

# **제 1회 서울아이티고등학교 해킹방어대회 Writeup**

Written by Safflower (st4rburst@naver.com)

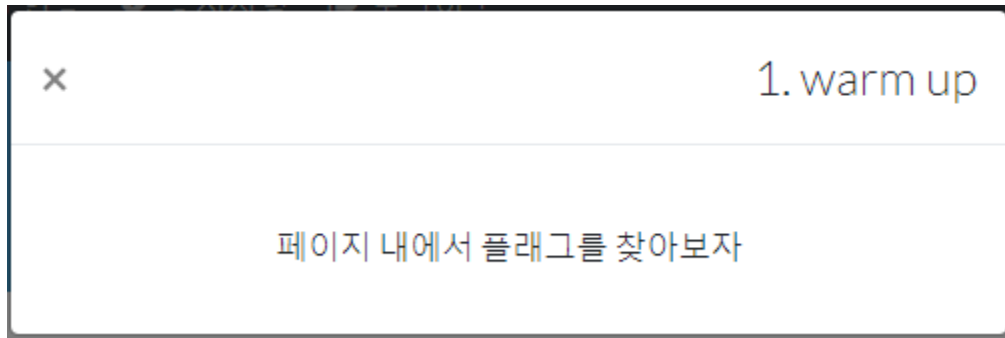
## Rank

#	Nickname	Score	School	Solve	Last solve time
1		1387	대덕소프트웨어마이스터고등학교	ALL CLEAR	2017-09-23 15:27:02
2	Safflower	1376	한국교통대학교	ALL CLEAR	2017-09-23 15:42:45
3		1320	서울아이티고등학교	o/o/x/o/o/o/o/o/o/o/o/o/o/o	2017-09-23 18:09:18
4	404N0tF0und	1320	한세사이버보안고등학교	o/o/x/o/o/o/o/o/o/o/o/o/o/o	2017-09-23 18:31:03
5	알약2봉지	1320	한세사이버보안고등학교	o/o/x/o/o/o/o/o/o/o/o/o/o/o	2017-09-23 17:12:40
6	KM_TM	1271	한세사이버보안고등학교	o/o/x/x/o/o/o/o/o/o/o/o/o/o	2017-09-23 12:53:40
7	닉네임 따워	1263	상계중학교	o/o/x/x/o/o/o/o/o/o/o/o/o/o	2017-09-23 10:13:10
8	anonymous-kr	1260	한국디지털미디어고등학교	o/o/x/x/o/o/o/o/o/o/o/o/o/o	2017-09-23 11:19:44
9	hihi	1260	선린인터넷고등학교	o/o/x/x/o/o/o/o/o/o/o/o/o/o	2017-09-23 11:21:59
10	M4ndU	1212	인현고등학교	o/o/o/x/o/o/o/x/o/o/o/o/o/o	2017-09-23 15:45:42
11	김태민	1184	한세사이버보안고등학교	o/o/x/x/o/o/o/o/o/o/o/o/o/x	2017-09-23 17:15:30
12	whitecat	1179	나는_학교가_없다	o/o/x/x/o/o/o/x/o/o/o/o/o/o	2017-09-23 10:11:59

<http://ctf.seoulit.kr/rank.php>**Nickname:** Safflower**Score:** 1376

최종 순위 2등 + 올클리어로 마무리 했다. www

## 1. warm up



소스를 읽어보자. Ctrl+U

```
<script src="frontend/js/sign-modal.js"></script>
<script src="frontend/js/pace.min.js"></script>
<script src="frontend/js/welcome.js"></script>
```

```
<script>
  (function (i, s, o, g, r, a, m) {
```

소스를 읽어봤는데 수상한 js 파일이 있다.

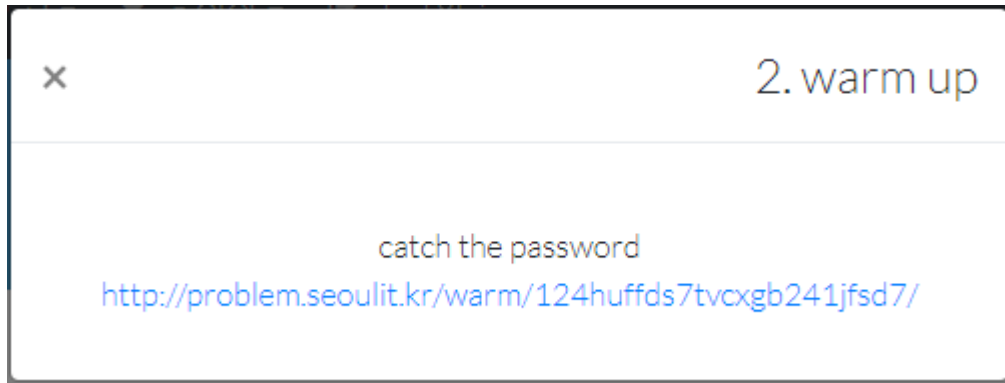
<http://ctf.seoulit.kr/frontend/js/welcome.js>

```
// alert("flag is im_jjang_hacker")
```

읽어보니 flag가 있다.

**flag: im\_jjang\_hacker**

## 2. warm up



로마의 군인이 다른사람들이 알아보지 못하도록 문자들을 다른 문자로 치환해  
친구들에게 비밀리에 편지를 보내곤했다. 아래 코드를 해독해보자.  
암호 전체가 flag 값이다.  
LP\_MMDQJ\_KDFN

암호 입력	편지 전송
-------	-------



로마 군인이라니까 Caesar Cipher를 썼을 것 같다.

## Caesar Cipher Online Decipher

Cipher text \*

LP\_MMDQJ\_KDFN

Decipher

Key	Plain text	Copy
01	KO_LLCPI_JCEM	
02	JN_KKBOH_IBDL	
03	IM_JJANG_HACK	
04	HL_IIZMF_GZBJ	
05	GK_HHYLE_FYAI	
06	FJ_GGXKD_EXZH	
07	EI_FFWJC_DWYG	
08	DH_EEVIB_CVXF	

<http://safflower.kr/crypto/caesar-cipher/decipher.php>

모든 키로 decipher해본 결과, 키는 3인 것 같다.

[http://problem.seoulit.kr/warm/124huffds7tvcxgb241jfsd7/?code=IM\\_JJANG\\_HACK](http://problem.seoulit.kr/warm/124huffds7tvcxgb241jfsd7/?code=IM_JJANG_HACK)

**flag: 5494dc6fe9c66a8b72733fbcbb473a86**

### 3. Mata Hari

x

3. Mata Hari

[악보 다운로드](#)

1. 마타하리->다른암호
2. 트리테미우스 암호



마타하리 암호다.

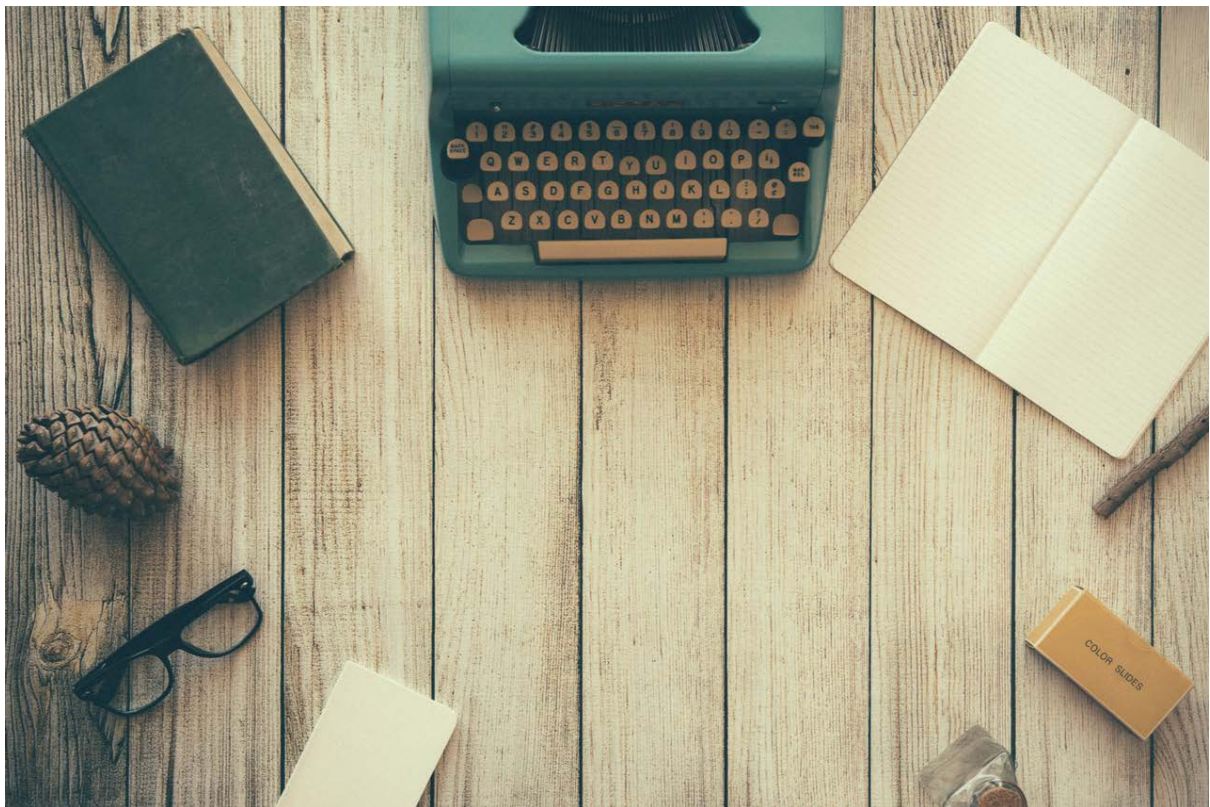
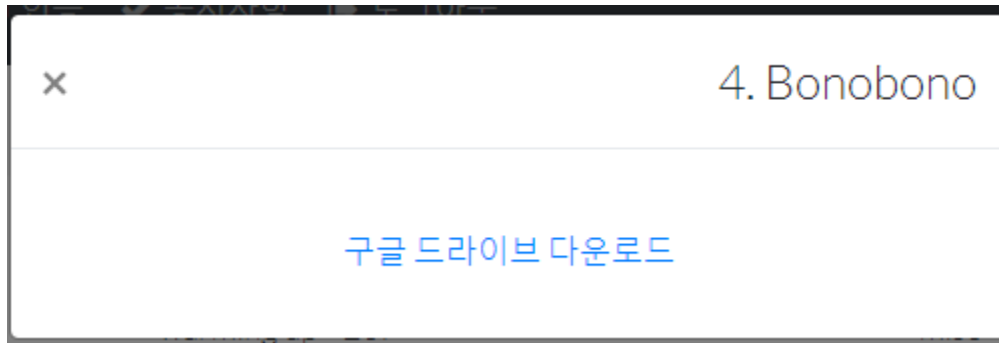
악보대로 복호화하면 "FMCJMXZYQCRPYVHSZ#HAYM" 라고 나온다.



