

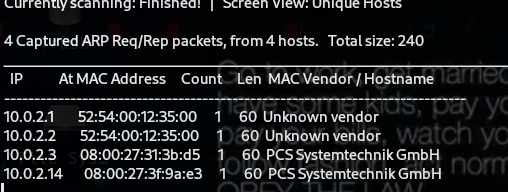
SPRINT 12

TEAM CHALLENGE

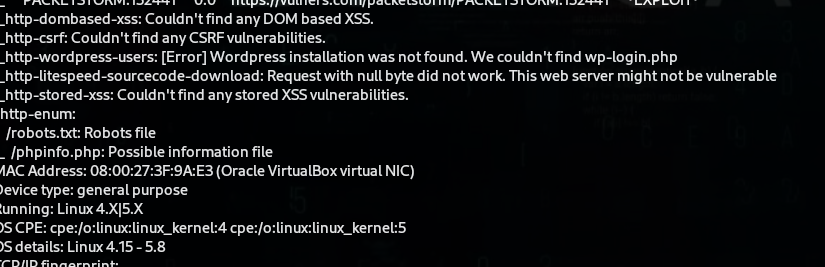
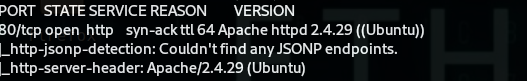
RETO SAR

Para la resolución de este reto, consistente en la búsqueda de posibles vulnerabilidades en la máquina *“sar.ova”,* realizando un ejercicio de pentesting, con la finalidad de llegar a obtener una Shell de la máquina objetivo, consiguiendo persistencia en el sistema, se han llevado a cabo las siguientes gestiones:

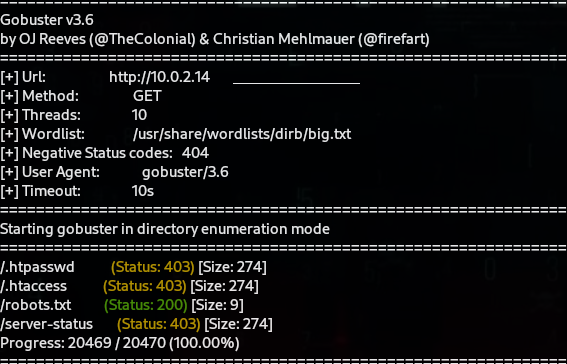
1. Una vez instalada la maquina y ejecutada en segundo plano en VirtualBox, se procede a conseguir la IP de la misma, usando para ello la aplicación Netdiscover:

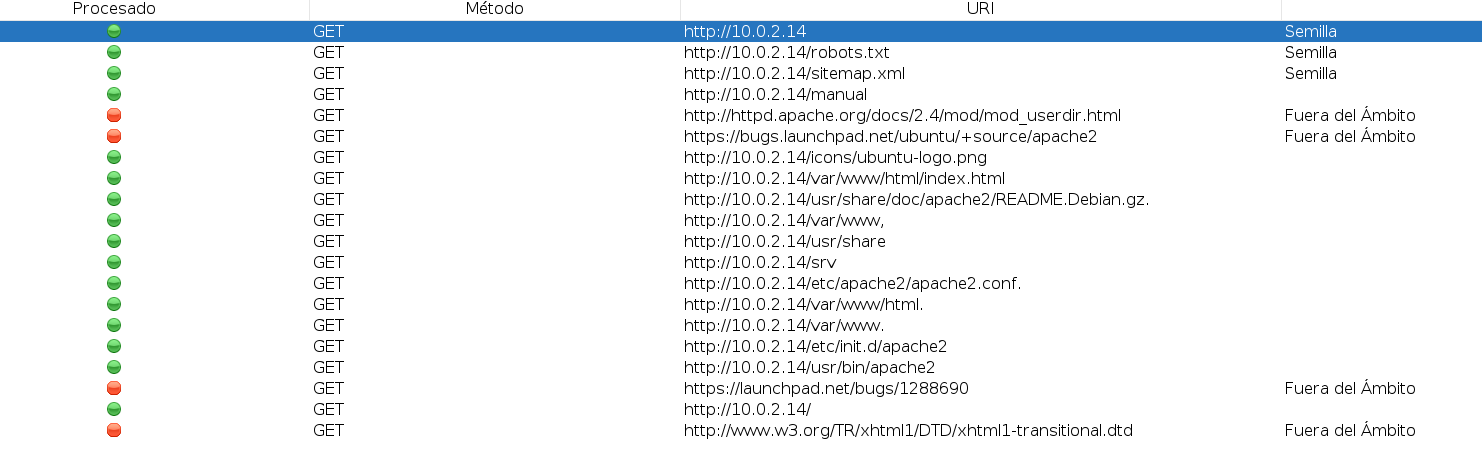


1. Se ejecuta la aplicación Nmap con la IP 10.0.2.14, ejecutándose con varias funciones para conseguir la mayor cantidad de información, destacando que el único puerto abierto es el 80, el cual tiene asignado el servicio Apache 2.4.29(Ubuntu) y como directorios del sistema: /robots y /phpinfo.php.

