Universidad de Sevilla

Grado en Ingeniería Informática – Ingeniería de Computadores

Seguridad en Sistemas Informáticos y en Internet

PRÁCTICA 4

Grupo 2

Alumnos: Silvia Castillo Ruiz, Amara Innocent Millán y Víctor Ramos Lara

Índice

1. Introducción 3

2. Descripción del sistema a analizar previo a realizar el análisis con OpenVAS 3

3. Detalles de instalación, configuración y tareas lanzadas de escaneos, así como los resultados y reportes pedido en las tareas 4

4. Definición del plan de mitigación de vulnerabilidades detectadas 10

5. Descripción del sistema a analizar previo a realizar la monitorización con Suricata 11

6. Descripción de la configuración y conjunto de reglas seleccionados 12

7. Detalles de las pruebas realizadas y los logs con las evidencias 14

8. Conclusiones 17

1. Introducción

La gestión de vulnerabilidades y la detección temprana de intrusos representan una parte importante dentro de cualquier Plan Director de Seguridad de la Información. Aunque gran parte de los esfuerzos de protección se han centrado tradicionalmente en asegurar las comunicaciones o los servicios expuestos públicamente, hay un notable desconocimiento por parte de usuarios y organizaciones sobre los riesgos debidos a fallos de configuración, software desactualizado o una validación insuficiente de entradas y salidas en los sistemas utilizados por los servicios corporativos.

En este informe se desarrolla un proceso completo de auditoría de vulnerabilidades mediante OpenVAS/Greenbone, así como la implementación de un sistema IDS basado en Suricata. El objetivo es demostrar la metodología adecuada para identificar, evaluar, priorizar y mitigar riesgos, complementándolo con la capacidad de detectar comportamientos sospechosos que comprometan los servicios desplegados en la infraestructura analizada.

2. Descripción del sistema a analizar previo a realizar el análisis con OpenVAS

El sistema seleccionado es una máquina virtual desplegada con Oracle VirtualBox. Se trata de Kali Linux en su versión más reciente.

Al tratarse de un sistema recién instalado, este mantiene la configuración de fábrica, incluyendo los servicios predeterminados y las políticas de red estándar de la distribución. El entorno cuenta con 10 GB de memoria RAM para garantizar el rendimiento durante las pruebas. El análisis de vulnerabilidades se realizará sobre la interfaz de red local (Localhost / 127.0.0.1), con el objetivo de establecer un estándar de seguridad e identificar posibles debilidades en la configuración por defecto del sistema operativo.

3. Detalles de instalación, configuración y tareas lanzadas de escaneos, así como los resultados y reportes pedido en las tareas

Vamos a trabajar con un sistema operativo recién instalado por lo que necesitaremos empezar ejecutando el comando **sudo apt update** en nuestra terminal.

Para la realización de las siguientes pruebas vamos a utilizar un entorno basado en contenedores de Docker, por lo que debemos instalar Docker. **sudo apt install -y Docker.io**

Descargamos la imagen oficial recomendada en la documentación (immauss/openvas) desde el repositorio de Docker Hub.

sudo docker pull immauss/openvas

Para asegurarnos que los datos de configuración y los feeds de vulnerabilidades no se pierdan al reiniciar el sistema, creamos un volumen de datos dedicado.

sudo docker volumen create openvas

Ejecutamos el contenedor en el puerto 9392 para desplegar web y estableciendo la contraseña necesaria posteriormente para loguearnos en la página.

sudo docker run --detach --publish 9392:9392 -e PASSWORD="admin" --volume openvas:/data --name openvas immauss/openvas

Tras el comando de ejecución, debes mantener el terminal abierto mientras se actualizan los feeds para test de vulnerabilidades y CVEs, en caso de fallo puede ejecutar los siguientes comandos.

