

Cheat Engine

With The Binding of Isaac : Rebirth

SCP | 이지훈 2024.10.29
2024-2학기 SCP 내부 세미나 B조



INDEX

01 | 치트 엔진 : 소개

Cheat Engine

04 | 아이작 게임 : 메모리 변조

In memory

02 | 치트 엔진 : 기능

Cheat Engine

05 | 아이작 게임 : 코드 인젝션

Code Injection

03 | 아이작 게임 : 분석 준비

The Binding of Isaac

06 | 아이작 치트 실행기

Hack On/off Program.exe



01 | 치트엔진 : 소개



Cheat Engine

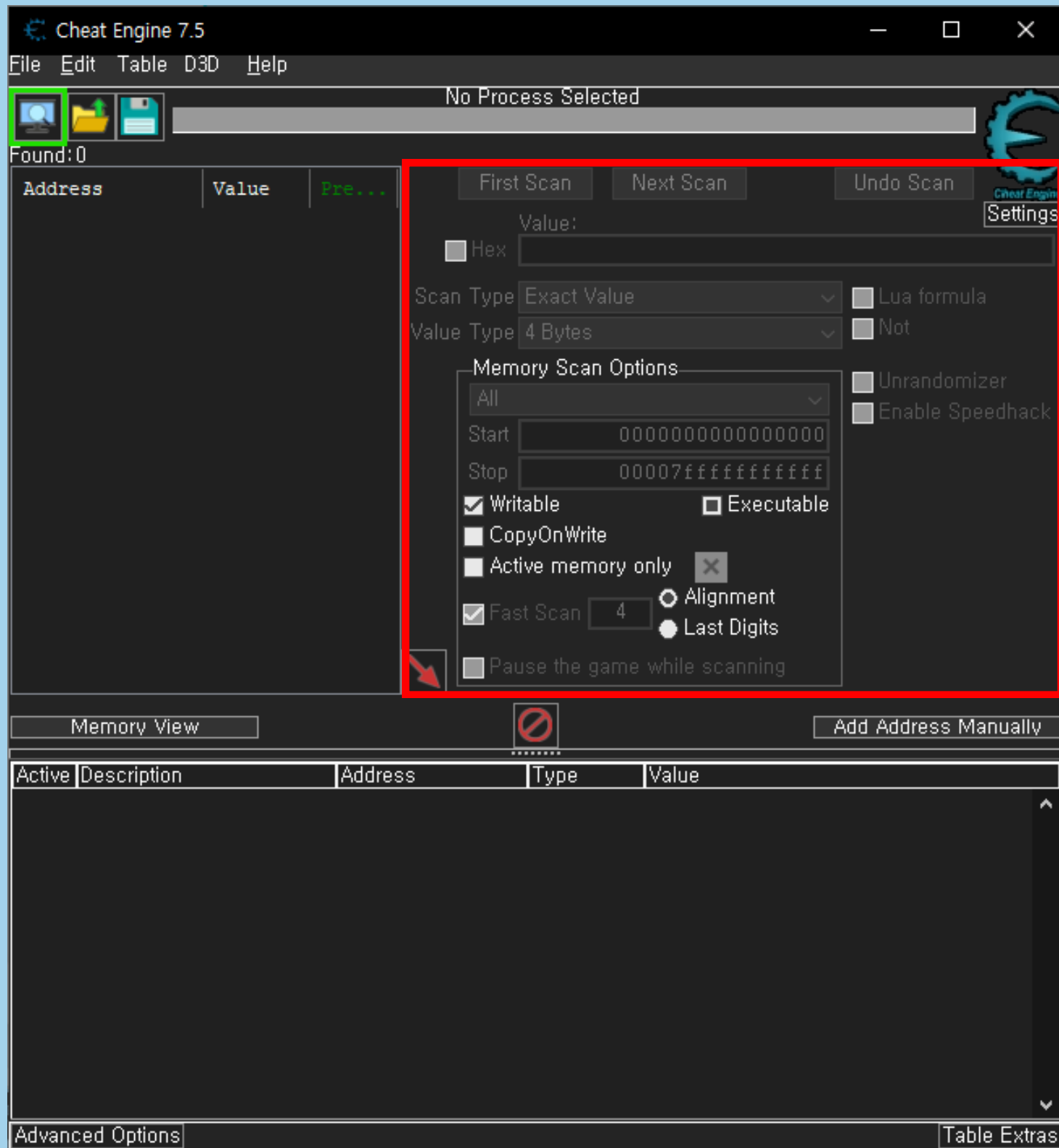
세상에서 제일 유명한(?) 메모리 에디터 / 디버거

메모리에서 원하는 값을 쉽게 찾아 접근하고 변경할 수 있음

그 외에도 디버깅, 레지스터 값 변경, 코드 패치, 코드 인젝션 등
여러 기능들 제공



02 | 치트엔진 : 기능



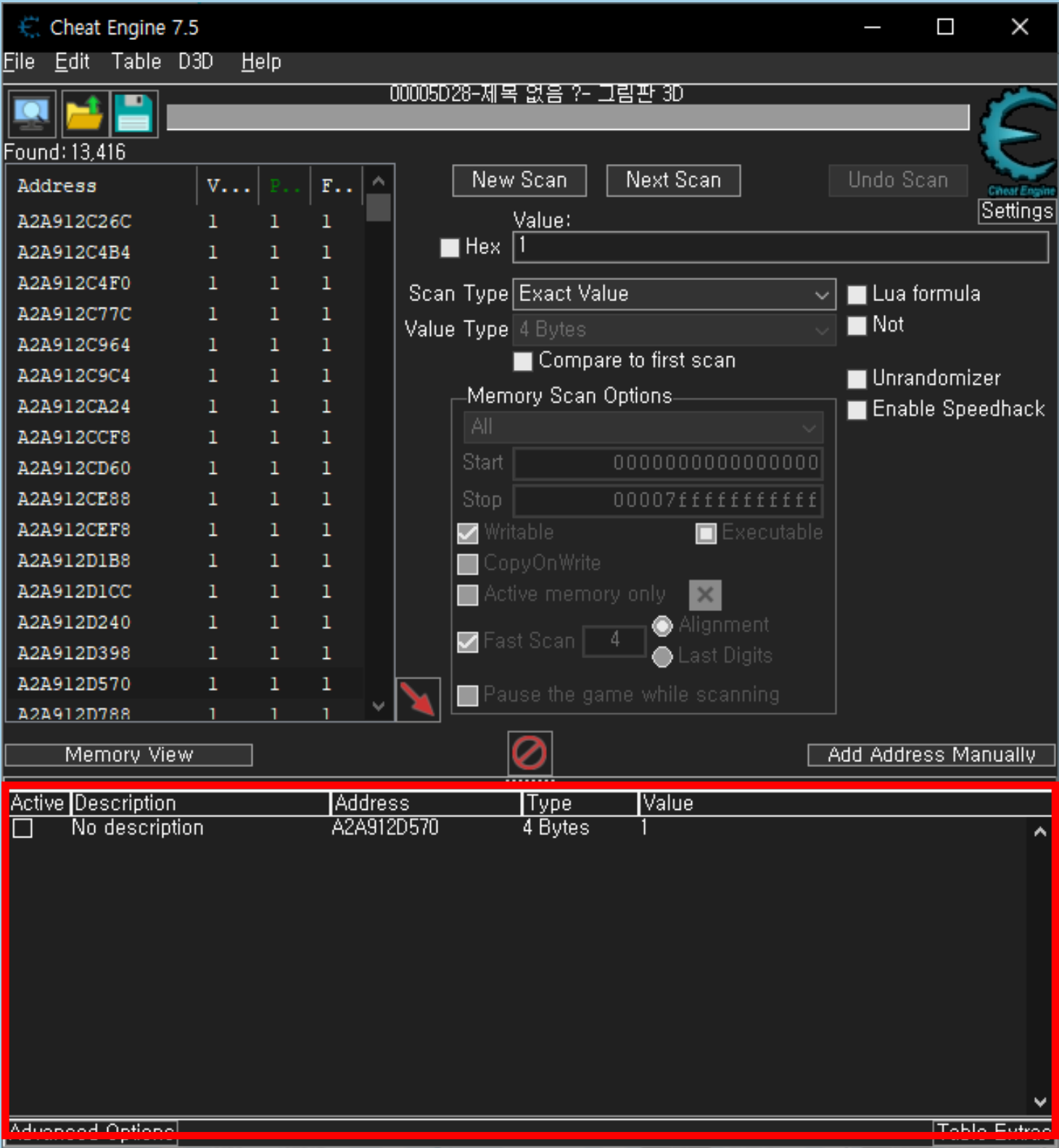
Memory Scan

원하는 값을 스캔하여 찾아주는 기능

- 원하는 타입(바이트수, 문자열, 실수형태, 이진수 등)
- 정확한 값, 그 이상/이하의 값, 그 사이의 값 등
- 스캔 이후 해당 값에서 다른 값으로 바뀐 것 또한 스캔 가능



