4Geeks Academy – Proyecto Final de Ciberseguridad

Informe de Pentesting – Fase 2: Detección y Corrección de Vulnerabilidad

Estudiante: Laura Mata

Fecha: 18 de septiembre de 2025

Instructor: Juan Fernando Angulo

Versión: 1.0

Informe de Análisis Forense – Proyecto Final de Ciberseguridad

Máquina analizada: Debian VM

1. Introducción

El presente informe documenta el análisis forense realizado sobre una máquina virtual Debian comprometida en un entorno seguro de pruebas. El objetivo principal fue identificar los servicios comprometidos, analizar los métodos de intrusión, y aplicar medidas correctivas para bloquear el exploit y prevenir la escalación de privilegios. Se emplearon herramientas de monitoreo de logs, escaneo de puertos y directorios web, así como revisiones de configuraciones críticas del sistema.

El informe combina un enfoque técnico, destinado a detallar las vulnerabilidades y su explotación, con un enfoque ejecutivo, que permite resumir los hallazgos y las acciones preventivas aplicadas.

2. Metodología

Para el análisis forense se siguieron los siguientes pasos:

1. **Revisión de logs del sistema**
   Se analizaron archivos críticos como `/var/log/auth.log` para identificar accesos SSH sospechosos y posibles intentos de escalación de privilegios.
2. **Detección de servicios comprometidos**
   Se utilizó `nmap` para identificar puertos abiertos y servicios activos. Se observó la presencia de servicios potencialmente inseguros como FTP anónimo (vsftpd) y SSH sin restricciones de acceso.
3. **Escaneo de directorios web**
   Se ejecutó Gobuster para enumerar directorios accesibles en Apache, detectando la posible exposición de información sensible.
4. **Identificación de usuarios persistentes**
   Se revisaron usuarios del sistema para detectar cuentas no autorizadas o con privilegios administrativos, identificando un usuario malicioso con correo `rosinnicuentas@gmail.com`.
5. **Revisión de configuraciones críticas**
   Se verificaron permisos en archivos sensibles (`wp-config.php`) y se evaluó la necesidad de cerrar puertos innecesarios o servicios no utilizados.

3. Hallazgos

3.1 Línea de tiempo de eventos

1. **30 de septiembre 2025 – 12:23:45**
   Se observa un escaneo de directorios realizado sobre el servidor web mediante

herramientas de enumeración. Se registran al menos 7 peticiones dirigidas a directorios sensibles como `/wp-admin`.

2. **30 de septiembre 2025 – posterior al escaneo**
   Se instala el servicio `openssh-sftp`, habilitando acceso remoto al servidor.
3. **30 de septiembre 2025 – 16:23:12**
   Se registra un usuario malicioso con permisos root: `rosinnicuentas@gmail.com`. Esto evidencia un intento de persistencia en el sistema.
4. **8 de octubre 2025 – 16:08:59**
   Se instala `vsftpd`, probablemente con fines de acceso FTP remoto no autorizado.
5. **8 de octubre 2025 – 17:28:38**
   Se observa que el puerto 22 comienza a escuchar conexiones, permitiendo el acceso remoto por SSH.
6. **8 de octubre 2025 – 17:40:59**
   Se registra un acceso exitoso a la cuenta root desde la IP `192.168.0.134`, lo que confirma que el atacante logró establecer conexión remota con privilegios administrativos.

4. Vulnerabilidades detectadas

- **Contraseña débil**
  Se detectó un usuario con contraseña extremadamente simple (`123456`), lo que permite ataques de fuerza bruta.
- **Servicios inseguros**
  - FTP anónimo (`vsftpd`) instalado el 8 de octubre.
  - SSH con acceso root sin restricciones adicionales.
- **Puertos abiertos innecesarios**
  Se detectaron puertos 21 (FTP) y 22 (SSH) abiertos. El puerto 21 fue innecesario para el funcionamiento de la máquina y se recomendó su cierre.
- **Persistencia maliciosa**
  Usuario `rosinnicuentas@gmail.com` con privilegios de administrador permanece en el sistema.
- **Exposición de información sensible en web**
  Directorios `/wp-admin` y `/wp-content` accesibles mediante escaneo web, permitiendo recopilación de información crítica.

