

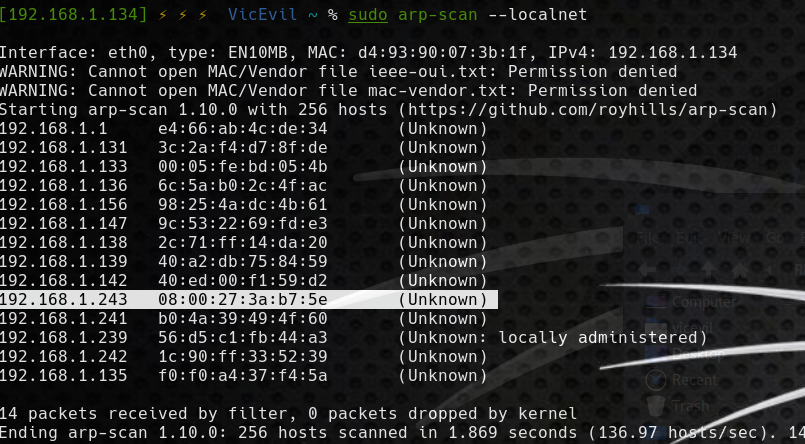
SPRING 16

TEAM CHALLENGE

BSIDE\_VANCOUVER\_2018

En el presente reto, haciendo uso de diferentes técnicas aprendidas en este Bootcamp, se han realizado una serie de gestiones encaminadas a conseguir elevar privilegios maximos y conseguir persistencia en el *“sistema Bsides\_vancouver”, siendo las siguientes:*

1. Una vez descargada la máquina en mi sistema Kali Linux, se procede a su instalación a traves de virtualbox, configurandola en la misma red de mi maquina anfitriona, realizando un “arp-scan” para conocer la IP asignada, comprobando la MAC asignada por virtualbox al sistema.



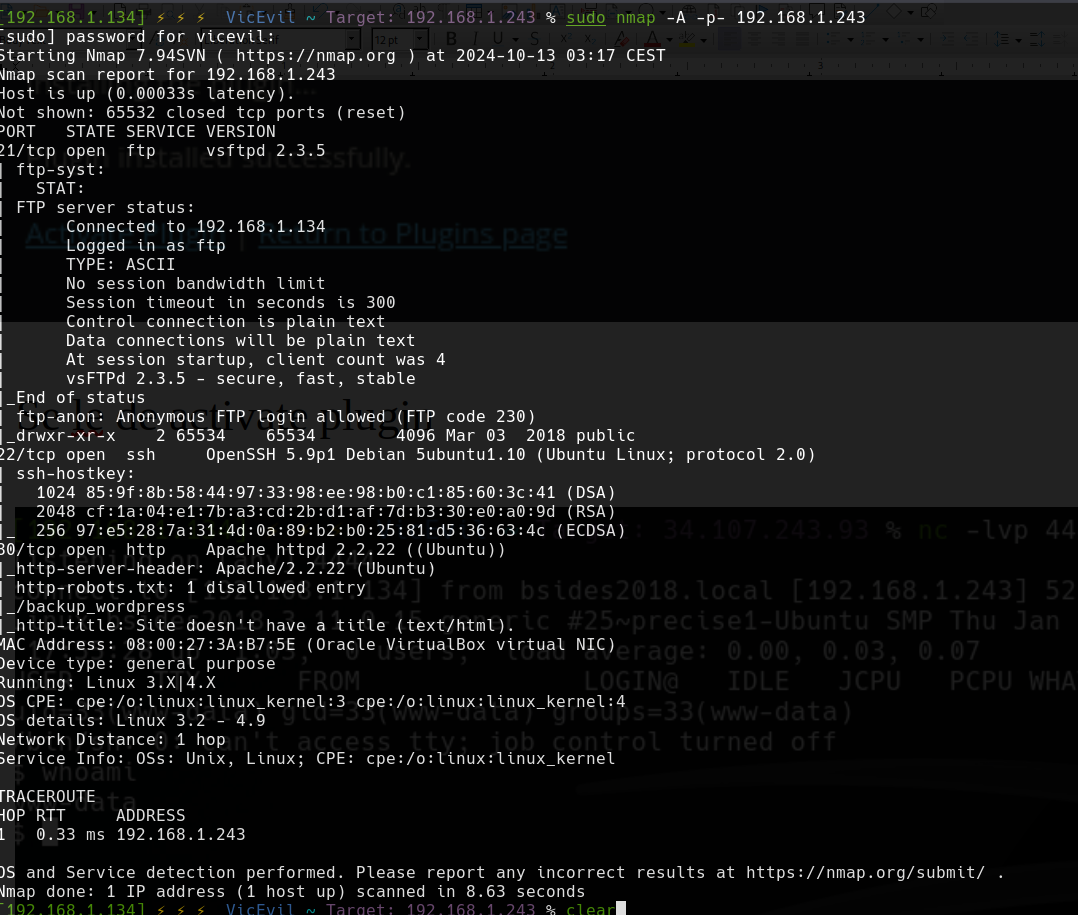
1. Se ejecuta un nmap para obtener la maxima información, destacando:

- Puertos abiertos:

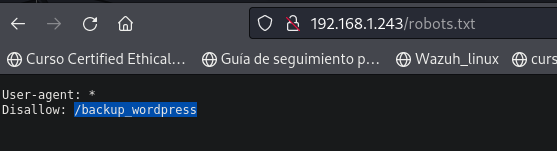
- 21(FTP) con versión **vsftpd 2.3.5**, que es interesante, ya que permite **autenticarse con usuario anonymous**.