02 | 치트엔진 : 기능



Memory List

원하는 메모리 주소를 저장하고 변조할 수 있는 곳

값 고정, 타입 변경, 값 변경 등 강력한 기능 제공



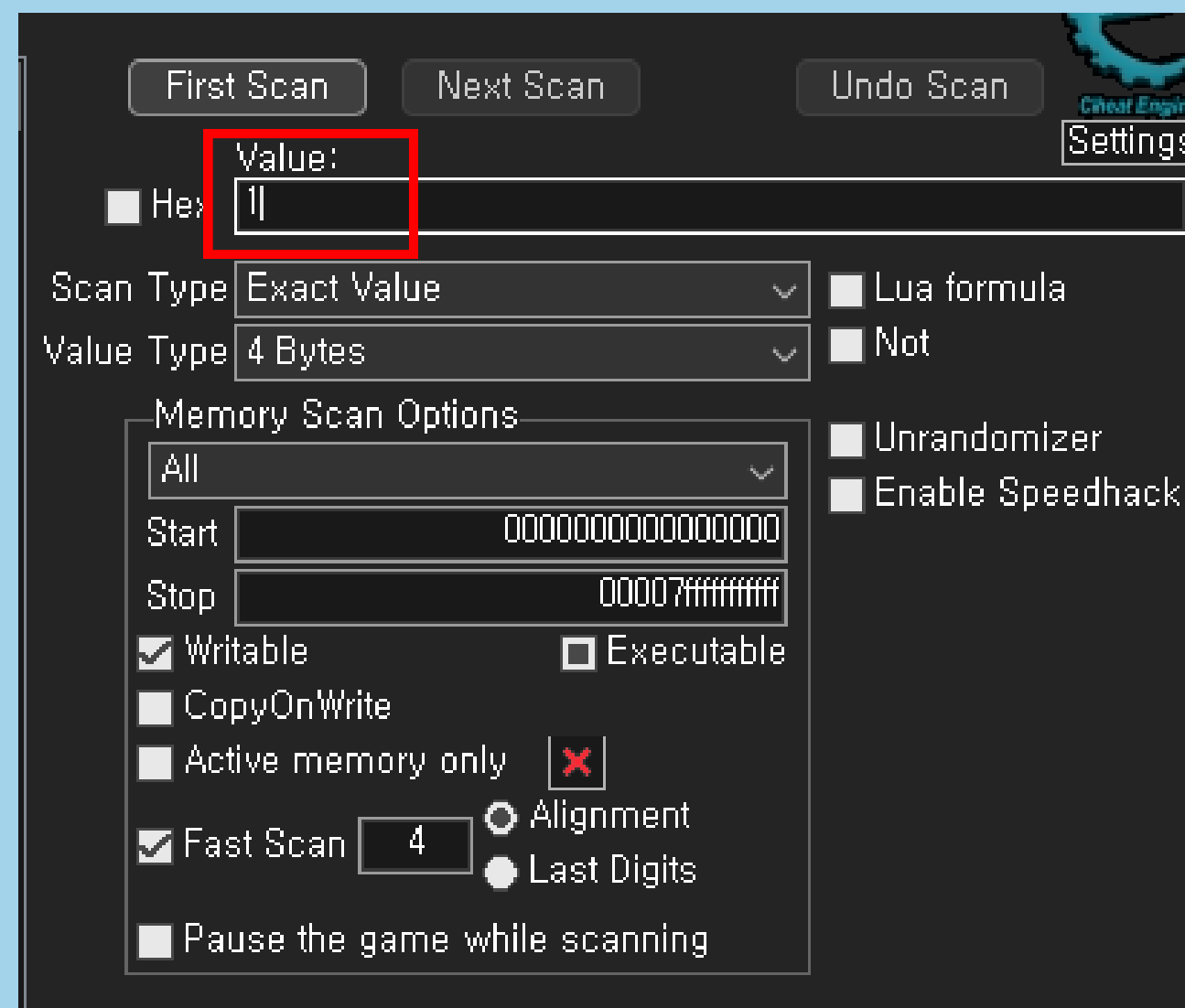
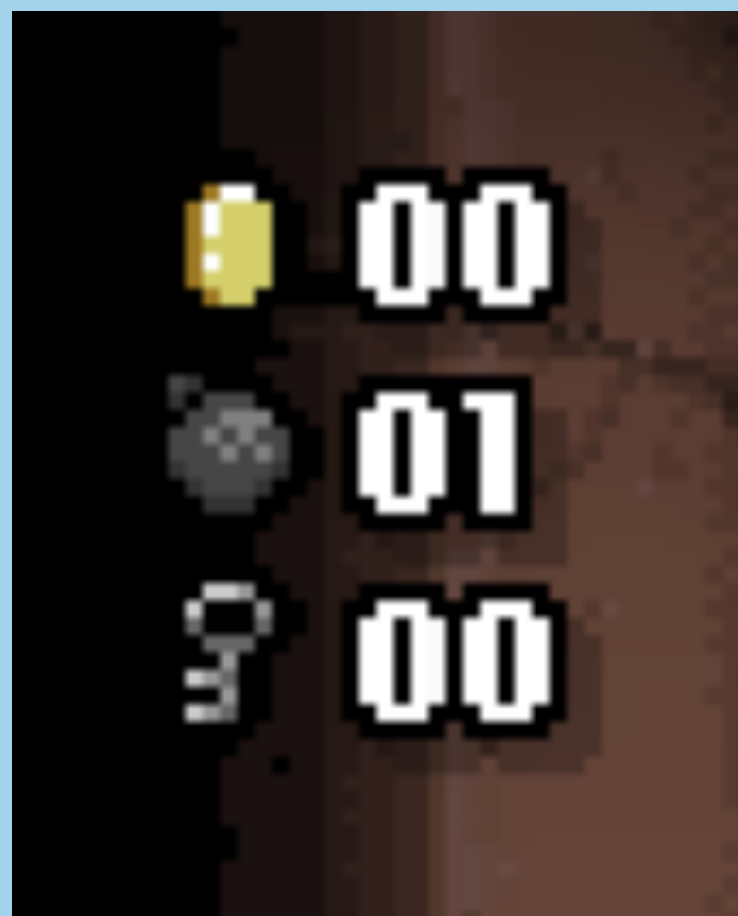
03 | 아이작 : 분석 준비



