Desarrollo: Hammered

Q1:

¿Qué servicio utilizaron los atacantes para acceder al sistema?

1) - Revise los registros (Logs) del sistema para detectar errores de autenticación. Estos registros pueden ayudarle a identificar el servicio que los atacantes intentaron usar para acceder.

2) - El archivo **auth.log** contiene registros de los intentos de autenticación. Busque fallos de autenticación repetidos para un servicio específico.

3) - Utilice **grep "fail" auth.log** para identificar los intentos de autenticación fallidos.

NOTA: El archivo auth.log registra eventos de autenticación y uso de privilegios en Linux. Se encarga de almacenar:

- **Intentos de login por SSH**
- **Intentos fallidos de contraseña**
- **Logins exitosos**
- **Uso de sudo**
- **Cambios de usuario (su)**
- **Creación de usuarios**
- **Actividad relacionada con credenciales**

Es el registro más importante para investigar accesos al sistema.

La salida de **`grep "fail" auth.log`** muestra intentos fallidos de autenticación SSH:

```
Apr 24 10:34:39 app-1 sshd[23375]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=124.51.108.68

Apr 24 10:37:32 app-1 sshd[23461]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=124.51.108.68
Apr 24 11:10:32 app-1 sshd[23539]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=121.11.66.70  user=root
```

Interpretación: Se registraron múltiples intentos fallidos de autenticación SSH desde direcciones IP externas durante la fase de validación PAM, incluyendo intentos dirigidos al usuario root.

Los registros muestran que `sshd` corresponde al servicios SSH que gestiona la conexión y `pam_unix(sshd:auth)` es el modulo PAM que valida credenciales.

Respuesta correcta de Q1: SSH ✓

Q2:

¿Cuál es la versión del sistema operativo del sistema de destino?

1) - El registro **dmesg** registra los mensajes del kernel y del sistema, incluidos detalles sobre la versión del sistema operativo durante el arranque.

2) - Ejecute **head dmesg** para obtener información sobre la versión de Linux. Busque la línea que contiene la **versión de Linux**, que revelará la versión del sistema operativo.

NOTA: El archivo dmesg muestra mensajes del kernel de Linux almacenados en memoria desde que el sistema arrancó. Estos mensajes describen:

- **Inicialización del kernel**
- **Detección de hardware**
- **Configuración de subsistemas internos**

La salida muestra información de la distribución base del sistema, en este caso, el sistema utiliza un Ubuntu 2.6.24-26.64-server**:**

```
checho@Windows-localhost:~/Hammered$ head dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 2.6.24-26-server (buildd@crested) (gcc version 4.2.4 (Ubuntu 4.2.4-1ubuntu3)) #1 SMP Tue Dec 1 18:26:43 UTC 2009 (Ubuntu 2.6.24-26.64-serve
r)
[    0.000000] Command line: root=UUID=a691743a-a4b7-482d-95ff-406e5acd83a3 ro quiet splash
[    0.000000] BIOS-provided physical RAM map:
[    0.000000]  BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
[    0.000000]  BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
[    0.000000]  BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
[    0.000000]  BIOS-e820: 00000000000dc000 - 00000000000e4000 (reserved)
[    0.000000]  BIOS-e820: 00000000000e8000 - 0000000000100000 (reserved)
```

El kernel de Linux fue compilado usando la version 4.2.4 de GCC**.**

Respuesta correcta de Q2: 4.2.4-1ubuntu3 ✓

Q3:

¿Cuál es el nombre de la cuenta comprometida?

1) - Consulte el archivo **auth.log** para encontrar información sobre las cuentas comprometidas. Céntrese en las cuentas con intentos de inicio de sesión inusuales o frecuentes.

2) - Busque intentos de autenticación relacionados con cuentas con **uid=0**, que normalmente se asocian con cuentas privilegiadas.

NOTA: UID 0 = usuario root (superusuario)

Al ejecutar **grep uid=0 auth.log** en el sistema, mostrara todos los eventos del sistema donde hubo participación directa del usuario root o privilegios root.

La salida muestra que la cuenta comprometida corresponde al usuario root en eventos de autenticación por SSH y ejecución de tareas programas con privilegios:

```
Apr 24 13:49:11 app-1 sshd[27801]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=8.12.45.242  user=root
```

```
Apr 25 10:06:01 app-1 CRON[9477]: pam_unix(cron:session): session opened for user root by (uid=0)
```

Respuesta correcta de Q3: root ✓

Q4:

¿Cuántos atacantes, representados por direcciones IP únicas, pudieron acceder con éxito al sistema después de intentos fallidos iniciales?

