

HTB Sherlock: APTNightmare-2	
Sistema Operativo:	Linux
Dificultad:	Hard
Release:	17/01/2025
<b>Tags</b>	
<ul style="list-style-type: none"> <li>● DFIR</li> </ul>	
<b>Skills Learned</b>	
<ul style="list-style-type: none"> <li>● Rootkit Analysis</li> <li>● Linux Memory Forensics</li> </ul>	

La presente resolución documenta de manera sistemática el análisis forense realizado sobre un entorno Linux comprometido, con el objetivo de identificar y caracterizar un módulo de kernel malicioso. A lo largo de las distintas fases se emplearon *plugins* especializados de Volatility y herramientas de desensamblado como **Ghidra**, **nm** y **readelf**, lo que permitió reconstruir la cronología del ataque, detectar técnicas de ocultación y establecer la correlación con tácticas recogidas en la matriz **MITRE ATT&CK**.

El trabajo se estructura en once preguntas clave que guían el proceso de investigación: desde la identificación de la conexión de reverse shell y su proceso asociado, hasta la detección del módulo oculto, el cálculo de su hash criptográfico y el contraste con el archivo legítimo. Cada hallazgo se contextualiza dentro de un marco metodológico riguroso, destacando cómo el adversario desplegó tácticas de **defense evasion**, **masquerading** y **rootkit** para garantizar persistencia y dificultar la atribución.



El análisis se inicia con la apertura del archivo comprimido protegido por contraseña, cuyo desbloqueo se efectuó mediante la clave *hacktheblue*. El material proporcionado corresponde a un volcado de memoria de un sistema Linux, lo que condiciona de manera significativa la elección de la herramienta forense.

```
└─(usuario㉿kali)-[~/HTB]
└─$ 7z l APTNightmare-2.zip
7-Zip 25.01 (x64) : Copyright (c) 1999-2025 Igor Pavlov : 2025-08-03
64-bit locale-es_ES.UTF-8 Threads:4 OPEN_MAX:1024, ASM

Scanning the drive for archives:
1 file, 309326246 bytes (295 MiB)

Listing archive: APTNightmare-2.zip

--
Path = APTNightmare-2.zip
Type = zip
Physical Size = 309326246

      Date    Time   Attr         Size   Compressed  Name
----- -----
2024-05-04 01:39:34 D....          0       0  APTNightmare-2
2024-04-27 02:28:16  ....  1242821    1242833  APTNightmare-2/Ubuntu_5.3.0-70-generic_profile.zip
2024-05-02 14:51:50  ....  1595816904  308082799  APTNightmare-2/dump.mem

2024-05-04 01:39:34          1597059725  309325632  2 files, 1 folders
```

Aunque **Volatility3** constituye la evolución natural del marco de análisis, caracterizado por una arquitectura modular, mayor extensibilidad y soporte nativo para Python 3, sus capacidades respecto a entornos Linux aún presentan limitaciones operativas que comprometen la exhaustividad del examen. Por ello, se optó por recurrir a **Volatility2**, versión que, pese a su dependencia de Python 2 y a la necesidad de configurar manualmente perfiles específicos para cada distribución, ofrece una cobertura más madura y estable en lo relativo a la disección de memorias Linux.

```
└─(isolated)─(usuario㉿kali)─[~/HTB]
└─$ git clone https://github.com/volatilityfoundation/volatility.git
Clonando en 'volatility'...
remote: Enumerating objects: 27414, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 27414 (delta 0), reused 0 (delta 0), pack-reused 27412 (from 2)
Recibiendo objetos: 100% (27414/27414), 21.11 MiB | 21.34 MiB/s, listo.
Resolviendo deltas: 100% (19758/19758), listo.
```

La instalación de Volatility2 se llevó a cabo mediante la clonación del repositorio oficial y la posterior incorporación del perfil requerido, contenido en el archivo facilitado junto con el reto. Dado que esta versión no incluye perfiles de Linux por defecto, fue preciso copiar el archivo comprimido correspondiente en la ruta *volatility/plugins/overlays/Linux*.

```
└─(usuario㉿kali)─[~/HTB/APTNightmare-2]
└─$ cp Ubuntu_5.3.0-70-generic_profile.zip volatility/plugins/overlays/linux/
└─(usuario㉿kali)─[~/HTB/APTNightmare-2]
└─$ ls volatility/plugins/overlays/linux/
elf.py  elf.pyc  __init__.py  __init__.pyc  linux.py  linux.pyc  Ubuntu_5.3.0-70-generic_profile.zip
```

Una vez completada esta operación, la ejecución de *python2 vol.py --info* permitió verificar la correcta carga del perfil, constatándose su aparición en la sección *Profiles*. Con esta preparación concluida, el entorno analítico quedó dispuesto para abordar la inspección sistemática del volcado de memoria y dar respuesta a las cuestiones planteadas en el desafío.

```
└─(usuario㉿kali)─[~/HTB/APTNightmare-2]
└─$ python volatility/vol.py --info
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (NameError: name 'distorm3' is not defined)

Profiles
-----
LinuxUbuntu_5_3_0-70-generic_profilex64 - A Profile for Linux Ubuntu_5.3.0-70-generic_profile x64
VistaSP0x64                                - A Profile for Windows Vista SP0 x64
VistaSP0x86                                - A Profile for Windows Vista SP0 x86
VistaSP1x64                                - A Profile for Windows Vista SP1 x64
VistaSP1x86                                - A Profile for Windows Vista SP1 x86
VistaSP2x64                                - A Profile for Windows Vista SP2 x64
VistaSP2x86                                - A Profile for Windows Vista SP2 x86
Win10x64                                    - A Profile for Windows 10 x64
Win10x64_10240_17770                         - A Profile for Windows 10 x64 (10.0.10240.17770 / 2018-02-10)
```



## 1. What is the IP and port the attacker used for the reverse shell?

Una vez configurado el entorno de análisis, el primer objetivo consistió en identificar los parámetros de conectividad empleados por el adversario para establecer la reverse shell. Para ello se recurrió al *plugin linux\_netstat* de Volatility2, cuya funcionalidad permite extraer un inventario exhaustivo de las conexiones activas presentes en el volcado de memoria. La inspección de estos registros reveló la dirección IP remota y el puerto de destino utilizados por el atacante, elementos que constituyen indicadores de compromiso de primer orden.