캐릭터를 조종할 수 있고, 상하좌우로 공격을 할 수 있고, 체력바, 아이템들의 개수 등이 보인다.



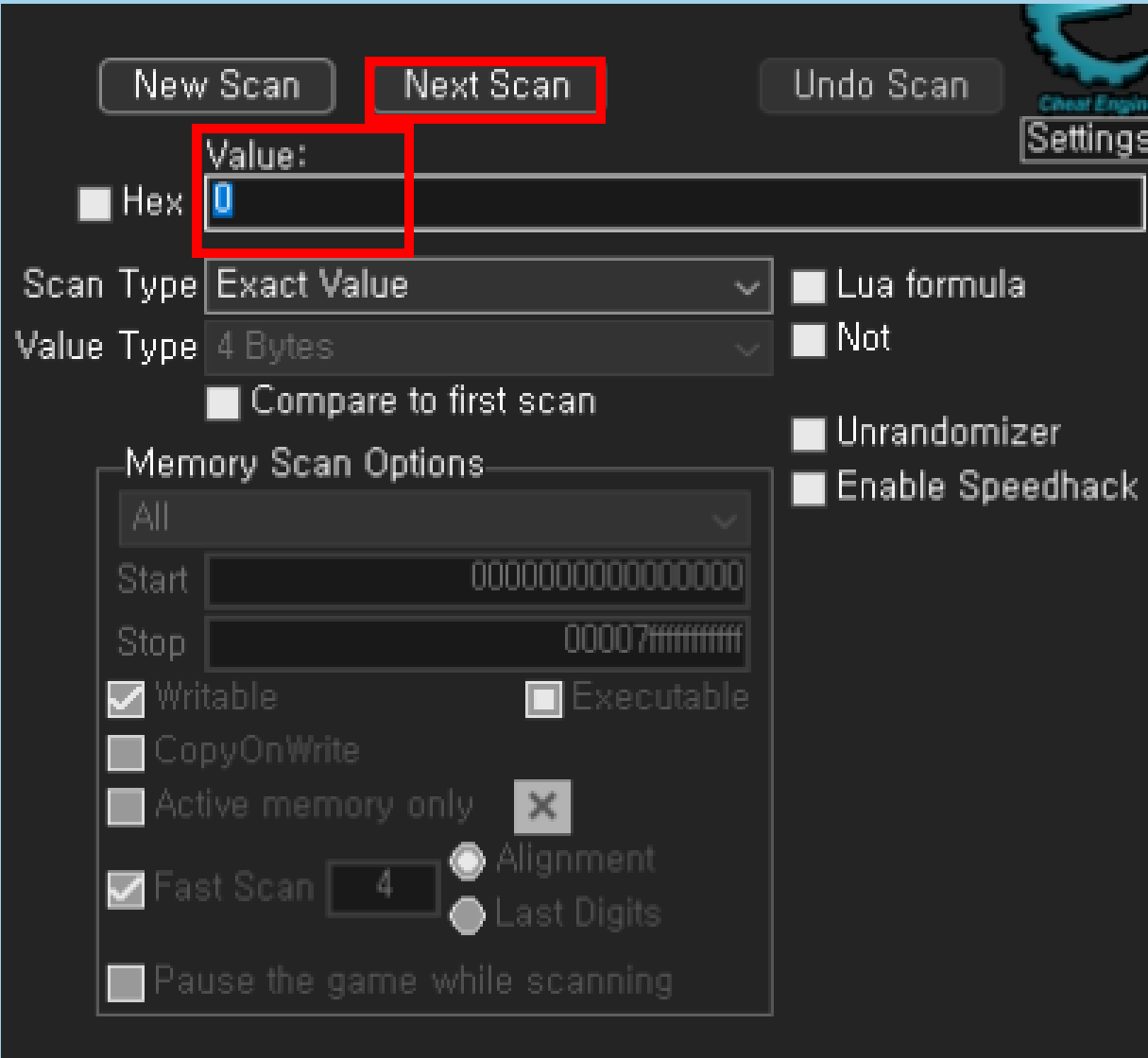
04 | 아이작 : 메모리 변조



폭탄의 개수를 스캔 : 수천-수만 개의 메모리 주소가 찾아진다.



04 | 아이작 : 메모리 변조



폭탄의 개수를 스캔 : 폭탄을 사용하여 0으로 만들고, 스캔한 값들 중 0으로 바뀐 값을 재스캔




04 | 아이작 : 메모리 변조

Found: 3

| Address | Value | Previous | First |
|----------|-------|----------|-------|
| 10A726CC | 0 | 0 | 0 |
| 10FD6050 | 0 | 0 | 0 |
| 154350A4 | 0 | 0 | 0 |

| Active | Description | Address | Type | Value |
|--------------------------|----------------|----------|---------|-------|
| <input type="checkbox"/> | No description | 154350A4 | 4 Bytes | 12 |



계속 줄이고 줄여서 찾은 3개의 값중 하나의 값을 변조하니 폭탄의 개수가 변하는 것을 확인



04 | 아이작 : 메모리 변조



