder a los directorios o recursos compartidos por NFS y con ello leer y escribir, archivos en el servidor. También es posible un robo de datos con información confidencial, la modificación de configuraciones y el daño al sistema.

Esta etapa de análisis de vulnerabilidades, nos sirve para preparar la siguiente etapa dentro de un pentesting, que sería la explotación de vulnerabilidades. Debemos revisar y analizar los informes para priorizar estas vulnerabilidades según su gravedad y potencial impacto. Es muy importante la verificación de falsos positivos, ya que algunas herramientas de escaneo pueden informar de vulnerabilidades que en realidad no existen. Es importante verificar cada una para asegurarse de que sea una vulnerabilidad real. En la etapa de explotación, intentaremos explotar las vulnerabilidades confirmadas, con ataques de inyección de SQL o pruebas de secuencias de comandos entre sitios (XSS), por ejemplo.