La identificación de la dirección IP y el puerto empleado por el adversario para establecer la reverse shell no constituye únicamente un dato técnico aislado, sino que se inscribe en una tipología de comportamiento ofensivo ampliamente documentada. En la taxonomía de **MITRE ATT&CK**, este patrón se corresponde con la técnica **T1071 – Application Layer Protocols**, que describe el uso de protocolos de capa de aplicación —como HTTP, HTTPS, DNS o incluso servicios personalizados— para encapsular el tráfico de comando y control.

El valor estratégico de esta técnica radica en su capacidad para camuflar la comunicación maliciosa dentro de flujos legítimos, dificultando la detección por parte de mecanismos defensivos tradicionales. En el caso analizado, la reverse shell se articuló sobre un canal de red que, a primera vista, podía confundirse con una conexión ordinaria, lo que evidencia la intencionalidad del atacante de aprovechar la ubicuidad y permisividad de los protocolos estándar para garantizar persistencia y control remoto.

Desde una perspectiva defensiva, este hallazgo subraya la necesidad de implementar soluciones de inspección profunda de paquetes (*Deep Packet Inspection*) y correlación contextual de eventos, capaces de discriminar entre tráfico benigno y patrones anómalos asociados a actividad de comando y control.

```
[usuari@kali: ~/HTB/APTNightmare-2] $ python volatility.py --dump-mem --profile=linuxUbuntu_5_3_0-70-generic_profilex64 linux_netstat
[...]
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.apinhooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.shellbags (ImportError: No module named Crypto.Hash)

UNIX 38663 update-notifier/3578
TCP 10.0.2.15 :55704 10.0.2.6 : 443 ESTABLISHED bash/3633
UNIX 34615 deja-dup-monito/3646
UNIX 34616 deja-dup-monito/3646
UNIX 34618 deja-dup-monito/3646
UNIX 33679 deja-dup-monito/3646
UNIX 41248 dhclient/3799
UDP 0.0.0.0 : 68 0.0.0.0 : 0 dhclient/3799
```

## 2. What was the PPID of the malicious reverse shell connection?

El segundo eje de la investigación se centró en la determinación del **PPID** asociado a la conexión de reverse shell maliciosa. La aproximación inicial consistió en rastrear la genealogía de procesos mediante el *plugin linux\_pstree*, cuya finalidad es reconstruir la jerarquía de ejecución en el sistema. Sin embargo, el análisis reveló una anomalía significativa: el proceso correspondiente a la reverse shell aparecía desprovisto de progenitor, circunstancia que constituye un indicio inequívoco de técnicas de ocultación empleadas por el malware.

```
[usuari@kali: ~/HTB/APTNightmare-2] $ python volatility.py --dump-mem --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 linux_pstree
[...]
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.apinhooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)

.pulseaudio 2287 1000
.packagekitd 2734
.ibus-x11 2797 1000
.boltd 2900
.gsd-printer 3106 1000
.colord 3110 116
.fwupd 3373
.systemd 3497
.(sd-pam) 3498
.bash 3633
[kthreadd] 2
.[rcu_gp] 3
```

Por tanto, para responder con rigor a la cuestión del PPID, fue necesario recordar la mecánica de asignación de identificadores en sistemas Linux. Cada vez que se instancia un nuevo proceso, el núcleo le adjudica un **PID** a partir de un contador global que se incrementa secuencialmente.



De manera análoga, el **PPID** refleja la relación jerárquica con el proceso creador, constituyendo un metadato esencial para reconstruir la cadena de ejecución. La ausencia de este vínculo en el volcado analizado no obedece a un error del sistema, sino a una manipulación deliberada del adversario para encubrir la procedencia de la reverse shell y, con ello, obstaculizar la atribución del ataque.

### **3. Provide the name of the malicious kernel module.**

El tercer vector de análisis se orientó hacia la identificación del **módulo de kernel malicioso** cargado en el sistema. La primera aproximación se realizó mediante el *plugin linux\_lsmod*, cuya función consiste en recorrer la estructura enlazada *modules.list* a partir del símbolo *modules*, generando un inventario de los módulos presentes en memoria. Este procedimiento, en condiciones normales, permite detectar extensiones cargadas en el núcleo y constituye una técnica forense habitual para descubrir artefactos sospechosos.

Sin embargo, la inspección inicial no reveló ninguna anomalía aparente, lo que puso de manifiesto la sofisticación del implante. Ante esta ausencia de evidencias superficiales, fue necesario recurrir a un enfoque más incisivo: el *plugin linux\_hidden\_modules*, diseñado para escudriñar la memoria en busca de módulos que han sido deliberadamente ocultados mediante la manipulación de estructuras internas del kernel. La capacidad de este *plugin* para “carvear” regiones de memoria y detectar artefactos invisibilizados resultó crucial para desvelar la presencia del módulo malicioso, cuya ocultación constituye un claro ejemplo de técnicas avanzadas de evasión.

