

제 4회 한세사이버보안고 중학생 정보보안 콘테스트

Write-up

```
$ ./프라푸치노를_익스로_변환합니다  
top: 5  
points: 203  
name: 여준호  
school: 은여울중학교  
solved: [6, 7, 14, 15]
```

Roman-math (?)

사이사이에 로마 숫자가 있는 LOREM IPSUM 텍스트가
주어진다....!

Lorem ipsum dolor LXXXVI sit amet, consectetur adipiscing elit. Pellentesque elit sapien, aliquet CI fringilla ultricies nec, suscipit nec augue. Suspendisse blandit dapibus gravida. Donec CX aliquam auctor enim, scelerisque fermentum tortor tempus id. Suspendisse potenti. Integer bibendum, CV lorem in aliquet ornare, mauris neque lobortis ipsum, eget interdum velit velit ut augue. Nulla vehicula iaculis augue, in viverra quam fermentum at. Aliquam ultricies efficitur justo, eu XCV pellentesque odio.

Maecenas varius LXXXVI quis risus sit amet pellentesque. Cras diam sem, scelerisque non ornare vitae, vehicula vitae CI C turpis. Cras molestie enim tortor, sit amet elementum sapien pulvinar eu. Phasellus

Proin CV ut sem ut enim ultrices rutrum scelerisque
luctus et netus et malesuada fames XCV ac turpis

Lorem ipsum dolor LXXXVI sit amet,

Etiam quis sem LXXIII quis diam imperdiet ultrices. In vulputate at odio in tincidunt. Fusce vel ante vitae erat convallis ultrices ac non dui. Nullam CXVI quis sem nec purus semper tempus euismod quis urna. Lorem ipsum dolor sit amet, consectetur XXXIX adipiscing elit. Integer at purus vel lacus gravida rhoncus quis laoreet nunc. Curabitur XCXV efficitur nulla tortor, et maximus XCV ex aliquam in. Donec tristique massa vitae quam rutrum dictum. In hac habitasse platea dictumst. Vestibulum in XCVII lectus quis massa auctor tincidunt vitae dignissim augue. Fusce varius ante nec lorem suscipit interdum.

Aenean non ipsum XCV velit. Sed posuere sapien id lorem pretium porttitor. Proin quis massa pretium, pharetra ante nec, tempus ligula. Vivamus quis bibendum turpis, in molestie sem. Nullam tincidunt dui vitae CII egestas ullamcorper. Nam vitae dui laoreet, tincidunt sapien id, porttitor ante. Curabitur blandit vulputate suscipit. Nam id pellentesque ipsum, dictum CVIII cursus nunc. Suspendisse id ex laoreet, congue eros nec, aliquet nisi. Aenean et bibendum tortor. Integer et efficitur erat. Etiam lacus lectus, semper in feugiat in, dictum sit amet ante.

Integer placerat mi justo, sit amet fermentum odio luctus at. Praesent eu massa nec XCVII mi bibendum mattis eget sit amet ipsum. In id venenatis diam. Nullam tincidunt, tellus vitae sodales porttitor, mauris erat malesuada diam, non maximus nulla ex porttitor CIII mi. Nam vulputate nec tellus non dapibus. Proin posuere mauris quis libero suscipit, cursus aliquam enim sagittis. Aenean at velit XXXIII urna. Aliquam elit ipsum, faucibus eget erat ut, auctor condimentum ante.

LXXXVI = 86
CHR(86) = V

Roman-math (?)

고로 플래그는 텍스트에 있는 로마숫자를 아스키로 바꾼 값?

roman.txt x

```
1 Lorem ipsum dolor LXXXVI sit amet, consectetur  
2  
3 Maecenas varius LXXXVI qu  
4 with open('roman.txt', 'r') as f:  
5 Etiam quis sem LXXIII qui  
6  
7 Aenean non ipsum XCV velit. Sed posuere sapien  
8  
9 Integer placerat mi justo, sit amet fermentum
```

```
msg = f.read().strip().replace('\n', ' ').split(' ')
```

갓_파(갓 파이썬이라는 뜻)로
roman.txt 파일을 읽어
단어별로 나누어진 리스트로 가져온다

Roman-math (?)

이걸 노가다 없이 구하기 위해서 노가다를 하는 거죠.

```
>>> msg  
['Lorem', 'ipsum', 'dolor', 'LXXXVI', 'sit', 'amet,', 'consectetur',  
'adipiscing', 'elit.', 'Pellentesque', 'elit', 'sapien,', 'aliquet',  
'CI', 'fringilla', 'ultricies', 'nec,', '이하생략']
```

그러니까, 이렇게 됩니다.

Roman-math (?)