이걸 Trithemius Cipher 로 복호화해주면 flag 가 나온다.

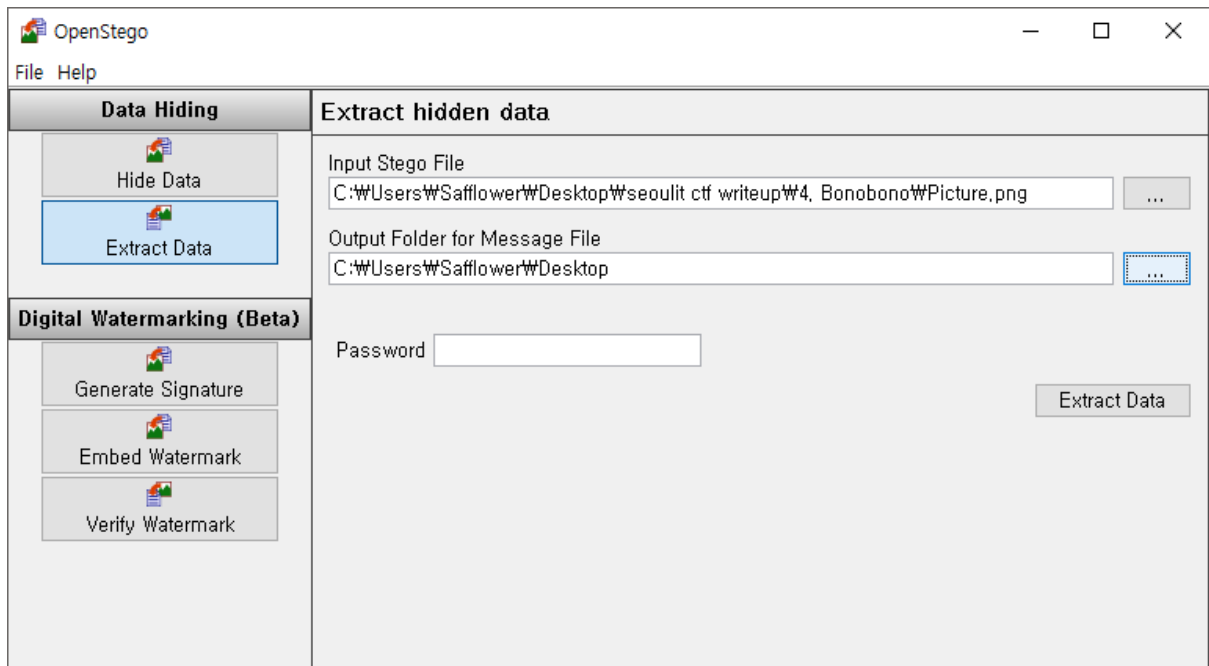
flag: TRITHEMIUS\_CIPHER

## 4. Bonobono

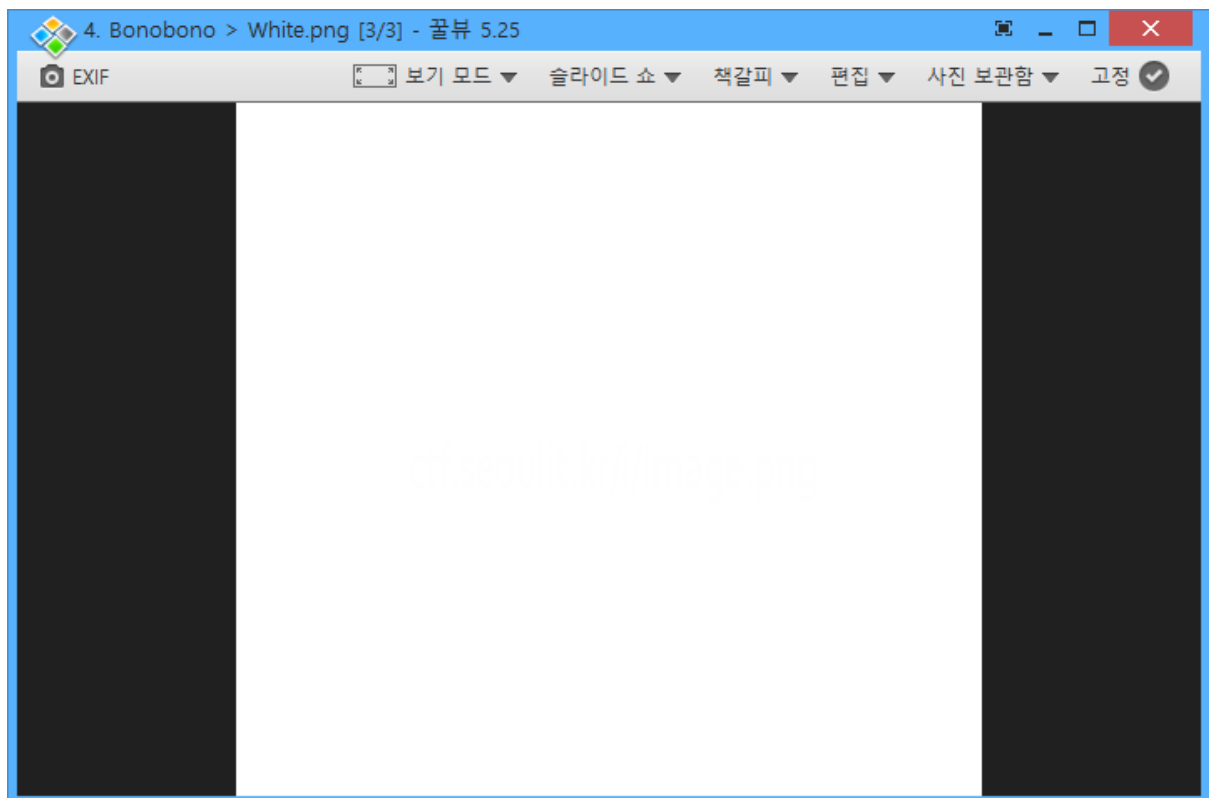


png 파일을 준다.