La ocultación deliberada del módulo de kernel identificado se inscribe en un conjunto articulado de tácticas ofensivas que la matriz **MITRE ATT&CK** sistematiza con notable claridad. En primer lugar, la técnica **T1014 – Rootkit** describe la inserción de componentes en el núcleo del sistema operativo con el fin de modificar su comportamiento y garantizar tanto la persistencia como la invisibilidad de la actividad adversaria. Los *rootkits* a nivel de kernel poseen la capacidad de interceptar llamadas al sistema, manipular estructuras internas y alterar la visibilidad de procesos, conexiones o archivos, lo que los convierte en uno de los mecanismos más sofisticados de evasión.



La inspección preliminar de los procesos activos no arrojó indicios de anomalías evidentes, circunstancia que, lejos de disipar las sospechas, reforzó la hipótesis de que el adversario había desplegado mecanismos avanzados de ocultación. Ante la ausencia de evidencias superficiales, resultó imprescindible recurrir a un enfoque más incisivo: el *plugin linux\_hidden\_modules*, cuya funcionalidad consiste en escudriñar la memoria mediante técnicas de *carving* para localizar módulos de kernel invisibilizados deliberadamente. Este procedimiento permitió desvelar la presencia del artefacto malicioso, cuya ocultación constituye un ejemplo paradigmático de tácticas de *defense evasion*.

El hallazgo del módulo oculto constituye un ejemplo paradigmático de esta técnica, evidenciando la intencionalidad del atacante de operar en el nivel más privilegiado del sistema.

```
(usuario㉿kali)-[~/HTB/APTNightmare-2]
└─$ python2 volatility/vol.py -f dump.mem --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 linux_hidden_modules
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apiohooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userrassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getsids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.eventlog (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.audit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpergistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apiohooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.registry.amcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mech.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.registrypi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apiohooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envvars (ImportError: No module named Crypto.Hash)
Offset (V)      Name
0xffffffffffffc053a280 nfentlink
```

#### 4. What time was the module loaded?

El cuarto eje de la investigación se orientó hacia la determinación de la **marca temporal asociada a la carga del módulo malicioso** en el núcleo. Para ello resultó imprescindible rastrear los registros del sistema, concretamente el archivo *kern.log*, cuya función es consignar eventos críticos relacionados con el kernel. La aproximación metodológica consistió en emplear el *plugin linux\_enumerate\_files*, con el fin de localizar el número de *inode* correspondiente a dicho registro dentro del volcado de memoria. Una vez identificado, se procedió a su volcado para disponer de una copia íntegra que permitiera un examen detallado.

```
(usuario㉿kali)-[~/HTB/APTNightmare-2]
└─$ python2 volatility/vol.py -f dump.mem --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 linux_enumerate_files | grep "/var/log"
Volatility Foundation Volatility Framework 2.6.1
0xfffff98ea352040e8 270157 /var/log
0xfffff98ea4247d1e8 262456 /var/log/Xorg.0.log
0xfffff98ea45df1a68 1058154 /var/log/mysql
0xfffff98ea45df2b68 1060107 /var/log/mysql/error.log
0xfffff98ea424ae668 1058146 /var/log/apache2
0xfffff98ea424533e8 1051726 /var/log/apache2/access.log
0xfffff98ea42450968 1058600 /var/log/apache2/other_vhosts_access.log
0xfffff98ea42453828 1058610 /var/log/apache2/error.log
0xfffff98ea593fa8 270499 /var/log/cups
0xfffff98ea4d00073e8 271057 /var/log/cups/access_log
0xfffff98ea4d092728 534923 /var/log/webkit
0xfffff98ea54730928 262161 /var/log/syslog
0xfffff98ea54732fa8 262162 /var/log/kern.log
0xfffff98ea5a670968 262163 /var/log/auth.log
```

Sin embargo, en este caso, la persistencia del evento en *kern.log* ofreció una ventana de visibilidad que permitió al analista forense desvelar la maniobra adversaria.

```
(usuario㉿kali)-[~/HTB/APTNightmare-2]
└─$ python2 volatility/vol.py -f dump.mem --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 linux_find_file -i 0xfffff98ea5a732fa8 -o kernel.log
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timeliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apiohooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.userrassist (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shellbags (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.eventlog (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.audit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpergistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apiohooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.mac.check_syscall_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.registrypi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apiohooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envvars (ImportError: No module named Crypto.Hash)
```