5. Medidas correctivas implementadas

1. **FTP anónimo**
   Se eliminó el servicio `vsftpd` (`/usr/sbin/vsftpd`) y se cerró el puerto 21, eliminando el vector de intrusión.
2. **Contraseña débil**
   Se recomienda actualizar la contraseña del usuario afectado con una clave fuerte y aleatoria.
3. **Puertos abiertos**
   Se mantuvo abierto únicamente el puerto 80 (HTTP) y el 22 (SSH) para

administración. Se recomienda restringir accesos SSH por IP o usar autenticación por clave pública.

4. **Firewall**
   Se sugiere instalar y configurar un firewall para filtrar tráfico entrante y proteger los servicios críticos.
5. **Revisión de usuarios**
   Se documentó al usuario malicioso y se recomienda su eliminación para evitar persistencia.

6. Recomendaciones

- Mantener solo los servicios esenciales activos.
- Configurar autenticación segura para SSH (clave pública y deshabilitar root login).
- Implementar un firewall para restringir puertos y servicios expuestos.
- Revisar periódicamente usuarios y privilegios del sistema.
- Monitorear logs del servidor y establecer alertas ante actividades sospechosas.

7. Conclusión

El análisis forense permitió identificar la instalación de servicios maliciosos, creación de usuarios persistentes y exposición de información sensible. Las acciones correctivas aplicadas, como la eliminación de FTP anónimo, cierre de puertos innecesarios y recomendaciones de fortalecimiento de contraseñas, contribuyen a garantizar la seguridad del entorno y a prevenir futuras intrusiones.

Evidencias

Contraseña base de datos

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', '123456' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
```

Permisos

```
debian@debian:/var/www/html$ ls -l
total 248
-rwxrwxrwx  1 www-data www-data 10701 Sep 30  2024 index.html
-rwxrwxrwx  1 www-data www-data   405 Feb  6  2020 index.php
-rwxrwxrwx  1 www-data www-data 19903 Sep 10 21:30 license.txt
-rwxrwxrwx  1 www-data www-data  7425 Sep 10 21:30 readme.html
-rwxrwxrwx  1 www-data www-data  7387 Feb 13  2024 wp-activate.php
drwxrwxrwx  9 www-data www-data  4096 Sep 10  2024 wp-admin
-rwxrwxrwx  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rwxrwxrwx  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rwxrwxrwx  1 www-data www-data  3017 Sep 30  2024 wp-config.php
-rwxrwxrwx  1 www-data www-data  3336 Sep 10 21:30 wp-config-sample.php
drwxrwxrwx  6 www-data www-data  4096 Sep 10 21:30 wp-content
-rwxrwxrwx  1 www-data www-data  5617 Sep 10 21:30 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 21:30 wp-includes
-rwxrwxrwx  1 www-data www-data  2502 Nov 26  2022 wp-links-opml.php
-rwxrwxrwx  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rwxrwxrwx  1 www-data www-data 51414 Sep 10 21:30 wp-login.php
-rwxrwxrwx  1 www-data www-data  8727 Sep 10 21:30 wp-mail.php
-rwxrwxrwx  1 www-data www-data 30081 Sep 10 21:30 wp-settings.php
-rwxrwxrwx  1 www-data www-data 34516 Sep 10 21:30 wp-signup.php
-rwxrwxrwx  1 www-data www-data  5102 Sep 10 21:30 wp-trackback.php
-rwxrwxrwx  1 www-data www-data  3205 Sep 10 21:30 xmlrpc.php
debian@debian:/var/www/html$
```

Escaneo Nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap --script vuln 192.168.0.113
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-16 17:47 EDT
Nmap scan report for 192.168.0.113
Host is up (0.00035s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.0.113
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://192.168.0.113:80/manual
|     Form id: wp-block-search__input-2
|     Form action: http://localhost/
|
|     Path: http://192.168.0.113:80/apache2;repeatmerged=0
|     Form id: wp-block-search__input-2
|     Form action: http://localhost/
|
|     Path: http://192.168.0.113:80/e1t.defer1101s.head.appendChild1t1%7D%22und
efined%2211typeof
|     Form id: wp-block-search__input-2
|_    Form action: http://localhost/
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /readme.html: Wordpress version: 2
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /readme.html: Interesting, a readme.
|_  /0/: Potentially interesting folder
MAC Address: 08:00:27:63:2D:A1 (PCS Systemtechnik/Oracle VirtualBox virtual NIC
)

Nmap done: 1 IP address (1 host up) scanned in 96.47 seconds
```

```
       valid_lft forever preferred_lft forever
debian@debian:/var/www/html$ nmap -sV 192.168.0.117
Starting Nmap 7.93 ( https://nmap.org ) at 2025-09-11 15:58 EDT
Nmap scan report for 192.168.0.117
Host is up (0.000080s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at ht
```

