2026 SSD CTF

후기 및 문제 풀이

IT 컨퍼런스

10216 제현승

Contents

01 SSD CTF 소개

02 대회 진행 및 참가 경험

03 Navigation 문제 풀이

04 참가 소감

SSD CTF 란?

3개 학교가 함께 진행한 CTF 대회
여러 분야의 문제를 풀고 Flag를 제출해 점수를 얻는 방식

대회 정보

- 일시: 2026.08.29 ~ 08.30
- 장소: HSPACE WHITE ROOM
- 분야: Misc, Reversing, Crypto, Pwnable, Web, Programming



02 참가 인원

선린인터넷고

15명

세명컴퓨터고

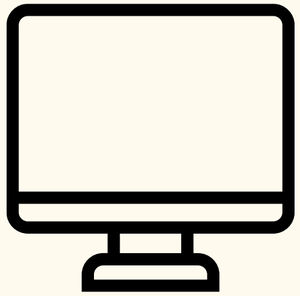
7명

한국디지털미디어고

15명

참가 학생 37명 · 스태프 포함 약 40명

03 대회 주요 일정



SSD CTF

분야별 문제 풀이



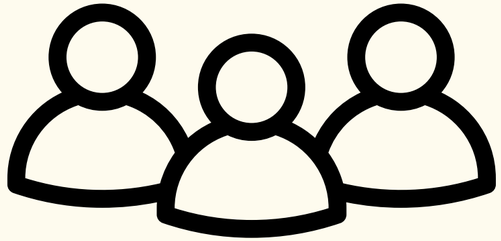
Live CTF

Pwn · Reversing · Web



Write-Up 발표

8월 30일 10:00~11:20



시상식 및 네트워킹

8월 30일 11:20~13:00

04 대회 진행 방식

Jeopardy CTF

분야별 문제를 자유롭게 선택

Flag를 제출하면 점수 획득

다른 학교가 먼저 풀면 해당 문제의 점수가 낮아짐

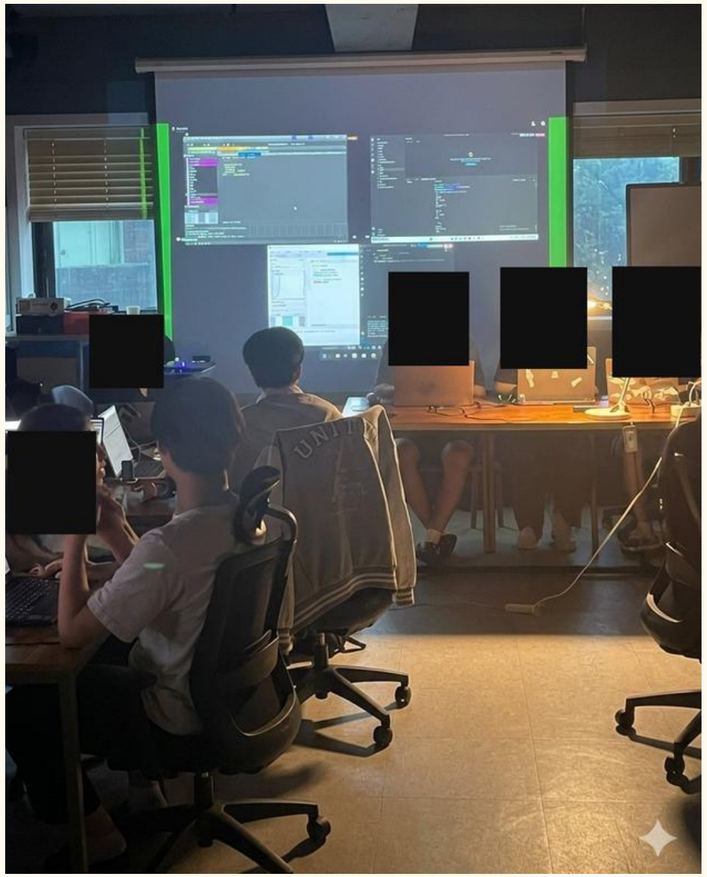
Live CTF

정해진 시간에 진행

Pwn · Reversing · Web

라운드별 제한 시간 1시간

05 Live CTF 참가 경험