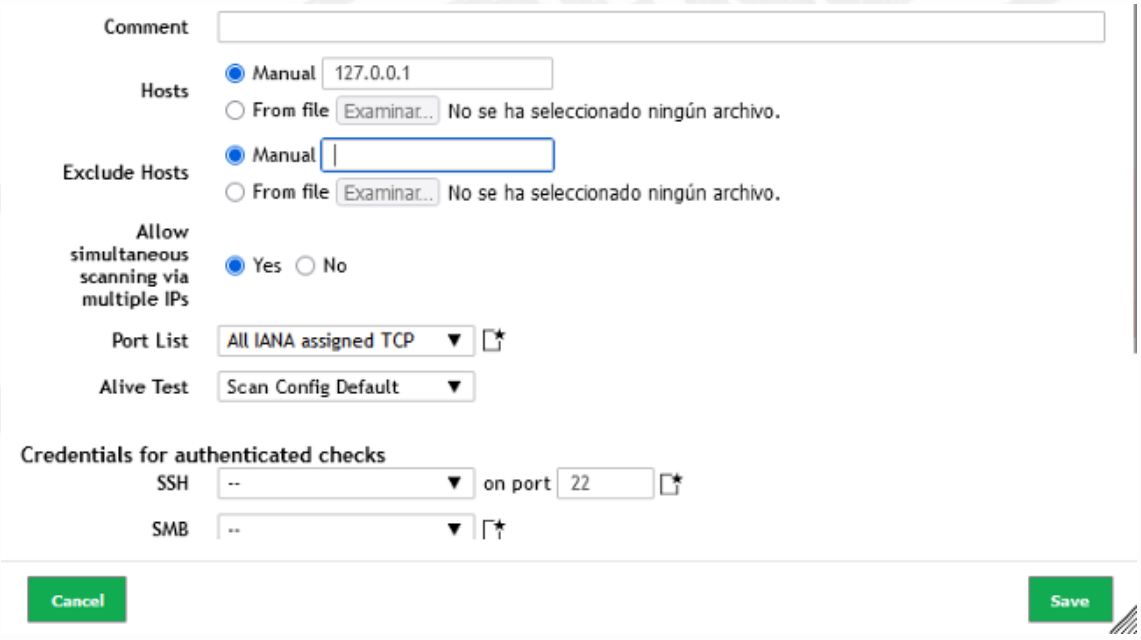
sudo docker restart openvas

sudo docker exec -it openvas /scripts/sync.sh

La actualización de los feeds puede tardar desde 30 minutos hasta entre 3 y 6 horas, es importante tener paciencia. Una vez se encuentre actualizado o haya transcurrido cierta cantidad de tiempo tras su inicialización, podrá acceder a la página web de OpenVAS desde el enlace http://127.0.0.1:9392, donde se encontrará con una pestaña de login, su usuario es "admin", y su contraseña será la utilizada durante el comando: sudo docker run … PASSWORD="(contraseña)".

Una vez dentro, iremos a la pestaña "Targets" dentro de "Scans"

Dentro del formulario nos centramos en poner 127.0.0.1 como Host para que realice el escaneo sobre Kali Linux.



Ejecutamos el escaneo y esperamos unos minutos a recibir el informe. En el caso de nuestra máquina hemos recibido estas vulnerabilidades:

Vulnerability	Severity	QoD	Host IP	Host Name	Location	EPSS Score	EPSS Percentile	Created
OpenVAS / Greenbone Vulnerability Manager (GVM) Default Credentials (OMP/GMP Protocol)	10.0 (Critical)	100 %	127.0.0.1	localhost	9390/tcp	N/A	N/A	Thu, Nov 20, 2025 11:36 AM Coordinated Universal Time
SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detection	5.0 (Medium)	99 %	127.0.0.1	localhost	25/tcp	N/A	N/A	Thu, Nov 20, 2025 11:33 AM Coordinated Universal Time
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3 (Medium)	98 %	127.0.0.1	localhost	9390/tcp	N/A	N/A	Thu, Nov 20, 2025 11:33 AM Coordinated Universal Time

Vulnerabilidad 1: Credenciales por defecto en OpenVAS / GVM, identificado con OID: 1.3.6.1.4.1.25623.1.0.103825, afecta al Puerto: 9390/tcp

OpenVAS ha detectado que el usuario admin tiene como contraseña admin. Esto permite a cualquier persona con acceso al puerto de gestión tomar el control total de la herramienta de escaneo.