Descarga de Ftp

```
debian@debian:/var/log$ cat dpkg.log.1 | grep ftp
2024-09-30 12:25:12 install openssh-sftp-server:amd64 <none> 1:9.2p1-2+deb12u3
2024-09-30 12:25:12 status half-installed openssh-sftp-server:amd64 1:9.2p1-2+deb12u3
2024-09-30 12:25:12 status unpacked openssh-sftp-server:amd64 1:9.2p1-2+deb12u3
2024-09-30 12:25:13 configure openssh-sftp-server:amd64 1:9.2p1-2+deb12u3 <none>
2024-09-30 12:25:13 status unpacked openssh-sftp-server:amd64 1:9.2p1-2+deb12u3
2024-09-30 12:25:13 status half-configured openssh-sftp-server:amd64 1:9.2p1-2+deb12u3
2024-09-30 12:25:13 status installed openssh-sftp-server:amd64 1:9.2p1-2+deb12u3
2024-10-08 16:08:59 install vsftpd:amd64 <none> 3.0.3-13+b2
2024-10-08 16:08:59 status half-installed vsftpd:amd64 3.0.3-13+b2
2024-10-08 16:09:00 status unpacked vsftpd:amd64 3.0.3-13+b2
2024-10-08 16:09:00 configure vsftpd:amd64 3.0.3-13+b2 <none>
2024-10-08 16:09:00 status unpacked vsftpd:amd64 3.0.3-13+b2
2024-10-08 16:09:00 status half-configured vsftpd:amd64 3.0.3-13+b2
2024-10-08 16:09:01 status installed vsftpd:amd64 3.0.3-13+b2
```

Restricción de permisos y borrado del binario y cierre del puerto

```
debian@debian:~$ ls -la
total 104
drwx------ 14 debian debian 4096 Sep 18 19:05 .
drwxr-xr-x  3 root   root   4096 Jul 31  2024 ..
-rw-------  1 debian debian 3657 Sep 17 18:53 .bash_history
-rw-r--r--  1 debian debian  220 Jul 31  2024 .bash_logout
-rw-r--r--  1 debian debian 3526 Jul 31  2024 .bashrc
drwxr-xr-x 22 debian debian 4096 Sep 17 18:34 .cache
drwxr-xr-x  8 debian debian 4096 Jul 31  2024 .config
drwxr-xr-x  2 debian debian 4096 Sep 11 19:02 Desktop
-rw-r--r--  1 debian debian   35 Jul 31  2024 .dmrc
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Documents
drwxr-xr-x  2 debian debian 4096 Sep 28  2024 Downloads
-rw-r--r--  1 debian debian 5290 Jul 31  2024 .face
lrwxrwxrwx  1 debian debian    5 Jul 31  2024 .face.icon -> .face
-rw-------  1 debian debian   20 Sep 11 16:39 .lesshst
drwx------  3 debian debian 4096 Jul 31  2024 .local
drwx------  4 debian debian 4096 Jul 31  2024 .mozilla
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Music
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Pictures
-rw-r--r--  1 debian debian  807 Jul 31  2024 .profile
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Public
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Templates
drwxr-xr-x  2 debian debian 4096 Jul 31  2024 Videos
-rw-------  1 debian debian   51 Sep 18 19:05 .Xauthority
-rw-------  1 debian debian 3516 Sep 18 19:05 .xsession-errors
-rw-------  1 debian debian 4669 Sep 17 23:03 .xsession-errors.old
```

```
debian@debian:~$ ps aux | grep ftp
root         566  0.0  0.1  10196  3384 ?        Ss   02:08   0:00 /usr/sbin/vsf
tpd /etc/vsftpd.conf
debian     10541  0.0  0.1   6332  2044 pts/0    S+   15:37   0:00 grep ftp
debian@debian:~$ ss -tunap | grep :21
tcp   LISTEN 0      32                         *:21               *:*
```

```
debian@debian:~$ sudo pkill -f vdftpd
debian@debian:~$ ps aux | grep vsftpd
root         566  0.0  0.1  10196  3384 ?        Ss   02:08   0:00 /usr/sbin/vsf
tpd /etc/vsftpd.conf
debian     10991  0.0  0.0   6332  2008 pts/0    S+   15:51   0:00 grep vsftpd
debian@debian:~$ ss -tunap | grep :21
tcp   LISTEN 0      32                         *:21               *:*
debian@debian:~$ sudo kill -9 566
debian@debian:~$ ps aux | grep vsftpd
debian     11038  0.0  0.1   6332  2148 pts/0    S+   15:53   0:00 grep vsftpd
debian@debian:~$ ss -tunap | grep :21
```