1) - Investigue intentos de inicio de sesión exitosos desde diferentes direcciones IP en los registros de autenticación.

2) - Utilice **auth.log** para contar cuántas direcciones IP únicas tuvieron inicios de sesión exitosos.

3) - Primero, enumere los intentos fallidos con **grep "authentication failure" auth.log**, y luego enumere los inicios de sesión exitosos usando **grep "Accepted password" auth.log** para identificar cuántas direcciones IP distintas accedieron al sistema.

Al ejecutar **grep "Accepted password" auth.log | cut -d ' ' -f 9 | sort | uniq** nos mostrara una lista sin repetir de los usuarios que lograron autenticarse por SSH con éxito:

NOTA: El campo 9 es el usuario que inició sesión y `sort | uniq` Ordena y elimina repetidos.

```
checho@Windows-localhost:~/Hammered$ grep "Accepted password" auth.log | cut -d ' ' -f 9 | sort | uniq
dhg
fido
for
root
user1
user2
user3
```

for no corresponde a un usuario, por lo que son 6.

Respuesta correcta de Q4: 6 ✓

Q5:

¿Qué dirección IP del atacante inició sesión exitosamente en el sistema la mayor cantidad de veces?

1) - Concéntrese en el archivo **auth.log** para determinar qué dirección IP tuvo la mayor cantidad de inicios de sesión exitosos.

2) - Ordene los intentos de inicio de sesión exitosos por dirección IP para ver cuál aparece con más frecuencia.

3) - Utilice **grep "Accepted password for" auth.log | awk '{print $11}' | sort | uniq -c** para contar cuántas veces inició sesión cada IP. Identifique la dirección IP con la mayor cantidad de inicios de sesión.

La salida muestra diversas direcciones IPs, pero una dirección en particular llamativa es la dirección: 219.150.161.20

```
checho@Windows-localhost:~/Hammered$ grep "Accepted password for" auth.log | awk '{print $11}' | sort | uniq -c
     14 10.0.1.2
      4 10.0.1.4
      2 121.11.66.70
      2 122.226.202.12
      1 151.81.204.141
      1 151.81.205.100
      1 151.82.3.201
      1 166.129.196.88
      1 188.131.22.69
      4 188.131.23.37
     23 190.166.87.164
      1 190.167.70.87
      3 190.167.74.184
      3 192.168.126.1
      1 193.1.186.197
      1 201.229.176.217
      2 208.80.69.69
      1 208.80.69.70
      7 208.80.69.74
      4 219.150.161.20
      1 222.169.224.197
      1 222.66.204.246
      1 61.168.227.12
      2 65.195.182.120
      6 65.88.2.5
      1 67.164.72.181
      5 71.132.129.212
     22 76.191.195.140
      2 94.52.185.9
```