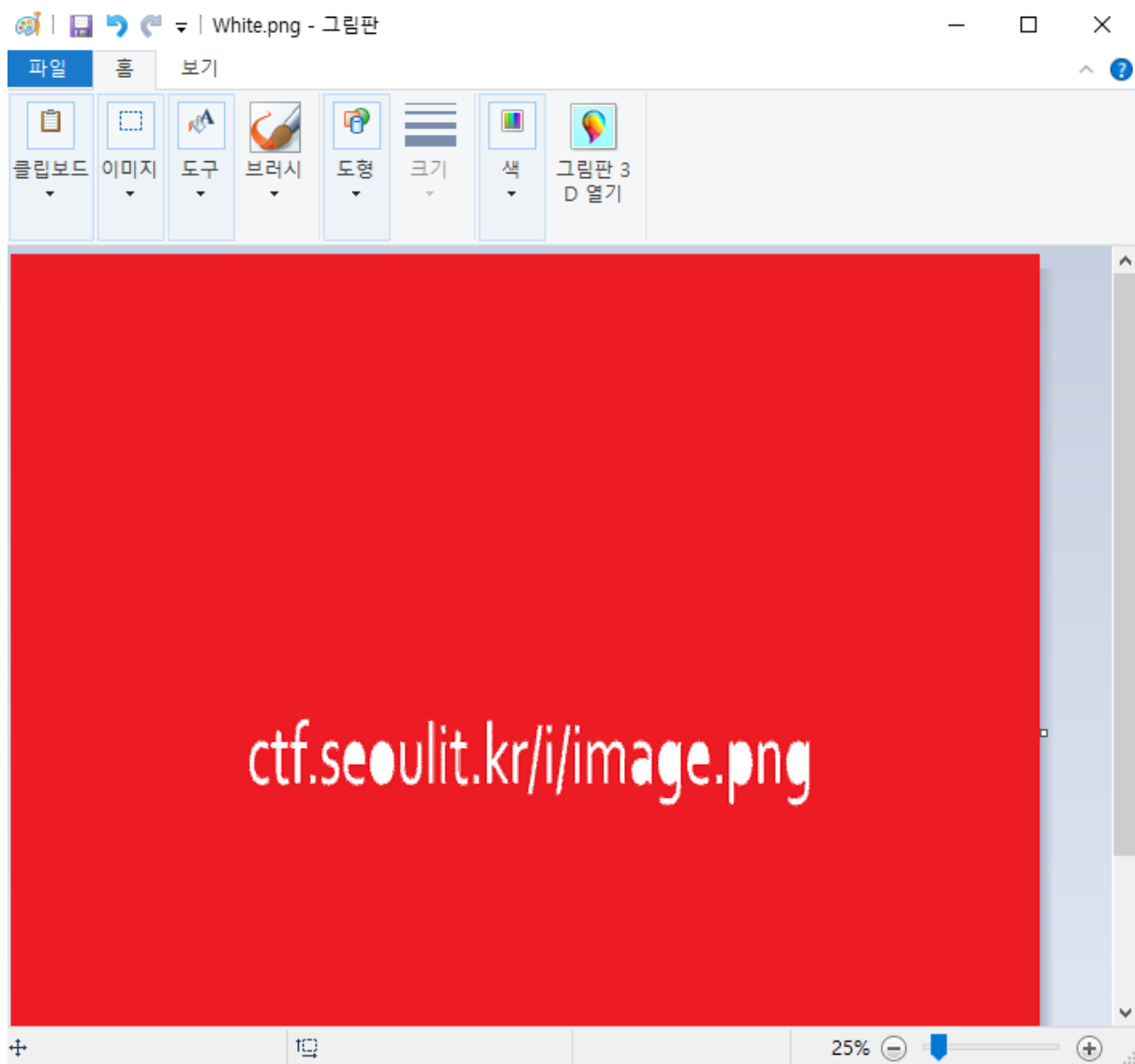




OpenStego로 [Extract Data] 하면 white.png 파일이 나온다.



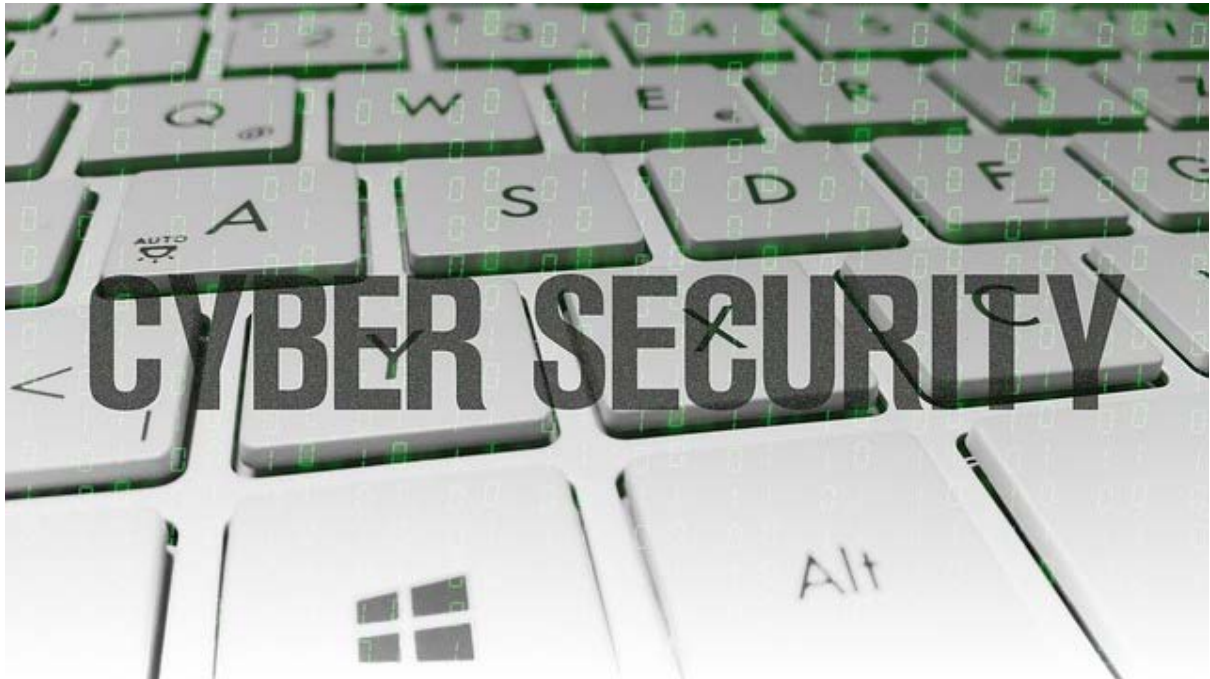
열어보니 그냥 하얀 사진이다.



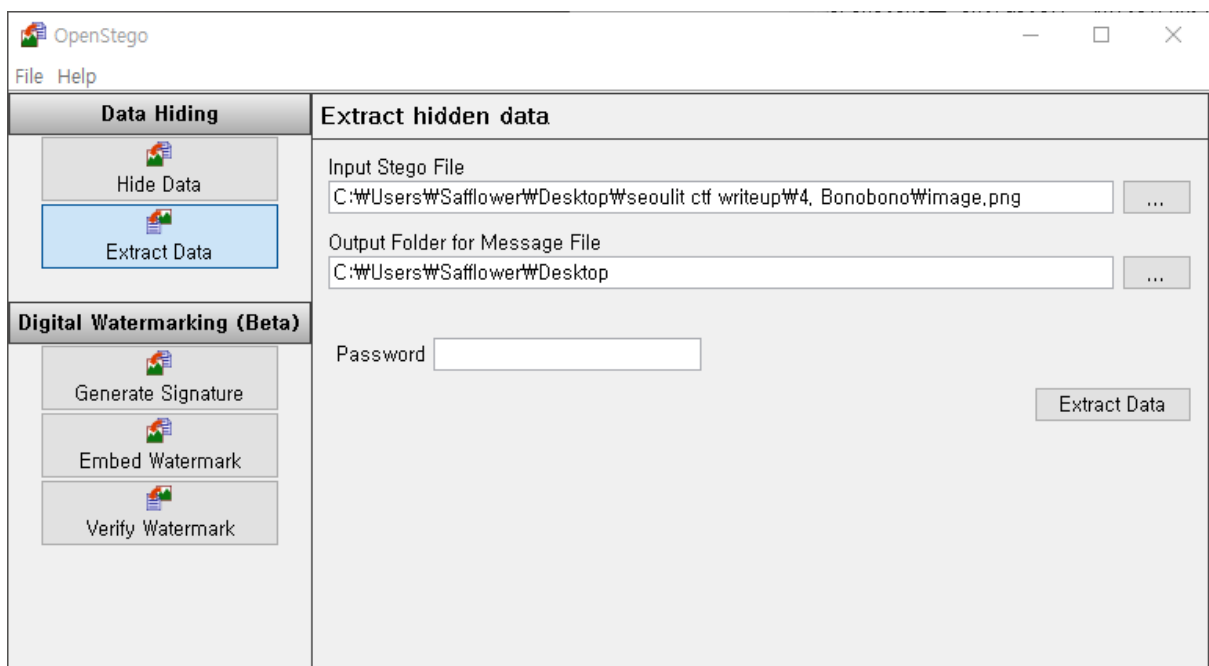
mspaint로 하얀 배경에 페인트통을 부어보면, 링크가 나온다.

다운로드 받아보자.

<http://ctf.seoulit.kr/i/image.png>



또 png 파일이다.