- 22(SSH) con la versión OpenSSH 5.9p1

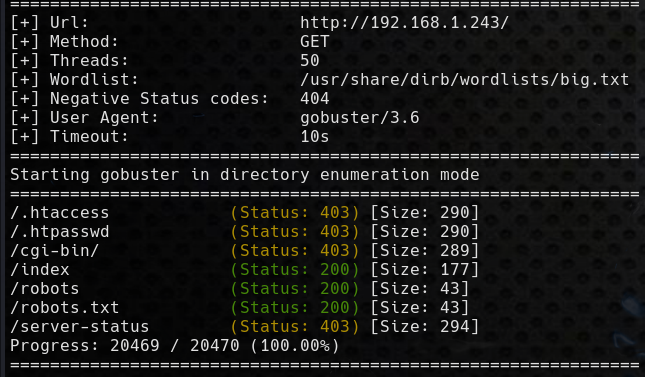
- 80(HTTP) teniendo instalado un servidor Apache 2.2.22, tiendo el archivo **robots.txt** visible y un directorio llamado **/backup\_wordpress.**

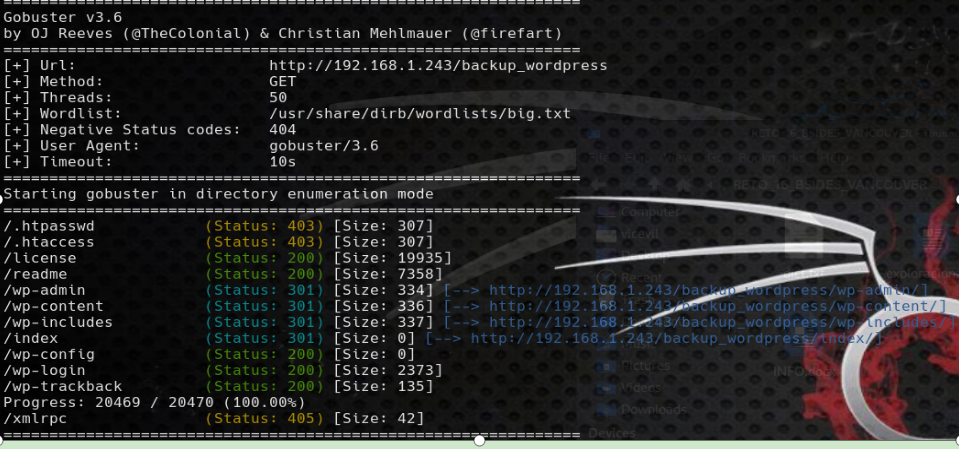


1. Se procede a revisar el archivo **robots.txt**, archivo de texto plano, ubicado en la raíz de un sitio web que sirve para dar instrucciones a los motores de búsqueda sobre qué páginas o secciones del sitio, que deben o no ser indexadas y debe estar oculto. En este caso, puede proporcionar información útil de directorios sensibles del servidor, pero finalmente unicamente aporta el directorio que ya conociamos.