이제 각 단어에 대해서 처리를 하면 됩니다.

```
roman = []
for word in msg:
    if word.isupper():
        roman.append(word)
```

string.isupper()는 string이
모두 대문자일 때만 True를 반환한다.
roman.txt의 로마 숫자는
모두 대문자로 이루어져 있으므로
로마 숫자를 따로 리스트에 저장할 수 있다.

Roman-math (?)

StackOverflow에서 검색하니 로마 문자 string을 int로 바꿔서 반환하는 함수가 있었습니다.

```
def roman_to_int(roman):
    values = [
        {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}[c]
        for c in roman
    ]
    return sum(
        +n if n >= next else -n
        for n, next in zip(values, values[1:] + [0])
    )
```

급하니깐 그냥 새로 바로 사용하겠습니다.

Roman-math(?)

이제 리스트 roman에 저장된 값을
roman_to_int()를 이용해 하나씩 문자로 바꿔서
플래그를 얻으면 됩니다.

```
flag = ''  
for n in roman:  
    flag += chr(roman_to_int(n))  
print(flag)
```

Roman-math(?)

최종 악스

```
def roman_to_int(roman):
    values = [
        {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}[c]
        for c in roman
    ]
    return sum(
        +n if n >= next else -n
        for n, next in zip(values, values[1:] + [0])
    )

with open('roman.txt', 'r') as f:
    msg = f.read().strip().replace('\n', ' ').split(' ')

roman = []
for word in msg:
    if word.isupper():
        roman.append(word)

flag = ''
for n in roman:
    flag += chr(roman_to_int(n))
print(flag)
```

Roman-math(?)

```
$ python roman.py
```

Veni_Vedi_I+I'i_a_flag!

이렇게 해서... 1등으로... 끝았습니다~

FOUND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO

FOUND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO
UND - FLAG-FOUND - FLAG-FO

Racing Car

IDA로 바이너리를 예리 FLAG가 보입니다.



IDA - race /Users/junhoyeo/Downloads/RacingCar/race
No debugger

Functions window

Function name

- sub_8049030
- _puts
- _exit
- _strlen
- __libc_start_main
- _write
- _creat
- _remove
- _close
- _gmon_start_
- _start
- sub_8049103
- _dl_relocate_static_pie
- _x86_get_pc_thunk_bx
- deregister_tm_clones
- register_tm_clones
- _do_global_dtors_aux
- frame_dummy
- main
- libc csu init