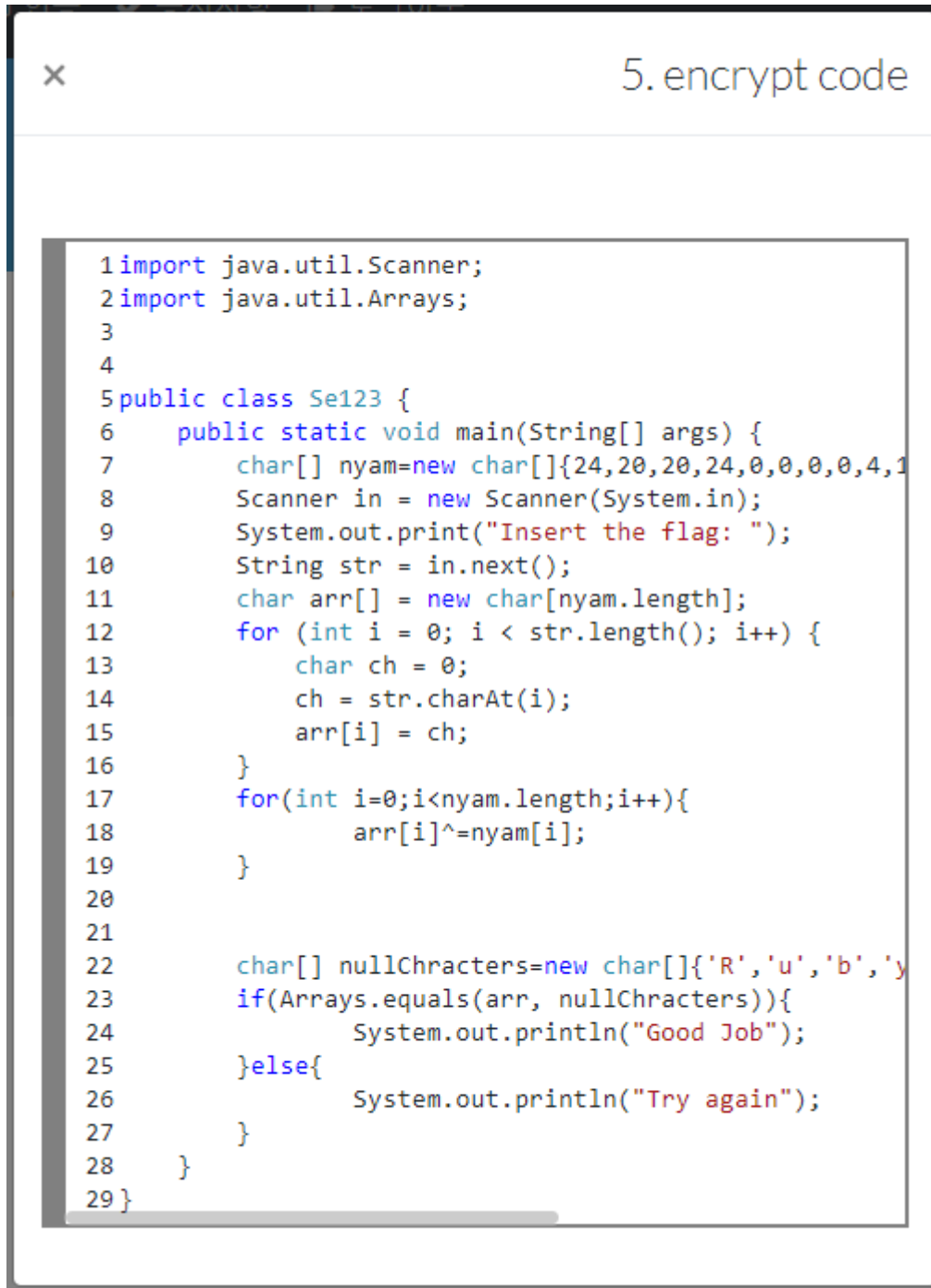


다시 한번 OpenStego를 써본다.

flag.txt 를 준다.

**flag: HE3E\_1S\_STEGANO\_GRAPHY**

## 5. encrypt code



```
1import java.util.Scanner;
2import java.util.Arrays;
3
4
5public class Se123 {
6    public static void main(String[] args) {
7        char[] nyam=new char[]{24,20,20,24,0,0,0,0,4,1
8        Scanner in = new Scanner(System.in);
9        System.out.print("Insert the flag: ");
10       String str = in.next();
11       char arr[] = new char[nyam.length];
12       for (int i = 0; i < str.length(); i++) {
13           char ch = 0;
14           ch = str.charAt(i);
15           arr[i] = ch;
16       }
17       for(int i=0;i<nyam.length;i++){
18           arr[i]^=nyam[i];
19       }
20
21
22       char[] nullChracters=new char[]{'R','u','b','y
23       if(Arrays.equals(arr, nullChracters)){
24           System.out.println("Good Job");
25       }else{
26           System.out.println("Try again");
27       }
28     }
29 }
```

Java 소스가 있다.

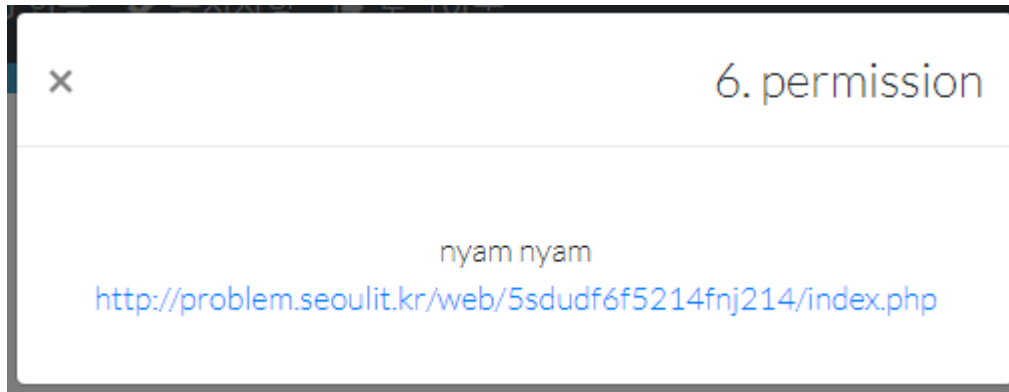
읽어보면 nyam이랑 flag랑 ascii를 xor한 후, 문자열로 바꿨을 때의 값이 nullChracters이면 정답인 것 같다.

```
1 nyam = [24,20,20,24,0,0,0,0,4,12,6,30,13]
2 char = ['R','u','b','y','_','i','s','_','l','i','g','h','t']
3
4 for i in range(len(nyam)):
5     print(chr(nyam[i] ^ ord(char[i])), end='')
6
```

역연산해주자.

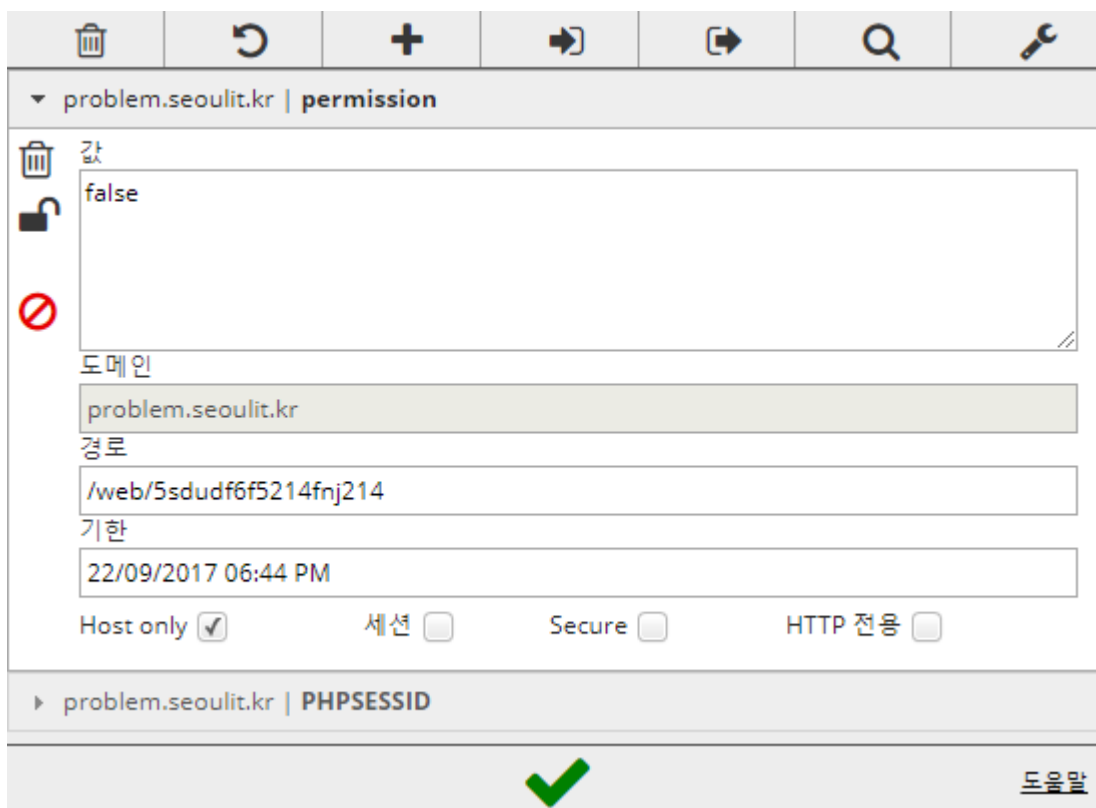
flag: Java\_is\_heavy

## 6. permission



고양이는 **냠냠**거리며 먹는걸 매우 좋아한다.  
이 동물이 고양이가 맞는지에 대해 **참거짓** 여부가 필요하다.

뭐라는지 모르겠다.



permission이라는 쿠키가 false로 되어있다.

true로 바꿔보자.