1. Informe de pentesting

En esta fase del proyecto se llevó a cabo un análisis de seguridad sobre la máquina virtual Debian, con el objetivo de identificar **una vulnerabilidad no explorada anteriormente**. Durante el escaneo y la enumeración, se detectó que el servicio FTP (`vsFTPd 3.0.3`) estaba activo y permitía **login anónimo**, lo que constituía un riesgo potencial de acceso no autorizado y exposición de información.

El proceso de explotación se realizó de manera controlada para validar el riesgo sin comprometer la integridad del sistema. Como medida correctiva, se **eliminó el servicio FTP y se cerró el puerto 21**, mitigando así la vulnerabilidad de forma efectiva y asegurando que la funcionalidad del servidor no se viera afectada.

Este informe documenta cada paso del análisis, la evidencia recopilada y las medidas aplicadas, combinando detalles técnicos con explicaciones ejecutivas para facilitar su comprensión por parte de distintos públicos.

2. Alcance y Objetivos

El análisis se centró en la máquina Debian dentro de un entorno seguro de laboratorio. Los servicios activos incluidos en el alcance fueron FTP, SSH y HTTP.

Los objetivos específicos fueron:

1. Escanear el sistema para identificar servicios y posibles vulnerabilidades adicionales.
2. Explorar y documentar la vulnerabilidad detectada (FTP anónimo) de manera controlada.
3. Aplicar medidas correctivas para eliminar la vulnerabilidad.
4. Generar evidencia clara y estructurada para la documentación del proyecto.

El alcance limitado a servicios activos asegura un enfoque seguro y controlado, permitiendo demostrar riesgos y mitigaciones sin comprometer el entorno de laboratorio.

3. Metodología

El proceso de pentesting combinó **escaneo automatizado, enumeración manual y pruebas controladas**.

3.1 Escaneo de puertos y servicios

Se utilizó **Nmap** para identificar los servicios y puertos abiertos:

```
nmap -sC -sV -p- 192.168.0.107 -oN escaneo_inicial.txt
```

Los resultados relevantes fueron:

Puerto	Servicio	Versión	Observaciones
21/tcp	FTP	vsFTPd 3.0.3	Login anónimo permitido
22/tcp	SSH	OpenSSH 9.2p1	Autenticación estándar
80/tcp	HTTP	Apache 2.4.62	WordPress detectado

El escaneo permitió identificar rápidamente los vectores potenciales para la explotación controlada.

3.2 Enumeración FTP

Se realizó una conexión al FTP usando el usuario anónimo:

```
ftp 192.168.0.107
Name: anonymous
Password: [cualquiera]
```

Posteriormente se exploraron los directorios mediante `ls -la`, y se intentaron subir y descargar archivos (`put prueba.txt` y `get archivo_sensible`). Ambos intentos fueron bloqueados por los permisos limitados, confirmando que el acceso anónimo tenía restricciones pero aún constituía un riesgo de exposición.