Pwnable Live CTF



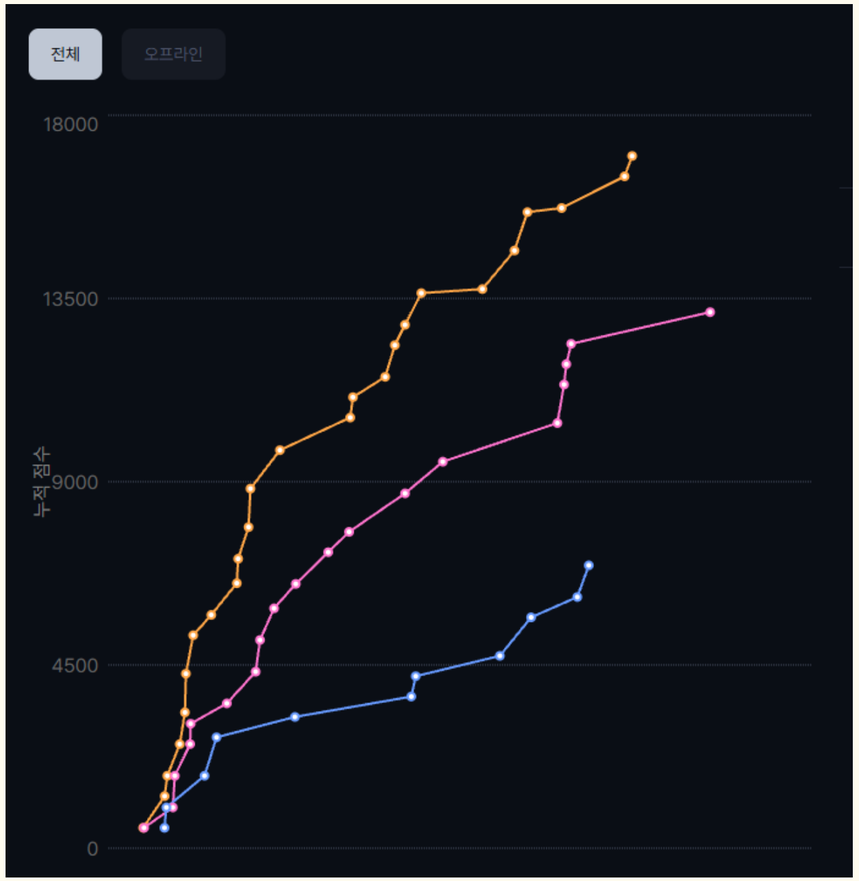
Reversing Live CTF

06 대회 문제 목록



대회 문제 목록

07 대회 결과



1	한국디지털미디어고등학교	15,954
2	선린인터넷고등학교	13,914
3	세명컴퓨터고등학교	7,696

세명컴퓨터고등학교 최종 3위 | 7,696점

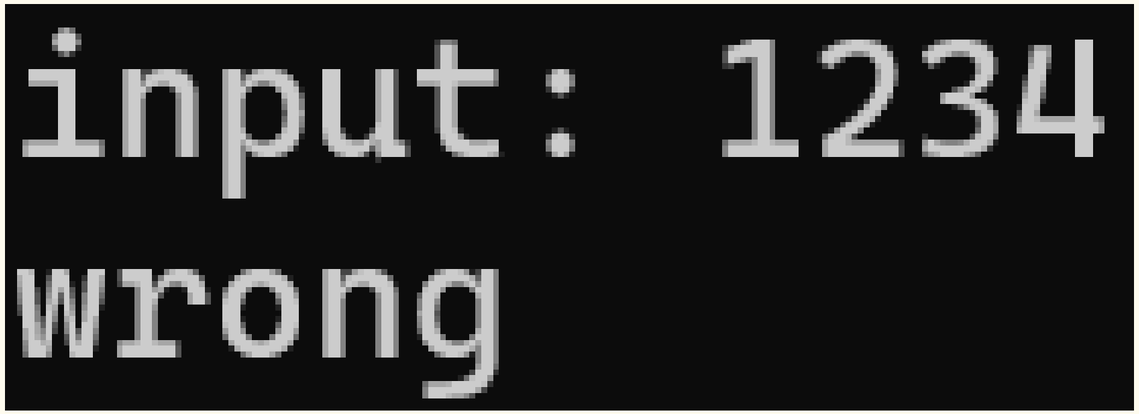
08 SSD CTF를 마치며



대회 종료 후 단체 사진

Navigation

- 분야 : Reversing
- 대회 당시 해결하지 못한 문제
- 대회 이후 직접 다시 분석



10 ltrace 실행

ltrace : 프로그램이 사용하는 라이브러리 함수를 확인하는 도구

```
ltrace ./navigation
setvbuf(0x7f7af1a055c0, 0, 2, 0)                                    = 0
write(1, "input: ", 7input: )                                              = 7
read(012345678901234567890123456789012
, "12345678901234567890123456789012"..., 32)                  = 32
fork()                                                              = 684
waitpid(684, 0, 0wrong
SSD{debugger_took_the_wrong_exit}
```