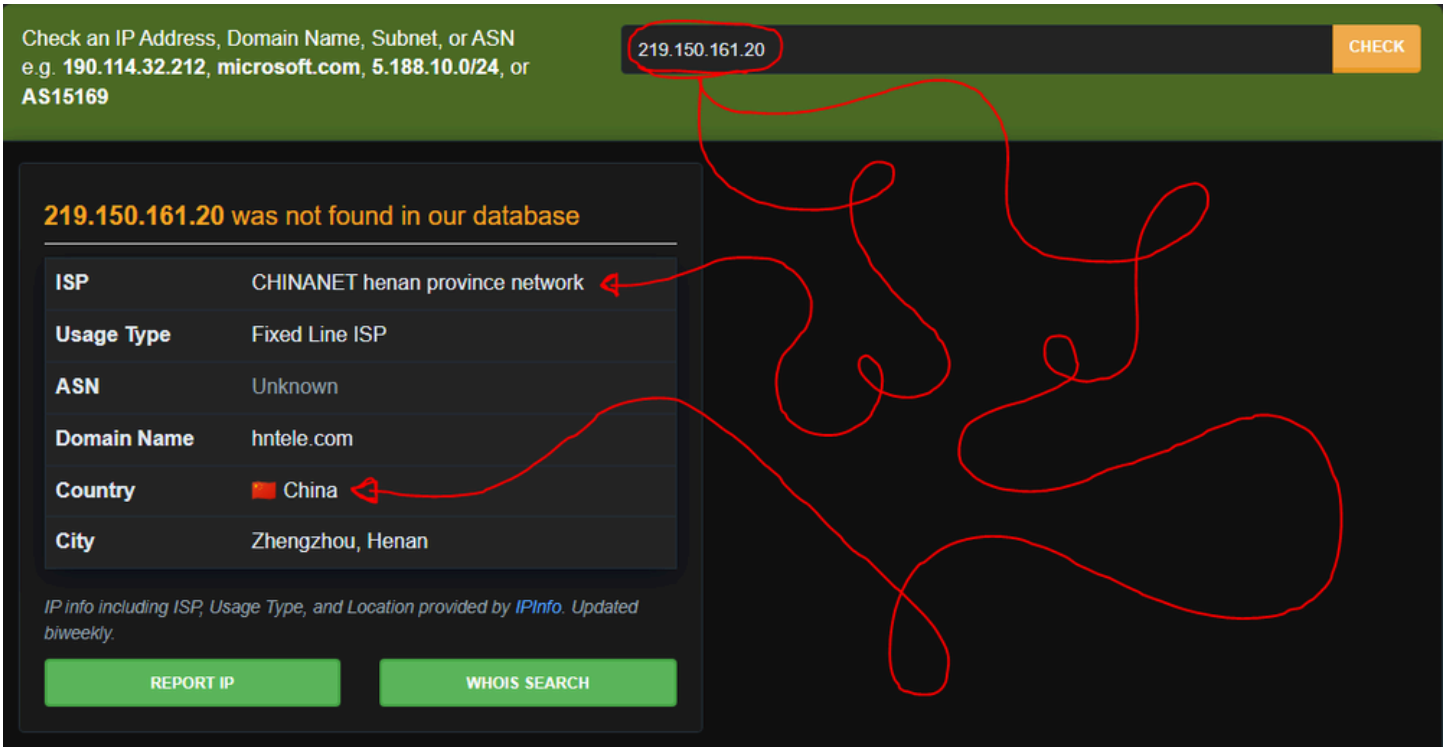
Esta dirección IP a realizado intentos de autenticación que en las cuales no a tenido éxito:

```
Apr 19 07:22:22 app-1 sshd[20935]: Invalid user darian from 219.150.161.20
Apr 19 07:22:22 app-1 sshd[20935]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:23 app-1 sshd[20937]: Invalid user eden from 219.150.161.20
Apr 19 07:22:23 app-1 sshd[20937]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:24 app-1 sshd[20935]: Failed password for invalid user darian from 219.150.161.20 port 54137 ssh2
Apr 19 07:22:25 app-1 sshd[20937]: Failed password for invalid user eden from 219.150.161.20 port 54511 ssh2
Apr 19 07:22:26 app-1 sshd[20939]: Invalid user dario from 219.150.161.20
Apr 19 07:22:26 app-1 sshd[20939]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:27 app-1 sshd[20941]: Invalid user edgar from 219.150.161.20
Apr 19 07:22:27 app-1 sshd[20941]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:28 app-1 sshd[20939]: Failed password for invalid user dario from 219.150.161.20 port 56816 ssh2
Apr 19 07:22:28 app-1 sshd[20941]: Failed password for invalid user edgar from 219.150.161.20 port 57062 ssh2
Apr 19 07:22:30 app-1 sshd[20943]: Invalid user darius from 219.150.161.20
Apr 19 07:22:30 app-1 sshd[20943]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:30 app-1 sshd[20944]: Invalid user eddy from 219.150.161.20
Apr 19 07:22:30 app-1 sshd[20944]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=219.150.161.20
Apr 19 07:22:32 app-1 sshd[20943]: Failed password for invalid user darius from 219.150.161.20 port 59164 ssh2
Apr 19 07:22:32 app-1 sshd[20944]: Failed password for invalid user eddy from 219.150.161.20 port 59270 ssh2
Apr 19 07:22:34 app-1 sshd[20947]: Invalid user darla from 219.150.161.20
```

En AbuseIPDB, se observa que la dirección 219.150.161.20 es una IP pública ubicada en China, asignada al proveedor:

- ISP (Proveedor): China Telecom
- ASN: 4134 (Chinanet)
- País: China
- Ciudad: información completa no disponible
- Zona horaria: Asia/Shanghai

Esto indica que la IP pertenece a un rango de direcciones públicas chinas usados por clientes finales o servicios de red de China Telecom.