Debido a esta configuración pobre un atacante remoto puede conectarse al servicio, obtener información sensible de auditorías previas, modificar la configuración del sistema o lanzar escaneos no autorizados contra otros objetivos desde esta máquina.

Se ha clasificado con prioridad crítica (Severity 10.0) porque otorga acceso administrativo total al sistema sin necesidad de herramientas avanzadas. Compromete la confidencialidad, integridad y disponibilidad del servicio.

Summary

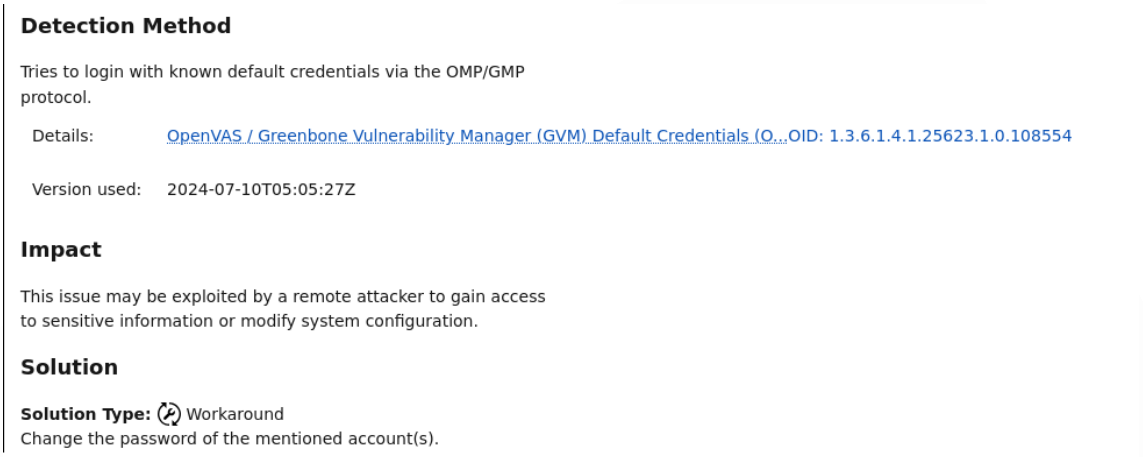
The remote OpenVAS / Greenbone Vulnerability Manager (GVM) is installed / configured in a way that it has account(s) with default passwords enabled.

Detection Result

```
It was possible to login using the following credentials (username:password:role):

admin:admin:Super Admin
```

Product Detection Result

Product	cpe:/a:greenbone:greenbone_vulnerability_manager:22.7
Method	OpenVAS / Greenbone Vulnerability Manager Detection (OMP/GMP) (OID: 1.3.6.1.4.1.25623.1.0.103825)
Log	View details of product detection



Detection Method

Tries to login with known default credentials via the OMP/GMP protocol.

Details: OpenVAS / Greenbone Vulnerability Manager (GVM) Default Credentials (O... OID: 1.3.6.1.4.1.25623.1.0.108554

Version used: 2024-07-10T05:05:27Z

Impact

This issue may be exploited by a remote attacker to gain access to sensitive information or modify system configuration.

Solution

Solution Type: Workaround
Change the password of the mentioned account(s).

Vulnerabilidad 2: Certificado SSL/TLS autofirmado / Autoridad no confiable, identificado con OID: 1.3.6.1.4.1.25623.1.0.103692, afecta al puerto: 25/tcp (Servicio de correo)

El certificado digital presentado por el servicio en el puerto 25 está firmado por una autoridad no reconocida o es autofirmado (Issuer: CN=localhost). Esto impide que los clientes puedan verificar la identidad real del servidor, generando alertas de seguridad en los navegadores o clientes de correo.

Este fallo facilita ataques de tipo Man-in-the-Middle. Un atacante podría interceptar el tráfico cifrado haciéndose pasar por el servidor, y el usuario no notaría la diferencia al no existir una cadena de confianza válida.

Se le ha asignado prioridad media. Aunque permite la interceptación, requiere que el atacante tenga acceso a la red para posicionarse en medio de la comunicación. Es una configuración habitual en entornos de prueba como el nuestro, pero es un gran problema en un campo real.