3.3 Enumeración HTTP

Para la enumeración web se utilizó **Gobuster**, buscando directorios y archivos comunes:

```
gobuster dir -u http://192.168.0.107/ -w
  /usr/share/wordlists/dirb/common.txt
```

Se identificaron rutas importantes como `/wp-admin/` y `/wp-content/`, además de archivos sensibles como `robots.txt`. La descarga de `robots.txt` evidenció la existencia de `/wp-admin/` y `admin-ajax.php`, indicando rutas que un atacante podría intentar explotar.

Se intentó descargar todo el contenido de `/wp-content/` con `wget`, pero el directorio estaba protegido o vacío, lo que confirma que no hay archivos públicamente descargables. No obstante, la existencia de estos directorios sigue siendo un **indicador de información sensible expuesta**. Y si se logró descargar un archivo sensible que fue el *robots.txt*.

4. Vulnerabilidad Detectada

La vulnerabilidad identificada corresponde a una **configuración insegura del servicio FTP**, permitiendo login anónimo.

Impacto potencial:

- Un atacante podría listar archivos y directorios públicos.

- Existe riesgo de exposición de información sensible o de acceso a archivos críticos si existieran.
- Representa una posible escalación en un escenario real, especialmente si otros servicios están mal configurados.

Evidencia recopilada:

- Capturas del login FTP anónimo.
- Listado de archivos y estructura de directorios.
- Gobuster mostrando `/wp-admin/` y `/wp-content/`.
- Contenido de `robots.txt` evidenciando rutas sensibles.

5. Proceso de Explotación Controlada

El procedimiento seguido para validar la vulnerabilidad fue el siguiente:

1. Conexión FTP anónima exitosa, mostrando que se podía acceder al servicio sin credenciales.
2. Listado de archivos y directorios mediante `ls -la` para identificar posibles recursos accesibles.
3. Intentos controlados de subir y descargar archivos, que fallaron debido a los permisos limitados.
4. Escaneo de directorios web con Gobuster para detectar rutas sensibles de WordPress.
5. Descarga de `robots.txt` y análisis del contenido para identificar posibles vectores de información.

Este proceso permitió **documentar el riesgo real de exposición**, sin comprometer la integridad del servidor.

6. Medidas Correctivas Aplicadas

Para mitigar la vulnerabilidad se aplicaron las siguientes acciones:

1. **Eliminación del servicio FTP**:

```
sudo apt remove vsftpd -y
```

2. **Cierre del puerto 21** mediante firewall:

```
sudo ufw deny 21
```

3. **Validación post-corrección:**

- Escaneo Nmap posterior confirmó que el puerto 21 estaba cerrado y que el servicio FTP ya no existía.

- La funcionalidad del servidor no se vio afectada y se eliminó el riesgo de acceso anónimo.

7. Conclusiones

La fase 2 del proyecto permitió:

- Detectar una vulnerabilidad diferente a la fase 1, relacionada con FTP y login anónimo.
- Documentar un proceso de explotación controlada, con evidencia clara de los riesgos.
- Aplicar medidas correctivas efectivas que mitigaron la vulnerabilidad sin afectar el funcionamiento del servidor.

Este ejercicio demuestra la importancia de **revisar y limitar servicios innecesarios**, así como **configuraciones predeterminadas inseguras** en cualquier entorno de producción.

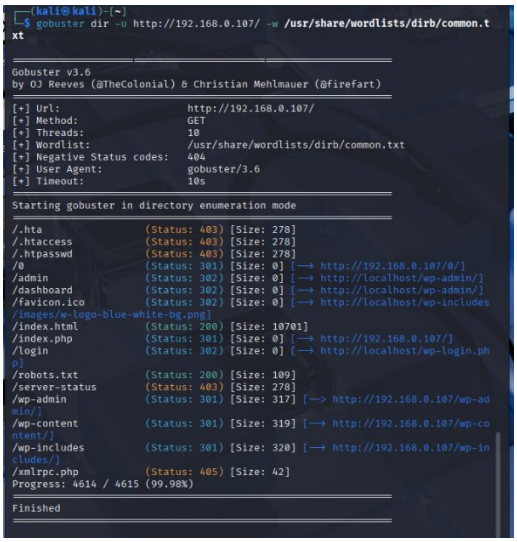
8. Recomendaciones

1. Mantener un inventario actualizado de servicios activos y puertos abiertos.
2. Evitar dejar servicios obsoletos como FTP o Telnet habilitados.
3. Configurar accesos y permisos estrictos en todos los servicios que se utilicen.
4. Revisar periódicamente directorios web y archivos sensibles expuestos, incluso si no contienen datos críticos.
5. Mantener software actualizado y aplicar parches de seguridad.