---

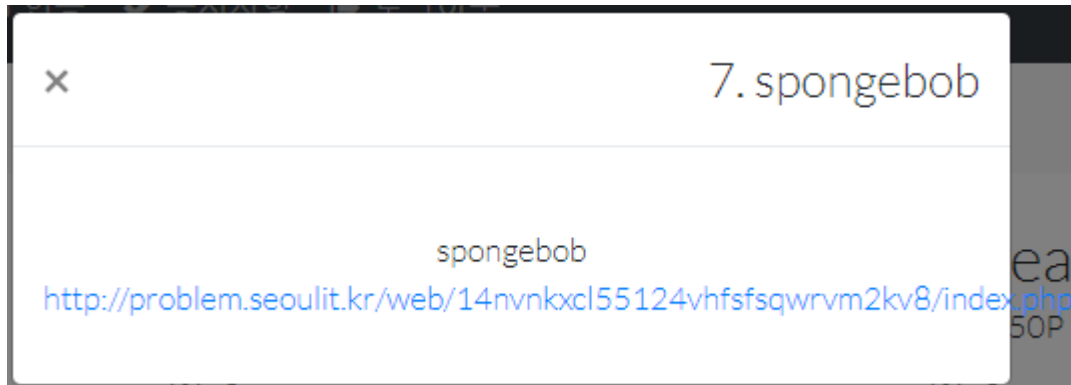
고양이는 ~~냠~~거리며 먹는걸 매우 좋아한다.  
이 동물이 고양이가 맞는지에 대해 ~~참~~거짓 여부가 필요하다.  
SCTF{NO\_CAT\_YES\_COOKIE}

flag를 준다.

**flag: NO\_CAT\_YES\_COOKIE**



## 7. spongebob



어딘가에 ....

뭘까. 소스를 보자.

```
418 <!-->
419 <!-->
420 <!-->
421 <!-->
422 <!-->
423 <!-- flag.php -->
424 <!-->
425 <!-->
426 <!-->
427 <!-->
```

소스 중 중간에 flag.php라고 주석이 있다.

들어가보자.

<http://problem.seoulit.kr/web/14nvnkxcl55124vhfsfsqwrvm2kv8/flag.php>



스펀지밥에게 편지를 보내봅시다.

[view-source](#)

소스보기가 생겼다.

들어가보자.

<http://problem.seoulit.kr/web/14nvnkxcl55124vhfsfsqwrvm2kv8/flag.phps>.

```
if(preg_match("/[a-zA-Z-0-9]/",$_GET[spongebob])) exit("no hack");
elseif(strlen($_GET[spongebob])>5) exit("no hack");
else echo $flag;
```

소스 읽어보니 영문자, 숫자가 아니고 5글자 이하로 spongebob를 주면 flag를 준다.

<http://problem.seoulit.kr/web/14nvnkxcl55124vhfsfsqwrvm2kv8/flag.php?spongebob=?>



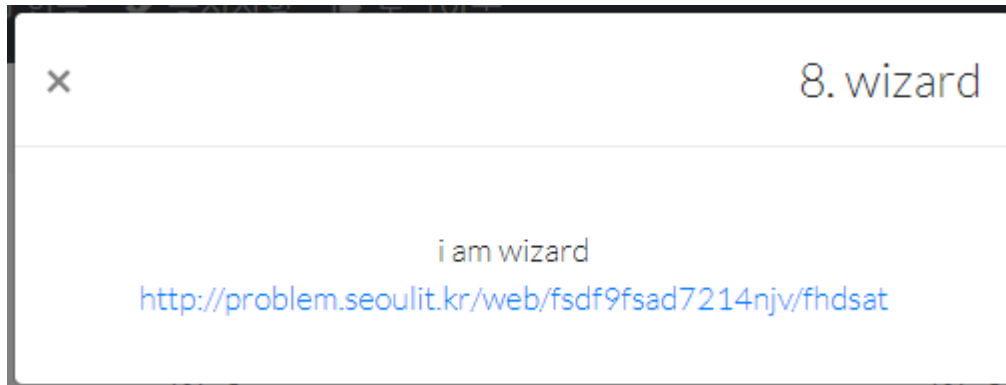
스펀지밥에게 편지를 보내주세요.

SCTF{SP0NG2B0B\_1S\_P1NGP1NG}

[view-source](#)

**flag: SP0NG2B0B\_1S\_P1NGP1NG**

## 8. wizard



change wizard

change weapon

shop

뭐지.

이것저것 눌러보니 [change wizard]나 [change weapon]은 안되고 [shop]은 되는 것 같다.



<http://problem.seoulit.kr/web/fsdf9fsad7214njv/fhdsat/?character=Henry>

Henry로 선택하고 들어가보니까 저렇게 뜬다.

remove하라고 한다.

대문자를 없애고 들어가보자

<http://problem.seoulit.kr/web/fsdf9fsad7214njv/fhdsat/?character=henry>



change wizard

change weapon

shop

없어졌다.

unknown 옵션을 선택하고 대문자를 없애서 들어가보자.

<http://problem.seoulit.kr/web/fsdf9fsad7214njv/fhdsat/?character=fitgerald>



SCTF{W1ZZARD\_POWER\_IS\_19999}

change wizard

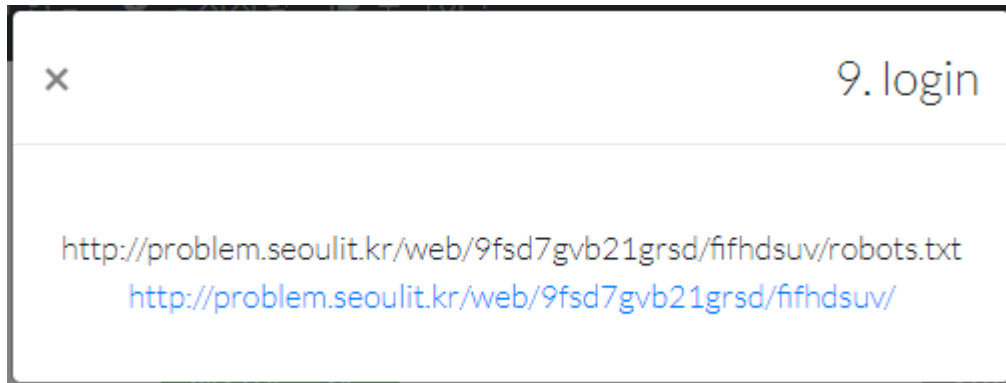
change weapon

shop

flag를 준다.

**flag: W1ZZARD\_POWER\_IS\_19999**

## 9. login



Disallow: /admin

Robots.txt 부터 들어가보니 /admin 페이지가 있다는걸 알았다.

<http://problem.seoulit.kr/web/9fsd7gvb21grsd/fifhdsuv/admin>

로 들어가보니까

<http://problem.seoulit.kr/web/9fsd7gvb21grsd/fifhdsuv/index.php>

로 리다이렉션된다.

이 경우 Location 헤더에 /index.php 값을 줘서 리다이렉션하게 하는 방식일건데,

리다이렉션되기 전에 패킷을 보면 뭔가 나올 것 같다.



```

1  import socket
2
3  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  s.connect(("problem.seoulit.kr", 80))
5
6  s.send("GET /web/9fsd7gvb2lgrsd/fifhdsuv/admin/index.php HTTP/1.1\r\n" \
7        "Host: problem.seoulit.kr\r\n" \
8        "\r\n".encode());
9
10 data = s.recv(4096).decode()
11
12 print(data)

```

```

Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: ../index.php
Set-Cookie: md5-cookie=d41d8cd98f00b204e9800998ecf8427e; expires=Fri, 22-Sep-2017 10:00:06 GMT; Max-Age=3600
Content-Length: 292
Content-Type: text/html; charset=UTF-8

admin only<!DOCTYPE html>
<head>
  <meta charset="utf-8">

```

admin only라고 한다.

md5-cookie라는 쿠키에 ""을 md5 해시한 값이 들어있다.

admin을 해시한 값으로 바꿔서 보내보자.

```

1  import socket
2
3  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  s.connect(("problem.seoulit.kr", 80))
5
6  s.send("GET /web/9fsd7gvb2lgrsd/fifhdsuv/admin/index.php HTTP/1.1\r\n" \
7        "Cookie: md5-cookie=21232f297a57a5a743894a0e4a801fc3;\r\n" \
8        "Host: problem.seoulit.kr\r\n" \
9        "\r\n".encode());
10
11 data = s.recv(4096).decode()
12
13 print(data)

```

```

Location: ../index.php
Set-Cookie: md5-cookie=d41d8cd98f00b204e9800998ecf8427e; expires=Fri, 22-Sep-2017 10:02:32 GMT; Max-Age=3600
Content-Length: 311
Content-Type: text/html; charset=UTF-8

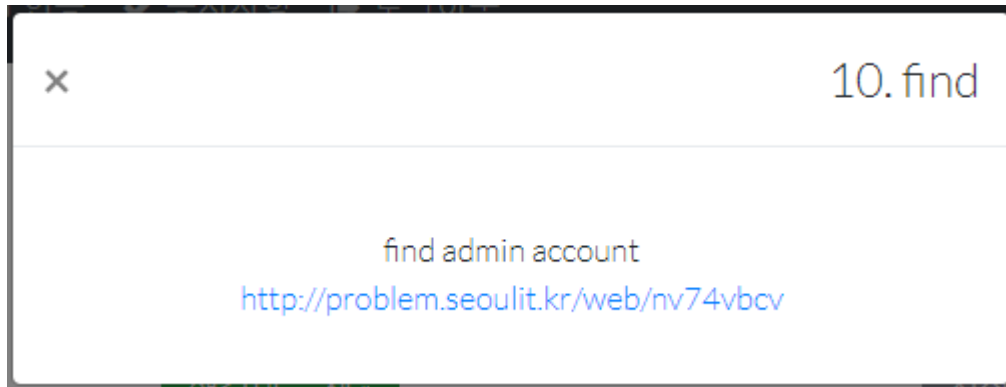
SCTF{C00K1E_1S_B2S7_S2CUR1TY}<!DOCTYPE html>
<head>
  <meta charset="utf-8">

```

flag를 준다.