Summary

The service is using an SSL/TLS certificate from a known untrusted and/or dangerous certificate authority (CA).

Detection Result

```
The certificate of the remote service is signed by the following untrusted and/or dangerous CA:

Issuer: CN=localhost

Certificate details:
fingerprint (SHA-1)             | B310FD54CDEED58B485F1422E03F0DCA8F325031
fingerprint (SHA-256)           | B538C63E514A50E63F4DECB0BF1A1EE560E365F327C1DA034D1099964F5A4E39
issued by                       | CN=localhost
public key algorithm            | RSA
public key size (bits)          | 2048
serial                          | 53CEC64CB39AB2335831213CC024E353EB478C30
signature algorithm             | sha256WithRSAEncryption
subject                         | CN=localhost
subject alternative names (SAN) | localhost
valid from                      | 2025-11-05 10:05:37 UTC
valid until                     | 2035-11-03 10:05:37 UTC
```

Product	cpe:/a:ietf:transport_layer_security
Method	SSL/TLS: Collect and Report Certificate Details (OID: 1.3.6.1.4.1.25623.1.0.103692)
Log	View details of product detection

Detection Method

The script reads the certificate used by the target host and checks if it was signed by a known untrusted and/or dangerous CA.

Details:	SSL/TLS: Known Untrusted / Dangerous Certificate Authority (CA) Detect...OID: 1.3.6.1.4.1.25623.1.0.113054
Version used:	2024-06-14T05:05:48Z

Impact

An attacker could use this for man-in-the-middle (MITM) attacks, accessing sensible data and other attacks.

Solution

Solution Type: Mitigation
Replace the SSL/TLS certificate with one signed by a trusted CA.

Vulnerabilidad 3: Protocolos de cifrado obsoletos (TLSv1.0 y TLSv1.1) , afecta al puerto: 9390/tcp.

CVEs asociados: CVE-2011-3389 (BEAST), CVE-2015-0204 (FREAK), CVE-2023-41928

El servidor acepta conexiones utilizando versiones antiguas del protocolo TLS (v1.0 y v1.1). Estos protocolos contienen debilidades criptográficas conocidas y han sido declarados obsoletos por los estándares de seguridad actuales.

Un atacante podría explotar vulnerabilidades conocidas como BEAST o realizar ataques de "downgrade" (FREAK) para forzar al servidor a usar un cifrado débil y posteriormente descifrar la comunicación, comprometiendo datos sensibles transmitidos en la sesión segura.

Ha recibido una prioridad media, menor al problema previo, debido a que la explotación de estas vulnerabilidades es compleja y requiere condiciones específicas, se debe mitigar actualizando la configuración del servidor web.

CVE	NVT	Hosts	Occurrences	Severity
CVE-2011-3389 CVE-2015-0204 CVE-2023-41928 CVE-2024-41270 CVE-2025-3200	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	1	1	4.3 (Medium)

1 - 1 of 1

(Applied filter: apply_overrides=0 levels=chml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 1 of 1

Summary

It was possible to detect the usage of the deprecated TLSv1.0 and/or TLSv1.1 protocol on this system.

Detection Result

```
In addition to TLSv1.2+ the service is also providing the deprecated TLSv1.0 and TLSv1.1 protocols and supports one or more ciphers. Those supported ciphers can be found in the 'SSL/TLS: Report Supported Cipher Suites' (OID: 1.3.6.1.4.1.25623.1.0.802067) VT.
```

Product Detection Result

Product	cpe:/a:ietf:transport_layer_security:1.0
Method	SSL/TLS: Version Detection (OID: 1.3.6.1.4.1.25623.1.0.105782)
Log	View details of product detection

Insight

The TLSv1.0 and TLSv1.1 protocols contain known cryptographic flaws like:

- CVE-2011-3389: Browser Exploit Against SSL/TLS (BEAST)

- CVE-2015-0204: Factoring Attack on RSA-EXPORT Keys Padding Oracle On Downgraded Legacy Encryption (FREAK)

Detection Method

Checks the used TLS protocols of the services provided by this system.