Respuesta correcta de Q5: 219.150.161.20 ✓

Q6:

¿Cuántas solicitudes se enviaron al servidor Apache?

1) - Debe analizar los registros de acceso del servidor Apache para encontrar la cantidad de solicitudes.

2) - Consulte el archivo **apache2/www-access.log**, que registra todas las solicitudes al servidor Apache.

3) - Ejecute **cat www-access.log | wc -l** para contar el número total de solicitudes en el registro.

La salida de **cat www-access.log | wc -l** muestra que el total de solicitudes hacia el servidor apache es 365:

```
checho@Windows-localhost:~/Hammered/apache2$ cat www-access.log | wc -l
365
```

Respuesta correcta de Q6: **365** ✓

Q7:

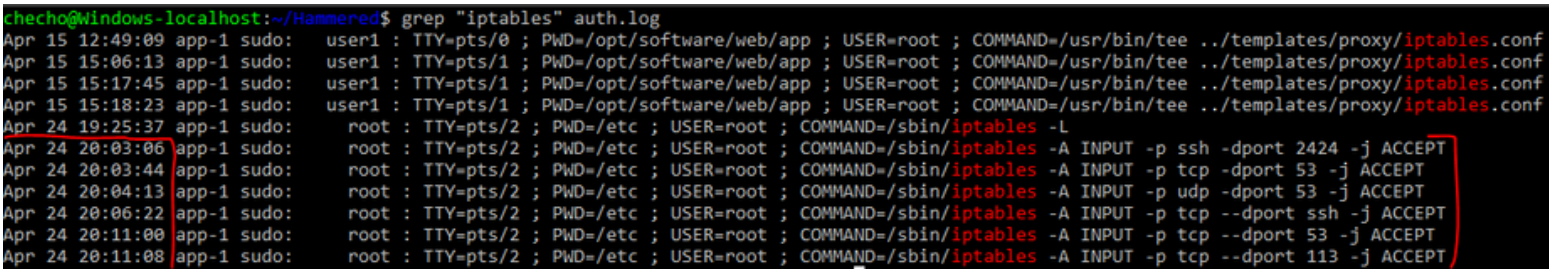
¿Cuántas reglas se han agregado al firewall?

1) - La actividad y los cambios del firewall suelen registrarse en archivos del sistema como **auth.log**. Busque entradas de registro relacionadas con **iptables**.

2) - Busque entradas relacionadas con las adiciones de reglas de firewall, específicamente **iptables**, en los archivos de registro.

3) - Utilice **grep "iptables" auth.log** para encontrar todas las entradas relacionadas con las reglas del firewall y contar la cantidad de reglas agregadas.

Al ejecutar **grep "iptables" auth.log** en la terminal, podemos observar que las reglas del firewall agregadas mas recientes son las del **24 de abril 20:03:06 - 20:11:08**.



Respuesta correcta de Q7: **6** ✓

Q8:

Uno de los archivos descargados en el sistema de destino es una herramienta de escaneo. ¿Cómo se llama?

1) - Revise los registros relacionados con la instalación o descarga de paquetes en el sistema. Es posible que el atacante haya instalado una herramienta de análisis.

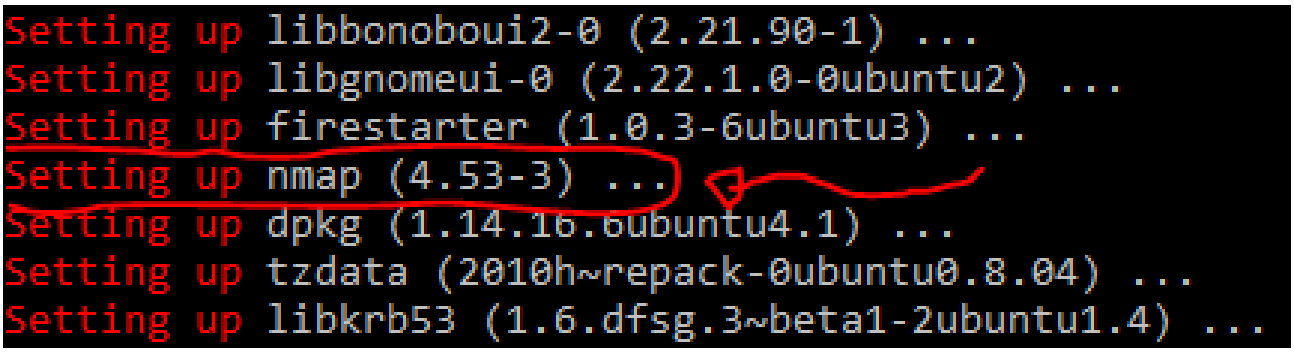
2) - El archivo **apt/term.log** registra los paquetes instalados. Busque en este registro las herramientas instaladas relacionadas con el análisis.