En un contexto donde los adversarios despliegan técnicas avanzadas de ocultación, la capacidad de reconstruir la cronología de la intrusión se convierte en un elemento esencial para la respuesta y la mitigación.

```
[user@host-hat] ~/HTB/APTNightmare-2]
[1] 1 kernel.log
ice [0x00000000] (multi-head: yes rom: no)
May 1 16:14:12 server1 kernel: [ 8.376825] ice: Video Bus as /devices/PCI0000:00/LNXGUS00/PNP0A03:00/LNXVIDEO:00/input/input6
May 1 16:14:12 server1 kernel: [ 8.376825] ice: probe: loading out-of-tree module taints kernel.
May 1 16:14:12 server1 kernel: [ 8.376833] ydvrheartbeatInit: Setting up heartbeat to trigger every 2000 milliseconds
May 1 16:14:12 server1 kernel: [ 8.380380] input: Unspecified device as /devices/pci0000:00/0000:00:10.0/input/input7
May 1 16:14:12 server1 kernel: [ 8.381946] vboxguest: misc device minor_5, IRQ 20, 1/0 port 0x400, MMIO at 0x000000f0e00000 (size 0x400000)
May 1 16:14:12 server1 kernel: [ 8.382000] vboxguest: vga redirection driver version 6.0.14-Unturned (interface 0x00001004)
May 1 16:14:12 server1 kernel: [ 8.469983] cryptd: max cu gen set to 1000
May 1 16:14:12 server1 kernel: [ 9.334163] AVX2 version of gcm_enc/dec engaged
May 1 16:14:12 server1 kernel: [ 9.334167] AES CTR mode byt optimized code enabled
May 1 16:14:12 server1 kernel: [ 20.747767] br0: port 1(veth) entered forwarding state
May 1 16:14:20 server1 kernel: [ 20.747767] br0: port 1(veth) left forwarding state
May 1 16:14:20 server1 kernel: [ 20.747767] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s5: Link becomes ready
May 1 16:14:28 server1 kernel: [ 28.210000] aufs 4.x-rch-20190805
May 1 16:14:31 server1 kernel: [ 31.424540] binder_print_skb: 1000 callbacks suppressed
May 1 16:14:31 server1 kernel: [ 31.424540] binder: binder_alloc_death(1716085271,111131): apparmor="STATUS" operation="profile_load" profile="" name="docker-default" pid=1613 comm="apparmor_parser"
May 1 16:14:34 server1 kernel: [ 34.664864] bridge: filtering via arp/dpitabletables is no longer available by default. Update your scripts to load br_netfilter if you need this.
May 1 16:14:34 server1 kernel: [ 34.73953] Bridge Firewalling registered
May 1 16:14:34 server1 kernel: [ 34.559619] bpftrace: Loaded module bpftrace with pid 1647
May 1 16:14:34 server1 kernel: [ 34.559619] bpftrace: loaded module bpftrace
May 1 20:41:06 server1 kernel: [ 46.027391] traps: appstreamd[2768]: trap int3 ip:7f7088193fc1 sp:7fd4d6f80 error:0 in liblbb-2.0.so.0.5600.4[7f70881a2000+114000]
May 1 20:41:11 server1 kernel: [ 50.810752] kfree: rmmod: handler disabled
May 1 20:42:57 server1 kernel: [ 156.007374] nfnetlink: module verification failed: signature and/or required key missing - tainting kernel
```

#### **5. What is the full path and name of the malicious kernel module file?**

El quinto eje de la investigación se orientó hacia la determinación de la **ruta completa y denominación del archivo correspondiente al módulo de kernel malicioso**. Para abordar esta cuestión resultó imprescindible detenerse en algunos aspectos fundamentales de la arquitectura de módulos en Linux. Los módulos de kernel, habitualmente compilados en formato .ko, constituyen extensiones dinámicas que el núcleo puede cargar para ampliar sus funcionalidades, desde controladores de dispositivos hasta subsistemas de red. La identificación de un módulo espurio exige, por tanto, una enumeración sistemática de los directorios que albergan estos artefactos, con el objetivo de localizar cualquier archivo que se aparte de la estructura legítima.

La inspección de los ficheros presentes en el sistema reveló la existencia de un archivo denominado **net/nfnetlink.ko**, cuya ubicación y nomenclatura resultaban sospechosas. El contraste con la ruta legítima **/net/netfilter/nfnetlink.ko** permitió constatar que el adversario había desplegado un módulo malicioso que se hacía pasar por el componente legítimo del subsistema de filtrado de red. Este hallazgo se inscribe en la técnica **T1036 – Masquerading** de la matriz **MITRE ATT&CK**, que describe la manipulación de nombres, rutas o atributos de archivos con el propósito de disfrazar artefactos maliciosos como componentes legítimos del sistema. La esencia de esta táctica radica en la explotación de la confianza implícita que los administradores y mecanismos defensivos depositan en la nomenclatura convencional de los binarios y módulos. Al replicar la apariencia de un archivo legítimo —en este caso, el módulo **nfnetlink.ko**— el adversario consigue que su implante se confunda con el subsistema de filtrado de red, reduciendo drásticamente las probabilidades de ser detectado en una inspección superficial.

La sofisticación de esta técnica reside en su capacidad para integrarse de manera verosímil en la estructura del sistema, dificultando la discriminación entre artefactos benignos y maliciosos. En términos defensivos, la detección de *masquerading* exige una correlación minuciosa entre rutas esperadas y ubicaciones reales, así como la verificación de firmas digitales y metadatos asociados. La discrepancia entre la ruta legítima y la ruta maliciosa constituye un indicador de compromiso de primer orden, que pone de relieve la intencionalidad del adversario de operar bajo el amparo de la apariencia legítima.

La elección de un nombre tan próximo al legítimo no obedece a la casualidad, sino a una intencionalidad ofensiva de camuflar el artefacto dentro de la estructura convencional del sistema, dificultando la identificación por parte de mecanismos defensivos automatizados.