ltrace로 실행하자 Flag처럼 보이는 문자열이 출력됨
하지만 실제 정답이 아니어서 IDA로 추가 분석

11 IDA 분석 과정

```
1  void __fastcall __noreturn sub_17B0(_QWORD *a1)
2  {
3    if ( ptrace(PTRACE_TRACEME, 0, 0, 0) != -1 )
4    {
5      raise(19);
6      qword_4060 = *a1 ^ 0x6A09E667F3BCC909LL;
7      qword_4068 = a1[1] ^ 0xBB67AE8584CAA73BLL;
8      qword_4070 = a1[2] ^ 0x3C6EF372FE94F82BLL;
9      qword_4078 = a1[3] ^ 0xA54FF53A5F1D36F1LL;
10     sub_1440(0);
11     sub_1440(1);
12     sub_1440(2);
13     sub_1440(3);
14     sub_15E0(a1);
15     _exit(0);
16   }
17   puts("wrong");
18   sub_14B0(&unk_2060, 33);
19   _exit(0);
20 }
```

```
1 _BOOL8 sub_13E0()
2 {
3   return (qword_4068 ^ 0xF5FF66F4D0A7F20ALL
4         | qword_4060 ^ 0xDE3F49775829D465LL
5         | qword_4070 ^ 0x3820642A064FB214LL
6         | qword_4078 ^ 0xE34A474287344A22LL) == 0;
7 }
```

ptrace로 프로그램 추적 여부를 확인하는 부분 발견

sub_1440을 4번 호출해 입력값을 계산

마지막에 정해진 값과 비교하는 부분 확인

12 입력값 계산 과정

```
13  v1 = a1;
14  v2 = qword_4060[a1];
15  v3 = qword_2140[a1];
16  v4 = ((_BYTE)v1 + 1) & 3;
17  qword_4060[v1] = qword_4060[v4] ^ __ROL8__(v3 + v2, byte_20E8[v1]);
18  v5 = sub_1280(qword_4060[v4] ^ v3, qword_4060, ((_BYTE)v1 + 3) & 3, &unk_20E0);
19  result = qword_4060[v8] + __ROL8__(v5, *(_BYTE *)(v6 + v7));
20  qword_4060[v8] = result;
21  return result;
```

원래 프로그램의 계산

입력값 → XOR · 비트 회전 · 덧셈 → 비교값

입력값을 구하는 과정

비교값 → 계산을 거꾸로 진행 → 입력값

13 Python으로 입력값 구하기

```python
for i in range(3,-1,-1):
    v1 = (i+1)%4
    v2 = (i+2)%4
    v3 = (i+3)%4

    x = rol(qword_2100[i] ^ numbers[v1],unk_20E0[i])
    numbers[v3] = (numbers[v3] - x) & 0xFFFFFFFFFFFFFFFF

    x = rol((qword_2120[i]+numbers[v2])& 0xFFFFFFFFFFFFFFFF, byte_20E4[i])
    numbers[i] = numbers[i] ^ x

    x = ror(numbers[i] ^ numbers[v1],byte_20E8[i])
    numbers[i] = (x-qword_2140[i]) & 0xFFFFFFFFFFFFFFFF

result = ""
for i in range(4):
    num = numbers[i] ^ xordata[i]
    for j in range(8):
        result += chr(num % 256)
        num //= 256

print(result)
```

```
NAVIGATION-PTRACE-ROUTE-2026!!!!
```

```
input: NAVIGATION-PTRACE-ROUTE-2026!!!!
correct
SSD{ptr4c3_tr4c3r_f1nd5_r34l_r0ut3}
```

마지막 비교값을 기준으로 XOR, 비트 회전, 덧셈 등을 거꾸로 계산
계산을 통해 32바이트 입력값을 구했고
프로그램에 입력하니 correct와 Flag가 출력됨

14 참가소감



- 처음 참가한 CTF라 모르는 문제도 많고 생각보다 어려웠음
- 다른 학교 학생들이 문제 푸는 모습을 보면서 배울 점이 많았음
- 대회 이후 못 풀었던 문제를 다시 분석하면서 부족한 부분을 알게 됨

감사합니다

IT 컨퍼런스

10216 제현승