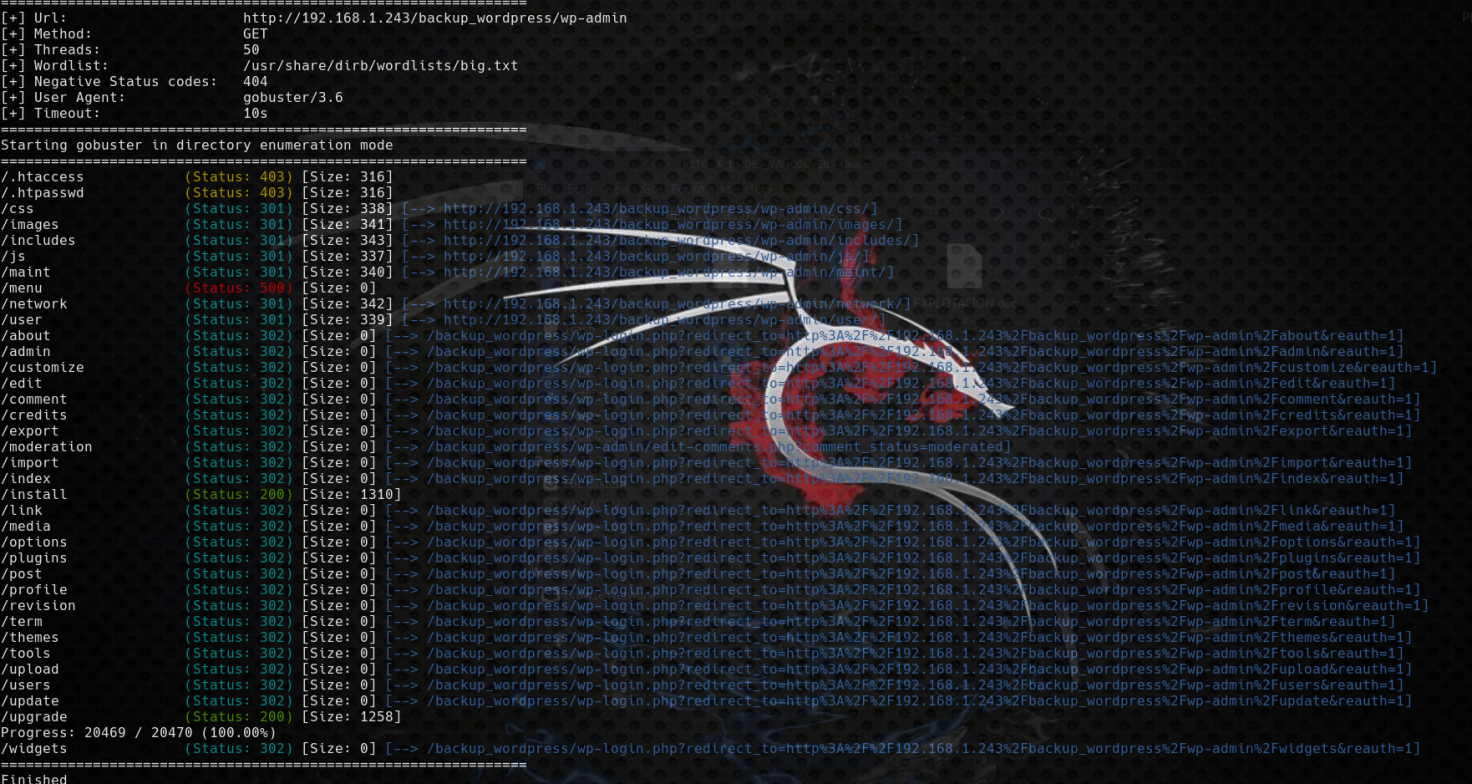


1. Se hace uso de la herramienta **Gobuster**, herramienta diseñada para enumerar directorios, archivos y subdominios en aplicaciones web o servidores, entre otras cosas, encontrando numerosos directorios y subdominios del sistema:

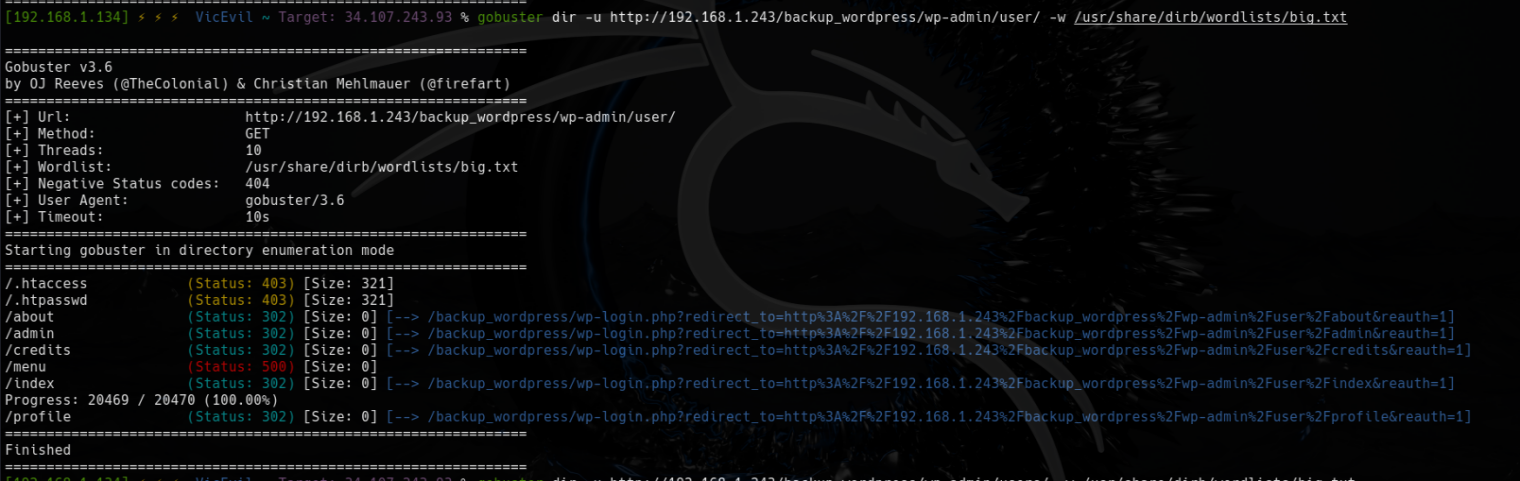
Gobuster a la IP



Gobuster al directorio principal

Gobuster al subdirectorio *“wp\_admin”*

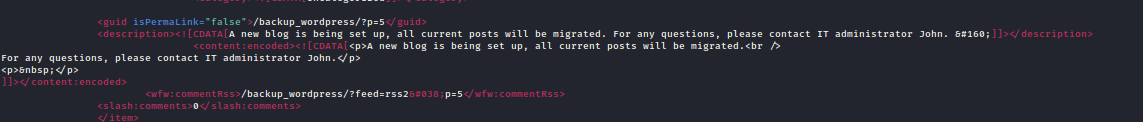
Gobuster sobre el subdirectorio *“wp-includes”*

Gobuster sobre el subdirectorio de wp-admin, llamado *“user”*

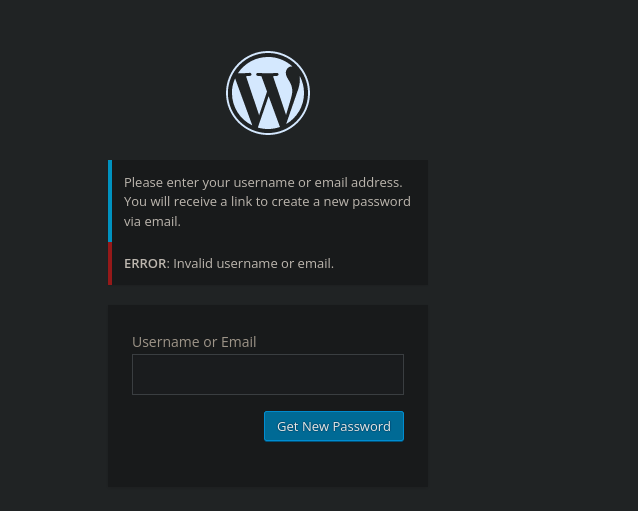
Se han consultado todos los dominos y subdominios, extrayengo una informacion general de la estructura del servidor, pero ningun dato concreto.

1. Paralelamente, se ha analizado el contenido de la web [http://192.168.1.243/backup\_wordpress/](https://192.168.1.243/backup_wordpress/) , sobre los diferentes cuadros de textos de comentarios, sobre la misma url, los enlaces, los archivos de descarga y sobre la sección para login con usuario y contraseña, destacando:

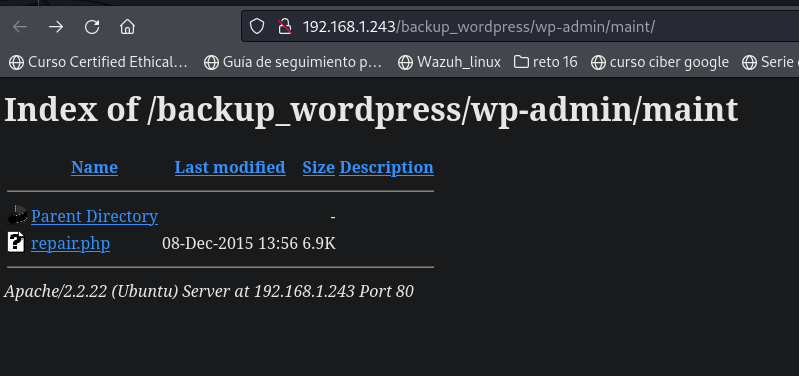
- Pulsando en un enlace de la web, descarag automaticamente un archivo, el cual contiene código HTML, que si se lee detenidamente, se puede observar que nombre a un *“adminitrador* ***john****”*



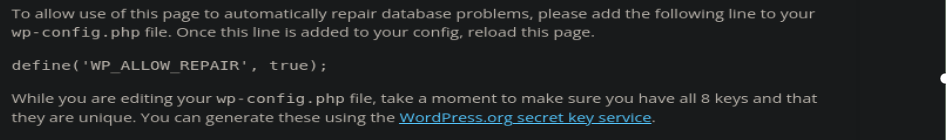
- En la página de login, hay un enlace para reestableciemiento de la contraseña . Si pulsamos en ese enlace, nos dirige a una seccion de la web, donde solicita introducir **“username o email**”, probando con **admin y john**, siendo **usuarios validos**, ya que te dirige a una web informado que no han podido enviar el email, y en otros usuarios probados sale inmediantamente un error.