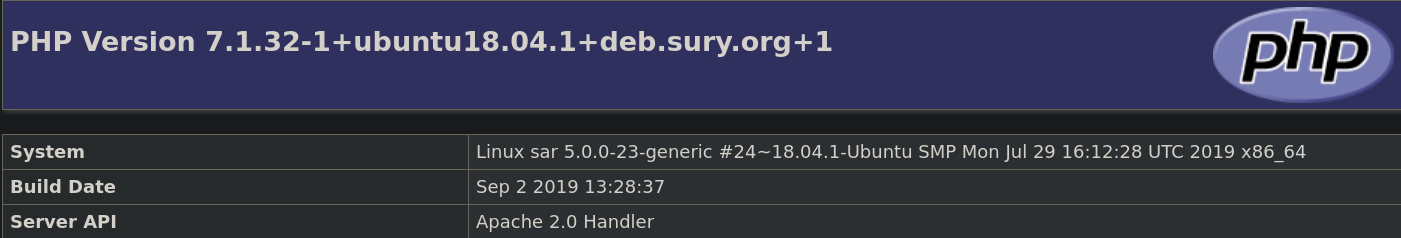


1. Se ejecutan Gobuster y Owasp Zas, escáneres de búsqueda de directorios posibles, confirmando la presencia del directorio /robots.txt, siendo archivo que los administradores de sitios web colocan en la raíz de su servidor, para dar instrucciones a los motores de búsqueda sobre cómo rastrear e indexar las páginas del sitio, permitiendo si es visible, aportar información de la estructura de la web:

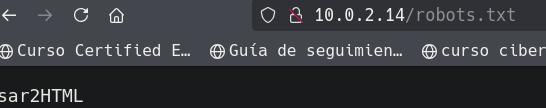




1. Se comprueba el directorio de la web <http://10.0.2.14/phpinfo.php>, el cual aporta aporta mucha información sobre versiones y sistema operativo:

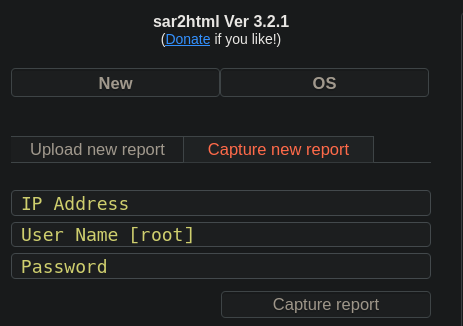
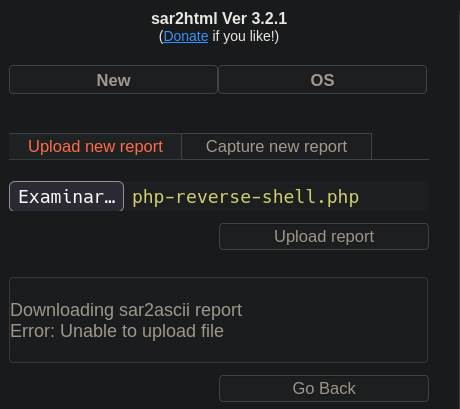


1. Se comprueba el directorio indicado anteriormente, el cual aporta una palabra “sar2HTML”. Se realiza una búsqueda por internet, siendo una interfaz basada en web para monitorización del rendimiento basada en python, convirtiendo datos binarios *“SAR”* *(System Activity Reporter)* a formato gráfico, siendo usada para recopilar, informar y guardar estadísticas de rendimiento del sistema. Además, llama la atención que el primer enlace que sale en Google al poner el nombre del programa sea una información de exploit-DB:

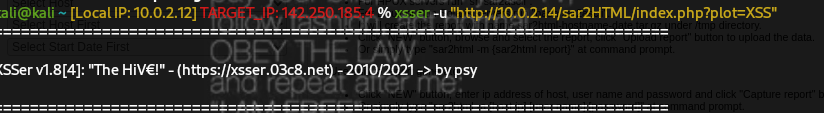


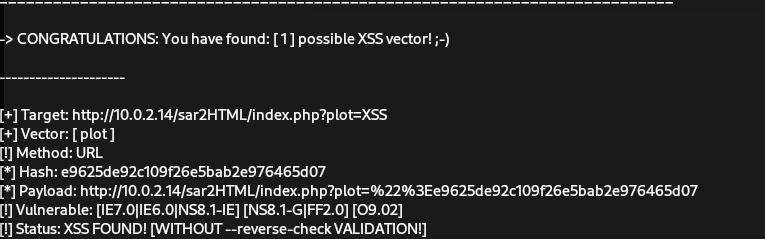


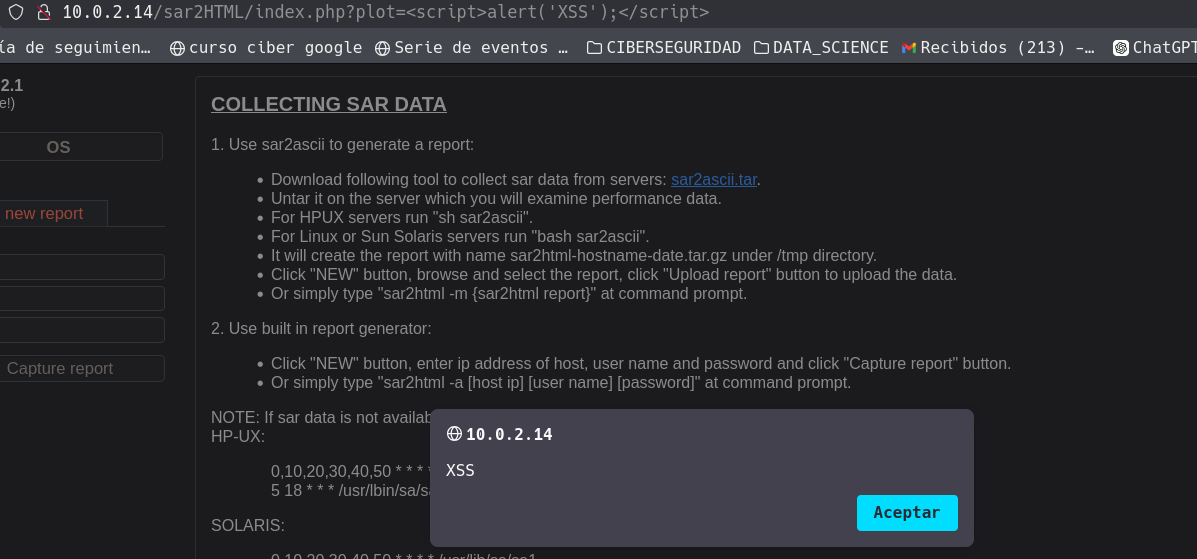
1. Se procede a probar si es un directorio de nuestra web <http://10.0.2.14/sar2HTML/> , siendo el resultado positivo, aportando información sobre su versión 3.2.1 y demás información, pero tras un análisis minucioso, se comprueba que a la derecha aparece un cuadro donde se puede introducir ip, user y password y además un apartada para subir archivos, cambiando la URL a: <http://10.0.2.14/sar2HTML/index.php?plot=NEW>