Details:	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection OID: 1.3.6.1.4.1.25623.1.0.117274
Version used:	2025-04-30T05:39:51Z

Affected Software/OS

- All services providing an encrypted communication using the TLSv1.0 and/or TLSv1.1 protocols

- CVE-2023-41928: Kiloview P1 4G and P2 4G Video Encoder

- CVE-2024-41270: Gorush v1.18.4

- CVE-2025-3200: Multiple products from Wiesemann & Theis

Impact

An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection.

Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates anymore.

Solution

Solution Type: ⮂ Mitigation
It is recommended to disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols.

4. Definición del plan de mitigación de vulnerabilidades detectadas

Para mitigar las 3 vulnerabilidades previamente mostradas realizaremos algunos cambios en nuestra configuración.

Comenzando por la primera vulnerabilidad, solo debemos cambiar la contraseña, utilizaremos en siguiente comando:

sudo docker exec -it openvas gvmd --user=admin (o tu usuario en otro caso) –new password = “(contraseña)”

La nueva contraseña deberá cumplir con las Politicas de Seguridad de la empresa.

Para la segunda vulnerabilidad, en entornos de producción, se debe adquirir un certificado firmado por una CA pública para evitar los errores de confianza. Para la tercera vulnerabilidad, hay que reconfigurar los servicios (web y correo) para deshabilitar el soporte a TLSv1.0 y TLSv1.1, forzando únicamente TLS 1.2 o superior.

Para los mecanismos de Recuperación hablaremos de mínimo 2 muy importantes; En caso de usar una máquina virtual, se programará una política de "Snapshots" diarios.

Esto permite revertir el sistema completo a un estado anterior conocido en minutos si una actualización falla o el sistema es comprometido. Se propone realizar backups de datos, una exportación semanal de la base de datos de OpenVAS y configuraciones críticas a un almacenamiento externo o nube cifrada.

Para la detección de intentos de intrusión hay 2 posibilidades claras; Realizar una configuración de rsyslog para enviar los registros de autenticación (/var/log/auth.log) y de servicios web a un servidor central. Hacer uso de un sistema IDS, despliegue de Suricata (como se explica en la siguiente sección del proyecto) para analizar el tráfico de red en tiempo real. Esto permitirá detectar si alguien intenta explotar los fallos SSL/TLS o realizar fuerza bruta antes de que logren entrar.

5. Descripción del sistema a analizar previo a realizar la monitorización con Suricata

Para la fase de monitorización con Suricata se utilizó un entorno basado en WSL2 ejecutando una distribución Ubuntu. Esto nos permitió disponer de un sistema operativo Linux dentro de Windows sin la necesidad de hacer uso de una máquina virtual, facilitando la instalación y ejecución del IDS.

Antes de desplegar Suricata identificamos los servicios más importantes del sistema a proteger:

- HTTP/HTTPS internos en los puertos 8083 y 8443
- MySQL sobre el puerto 3336
- SSH/SFTP en el puerto 2288

Estos servicios representan puntos críticos de acceso que deben ser monitorizados y por ello en el fichero suricata.yaml se ajustaron las variables de red:

- HOME_NET: Siendo la dirección del servidor protegido
- EXTERNAL_NET: Define todo tráfico no perteneciente a HOME_NET

Con esta configuración Suricata puede distinguir entre tráfico legítimo e intentos de conexión externos y sospechosos. Una vez definido este entorno, desplegamos y cargamos las reglas diseñadas específicamente para detectar accesos no autorizados hacia los servicios mencionados.

6. Descripción de la configuración y conjunto de reglas seleccionadas

Una vez definido el entorno de monitorización, se configuró Suricata para su ejecución en modo IDS dentro de Ubuntu en WSL2. El objetivo principal de la configuración fue permitir la detección de tráfico sospechoso dirigido hacia ciertos servicios del sistema, diferenciando el tráfico interno (HOME_NET) del externo (EXTERNAL_NET).