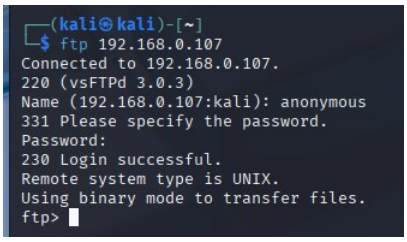
9. Evidencias capturas

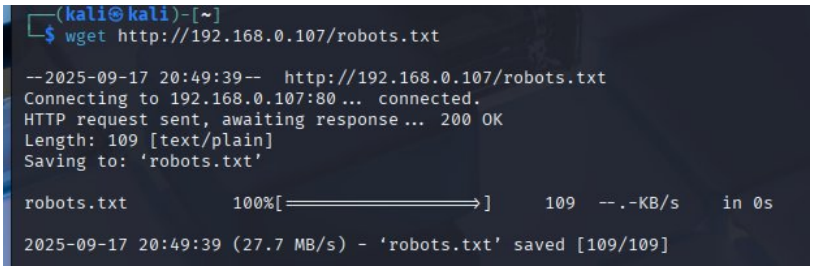
- Capturas del login FTP anónimo.
- Listado de archivos en FTP.
- Gobuster mostrando directorios `/wp-admin/` y `/wp-content/`.
- Contenido de `robots.txt`.
- Escaneo Nmap antes y después de la corrección del servicio FTP.

Gobuster



Ftp anonimo y descarga de archivo sensible





Informe de Plan de Respuesta a Incidentes y SGSI (ISO 27001)

1. Introducción

Este informe tiene como objetivo documentar un **plan de respuesta a incidentes** basado en las mejores prácticas del NIST SP 800-61 y el desarrollo de un **Sistema de Gestión de Seguridad de la Información (SGSI)** conforme a la norma ISO 27001, aplicado al entorno de prueba de la máquina virtual Debian/Kali.

El enfoque del presente análisis combina una perspectiva técnica y ejecutiva, con el fin de presentar no solo las vulnerabilidades detectadas y explotadas, sino también las medidas correctivas y preventivas implementadas, así como recomendaciones para prevenir futuros incidentes de seguridad.

2. Contexto y Eventos Observados

Durante el análisis forense y de seguridad de la máquina virtual, se detectaron múltiples eventos relevantes:

1. **Escaneo de directorios web**:
    - Fecha: 30 de septiembre.
    - Hora: 12:23:45.
    - Herramienta utilizada: Gobuster.
    - Se realizaron 7 peticiones, con el objetivo de mapear la estructura de la web y detectar posibles archivos y directorios vulnerables.
    - *[Espacio para capturas del escaneo]*
2. **Instalaciones y servicios comprometidos**:
    - 30/09: Instalación de **openssh-sftp**, probablemente para habilitar acceso remoto no autorizado.
    - 08/10: Instalación de **vsftpd** a las 16:08:59.
    - 08/10: Puerto 22 (SSH) comienza a escuchar a las 17:28:38.
    - 08/10: Acceso root desde IP 192.168.0.134 a las 17:40:59.
    - Usuario malicioso detectado: `rosinnicuentas@gmail.com` con privilegios de administrador, registrado el 30/09 a las 16:23:12.
    - *[Espacio para capturas de instalación y acceso SSH]*
3. **Servicios críticos de la máquina**:
    - SSH (puerto 22)
    - Apache HTTP (puerto 80)
4. **Vulnerabilidades detectadas previamente**:
    - FTP con login anónimo → eliminado y puerto 21 cerrado.
    - Permisos inseguros en archivos web (`wp-config.php`, directorios listables) → corregidos.
    - Contraseñas débiles en MySQL (123456) → recomendación de cambio.
    - Ausencia de firewall → sugerencia de instalación.