Desde la perspectiva forense, este hallazgo subraya la necesidad de una inspección minuciosa de los directorios de módulos y de una correlación con las rutas esperadas, ya que la detección de discrepancias constituye un indicador de compromiso de primer orden.

```
[~] $ python2 volatility/vol.py -f dump.mem --profile=linuxUbuntu_5.3.0-70-generic_profilex64 linux_enumerate_files | grep "\.ko$"
Volatility Foundation Volatility Framework 2.6.1
0xfffff908ea4275e2e8 416211 /lib/modules/5.3.0-70-generic/kernel/drivers/net_ne_failover.ko
0xfffff908ea4275d628 416229 /lib/modules/5.3.0-70-generic/kernel/drivers/net_vxlan.ko
0xfffff908ea4275c968 416129 /lib/modules/5.3.0-70-generic/kernel/drivers/net_dummy.ko
0xfffff908ea4275d668 416201 /lib/modules/5.3.0-70-generic/kernel/drivers/net_mii.ko
0xfffff908ea4275d152 416208 /lib/modules/5.3.0-70-generic/kernel/drivers/net_macvtap.ko
0xfffff908ea42661452 416209 /lib/modules/5.3.0-70-generic/kernel/drivers/net_mmc.ko
0xfffff908ea42663468 2352708 /lib/modules/5.3.0-70-generic/kernel/drivers/net_nfnetlink.ko
0xfffff908ea42667468 416200 /lib/modules/5.3.0-70-generic/kernel/drivers/net_genetic.ko
0xfffff908ea426598068 416224 /lib/modules/5.3.0-70-generic/kernel/drivers/net_tap.ko
0xfffff908ea42659a6b8 416203 /lib/modules/5.3.0-70-generic/kernel/drivers/net_ifb.ko
0xfffff908ea426599ea8 416207 /lib/modules/5.3.0-70-generic/kernel/drivers/net_macvlan.ko
0xfffff908ea426599e8 416212 /lib/modules/5.3.0-70-generic/kernel/drivers/net_ncconsole.ko
0xfffff908ea426599e8 416225 /lib/modules/5.3.0-70-generic/kernel/drivers/net_vx.ko
0xfffff908ea426599e8 530213 /lib/modules/5.3.0-70-generic/kernel/net/netfilter/nf_log_netdev.ko
0xfffff908ea426599e8 530286 /lib/modules/5.3.0-70-generic/kernel/net/netfilter_xt_cluster.ko
0xfffff908ea426599e8 530285 /lib/modules/5.3.0-70-generic/kernel/net/netfilter/xt_cgroupl.ko
0xfffff908ea4272de728 530203 /lib/modules/5.3.0-70-generic/kernel/net/netfilter/nf_conntrack_netlink.ko
0xfffff908ea4272dcda8 530223 /lib/modules/5.3.0-70-generic/kernel/net/netfilter/nfnetlink.ko
0xfffff908ea42778968 529095 /lib/modules/5.3.0-70-generic/kernel/net/bridge/br_nfnetfilter.ko
0xfffff908ea42778968 529144 /lib/modules/5.3.0-70-generic/kernel/net/bridge/bridge.ko
0xfffff908ea42778528 530157 /lib/modules/5.3.0-70-generic/kernel/net/l1c/l1c.ko
0xfffff908ea42779ea8 528229 /lib/modules/5.3.0-70-generic/kernel/net/802/stp.ko
```

En este punto emergió una cuestión aparentemente paradójica: mientras que el archivo malicioso identificado respondía a la denominación **nfnetlink.ko**, el módulo cargado en el núcleo aparecía bajo el nombre **nfentlink**. La reconciliación de esta discrepancia exigió un análisis más profundo del binario, recurriendo a técnicas de descompilación y a la inspección de su tabla de símbolos mediante la utilidad **readelf**. El examen reveló la presencia de la cadena **nfentlink.c**, probablemente utilizada como referencia durante la fase de compilación, lo que explica la divergencia entre el nombre del archivo y el identificador interno del módulo.

```
[~] $ readelf -s nfnetlink.ko
Symbol table '.symtab' contains 89 entries:
Num: Valor Tam Tipo Unión Vis Nombre Ind
0: 0000000000000000 0 NOTYPE LOCAL DEFAULT UND
1: 0000000000000000 0 SECTION LOCAL DEFAULT 1 .note.gnu.build-id
2: 0000000000000000 0 SECTION LOCAL DEFAULT 2 .text
3: 0000000000000000 0 SECTION LOCAL DEFAULT 4 .text.unlikely
4: 0000000000000000 0 SECTION LOCAL DEFAULT 6 .init.text
5: 0000000000000000 0 SECTION LOCAL DEFAULT 8 .exit.text
6: 0000000000000000 0 SECTION LOCAL DEFAULT 10 .parainstructions
7: 0000000000000000 0 SECTION LOCAL DEFAULT 12 .rodata.str1.1
8: 0000000000000000 0 SECTION LOCAL DEFAULT 13 .rodata.str1.2
9: 0000000000000000 0 SECTION LOCAL DEFAULT 15 .rodata.str1.8
10: 0000000000000000 0 SECTION LOCAL DEFAULT 16 .smp.locks
11: 0000000000000000 0 SECTION LOCAL DEFAULT 18 .rodata
12: 0000000000000000 0 SECTION LOCAL DEFAULT 19 .modinfo
13: 0000000000000000 0 SECTION LOCAL DEFAULT 20 .note.Linux
14: 0000000000000000 0 SECTION LOCAL DEFAULT 21 .data
15: 0000000000000000 0 SECTION LOCAL DEFAULT 23 __bug_table
16: 0000000000000000 0 SECTION LOCAL DEFAULT 25 .gnu.linkonce.th[...]
17: 0000000000000000 0 SECTION LOCAL DEFAULT 27 .bss
18: 0000000000000000 0 SECTION LOCAL DEFAULT 28 .comment
19: 0000000000000000 0 SECTION LOCAL DEFAULT 29 .note.GNU-stack
20: 0000000000000000 0 FITF ILOCAL DEFAULT ABS nfentlink.c
21: 0000000000000000 13 FUNC LOCAL DEFAULT 4 read_cr0 ..
```

La ejecución de **strings** sobre el artefacto malicioso y la posterior búsqueda de referencias a **nfentlink** permitió corroborar esta hipótesis. El hallazgo más relevante fue la presencia de la directiva **name=nfentlink**, que constituye un atributo fundamental en la definición de módulos de kernel. En términos técnicos, este parámetro suele establecerse mediante macros como **MODULE\_INFO(name, "nfentlink")**; lo que habilita al módulo para autodefinirse con un nombre específico independientemente de la nomenclatura del archivo que lo contiene.

```
[~] $ strings nfentlink.ko | grep "nfentlink"
name=nfentlink
nfentlink
nfentlink.c
nfentlink.mod.c
```