6.1. Configuración inicial

Tras instalar Suricata con los comandos:

```
silvi@LAPTOP-SILVIA:~$ sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt update
sudo apt install suricata
```

Se configuró el archivo *suricata.yml*:

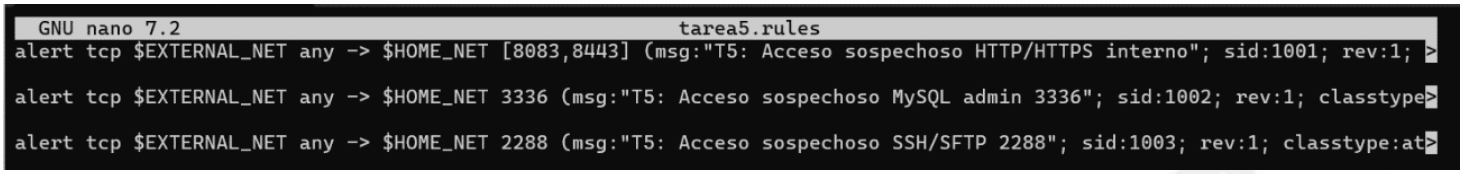
```
suricata-version: "8.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    HOME_NET: "[172.22.96.0/20]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
```

Ajustando el valor de los parámetros:

- HOME_NET: Dirección de la red del servidor
- EXTERNAL_NET: Cualquiera que no sea HOME_NET (!$HOME_NET)

6.2. Reglas personalizadas

Para definirlas, creamos un archivo llamado *tarea5.rules*, donde incluimos las diferentes reglas personalizadas para detectar accesos sospechosos a SSH/SFTP, tráfico hacia puertos HTTP internos y conexiones externas a MySQL:



Cada regla se describe a continuación:

- **Acceso sospechoso a HTTP/HTTPS interno:** Esta regla genera una alerta cuando se detecta tráfico TCP desde cualquier dirección externa a la HOME_NET hacia los puertos 8083 o 8443. Esto se entiende como un intento de acceso a servicios HTTP/HTTPS internos.
- **Acceso sospechoso a MySQL externo (puerto 3336):** Esta regla genera una alerta cuando una red externa intenta establecer conexión hacia el puerto 3336, utilizado para servicios MySQL. Esto se entiende como un intento de administración no autorizado.
- **Acceso sospechoso a SSH/SFTP (puerto 2288):** Esta regla genera una alerta cuando una red externa intenta establecer conexión hacia el puerto 2288, utilizado para alojar servicios SSH y SFTP. Esto se entiende como un intento de acceso no autorizado.

Finalmente cargamos las reglas y pudimos desplegar suricata de la siguiente forma:

```
root@LAPTOP-SILVIA:/etc/suricata/rules# sudo suricata -c /etc/suricata/suricata.yaml -i eth0 --init-errors-fatal -v -S /
etc/suricata/rules/tarea5.rules
Notice: suricata: This is Suricata version 8.0.2 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 8
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: suricata: Preparing unexpected signal handling
Info: conf: Running in live mode, activating unix socket
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 1 rule files processed. 3 rules successfully loaded, 0 rules failed, 0 rules skipped
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 3 signatures processed. 0 are IP-only rules, 0 are inspecting packet payload, 0 inspect application layer,
 0 are decoder event only
Notice: mpm-hs: Rule group caching - loaded: 0 newly cached: 0 total cacheable: 0
Info: unix-manager: unix socket '/var/run/suricata/suricata-command.socket'
Info: unix-manager: created socket directory /var/run/suricata/
Info: af-packet: eth0: unable to find af-packet config for interface "eth0" or "default", using default values
Warning: af-packet: eth0: AF_PACKET tpacket-v3 is recommended for non-inline operation
Info: runmodes: eth0: creating 8 threads
Notice: threads: Threads created -> W: 8 FM: 1 FR: 1   Engine started.
```

7. Detalles de las pruebas realizadas y los logs con las evidencias

Para verificar el correcto funcionamiento de las reglas personalizadas implementadas en Suricata, realizamos pruebas de acceso a los servicios configurados y también analizamos los logs generados.