**flag: C00K1E\_1S\_B2S7\_S2CUR1TY**

## 10. find

A screenshot of a login interface on a black background. It consists of two white input fields. The first field is preceded by the label 'ID :' and the second by 'PW :'. To the right of the second input field is a rectangular button with the text 'login' in a light gray font.

소스를 보자.

```
20 </form>
21 </body>
22 <!-- hint :
23 id : admin
24 pw : 0~9999
25 -->
26 </html>
27
```

주석으로 계정에 대한 힌트가 나와있다.

```

1  import urllib.parse, urllib.request
2
3  def go(i):
4      url = 'http://problem.seoulit.kr/web/nv74vbcv/1.php'
5      post_data = {
6          "id": "admin",
7          "pw": str(i)
8      }
9      headers = {
10         'User-Agent': 'Mozilla/5.0',
11         'Content-Type': 'application/x-www-form-urlencoded',
12         "Cookie": "PHPSESSID=ieu6vnlv66dd5sj0062ne94qu3"
13     }
14
15     req = urllib.request.Request(
16         url,
17         urllib.parse.urlencode(post_data).encode(),
18         headers
19     )
20     res = urllib.request.urlopen(req).read().decode()
21
22     return res != "not account"
23
24  if __name__ == '__main__':
25      for i in range(5949, 9999):
26          print(i)
27          if go(i):
28              print("Find")
29              break
30

```

브루트포스해보니 패스워드는 7780이다.

SCTF{ADM1N\_1S\_G00D}

admin:7780 으로 로그인하니 flag를 준다.

flag: ADM1N\_1S\_G00D

## 11. noob test

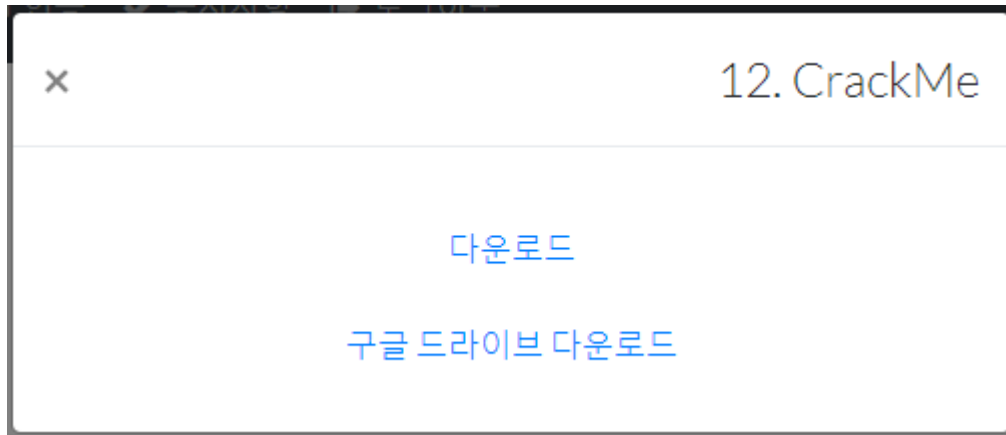


```
IDA View-A x Pseudocode-B x Pseudocode-A x Hex View
1 int __cdecl main_0()
2 {
3     char v1; // [sp+Ch] [bp-B4h]@1
4     int v2; // [sp+4Ch] [bp-74h]@2
5     int v3; // [sp+50h] [bp-70h]@1
6     int v4; // [sp+B8h] [bp-8h]@1
7     __int16 v5; // [sp+BCh] [bp-4h]@1
8
9     memset(&v1, 0xCCu, 0xB4u);
10    v4 = 1819043144;
11    v5 = 111;
12    v3 = 20170824;
13    printf("-----password-----\n");
14    while ( 1 )
15    {
16        printf("input : ");
17        scanf("%d", &v2);
18        if ( v3 == v2 )
19            break;
20        printf("this password wrong\n");
21    }
22    printf("Seoulit_is_best\n");
23    system("pause");
24    return 0;
25 }
```

Exe 파일을 IDA hex스레이 측 열어보면 flag가 뭔지 알 수 있다.

flag: Seoulit\_is\_best

## 12. CrackMe

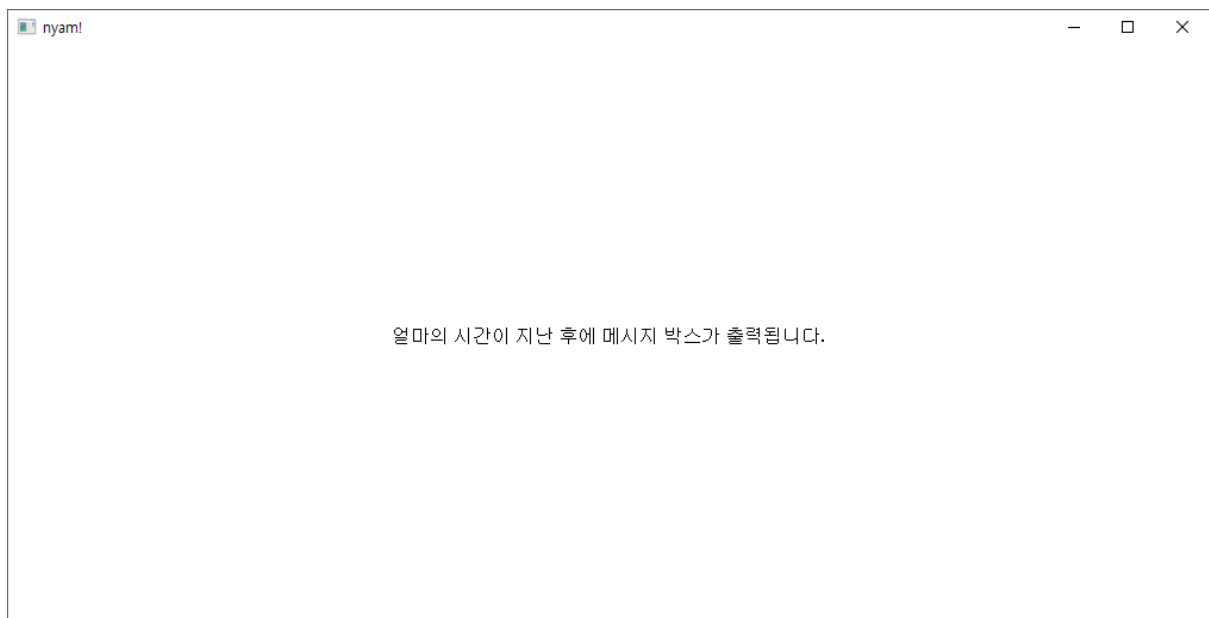
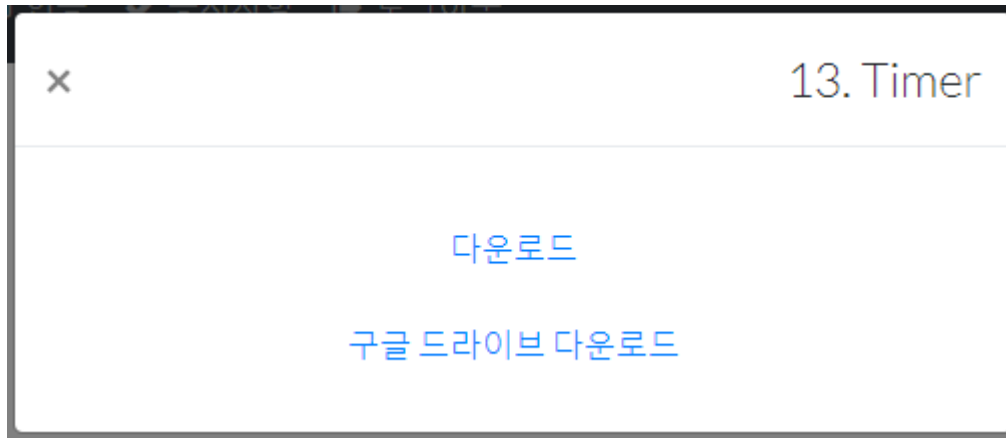


```
tion Unexplored Instruction External symbol
IDA View-A Pseudocode-A Hex View-1 Structures
1 int __cdecl main_0()
2 {
3     char v1; // [sp+Ch] [bp-58h]@1
4     char v2[4]; // [sp+4Ch] [bp-18h]@1
5     char v3; // [sp+58h] [bp-Ch]@2
6
7     memset(&v1, 0xCCu, 0x58u);
8     strcpy(v2, "s68h90i50n");
9     printf("Crack Me\n");
10    printf("Password is a series of alphabets and numbers\n");
11    while ( 1 )
12    {
13        printf("Password : ");
14        fgets(&v3, 11, &stru_422A40);
15        fflush(&stru_422A40);
16        if ( !strcmp(&v3, v2) )
17            break;
18        printf("Wrong password! Try again\n");
19    }
20    printf("Good Job!!\n");
21    return 0;
22 }
```

이 역시 Exe 파일을 IDA hex스레이 속 열어보면 flag가 뭔지 알 수 있다.

flag: s68h90i50n

## 13. Timer



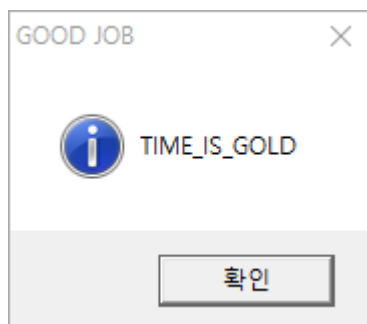
일정 시간 뒤 메시지 박스가 출력된다고 한다.