## 6. What's the MD5 hash of the malicious kernel module file?

El sexto eje de la investigación se centró en la obtención del **hash criptográfico MD5** correspondiente al archivo del módulo de kernel malicioso previamente identificado. Partiendo del número de *inode* recuperado en la fase anterior, se procedió al volcado íntegro del fichero, garantizando así la preservación de su contenido para un examen más detallado.

```
[usuario@kali: ~/HTB/APTNightmare-2]
$ ./pythont2 volatility/vol --nofile=LinuxUbuntu_5_3_0-70-generic_profile64 linux_find_file -i 0xfffff98ea2665a68 -o nfnetlink.ko
Volatility Framework Version: 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservicesids (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.timelines (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apiohooks (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.servicediff (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.uservassis (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.getservices (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apimatch (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.evlogts (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.lsadmp (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.malware.apimatch (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.apimatch (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.apimatch (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware.svcscan (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.auditpol (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registerapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apimatch (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.getenvs (ImportError: No module named Crypto.Hash)
```

Una vez aislado el artefacto, se generó su huella digital mediante el algoritmo MD5, obteniéndose un identificador único que permite caracterizar el archivo con independencia de su nombre o ubicación.

```
[usario@kali] -~/HTB/APTNightmare-2]
└─$ file nfnetlink.ko
nfnetlink.ko: ELF 64-bit LSB relocatable, x86_64, version 1 (SYSV), BuildID[sha1]=6ff658cc4a1ee80529ca205c7548169e79d677d3, not stripped
[usario@kali] -~/HTB/APTNightmare-2]
└─$ md5sum nfnetlink.ko
35bd8664b021b862a0e50b13e0a57f7 nfnetlink.ko
```

La generación de este hash constituye un paso esencial en la praxis forense, ya que posibilita la correlación del artefacto con bases de datos de inteligencia de amenazas. En este caso, la consulta del valor en **VirusTotal** proporcionó evidencia adicional de la naturaleza maliciosa del módulo, al asociarlo con firmas previamente reportadas por motores antivirus y sistemas de detección.

www.virustotal.com/gui/file/3cd556862470b38503f79d35e21008b11f19639a92538ee14dccea228817

Community Score: 29 / 100

29/65 security vendors flagged this file as malicious

3cd556862470b38503f79d35e21008b11f19639a92538ee14dccea228817  
infnetlink.ko

File · Retocable · Details

Size: 13.04 KB · Last Analysis Date: 8 days ago · AV-ELF

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label	Trojan.Rootkit.netlink	Threat categories	Trojan	Family labels	rootkit netlink Kali
Security vendors' analysis	Do you want to automate checks?				
AllCloud	Rootkit.Linux.Bifrp.vB	AIYec	Trojan.Generic.37973914		
Arcabit	Trojan.Generic.D2436f9A	Avast	ELF:Netlink-A [Rtk]		
AVG	ELF:Netlink-A [Rtk]	Avira [no cloud]	LINUX/AVI.Netlink.xgph		
Bitdefender	Trojan.Generic.37973914	CTX	ElfTrojan.netlink		
Cynet	Malicious (Score: 99)	DrWeb	Linux.Rootkit.Diamorphine.18		
Elastic	Linux.Generic.Threat	Emsisoft	Trojan.Generic.37973914 (8)		
eScan	Trojan.Generic.37973914	ESET-NOD32	Linux/Rootkit.Agent.PW.Trojan		
Fortinet	Possible.Threat	GData	Trojan.Generic.37973914		
Google	Patented	Ikarus	Trojan.I Linux Bookie		



## 7. What is the full path and name of the legitimate kernel module file?

El séptimo eje de la investigación se centró en la identificación del **archivo legítimo de kernel** frente al cual se produjo la suplantación. Tal como se expuso en la quinta pregunta, la inspección de los directorios reveló la existencia del artefacto malicioso **net/nfnetlink.ko**, cuya nomenclatura buscaba imitar al componente auténtico. En la captura correspondiente a esa fase del análisis puede observarse con claridad la ruta legítima **/net/netfilter/nfnetlink.ko**, que constituye la referencia original del subsistema de filtrado de red.

La comparación entre ambas rutas —ya ilustrada en la evidencia gráfica de la quinta pregunta— permite constatar la intencionalidad ofensiva de camuflar el implante bajo la apariencia de un módulo legítimo. Este hallazgo refuerza la correlación con la técnica **T1036 – Masquerading** de la matriz **MITRE ATT&CK**, en tanto que el adversario replicó la apariencia de un componente legítimo para evadir la detección.

## 8. What is the single character difference in the author value between the legitimate and malicious modules?

El octavo eje de la investigación se orientó hacia la identificación de discrepancias en los metadatos asociados al módulo malicioso, concretamente en el campo **author**. Para ello se recurrió al comando **modinfo**, cuya funcionalidad permite extraer información detallada sobre los módulos de kernel, incluyendo atributos como versión, descripción, dependencias y autoría.