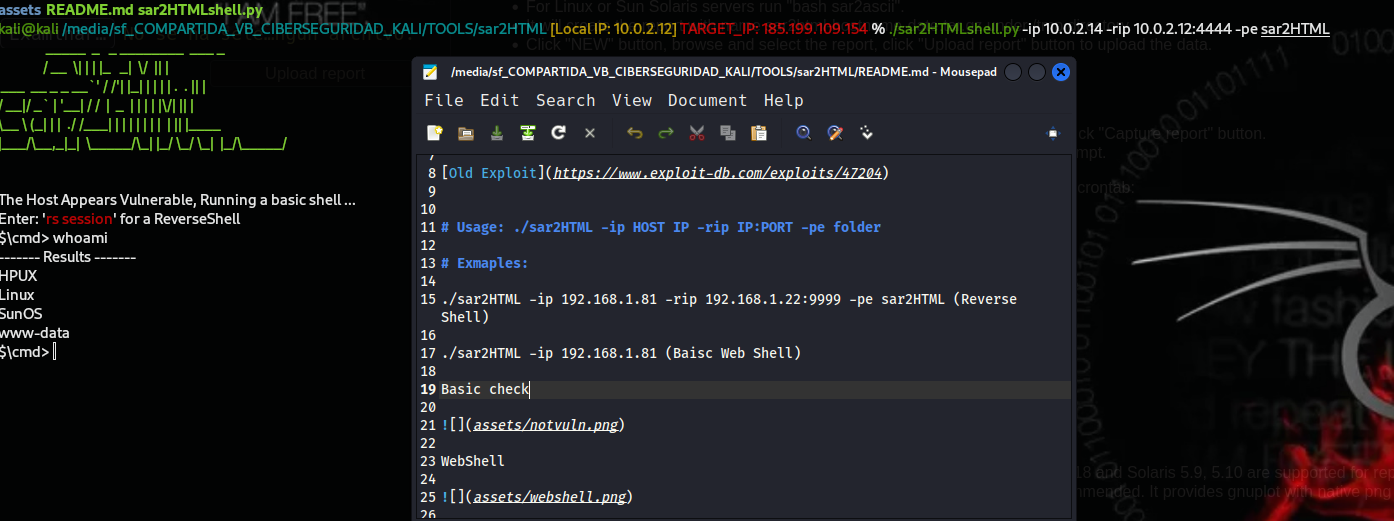
1. Con la nueva URL, se procede a comprobar posibles vulnerabilidades (SQL, Path Traversal, LFI...), encontrando una vulnerabilidad XSS (Cross Site Scripting) de tipo reflejado, ya que se ejecuta como respuesta del servidor cuando el usuario interactúa con el URL, utilizando para ello la aplicación **“XSSer”,** la cual,es herramienta automatizada para detectar y explotar este tipo de vulnerabilidades en aplicaciones web, buscando formas de inyectar código malicioso y detectar XSS.

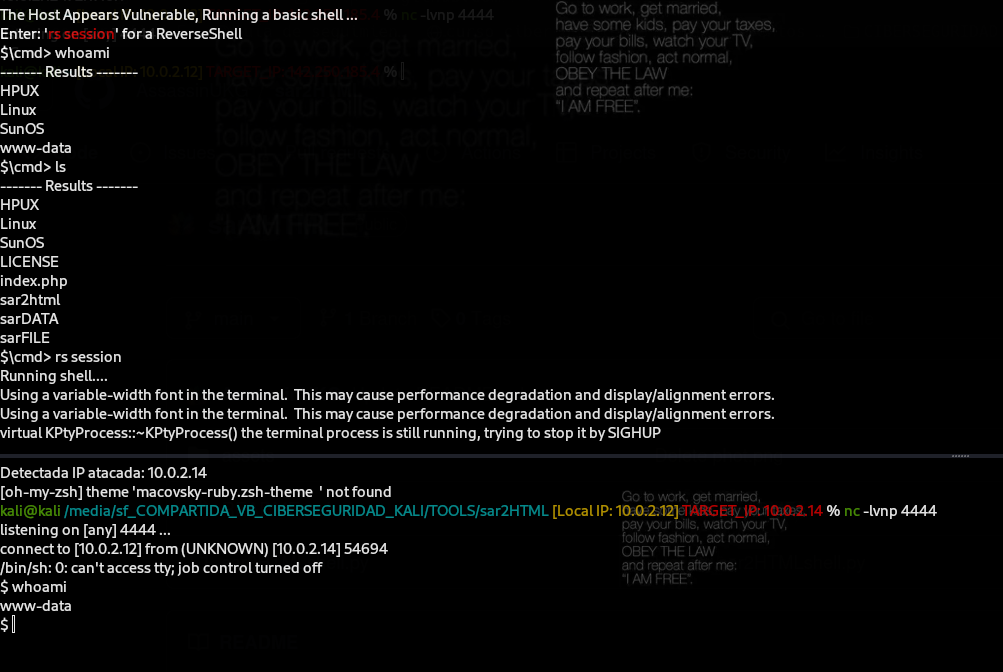






1. Por todo lo anterior, y recordando el primer enlace que aparecía en Google la buscar sar2HTML, y ya habiendo verificado que nuestra versión es la 3.2.1, se procede a seguir con la búsqueda de algún exploit, encontrando en GitHub: <https://github.com/AssassinUKG/sar2HTML> , un script de python con el que es posible abrir una shell o reverse shell, por lo que se procede a su descarga y ejecución, consiguiendo acceso a una shell básica a la maquina objetivo con privilegios básicos.