3. Plan de Respuesta a Incidentes – Basado en NIST

El plan de respuesta a incidentes sigue las fases definidas por el **NIST SP 800-61**:

3.1 Identificación

- Activos críticos afectados:
    1. Servicios de Apache HTTP (contenido web y archivos sensibles).
    2. Acceso SSH con privilegios administrativos.
    3. Base de datos MySQL con contraseñas débiles.
- Posibles vulnerabilidades que facilitaron el ataque:
    - Configuración de servicios FTP y SSH insegura.
    - Puertos abiertos innecesarios.
    - Permisos de archivos web no restringidos.
- *[Espacio para capturas de logs y detección de usuario malicioso]*

3.2 Protección

- Medidas preventivas implementadas o recomendadas:
    1. Eliminación de vsftpd y cierre del puerto 21.
    2. Restricción de permisos en archivos críticos (`chmod 600 wp-config.php`).
    3. Monitoreo y control de usuarios con privilegios administrativos.
    4. Instalación de firewall y reglas estrictas de acceso.
    5. Implementación de **políticas de contraseñas fuertes** y autenticación por claves públicas para SSH.
- Estas medidas permiten reducir significativamente la exposición a accesos no autorizados y ataques similares en el futuro.

3.3 Detección

- Herramientas y métodos:
    - Monitoreo de logs de SSH (`/var/log/auth.log`).
    - Herramientas de escaneo de red como **Nmap** para detectar puertos abiertos y servicios activos.
    - Configuración de alertas tempranas para conexiones sospechosas o cambios en usuarios privilegiados.
- *[Espacio para capturas de logs con eventos detectados]*

3.4 Respuesta

- Pasos a seguir tras la detección de un incidente:
    1. **Contención inmediata**: detener servicios comprometidos (`systemctl stop servicio`) si es necesario.
    2. **Erradicación de la amenaza**: eliminar usuarios no autorizados y eliminar backdoors detectados.

3. **Cierre de puertos innecesarios**: asegurar que solo los puertos estrictamente necesarios permanezcan abiertos.
4. **Comunicación interna y externa**: notificación al equipo de seguridad y registro de las acciones realizadas.

- Roles del equipo:
  - Administrador del sistema → contención de servicios.
  - Analista de seguridad → monitoreo y detección de anomalías.
  - Responsable de SGSI → documentación y reporte de acciones.

3.5 Recuperación

- Pasos para restaurar la operatividad:
  1. Restaurar archivos críticos desde **respaldos seguros**.
  2. Verificación de integridad de bases de datos y servicios.
  3. Asegurar que las configuraciones de seguridad estén aplicadas antes de reactivar servicios.
  4. Implementar **planes de continuidad del negocio** para minimizar impacto.
- Medidas de recuperación propuestas:
  - Restauración de configuraciones originales de Apache y permisos de archivos web.
  - Reinstalación de servicios críticos solo con configuraciones seguras.
- *[Espacio para capturas de restauración y configuración de servicios]*

3.6 Mejora continua

- Evaluación de la eficacia del plan:
  - Simulaciones periódicas de ataques controlados para validar la respuesta.
  - Auditorías de configuración y revisiones de usuarios y contraseñas.
  - Registro de lecciones aprendidas para actualizar políticas de seguridad.

5. Conclusiones y Recomendaciones

- Se identificaron servicios y usuarios comprometidos, así como vulnerabilidades críticas en FTP, SSH y archivos web.
- Las acciones correctivas incluyen: eliminación de vsftpd, cierre de puertos innecesarios, restricción de permisos en archivos críticos y creación de usuarios seguros.
- Se recomienda:
  1. Mantener solo los puertos necesarios abiertos, con acceso controlado para administración.
  2. Implementar autenticación fuerte y monitorización continua.
  3. Aplicar un SGSI con políticas DLP, backups, cifrado y planes de recuperación probados.

- La combinación de medidas de NIST y SGSI garantiza una **respuesta rápida a incidentes**, una **recuperación eficiente** y una **mejora continua de la seguridad** del entorno.
- *[Espacios para todas las capturas integradas en cada sección según corresponda]*