NOTA: `apt/term.log` Pertenece al sistema de gestión de paquetes APT. `term.log` guarda exactamente lo que se mostró en la terminal cuando alguien ejecutó comandos como:

- `apt install`
- `apt remove`
- `apt upgrade`
- `apt purge`

Es decir: La salida completa en pantalla de las instalaciones y desinstalaciones de paquetes.

Al ejecutar grep “Setting up” term.log, la salida muestra una herramienta de escaneo llamada `nmap`.



Nmap (Network Mapper) es una herramienta de código abierto gratuita utilizada para la exploración de redes, auditoría de seguridad y escaneo de puertos.

Respuesta correcta de Q8: nmap ✓

Q9:

¿Cuándo fue el último inicio de sesión del atacante con la IP 219.150.161.20? Formato: MM/DD/AAAA HH:MM:SS AM

19/04/2010 05:56:05 AM

1) - Utilice **auth.log** para filtrar las entradas de registro que mencionan la dirección IP **219.150.161.20** e inicios de sesión exitosos.

2) - Ejecute **grep "Accepted password" auth.log | grep "219.150.161.20"** para encontrar la hora exacta del último inicio de sesión. Combine esto con el año de los metadatos del archivo para formatear la marca de tiempo correctamente.

La salida muestra que el ultimo inicio de sesión exitoso fue el: **2010/04/19 05:56:05**

```
checho@Windows-localhost:~/Hammered$  grep "Accepted password" auth.log | grep "219.150.161.20"
Apr 19 05:41:44 app-1 sshd[8810]: Accepted password for root from 219.150.161.20 port 51249 ssh2
Apr 19 05:42:27 app-1 sshd[9031]: Accepted password for root from 219.150.161.20 port 40877 ssh2
Apr 19 05:55:20 app-1 sshd[12996]: Accepted password for root from 219.150.161.20 port 55545 ssh2
Apr 19 05:56:05 app-1 sshd[13218]: Accepted password for root from 219.150.161.20 port 36585 ssh2
```

Interpretación: El 19 de abril a las 05:56:05 el usuario administrador (root) inició sesión con contraseña en el servidor "app-1" desde la IP 219.150.161.20 el 19 de abril.

Respuesta correcta de Q9: 2010-04-19 05:56 ✓

Q10:

La base de datos mostró dos mensajes de advertencia. Indique el más crítico y potencialmente peligroso.

1) - Las advertencias de la base de datos suelen registrarse en los registros del sistema, como **daemon.log**. Céntrese en las entradas relacionadas con el servicio de base de datos.

2) - Utilice la palabra clave **mysql** para buscar en los registros advertencias relacionadas con la base de datos.

3) - Busque advertencias con **grep "mysql" daemon.log | grep "WARNING"**. Una de las advertencias críticas se refiere a cuentas sin contraseña, lo cual supone un riesgo de seguridad significativo.