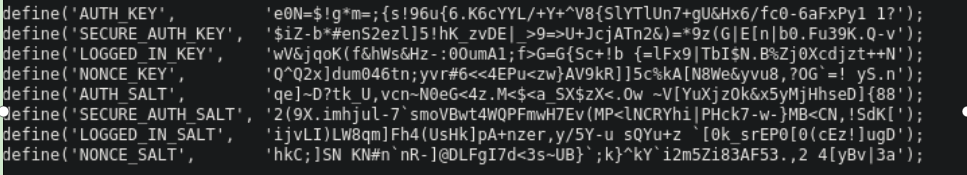
Parte de la pagina para el reestablecimiento de la cuenta

1. Continuando la explotación web, se localiza en la página <http://192.168.1.243/backup_wordpress/wp-admin/maint/> , un directorio del servidor web que lista automáticamente los archivos y carpetas disponibles en esa ubicación, permitiendo a los usuarios navegar, leer y en muchos casos descargar archivos (en nuestro caso no), pudiendo exponer información sensible y representar un riesgo de seguridad significativo.

Pagina “Index of” donde se observa el archivo *“repair.php”*

Si pulsamos sobre el archivo **repair.php**, nos redirige a una pagina donde informa lo que debes de hacer en caso de problemas con laa base de datos, aportando el directorio “wp-config”, una liena de código y un enlace con las 8 claves maestras del servidor, las cuales, son generadas aleatoriamente y se usan para mejorar la seguridad de WordPress en su autenticación, en tareas como: cifrando las cookies de autenticación de los usuarios, asegurando que las sesiones no puedan ser falsificadas y protección los tokens de autenticación, pero no pudiendo extraer de ellas las contraseñas de los usuarios.



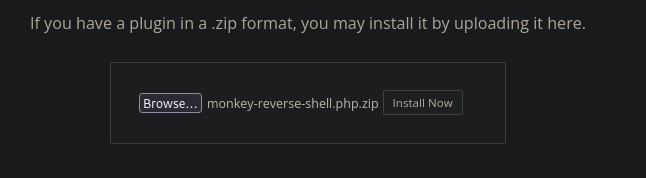


1. Llegados a este punto, con los dos nombres de usuario validos, se procede a usar la herramienta **Hydra**, siendo usada para realizar ataques de autenticación en varios servicios y protocolos, permitiendo probar rápidamente diferentes combinaciones de usuarios y contraseñas en múltiples servicios, como SSH, FTP, HTTP, LOGIN,entre otros.

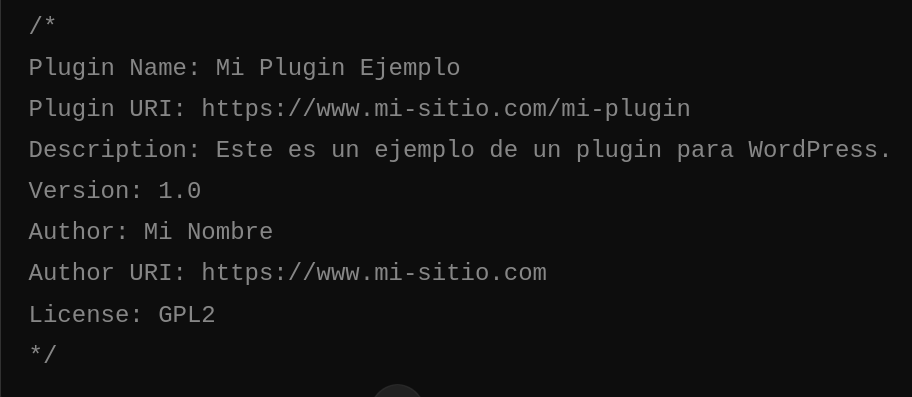
comando hydra : ruta para los users, el diccionario, especificción de ataque para formulario web (metodo post), numero de hilos a lavez y que solo imprima en terminal las que sean correctas.

Finalmente se consigue la contraseña para las credenciales **john:enigma**

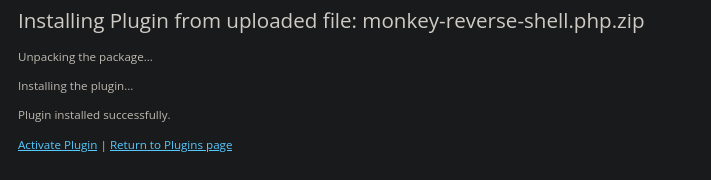
1. Con esta información, me dirijo a la web de acceso de login, consiendo el acceso al administrador de la web de wordpress las citadas credenciales. Una vez en su intenrior, procedo a realziar una inspaccion sobre las diferentes opciones de la pagina con la intención de obtener una conexión remota, encontrando una seccion de plugin, donde te permite instalar plugin por defectoo o personalizados desde su red local, por lo que se procede a realizar una reverse\_shell, incluyendo al inicio de la misma, información básica del plugin, con la finalidad que una vez comprimido en .zip, lo reconozca wordpress como plugin válido.