Los registros en Suricata se encuentran en /var/log/suricata/

```
root@LAPTOP-SILVIA:~# cd /var/log/suricata
root@LAPTOP-SILVIA:/var/log/suricata# ls -l
total 110276
drwxrwxr-x 2 root     suricata     4096 Nov  6 11:23 certs
drwxrwxr-x 2 root     suricata     4096 Nov  6 11:23 core
-rw-r--r-- 1 suricata suricata 63027515 Nov 22 17:17 eve.json
-rw-r--r-- 1 suricata suricata     4394 Nov 17 16:11 fast.log
drwxrwxr-x 2 root     suricata     4096 Nov  6 11:23 files
-rw-r--r-- 1 suricata suricata 49814256 Nov 22 17:17 stats.log
-rw-r--r-- 1 suricata suricata    42488 Nov 22 17:17 suricata.log
```

- **Fast.log:** Es el registro rápido de las alertas
- **Eve.log:** Es el registro detallado en formato JSON
- **Stats.log:** Almacena las estadísticas de Suricata
- **Suricata.log:** Logs generales del sistema de detección

Para monitorear las alertas usamos el comando tail –f fast.log y así poder ver las alertas en tiempo real durante las pruebas:

```
root@LAPTOP-SILVIA:/var/log/suricata# sudo tail -f fast.log
11/16/2025-18:15:44.420027  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
11/16/2025-18:15:44.420027  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
11/16/2025-18:15:44.420027  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
11/16/2025-18:15:44.420027  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
11/16/2025-18:15:44.420027  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
11/16/2025-18:15:46.244382  [**] [1:2013504:6] ET INFO GNU/Linux APT User-Agent Outbound
 likely related to package management [**] [Classification: Not Suspicious Traffic] [Pri
ority: 3] {TCP} 172.22.106.241:34866 -> 185.125.190.83:80
```

Para probar que nuestras reglas funcionaban correctamente realizamos pruebas de conexión desde una terminal Windows:

Acceso a HTTP/HTTPS por el puerto 8083

```
PS C:\Users\silvi> Test-NetConnection -ComputerName 172.22.106.241 -Port 8083
ADVERTENCIA: TCP connect to (172.22.106.241 : 8083) failed


ComputerName           : 172.22.106.241
RemoteAddress          : 172.22.106.241
RemotePort             : 8083
InterfaceAlias         : vEthernet (WSL (Hyper-V firewall))
SourceAddress          : 172.22.96.1
PingSucceeded          : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded       : False
```

Acceso a HTTP/HTTPS por el puerto 8443

```
PS C:\Users\silvi> Test-NetConnection -ComputerName 172.22.106.241 -Port 8443
ADVERTENCIA: TCP connect to (172.22.106.241 : 8443) failed


ComputerName           : 172.22.106.241
RemoteAddress          : 172.22.106.241
RemotePort             : 8443
InterfaceAlias         : vEthernet (WSL (Hyper-V firewall))
SourceAddress          : 172.22.96.1
PingSucceeded          : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded       : False
```

Acceso a MySQL por el puerto 3336

```
PS C:\Users\silvi> Test-NetConnection -ComputerName 172.22.106.241 -Port 3336
ADVERTENCIA: TCP connect to (172.22.106.241 : 3336) failed


ComputerName           : 172.22.106.241
RemoteAddress          : 172.22.106.241
RemotePort             : 3336
InterfaceAlias         : vEthernet (WSL (Hyper-V firewall))
SourceAddress          : 172.22.96.1
PingSucceeded          : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded       : False
```

Acceso a SSH/SFTP por el puerto 2288

```
PS C:\Users\silvi> Test-NetConnection -ComputerName 172.22.106.241 -Port 2288
ADVERTENCIA: TCP connect to (172.22.106.241 : 2288) failed


ComputerName            : 172.22.106.241
RemoteAddress           : 172.22.106.241
RemotePort              : 2288
InterfaceAlias          : vEthernet (WSL (Hyper-V firewall))
SourceAddress           : 172.22.96.1
PingSucceeded           : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded        : False
```