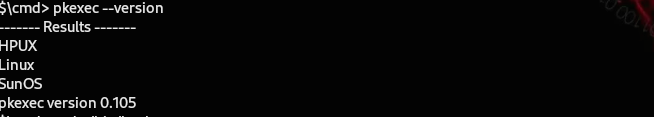
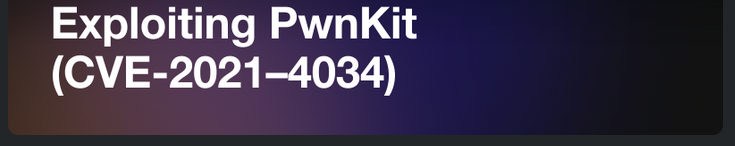
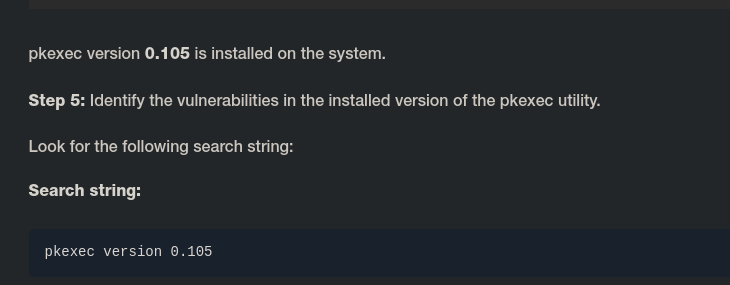




1. Se procede a analizar posibles ficheros que se puedan ejecutar con permisos root, si no están correctamente configurados, los llamados *“bit SUID” (bit (4) Set User ID*), los cuales, pueden dar la capacidad de ejecutarse con los permisos del propietario del archivo, en lugar de con los permisos del usuario que lo ejecuta, es decir, si el usuario es root, podrías ejecutar archivos con ese permiso:

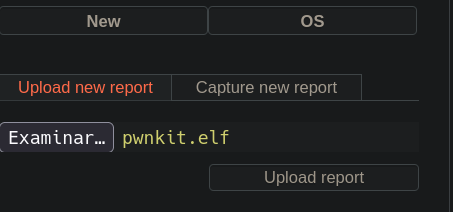


1. Se realiza una búsqueda de información de los archivos en la web, encontrando que *“pkexec”* es una herramienta que permite a un usuario ejecutar programas con los privilegios de otro usuario, típicamente root, comprobando la versión en el sistema objetivo y buscando en la web posibles exploit, encontrando en este enlace: <https://ine.com/blog/exploiting-pwnkit-cve-20214034> , una vulnerabilidad en esa versión, con CVE-2021-4034.



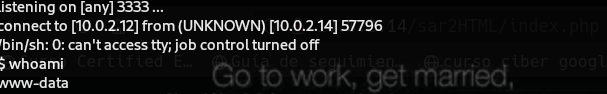
1. Con la herramienta MSFvenom, se realiza un payload con una shell interactiva, el cual procedemos a subir a través del enlace de la web <http://10.0.2.14/sar2HTML/index.php?plot=NEW>

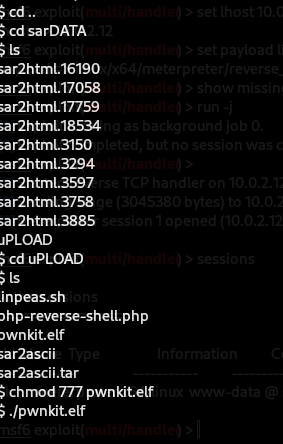




1. A través de la shell básica anteriormente explotada mediante un script de python para sar2HTML y cambiando a reverse shell con la opción que viene descrita, accediendo a “*/sarDATA/uPLOAD”* encontrando en esa carpeta el payload subido, dándole permisos adecuados y ejecutando el script “*./pwnkit.elf”*, no sin antes haber abierto un handler en Metaesploit a la escucha, consiguiendo la conexión con una meterpreter,



