




성원이형 과제

2017년 2월 12일 일요일 오후 4:37

문제1

우선은 중에 가장 쉬워 보이는 것 3문제를 다운받았다.

	FortuneCookie	20
	CDC8501BE6FEC4B2D1FCC7B80DD129F...	20
	F2EE31BCB5BD7DE78280E949877018AA...	20

FortuneCookie

```
scriptkid@ubuntu:~/Downloads/sweetchip/fortunecookie$ ./346fb9a0df23fe119fdb62d7a197a66c
Segmentation fault (core dumped)
```

열자마자 크래쉬가 난다.

ida 32bit로 열어보았다.

```
v12 = &argc;
v11 = *MK_FP(__GS__, 20);
setvbuf(stdout, 0, 2, 0);
loadkey();
v9 = 100;
selector = 0;
puts("== [Fortune Cookie] ==");
puts("== ADVANCED Memory Corruption Detector. ==");
puts("== Can you break this one? ==");
seed = time(0);
srand(seed);
while ( 1 )
{
    print_menu();
    memset(&s, 0, 0x64u);
    v3 = rand();
    v4 = rand() * v3;
    v5 = rand();
    v10 = v4 * v5;
    g_canary = v4 * v5;
    printf("> ");
    __isoc99_scanf("%d", &selector);
    __fpurge(stdin);
    if ( selector != 1 )
        break;
    printf("Input your string : ");
    input_wrap(&s, v9);
    printf("This is your string : %s\n", &s);
    sleep(1u);
    if ( check_canary(&g_canary, &v10, 4) != 1 )
    {
        puts("[!] Attack Detected.\nBye :pp");
        sleep(1u);
        exit(0);
    }
}
if ( selector == 2 )
    printf("Good bye :p");
else
    puts("Wrong Number..");
sleep(1u);
result = 0;
v7 = *MK_FP(__GS__, 20) ^ v11;
```

와...심볼이 다 있다 ㅎㅎ

열었을 때 크래시가 나던 건

```
int loadkey()
{
    FILE *stream; // ST1C_4@1

    stream = fopen("/home/fortune_cookie/flag", "r");
    fread(&key, 0x64u, 1u, stream);
    return fclose(stream);
}
```

여기서 처음에 플래그를 불러오는데 플래그 파일이 없어서 그렇다.

```
scriptkid@ubuntu:~/Downloads/sweetchip/fortunecookie$ ./346fb9a0df23fe119fdb62d7a197a66c
== [Fortune Cookie] ==
== ADVANCED Memory Corruption Detector. ==
== Can you break this one? ==
=====
1. Try Exploit.
2. Give up.
=====
>
```

플래그 파일을 만들어 준 후 프로그램을 다시 실행해 보았다.

프로그램을 조금 분석해 보니

처음에 파일에서 플래그 값을 읽어와 bss영역에 저장한다.

그 후 카나리를 3개의 rand함수를 이용하여 만든 후, 곱셈 연산을 하여 전역변수(bss영역)에 저장한다. (플래그 값이 들어있는 변수와 들어있는 영역이 같아서 수상하다)

그리고 다음 함수를 통해 string값을 100개까지 입력 받는다.

```
int __cdecl input_wrap(int a1, int a2)
{
    char v3; // [sp+7h] [bp-11h]@2
    int v4; // [sp+8h] [bp-10h]@0
    int i; // [sp+Ch] [bp-Ch]@1

    for ( i = 0; i <= a2; ++i )
    {
        v3 = getchar();
        if ( v3 == 10 )
            return i;
        *(_BYTE *)(a1 + i) = v3;
    }
    return v4;
}
```

이 함수는 v3==10일때(Line Feed) 일 때 까지 입력을 받아 배열에 저장한다.

10	0x0A	LF
----	------	----

이곳이 메인의 변수들인데, s가 딱 100개이고, v9에는 100이라는 숫자가 들어가며, v10에는 우리가 원하는 캐너리 값이 들어있다.

```
char s; // [sp+14h] [bp-78h]@2
int v9; // [sp+78h] [bp-14h]@1
int v10; // [sp+7Ch] [bp-10h]@2
int v11; // [sp+80h] [bp-Ch]@1
int *v12; // [sp+88h] [bp-4h]@1
```

이때 Line Feed를 넣지 않고 100개의 문자를 input_wrap에 넣어 준다면

```
printf("This is your string : %s\n", &s);
```

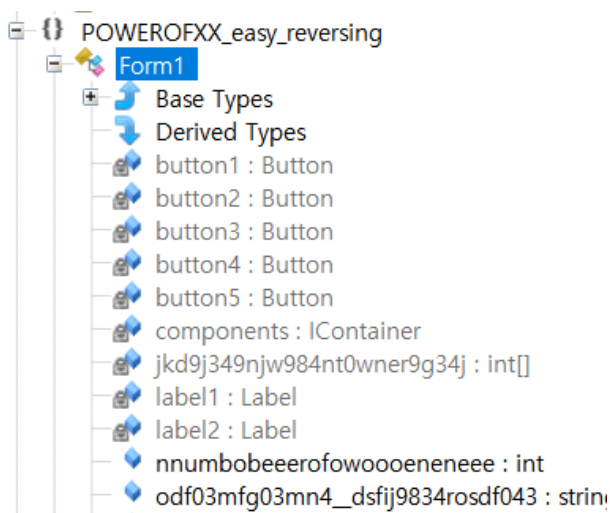
이 printf문에서 v9변수의 값이 찍힌다. 그러나 우리가 원하는 값은 v10이다.

어떻게 해야할까 $\pi\pi$ 아직은 감이 잘 오진 않지만 뭔가 풀 수 있을 거 같은 느낌이 오는 문제였다.

Easy.Net

이 문제는 C#리버싱 문제다.

ILSpy를 통해 디컴파일 해 보았다.



음...무언가 코드가 많이 있다. 일단 조금 쓸 줄 아는 java랑 형태가 비슷해서 분석하기는 편해 보인다.

다만 변수명이 많이 이상했다 ㅎㅎ