En nuestro fast.log nos aparecen alertas como las siguientes, y por tanto podemos comprobar que nuestras reglas y alertas si funcionan como deberían:

```
11/22/2025-17:38:23.560445  [**] [1:1001:1] T5: Acceso sospechoso HTTP/HTTPS interno [**
] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 172.22.96.1:34152 ->
172.22.106.241:8083
11/22/2025-17:39:07.728113  [**] [1:1001:1] T5: Acceso sospechoso HTTP/HTTPS interno [**
] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 172.22.96.1:37607 ->
172.22.106.241:8443
11/22/2025-17:39:22.766852  [**] [1:1002:1] T5: Acceso sospechoso MySQL admin 3336 [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.22.96.1
:37609 -> 172.22.106.241:3336
11/22/2025-17:43:21.149531  [**] [1:1003:1] T5: Acceso sospechoso SSH/SFTP 2288 [**] [Cl
assification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.22.96.1:18
669 -> 172.22.106.241:2288
```

Para hacer un análisis aún más profundo, verificamos la cantidad de alertas por regla que se generaron con el comando grep y wc –l:

```
root@LAPTOP-SILVIA:/var/log/suricata# grep "1001" fast.log | wc -l
grep "1002" fast.log | wc -l
grep "1003" fast.log | wc -l
2
2
1
```

```
root@LAPTOP-SILVIA:/var/log/suricata# grep "1001" fast.log
grep "1002" fast.log
grep "1003" fast.log
11/22/2025-17:38:23.560445  [**] [1:1001:1] T5: Acceso sospechoso HTTP/HTTPS interno [**
] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 172.22.96.1:34152 ->
172.22.106.241:8083
11/22/2025-17:39:07.728113  [**] [1:1001:1] T5: Acceso sospechoso HTTP/HTTPS interno [**
] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 172.22.96.1:37607 ->
172.22.106.241:8443
11/17/2025-16:11:23.100207  [**] [1:2027397:1] ET INFO Spotify P2P Client [**] [Classifi
cation: Not Suspicious Traffic] [Priority: 3] {UDP} 172.22.96.1:57621 -> 172.22.111.255:
57621
11/22/2025-17:39:22.766852  [**] [1:1002:1] T5: Acceso sospechoso MySQL admin 3336 [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.22.96.1
:37609 -> 172.22.106.241:3336
11/22/2025-17:43:21.149531  [**] [1:1003:1] T5: Acceso sospechoso SSH/SFTP 2288 [**] [Cl
assification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.22.96.1:18
669 -> 172.22.106.241:2288
```

Esto se corresponde con los dos intentos de acceso a servicios HTTP/HTTPS, el acceso al servicio de MySQL y el acceso a servicios SSH/SFTP. También detecta otro

acceso, aunque no sospechoso a Spotify. Esto se debe a que una de nuestras reglas, la de acceso sospechoso a MySQL y ese acceso comparten una parte de identificador 1002.

8. Conclusiones

La realización de esta práctica nos ha permitido implementar una metodología para la gestión de la ciberseguridad, mezclando la auditoría pasiva con la detección de intrusiones en tiempo real. El análisis inicial de vulnerabilidades con OpenVAS/Greenbone identificó fallas, siendo la más grave la existencia de credenciales por defecto en el propio gestor, lo que proporcionaba acceso administrativo total al sistema. También se detectaron riesgos de prioridad asociados al cifrado, incluyendo el uso de un certificado SSL/TLS autofirmado y la compatibilidad con protocolos TLS obsoletos.

El plan de mitigaciones se diseñó de forma que se pudieran solucionar las vulnerabilidades de forma inmediata, priorizando la eliminación del riesgo crítico al cambiar la contraseña por defecto de OpenVAS. Para las fallas de comunicación, sustituimos el certificado no confiable y reconfiguramos los servicios para deshabilitar los protocolos inseguros para que utilicen TSL 1.2 o superior. También se añadió un mecanismo de recuperación.

La implementación del sistema de detección de intrusos con Suricata fue exitosa, configurando correctamente el entorno y las variables de red para distinguir el tráfico interno y legítimo del sospechoso. La creación de reglas personalizadas para monitorear puertos críticos nos ha permitido validar la capacidad del sistema. Todas las pruebas resultaron exitosas ya que generaron las alertas esperadas en el fast.log del IDS. Esto prueba que Suricata funciona de manera eficaz como herramienta de alerta temprana contra intentos de acceso no autorizados.