El análisis reveló una diferencia sutil pero significativa entre el módulo legítimo y el artefacto malicioso: mientras el primero consignaba la dirección de correo **laforge@netfilter.org**, el segundo mostraba la variante alterada **laforge@netflter.org**, en la que se había suprimido la letra *i* en el término *netfilter*. Esta modificación mínima constituye un ejemplo paradigmático de cómo los adversarios emplean técnicas de **masquerading** (MITRE ATT&CK **T1036**) para camuflar sus artefactos bajo la apariencia de legitimidad, introduciendo alteraciones apenas perceptibles que pueden pasar inadvertidas en una inspección superficial.

```
(usuario㉿kali)-[~/HTB/APTNightmare-2]
└─$ modinfo nfnetlink.ko
filename:      /home/usuario/HTB/APTNightmare-2/nfnetlink.ko
description:   Netfilter messages via netlink socket
author:        Harald Welte <laforge@netfilter.org>
license:       GPL
srcversion:    ABB5B6039A88BF0DBAE6939
depends:
retpoline:     Y
name:          nfnetlink
vermagic:      5.3.0-70-generic SMP mod_unload

((usuario㉿kali)-[~/HTB/APTNightmare-2]
└─$ modinfo nfnetlink
filename:      /lib/modules/6.16.8+kali-amd64/kernel/net/netfilter/nfnetlink.ko.xz
description:   Netfilter messages via netlink socket
alias:         net-pf-16-proto-12
author:        Harald Welte <laforge@netfilter.org>
license:       GPL
depends:
intree:        Y
name:          nfnetlink
retpoline:     Y
vermagic:      6.16.8+kali-amd64 SMP preempt mod_unload modversions
sig_id:        PKCS#7
signer:        Build time autogenerated kernel key
sig_key:       6A:9A:F7:D9:0F:2B:6B:F6:55:D5:D1:67:D5:77:B7:90:01:C6:DC:BF
sig_hashalgo:  sha256
signature:    30:65:02:31:00:82:B7:3B:4F:3B:63:DF:6B:A6:52:1B:57:14:0B:DE
              EE:19:0B:43:56:EF:A3:60:C2:CA:60:AB:5C:3E:BD:B5:46:AE:23:56
              6A:06:AA:80:9D:E2:D4:55:F2:32:1E:FA:35:02:30:7F:58:9C:9C:93:
              E5:51:9F:CE:0B:30:90:F0:B1:75:9A:96:27:7F:25:BF:D:83:CA:FB:
              94:93:72:FC:9A:A0:89:0B:53:24:34:62:DA:06:A0:B3:D6:73:3B:D1:
              50:97:4E
```

La relevancia forense de este hallazgo radica en que la discrepancia nominal en el campo de autoría no solo confirma la manipulación deliberada del módulo, sino que también aporta un indicador de compromiso de alto valor. La intencionalidad ofensiva se manifiesta en la creación de ambigüedad semántica, donde un único carácter ausente basta para desdibujar la frontera entre lo legítimo y lo malicioso. Desde la perspectiva defensiva, este hallazgo subraya la necesidad de una verificación exhaustiva de metadatos y de la correlación con fuentes de confianza, ya que incluso las alteraciones más sutiles pueden constituir la clave para desvelar un implante oculto.



## 9. What is the name of initialization function of the malicious kernel module?

El noveno eje de la investigación se centró en la identificación de la **función de inicialización del módulo de kernel malicioso**. En la arquitectura de Linux, todo módulo requiere un punto de entrada denominado **init\_module**, que cumple un rol equivalente al de la función *main()* en los ejecutables convencionales. Este mecanismo garantiza que, al cargarse el módulo, se ejecute la rutina de inicialización definida por el desarrollador.

```
(usuario@kali)-[~/HTB/APTNightmare-2]
└$ readelf -W -s nfnetlink.ko

Symbol table 'symtab' contains 89 entries:
Num: Value          Type            Name/Location/Symbol Flags
0: 0000000000000000     OBJECT LOCAL  DEFAULT    .UND
1: 0000000000000000     SECTION LOCAL  DEFAULT    .NOTE.GNU.BUILD-ID
2: 0000000000000000     SECTION LOCAL  DEFAULT    .TEXT
3: 0000000000000000     SECTION LOCAL  DEFAULT    .TEXT.UNLIKELY
4: 0000000000000000     SECTION LOCAL  DEFAULT    .INIT.TEXT
5: 0000000000000000     SECTION LOCAL  DEFAULT    .EXIT.TEXT
6: 0000000000000000     SECTION LOCAL  DEFAULT    .PARADESTRUCTIONS
7: 0000000000000000     SECTION LOCAL  DEFAULT    .RMANAGERCTRL.1
8: 0000000000000000     SECTION LOCAL  DEFAULT    .MCOUNT.LOC
9: 0000000000000000     SECTION LOCAL  DEFAULT    .RODATA.STRL.8
10: 0000000000000000     SECTION LOCAL  DEFAULT    .SMP.LOCKS
11: 0000000000000000     SECTION LOCAL  DEFAULT    .RODATA
12: 0000000000000000     SECTION LOCAL  DEFAULT    .MODINFO
26: 0000000000000008     OBJECT LOCAL  DEFAULT    .FUNC_.5154
27: 0000000000000000     OBJECT LOCAL  DEFAULT    .OPCODE.5154
29: 0000000000000010     OBJECT LOCAL  DEFAULT    .OPCODE.5154
29: 0000000000000040     70 FUNC  LOCAL  DEFAULT    .WITHIN_MODULE.CONSTPROP.6
30: 0000000000000080     42 FUNC  LOCAL  DEFAULT    .FH.FTRACE.THUNK
31: 0000000000000040     55 FUNC  LOCAL  DEFAULT    .LIST.DEL.CONSTPROP.7
32: 0000000000000000     37 FUNC  LOCAL  DEFAULT    .LIST.ADD.CONSTPROP.8
33: 000000000000000d     11 FUNC  LOCAL  DEFAULT    .CLEAR_BIT.CONSTPROP.9
34: 0000000000000018     11 FUNC  LOCAL  DEFAULT    .SET_BIT.CONSTPROP.10
34: 0000000000000000     248 FUNC LOCAL  DEFAULT    .CONSTPROP.11
36: 0000000000000000     248 FUNC LOCAL  DEFAULT    .NFNETLINK_INIT
38: 00000000000000120    8 OBJECT LOCAL  DEFAULT    .SYS_CALL_TABLE
39: 0000000000000000     18 FUNC  LOCAL  DEFAULT    .NFNETLINK_EXIT
40: 0000000000000000     50 OBJECT LOCAL  DEFAULT    .UNIQUE_ID_DESCRIPTION106
41: 0000000000000000     43 OBJECT LOCAL  DEFAULT    .UNIQUE_ID_AUTHOR105
42: 000000000000005d     12 OBJECT LOCAL  DEFAULT    .UNIQUE_ID_LICENSE104
43: 0000000000000010     0 NOTYPE LOCAL  DEFAULT    .LC2
```