근데 한참 기다려도 안뜬다.

크랙하자.

주소	Hex	Assembly
00401110	8B 44 24 08	mov eax,dword ptr ss:[esp+0x8]
00401114	83 EC 50	sub esp,0x50
00401117	83 F8 0F	cmp eax,0xF
0040111A	56	push esi
0040111B	0F 87 B5 00 00 00	ja timer.401106
00401121	74 3A	je timer.401150
00401123	8B C8	mov ecx,eax
00401125	49	dec ecx
00401126	74 18	je timer.401140
00401128	49	dec ecx
00401129	0F 85 AE 00 00 00	jne timer.40110D
0040112F	6A 00	push 0x0
00401131	FF 15 9C 50 40 00	call dword ptr ds:[<&PostQuitMessage>]
00401137	33 C0	xor eax,eax
00401139	5E	pop esi
0040113A	83 C4 50	add esp,0x50
0040113D	C2 10 00	ret 0x10
00401140	8B 44 24 58	mov eax,dword ptr ss:[esp+0x58]
00401144	6A 00	push 0x0
00401146	6A 10	push 0x10
00401148	90	nop
00401149	90	nop
0040114A	90	nop
0040114B	6A 01	push 0x1
0040114D	50	push eax
0040114E	FF 15 A0 50 40 00	call dword ptr ds:[<&SetTimer>]
00401154	33 C0	xor eax,eax
00401156	5E	pop esi
00401157	83 C4 50	add esp,0x50
0040115A	C2 10 00	ret 0x10

SetTimer() API를 호출하는데, 그 인터벌 값을 낮게 줘보자.

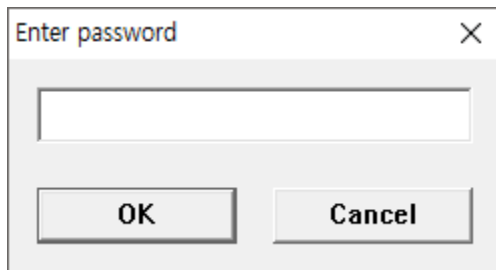


flag를 바로 알려준다.

**flag: TIME\_IS\_GOLD**



## 14. description



분석해보자.

```
1 BOOL __stdcall DialogFunc(HWND hDlg, UINT a2, WPARAM a3, LPARAM a4)
2 {
3     BOOL result; // eax@5
4
5     if ( a2 == 272 )
6     {
7         result = 1;
8     }
9     else
10    {
11        if ( a2 != 273 )
12            return 0;
13        if ( (unsigned __int16)a3 == 1 )
14        {
15            if ( sub_401060(hDlg) )
16                EndDialog(hDlg, 0);
17            result = 0;
18        }
19        else
20        {
21            if ( (unsigned __int16)a3 != 2 )
22                return 0;
23            EndDialog(hDlg, -1);
24            result = 0;
25        }
26    }
27    return result;
28 }
```

[OK] 버튼을 누르면 0x401060 주소의 함수를 호출한다.

```
24  if ( GetDlgItemTextA(hDlg, 1000, &String, 256) )
25  {
26      v8 = 0;
27      v9 = 0;
28      v10 = 0;
29      v11 = 0;
30      v7 = 0;
31      v12 = 0;
32      v13 = 0;
33      v1 = sub_401160(&v7, &unk_4050B0, 20);
34      if ( (signed int)v1 > 0 )
35      {
36          if ( v1 > 0x100 )
37              v1 = 256;
38          if ( !memcmp(&String, &v7, v1) )
39          {
40              v4 = 0;
41              v5 = 0;
42              Text = 0;
43              v6 = 0;
44              if ( sub_401160(&Text, &unk_4050C4, 12) > 0 )
45              {
46                  MessageBoxA(hDlg, &Text, Caption, 0x40u);
47                  return 1;
48              }
49          }
50      }
    else
```

거기서 입력 받은 패스워드 값을 비교(memcmp)한다.

비교하는 패스워드 값은 SECURITY\_IN\_SEOUL\_IT 이다.

패스워드가 올바르게 입력되면 MADE\_BY\_NYAM 이라고 뜬다.

#### ☆ CTF 문제 질문

보낸사람 ☆dev<pushesp@naver.com>  
받는사람 <seoulitctf@gmail.com>

Description 문제에 이상 없나요?  
Answer이 "MADE\_BY\_NYAM"라고 하는데, 인증해보면 답이 아니라고 합니다.  
만약 이상이 없는거라면, 문제 의도를 모르겠네요.. Answer랑 flag는 다른건가요?

근데 SECURITY\_IN\_SEOUL\_IT 랑 MADE\_BY\_NYAM를 둘 다 auth해봐도 flag가 아니라길래, 뭔가 놓친게 있는지 한참 해매다가, 결국 문의 메일을 보내봤다.

☆ Re: CTF 문제 질문

보낸사람 ☆seoulit.ctf<seoulitctf@gmail.com>

받는사람 dev<pushesp@naver.com>

---

안녕하세요. 서울아이티고 해킹방어대회 운영팀 입니다.

문제에 이상은 없습니다.

첫번째로 구한 flag\_두번째로 구한 flag

언더바로 구분 됩니다.

ex) FIRST\_SECOND

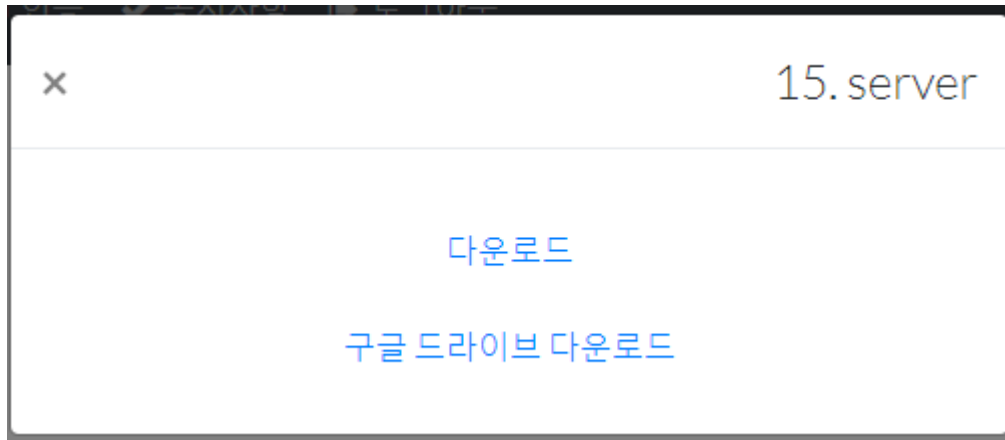
감사합니다.

그 결과, flag1\_flag2로 입력하라고 한다.