Pseudocode-A

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     size_t v3; // eax
4     char s[4]; // [esp+1h] [ebp-2Dh]
5     int fd; // [esp+1Eh] [ebp-10h]
6     char *file; // [esp+22h] [ebp-Ch]
7     int *v8; // [esp+26h] [ebp-8h]
8
9     v8 = &argc;
10    strcpy(s, "flag_is_{K@rt_ Rider_ G00d}\n");
11    file = "/home/flag.tmp";
12    fd = creat("/home/flag.tmp", 0x180u);
13    if ( fd < 0 )
14    {
15        puts("can not creat a temporary file.");
16        remove(file);
17        exit(0);
18    }
19    v3 = strlen(s);
20    write(fd, s, v3);
21    close(fd);
22    remove(file);
23
24 }
```

"flag_is_{K@rt_ Rider_ G00d}\n"

Output window

384

Python

AU: idle Down Disk: 15GB

Racing Car

그냥 보입니다.

flag{K@rt_Rider_Good}

아호~

FOUND - FLAG-FOUND - FLAG-FOU

ND - FLAG-FOUND - FLAG-FOU

Bonus - 1

나중에 딱 종료전에 들어와보니 생긴거

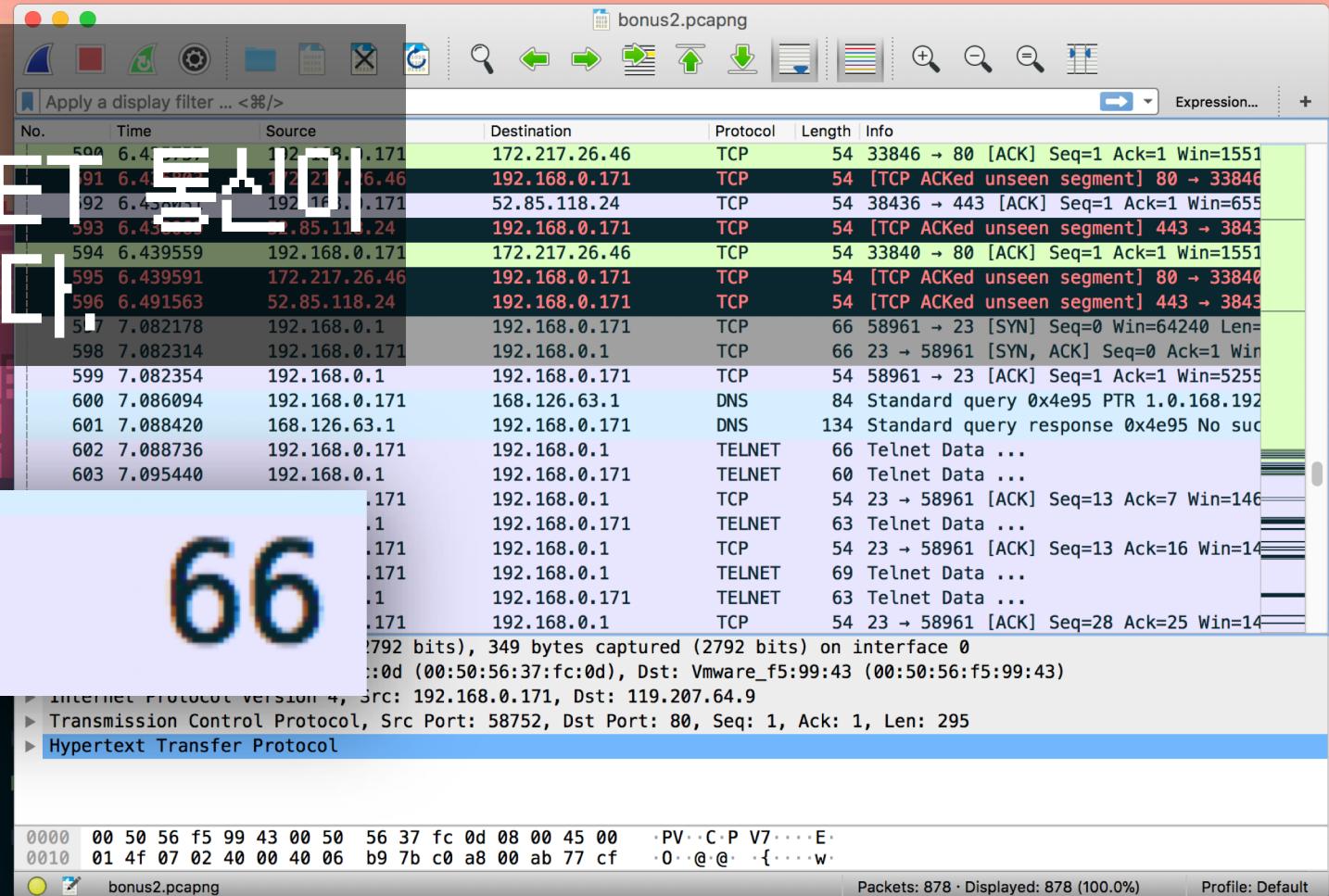


웹사이트 링크가 주어지는데 페이지 주석에 FLAG가 있습니다.
플래그 기억안남,,,

Bonus -2

나중에 딱 종료전에 들어와보니 생긴거 2

패킷 파일이 주어진다.
와이어샤크로 열면 TELNET 통신이
미루어진 흔적을 볼 수 있다



Bonus -2

와우~~~~!

Password: what_a_bonus

아무거나 하나 끌라서

Follow TCP Stream하면
뭐뭐 했는지 나온다.

Wireshark · Follow TCP Stream (tcp.stream eq 18) · bonus2.pcapng

..... #.'..... #.'.....x.....'.....ANSI.....!.....CentOS release 6.9
(Final)
Kernel 2.6.32-696.el6.x86_64 on an x86_64
...localhost.localdomain login: ...ggoooddhhaannsseeeii

Password: what_a_bonus

Last login: Sat Nov 3 13:39:05 from 192.168.0.1
[godhansei@localhost ~]\$ llss --ll

total 0
[godhansei@localhost ~]\$ ccdd //

|\$ llss --ll

File	Permissions	Last Modified	Size	Type
..	drwxr-xr-x.	root root	4096 Feb 2 2018	[01;34m.[0m
.	drwxr-xr-x.	root root	1024 Jan 31 2018	[01;34mbin.[0m
urwxr-xr-x.	root root	4096 Mar 22 2017	[01;34mcgroup.[0m	
drwxr-xr-x.	21 root root	3840 Nov 3 13:32	[01;34mdev.[0m	
drwxr-xr-x.	118 root root	12288 Nov 3 13:36	[01;34metc.[0m	
drwxr-xr-x.	4 root root	4096 Nov 3 13:36	[01;34mhome.[0m	
dr-xr-xr-x.	11 root root	4096 Jan 31 2018	[01;34mlib.[0m	
dr-xr-xr-x.	9 root root	12288 Feb 2 2018	[01;34mlib64.[0m	
drwx-----.	2 root root	16384 Jan 31 2018	[01;34mlost+found.[0m	
drwxr-xr-x.	2 root root	4096 Aug 29 11:51	[01;34mmedia.[0m	
drwxr-xr-x.	2 root root	0 Nov 3 13:32	[01;34mmisc.[0m	
drwxr-xr-x.	3 root root	4096 Feb 1 2018	[01;34mmnt.[0m	
drwxr-xr-x.	2 root root	0 Nov 3 13:32	[01;34mnet.[0m	
drwxr-xr-x.	3 root root	4096 Jan 31 2018	[01;34mopt.[0m	
dr-xr-xr-x.	179 root root	0 Nov 3 13:32	[01;34mproc.[0m	
dr-xr-x---	20 root root	4096 Nov 3 13:39	[01;34mroot.[0m	
dr-xr-xr-x.	2 root root	12288 Feb 1 2018	[01;34msbin.[0m	
drwxr-xr-x.	7 root root	0 Nov 3 13:32	[01;34mselinux.[0m	
drwxr-xr-x.	2 root root	4096 Sep 23 2011	[01;34msrv.[0m	