El análisis mediante herramientas de desensamblado permitió observar la presencia de una función denominada **nfnetlink\_init**, distinta de *init\_module*. Para profundizar en esta aparente discrepancia, se recurrió a la utilidad **nm** con la opción *-n*, que ordena los símbolos de manera numérica. El resultado mostró que *init\_module* aparecía con la etiqueta de **Global Function (T)** en la misma dirección de memoria que *nfnetlink\_init*, marcada como **Local Function (t)**. Esta coincidencia sugiere que, en tiempo de ejecución, la invocación de *init\_module* deriva en la ejecución de *nfnetlink\_init*, lo que constituye una estrategia de redirección interna.

```
(usuario@kali)-[~/HTB/APTNightmare-2]
└$ nm -n nfnetlink.ko | grep init
0000000000000000 T init_module
0000000000000000 t nfnetlink_init
└$
```

La confirmación de este comportamiento se obtuvo al abrir el archivo **nfnetlink.ko** en **Ghidra**, donde el análisis heurístico y la resolución de símbolos mostraron ambas denominaciones coexistiendo en el binario. Este hallazgo evidencia la intencionalidad del adversario de manipular la semántica interna del módulo, camuflando la función real de inicialización bajo la apariencia del punto de entrada estándar.

```
Listing: nfnetlink.ko
=====
* FUNCTION
=====
undefined undefined init_module()
undefined undefined nfnetlink_init
Stack[-0x10]@8:local_18
Stack[-0x20]@8:local_20
Stack[-0x30]@8:local_30
Stack[-0x38]@8:local_38
Stack[-0x40]@8:local_40
XREF[2]: 001000a0(W),
00100044(R)
XREF[1]: 001000ae(W)
XREF[1]: 00100090(W)
XREF[1]: 0010007f(W)
XREF[2]: 0010007f(*),
00100078(W)
XREF[6]: 00100095(*),
00100097(W),
00100092(R),
00100030(*),
00100034(W),
00100039(R)
Entry Point (*)
_elfSectionHeaders::00000190(*)
undefined _fentry_()

0010076c e8 bf 1b    CALL    <EXTERNAL>:_fentry_
00100771 50          PUSH    RBP
00100772 b0 01 00    MOV     ECX,0x1
00100773 00          XOR    EDX,EDX
00100774 31 d2        XOR    EDX,EDX
00100775 48 c7 c7    MOV     RDI,$./bin/bash_0010008b0
00100776 48 89 00 00  mov    rax,0x0
00100777 31 d2        XOR    EDX,EDX
00100778 48 c7 c7    MOV     RDI,RSP
00100779 48 89 00 00  mov    rax,0x0
00100780 48 99 ec 30  sub    RSP,RSP
00100783 48 93 ec 30  sub    RSP,RSP
00100787 48 8d 75 d8  lea    RSI=>local_38,[RBP + local_38]
00100788 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100789 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100790 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100793 48 93 ec 30  sub    RSP,RSP
00100797 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100798 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100799 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079a 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079b 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079c 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079d 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079e 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
0010079f 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a4 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a5 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a6 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a7 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a8 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a9 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a0 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a1 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a2 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
001007a3 48 8d 7d d8  lea    RDI=>_sys_call_table+0x28
00100
```

10. There is a function for hooking syscalls. What is the last syscall from the table?

El décimo eje de la investigación se orientó hacia la identificación de la **última llamada al sistema (syscall) enganchada por el módulo malicioso**. Para abordar esta cuestión se recurrió nuevamente al análisis del binario **nfnetlink.ko** mediante la herramienta **Ghidra**, iniciando la inspección en la función **init\_module**, que constituye el punto de entrada del artefacto. Desde allí se navegó hacia la estructura **sys\_call\_table**, donde se encuentran referenciadas las funciones del núcleo que pueden ser interceptadas o redirigidas por módulos cargados dinámicamente.

La revisión detallada de esta tabla permitió constatar la presencia de una rutina de *hooking* que manipulaba las llamadas al sistema, alterando su comportamiento legítimo.

**11. What signal number is used to hide the process ID (PID) of a running process when sending it?**

El undécimo eje de la investigación se centró en la identificación del **número de señal empleado por el módulo malicioso para ocultar el PID de un proceso en ejecución**. El análisis del código desensamblado reveló la presencia de la función **hook\_kill**, diseñada para interceptar las invocaciones a la syscall kill. Esta rutina recibe un objeto y evalúa el número de señal en el desplazamiento **0x68**.

La lógica implementada es clara: si el valor de la señal no corresponde a **0x40**, la función delega la ejecución en la syscall original y retorna el resultado esperado. Sin embargo, cuando el valor coincide con **0x40**, el módulo almacena el PID del proceso objetivo en una estructura interna destinada a ocultarlo y retorna **0**, evitando así que el proceso sea visible en listados convencionales. La conversión de **0x40** a formato decimal revela el número de señal **64**, que constituye el identificador utilizado por el adversario para activar la rutina de ocultación.