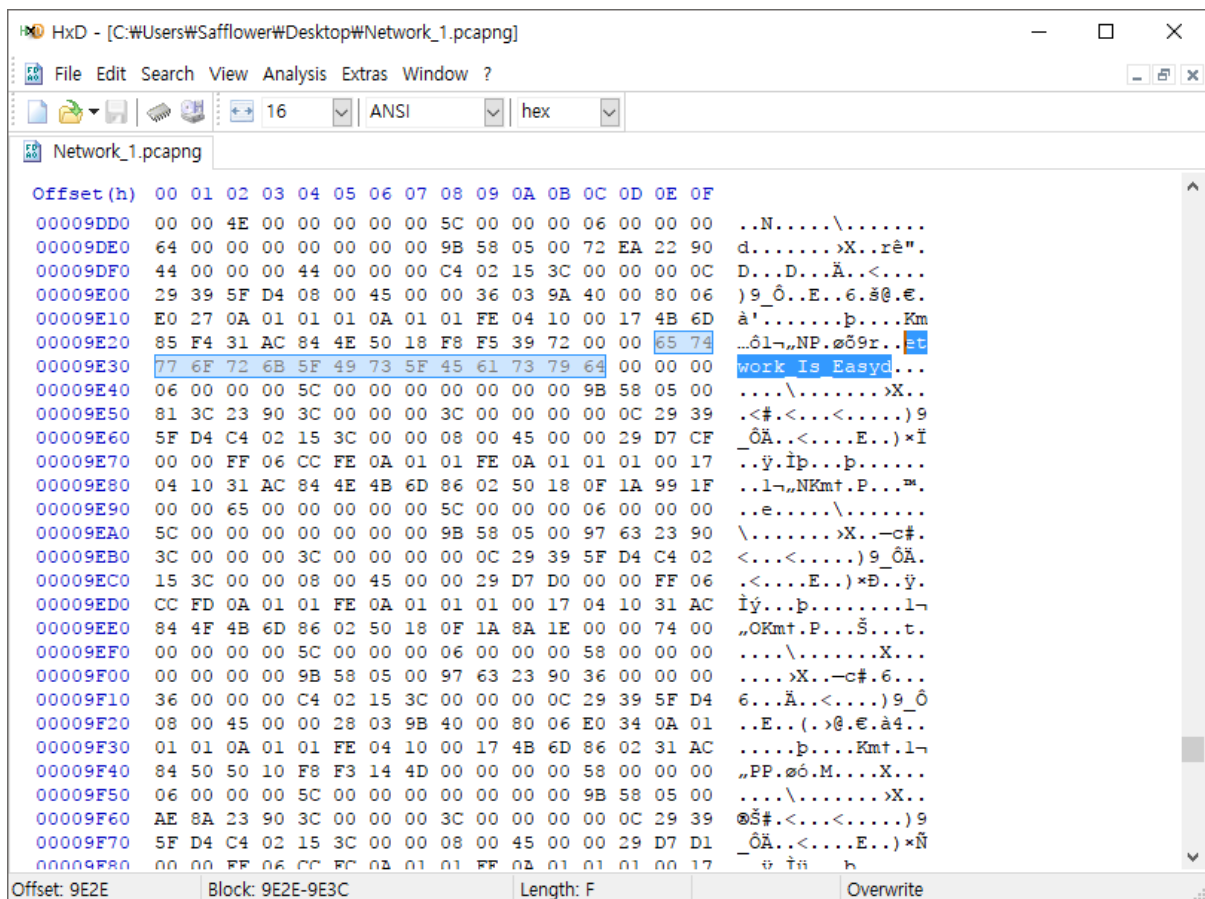
~~진작 알려줬어야지!~~

**flag: SECURITY\_IN\_SEOUL\_IT\_MADE\_BY\_NYAM**

## 15. server



패킷 덤프 파일(pcap)을 준다.



그냥 적당히 속 둘러보다가 flag 같아보이는 문자열을 발견했다.

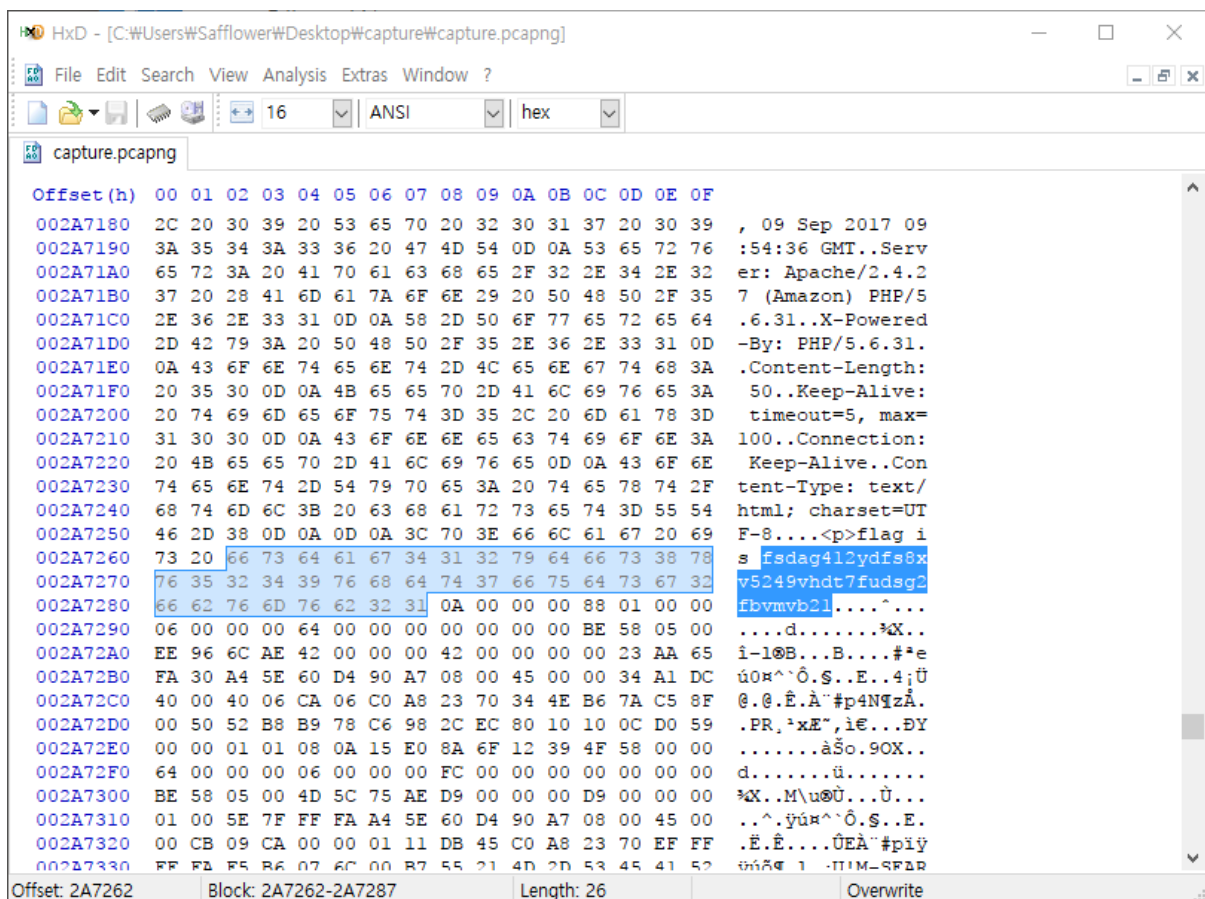
맨 앞에 N이 잘린건 계상으로 끼워맞춘다.

**flag: Network\_Is\_Easy**

## 16. capture



패킷 덤프 파일(pcap)을 준다.



그냥 평문으로 flag가 담겨있다.

**flag: fsdag412ydfs8xv5249vhdt7fudsg2fbvmvb21**

## 17. unknown



ADMIN\_PAGE

버튼이 있는 페이지다.

```

1 <html>
2   <head>
3   </head>
4   <body>
5     <script>
6       function admin_page(check){
7         if(check){
8           window.open('./hint.txt','hint');
9
10          location.href="'_11111111111111111111111111111111'+ '_11111111111111111111111111111111'+'_11111111111111111111111111111111'+'_11111111111111111111111111111111'";
11        }
12
13        else{
14          alert("Page Error\n[Permission Denied]")
15        }
16      }
17    </script>
18    <button onclick="admin_page(0)">ADMIN_PAGE
19  </button>
20 </body>
21 </html>
22
```



소스를 보니 뭔가 괴상한 스크립트가 있다.

XzE9I mEiLF8xMT0iYiIsXzExMT0iYiYIsXzExMTE9ImQiLF8xMTExMT0iZSiSxZExMTExMT0iZiIsXzExMTExMTE9Imc iLF8xMTExMTExMT0iaCl sXz MTExMTExMTExMTExMT0ibYIsXzExMTExMTExMTExMTExMTE9Ina iLF8xMTExMTExMTExMTExMT0icSiSxZExMTExMTExMTExMTExMT0ic iLF8xMTExMTExMTExMTExMTExMTExMT0idyIsXzExMTExMTExMTExMTExMTExMTExMT0ieClSxZExMTExMTExMTExMTExMTExMTExMT

Hint.txt 를 열어보니, 이상한 문자열들이 있다.

[illegible]

Base64로 디코드해봤다.

[illegible]

이런 소스가 나온다.

```
_1="a",_11="b",_111="c",_1111="d",_11111="e",_111111="f",_1111111="g",_11111111="h",_1111111
```

```
111="i",_1111111111="j",_1111111111="k",_1111111111="l",_1111111111="m",_1111111111
11111="n",_11111111111111="o",_11111111111111="p",_11111111111111="q",_1111111111
111111111="r",_111111111111111111="s",_111111111111111111="t",_111111111111111111
1="u",_111111111111111111111111="v",_111111111111111111111111="w",_111111111111111111
111="x",_111111111111111111111111="y",_111111111111111111111111="z";
```

```
_11111111111111111111+_11111111+_11111111+_111111111111111111+_11111111+_111111
11111111111111+_1+_1111+_11111111111111+_11111111+_11111111111111+_11111111111111
11+_1+_1111111+_11111+" .php"
```

"thisisadminpage.php"

아까 전의 소스랑 합쳐서 실행해보니 파일명을 알려준다.

---

[downloadp](#)

[hint]

id : admin

password : \*\*\*\*

id

password

<http://problem.seoulit.kr/web/sfhfsau124ifi/thisisadminpage.php>

접속해봤더니 로그인 폼이 있다.

---

[downloadp](#)

[hint]

id : admin

password : \*\*\*\*

id

password

SCTF{LOST\_MY\_PASSWORD\_T\_T}

패스워드는 \*\*\*\*이라고 한다.

근데 아이디를 admin으로 하고, 그냥 아무 패스워드나 치고 제출하니 flag를 준다.

???

**flag:** LOST\_MY\_PASSWORD\_T\_T