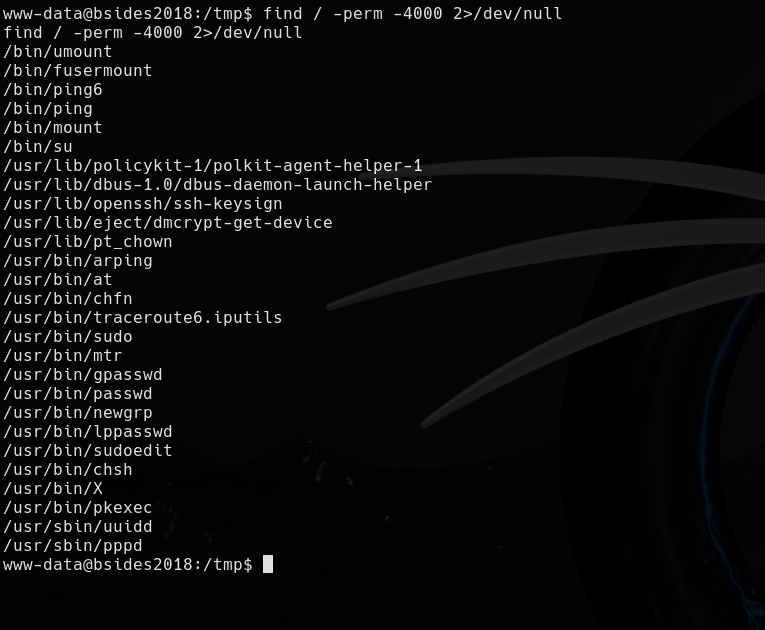
El encabezado o información inicial es crucial para que wordpress lo reconozca como plugin, siendo un ejemplo de este:



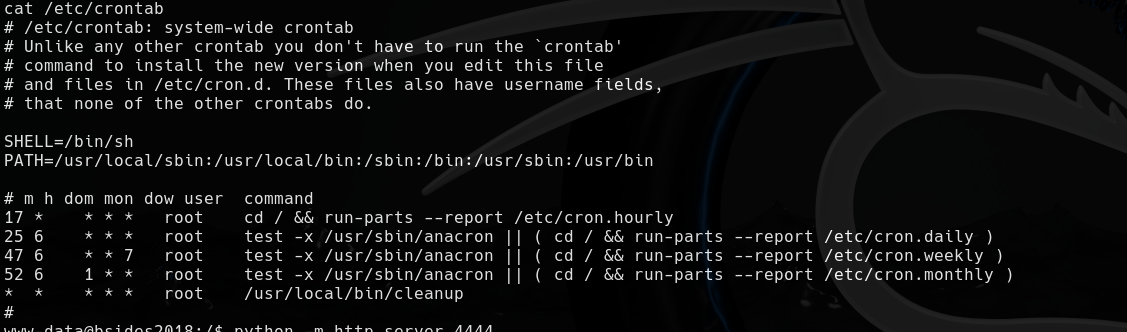
1. Una vez subido al servidor, no muestra el mensaje que es satisfactoria por lo que procedemos a activar nuestro plagin malicioso, no sien antes abrir un netcat en nuestra maquina Kali, consiguiendo acceso al sistema con usuario con permisos limitados.



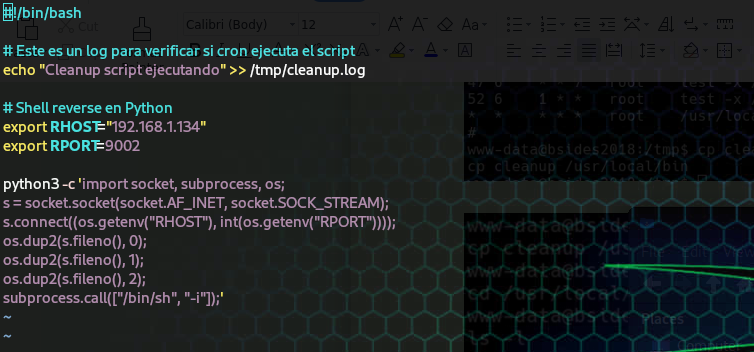
1. Se consulta los archivos SUID del sistema, habiendo bastantes, siendo interesantes pxec, sudoers, sudo y at, probando diferentes opciones pero siempre me solicitan la contraseña de usuarios del sistema que no posseo, por lo que de momento, dejo esta liena de investigacion a un lado.



1. Se procede a consultar el archivo crontab, observando que hay un directorio con un archivo “cleanup”, el cual tiene permisos root, por lo que se prueba si podemos escribir en ese archivo, probando copiar el archivo desde su origen, al directorio /tmp, siendo positivo.