La salida muestra mensajes de advertencia indicando que mysql.user contiene 2 cuentas root sin contraseña:

```
checho@Windows-localhost:~/Hammered$ grep "mysql" daemon.log | grep "WARNING"
Mar 18 10:18:42 app-1 /etc/mysql/debian-start[7566]: WARNING: mysql.user contains 2 root accounts without password!
Mar 18 17:01:44 app-1 /etc/mysql/debian-start[14717]: WARNING: mysql.user contains 2 root accounts without password!
Mar 22 13:49:49 app-1 /etc/mysql/debian-start[5599]: WARNING: mysql.user contains 2 root accounts without password!
Mar 22 18:43:41 app-1 /etc/mysql/debian-start[4755]: WARNING: mysql.user contains 2 root accounts without password!
Mar 22 18:45:25 app-1 /etc/mysql/debian-start[4749]: WARNING: mysql.user contains 2 root accounts without password!
Mar 25 11:56:53 app-1 /etc/mysql/debian-start[4848]: WARNING: mysql.user contains 2 root accounts without password!
Apr 14 14:44:34 app-1 /etc/mysql/debian-start[5369]: WARNING: mysql.user contains 2 root accounts without password!
Apr 14 14:44:36 app-1 /etc/mysql/debian-start[5624]: WARNING: mysqlcheck has found corrupt tables
Apr 18 18:04:00 app-1 /etc/mysql/debian-start[4647]: WARNING: mysql.user contains 2 root accounts without password!
Apr 24 20:21:24 app-1 /etc/mysql/debian-start[5427]: WARNING: mysql.user contains 2 root accounts without password!
Apr 28 07:34:26 app-1 /etc/mysql/debian-start[4782]: WARNING: mysql.user contains 2 root accounts without password!
Apr 28 07:34:27 app-1 /etc/mysql/debian-start[5032]: WARNING: mysqlcheck has found corrupt tables
May  2 23:05:54 app-1 /etc/mysql/debian-start[4774]: WARNING: mysql.user contains 2 root accounts without password!
```

Desde un ángulo técnico y de administración de sistemas, tener cuentas de root sin contraseña en MySQL no es solo un descuido, es una vulnerabilidad de severidad crítica que permiten la escalada de Privilegios y control Total.

Respuesta correcta de Q10: mysql.user contains 2 root accounts without password! ✓

Q11:

Se crearon varias cuentas en el sistema de destino. ¿Cuál de ellas se creó el 26 de abril a las 04:43:15?

1) - Los registros del sistema, como **auth.log**, rastrean los eventos de creación de cuentas. Se centran en los registros de la fecha y hora especificadas.

2) - Busque el comando **useradd** en los registros para encontrar detalles sobre cuándo se crearon las cuentas.

3) - Utilice **grep "useradd" auth.log | grep "Apr 26 04:43:15"** para encontrar la cuenta que se creó en ese momento.

EL **26 de abril a las 04:43:15**, el sistema registra que en el host **`app-1`**, se ejecutó el comando **`useradd`** (PID 20115) para crear una nueva cuenta local llamada **`wind3str0y`**, asignándole un UID y GID propios de usuario interactivo, generando su directorio personal en **`/home/wind3str0y`** y habilitándole acceso mediante la **`shell /bin/bash`**.

```
checho@Windows-localhost:~/Hammered$ grep "useradd" auth.log | grep "Apr 26 04:43:15"
Apr 26 04:43:15 app-1 useradd[20115]: new user: name=wind3str0y, UID=1004, GID=1005, home=/home/wind3str0y, shell=/bin/bash
```

Respuesta correcta de Q11: wind3str0y ✓

Q12:

Algunos atacantes usaban un proxy para ejecutar sus análisis. ¿Cuál es el agente de usuario correspondiente que utiliza este proxy?

1) - Verifique los registros de acceso del servidor web para detectar cadenas de agente de usuario inusuales, ya que pueden revelar el uso de un proxy o una herramienta de escaneo.

2) - El archivo **apache2/www-access.log** contiene información del agente de usuario para las solicitudes entrantes.

3) - Extraiga y analice agentes de usuario usando **cat apache2/www-access.log | cut -d ' ' -f 12 | sort | uniq.**

```
checho@Windows-localhost:~/Hammered/apache2$ cat www-access.log | cut -d ' ' -f 12 | sort | uniq
"-"
"Apple-PubSub/65.12.1"
"Mozilla/4.0
"Mozilla/5.0
"WordPress/2.9.2;
"pxyscand/2.1"
```

El análisis de los **`User-Agent`** únicos del archivo **`www-access.log`** evidencia accesos tanto de clientes legítimos como de herramientas automatizadas y escáneres web, destacando la presencia de **`pxyscand/2.1`**, indicativo de actividades de reconocimiento automatizado sobre el servidor.

Respuesta correcta de Q12: pxyscand/2.1 ✓

Documento elaborado con fines educativos y de entrenamiento SOC – CyberDefenders