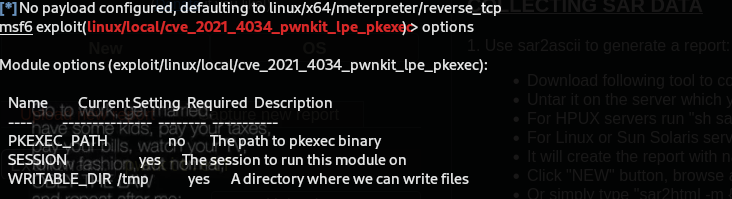


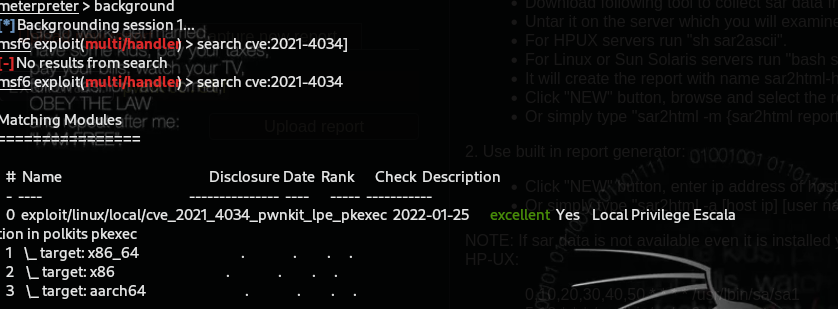




1. Se procede a buscar la vulnerabilidad detectada anteriormente con CVE 2021-4034, siendo conocida como *“Pwnkit”,* la cual, esvulnerabilidad crítica para la herramienta pkexec, que se produce por un error de desbordamiento de variables, permitiendo a un atacante manipular la forma en que se pasan los argumentos al programa, lo que puede ser explotado para ejecutar comandos maliciosos como root.

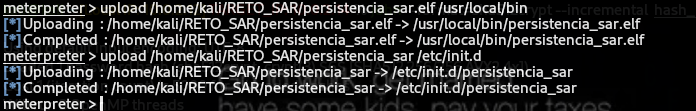
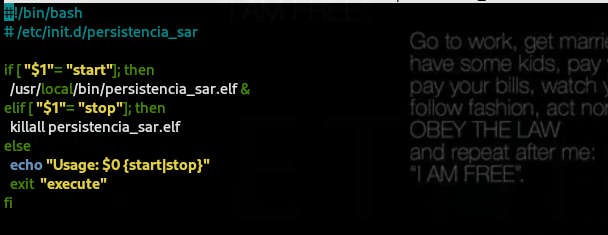
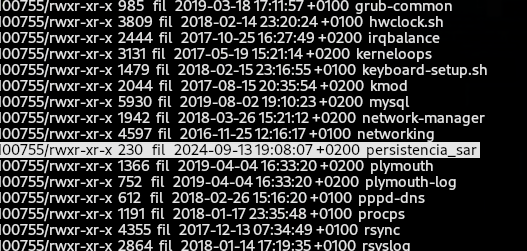
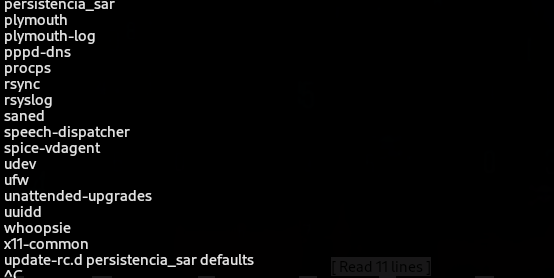
Este CVE se encuentra en Metaexploit, necesitando tener una sesión previa abierta, por lo que, procedemos a configurarla con la sesión de la meterpreter con privilegios limitados, para conseguir una meterpreter con privilegios root a través de la explotación de la vulnerabilidad CVE descrita.





1. Ahora que hemos conseguido una shell interactiva con privilegios root, vamos a ganar persistencia, aprovechando para ello el archivo “*/etc/init.d”,* en el cual están las aplicaciones que se ejecutan al inicio de sesión de cualquier usuario. Para ello, realizamos un nuevo payload en otro puerto a la escucha, el cual será subido a “*usr/etc/bin”* y un script de Bash, para activar el payload como si fuera un servicio del sistema, que será el que se colocara en “*etc/init.d”*, estableciendo a ambos archivos los permisos necesarios de ejecución.

Cuando se inicie el sistema objetivo, los servicios y aplicaciones ubicadas en esa ruta, se iniciarán automáticamente(start), incluyendo el script, el cual establece que si el servicio se encuentra en “*start”* hace que se ejecute el payload malicioso, si está en *“stop”* que se pare el servicio y en cualquier otro caso, que pinte por pantalla: “execute”



1. Finalmente, lo activo para que se inicie de manera automática en la configuración por defecto en el sistema objetivo *(“update-rc.d persitencia\_sar defaults”)* y reiniciamos con “reboot”, no sin antes haber preparado un handler a la escucha de nuestro payload en el puerto 5555, resultando positivo, consiguiendo la persistencia en el sistema.