drwxr-xr-x.	13 root root	0 Nov 3 13:32	[01;34msys.[0m	

Packet 765. 86 client pkt(s), 78 server pkt(s), 140 turn(s). Click to select.

Enter the current selection (2136 bytes)

Show and save data as ASCII

Stream 18

Find:

Help Filter Out This Stream Print Save as... Back Find Next Close

Bonus - 2

FLAG는 Telnet 접속에 사용한 비밀번호다.

what_a_bonus

Telnet 취약점을 이용한 문제였다.
(기본적으로 패킷을 암호화하지 않음)

DU-DU-DDU-DU

끌나고 끈 뚜드뚜드

Packet list에서 'flag' string으로 검색 ->

Follow TCP Stream으로 FTP로 flag 파일이 이동한 것을
알 수 있음을

Wireshark · Follow TCP Stream

PORT 192,168,55,212,238,135

200 PORT command successful. Consider using PASV.

STOR flag

150 Ok to send data.

226 Transfer complete.

QUIT

221 Goodbye.

D D U - D U - D D U - D U

힛츄잇대 뚜뚜뚜뚜뚜

FTP-DATA로 필터링해서
실제 데이터가 전송된 때를 확인

The screenshot shows a Wireshark interface with the following details:

- File:** ftp-pink.pcapng
- Selected Filter:** ftp-data
- Table Headers:** No., Time, Source, Destination, Protocol, Length, Info
- Table Data:** A list of 298 rows, each representing a network frame. The 'Info' column for most frames reads "FTP Data: 1448 bytes (PORT) (STOR fla...)".
- Bottom Status Bar:** Frame 29867: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
- Bottom Details:** Ethernet II, Src: AsustekC_3e:6e:6e (d8:50:e6:3e:6e:6e), Dst: EfmNetwo_0a:1c:41 (88:36:6c:0a:1c:41)
Internet Protocol Version 4, Src: 192.168.55.212, Dst: 192.168.55.91
Transmission Control Protocol, Src Port: 61063, Dst Port: 20, Seq: 1, Ack: 1, Len: 1448

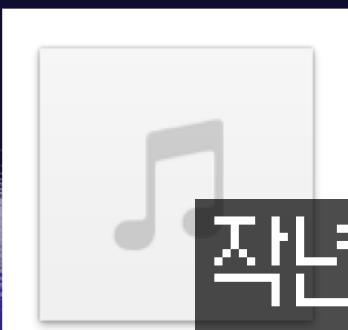
D D U - D U - D D U - D U

전송된 파일을 열어파

파일 헤더에 PK → 암축파일?

RAW로 저장하고 압축하지 않는

flag.wma II | 일이 있음



flag.wma

The screenshot shows a Wireshark window with the title "Wireshark · Follow TCP Stream (tcp.stream eq 150) · ftpink.pcapng". The main pane displays a sequence of Morse code characters (dots and dashes) in ASCII hex dump format. The sequence starts with "MORSE" followed by a series of dots and dashes. The interface includes standard Wireshark controls like "Show and save data as ASCII", "Stream 150", and "Find" buttons.

MORSE...
P.....R.....O.....S.....E.....
Pkt 127 of 68. Click on pkts, or double click on pkts, to turn(s). Click to select.
Entire conversation (84 kB)

Show and save data as ASCII Stream 150

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

DU-DU-DDU-DU

탁따 딱따따따 탁탁따따따

Translate a Message

Input:

.-

Output:

N<AS>7W0RK!

Translate 



한자리 게임
 

mp3으로 변환한 뒤
음만 따서 직접 디코딩했다.

N<AS>7W0RK !

DU-DU-DDU-DU

영영 알 수 없는 것은 아니였다.

FOUND - FLAG-FOUND - FLAG-FOU

NETWORK !

야마 중간엔 ELT 301 있어겠지?

FOUND - FLAG-FOUND - FLAG-FOU

The End

좋은 대회 감사합니다.

감사합니다~!