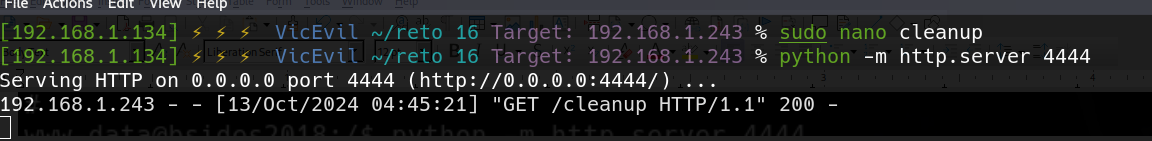
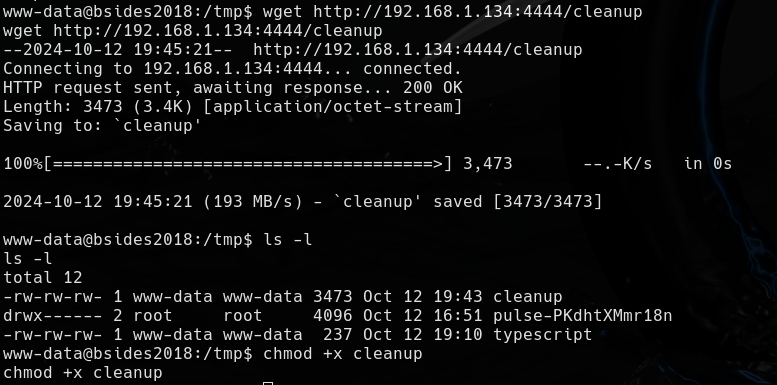


1. Con esta via descubierta, dado que tenemos privilegios sobre un archivo root del crontab que se ejecuta cada minuto, vamos a procede a realziar una shell, acompañandolo de un archivo de verificación de la ejecución para saber si esta funcionando, ya que se han probado con varias shell diferentes hasta encontrar la apropiada:

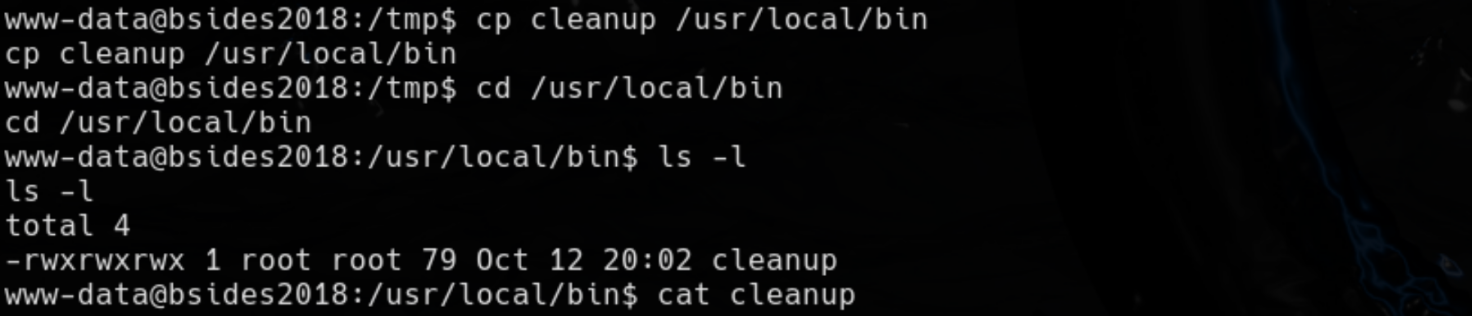


archivo cleanup malicioso con un verificación de ejecución y una reverse shell de python

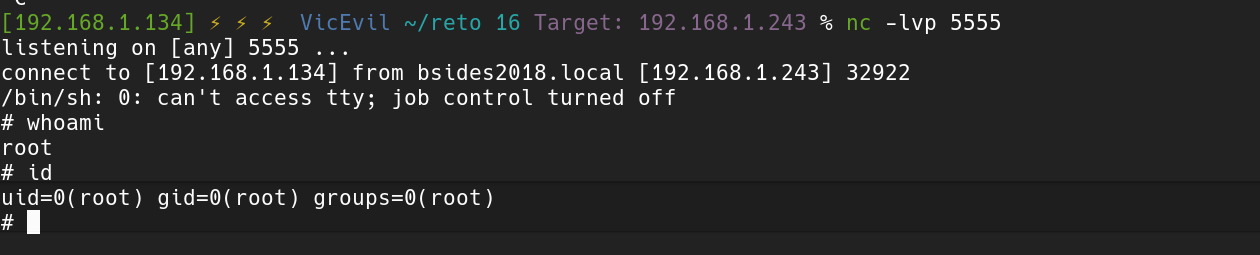
1. Se apertura un servidor python en mi kali , en el directorio donde se encuentra el archivo **“cleanup malicioso”**, y en el sistema objetivo se realiza un wget al servidor, descargando satisfactoriamente el archivo en la carpeta *“tmp”* de la maquina atacada.



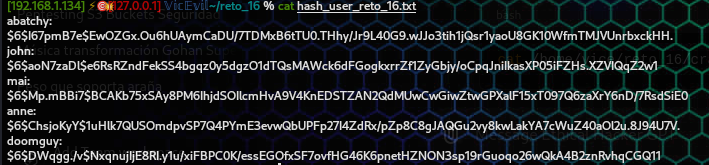
1. Una vez en la maquina objetivo, se procede a darle permisos de ejecución con el comando chmod, y se copia el archivo cleanup malicioso directamente en la ruta “*/usr/local/bin”*, donde se haya el archivo original cleanup, siendo positivo el copiado, no sin antes haber abierto un netcat a la escucha en la maquina kali.



1. Una vez esperado el tiempo indicado en el crontab, conseguimos la reverse shell con maximos privilegios en el sistema.



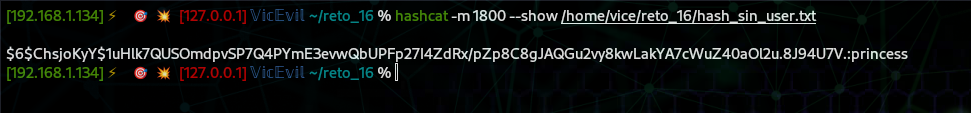
1. Como root, se procede a realziar un cat de l archivo “shadow” donde estan todas las contraseñas cifradas del sistema, con la finalidad de intentar descifrarla con hashcat.



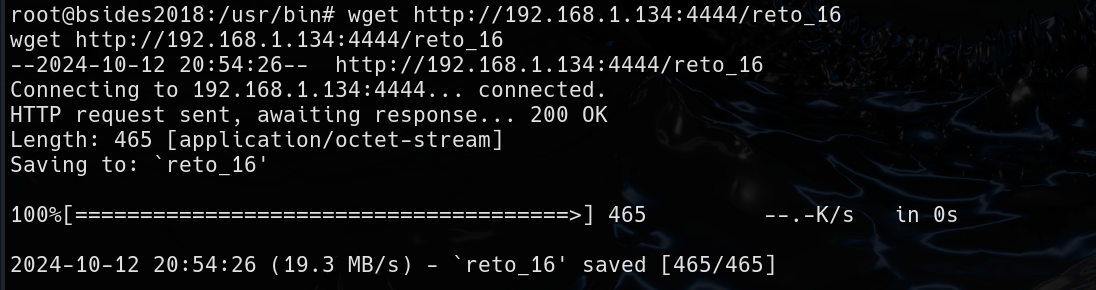
Se realzia un archivo sin los nombre de usuario y se le pasa al comando hashcat, consiguiendo descifrar una contraseña: **anne:princess**



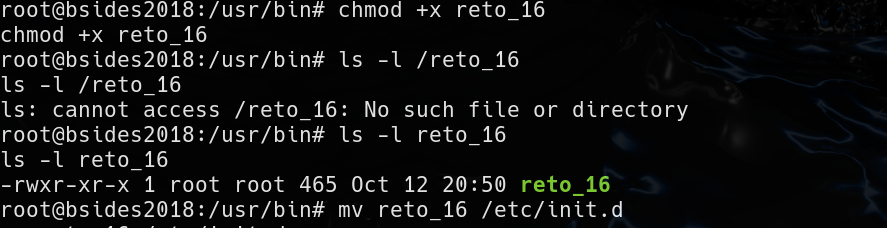
comando hashcat:archivo de resultados, -m 1800=SHA-512, -a 0 (modo diccionario), archivo de los hashes, diccionario, -qui, que no salga en terminal. El error es porque no es compatible con la funcion de monitoreo del ventilador de la GPU.

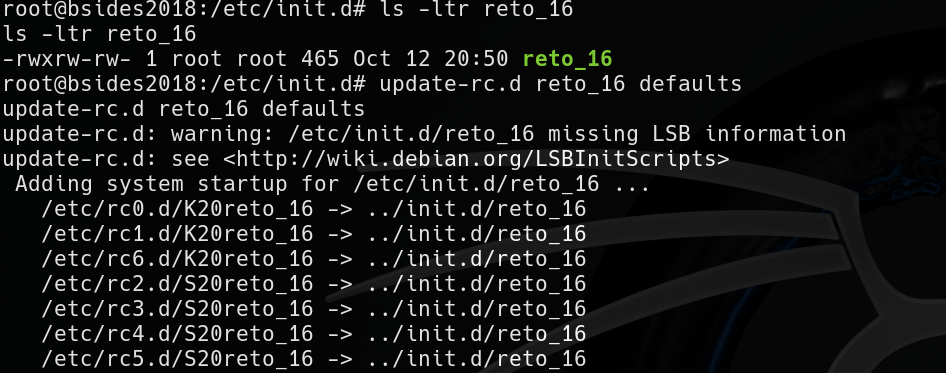


PERSISTENCIA

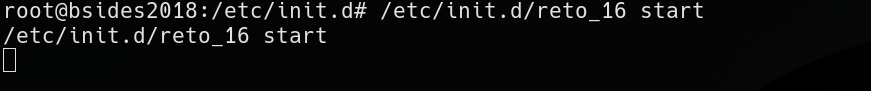


USR/BIN → LO SUBO





comprobacion

persitencia