이와 같은 방식으로 다른 메모리 값 또한 찾아내어 변조를 할 수 있다.



04 | 아이작 : 메모리 변조

| | | | |
|----------|----------|---------|----|
| COIN_MEM | 10EC516C | 4 Bytes | ?? |
| BOMB_MEM | 10EC5170 | 4 Bytes | ?? |
| KEY_MEM | 10EC5164 | 4 Bytes | ?? |

이후 게임을 재실행하면..



감사합니다.

????????????



04 | 아이작 : 메모리 변조

ASLR : Address Space Layout Randomization

메모리 손상 취약점 공격을 방지하기 위한 기술

라이브러리, 힙, 스택 영역 등의 주소를
바이너리가 실행될 때마다 랜덤하게 바꿔
정해진 주소를 이용한 공격을 막는 보호기법이다.



04 | 아이작 : 메모리 변조

01 | 실행할 때마다 메모리 찾기

너무나도 비효율적인 방식.

02 | 오프셋 찾기

실행되는 주소는 바뀌더라도 코드의 구조는 변하지 않음

03 | 코드 인젝션

해당 값을 이용하는 코드 부분에 인젝션을 하는 방식



05 | 아이작 : 코드 인젝션

The following opcodes write to 1110625C

| Co... | Instruction |
|-------|--|
| 1 | 010511C0 - FF 88 54110000 - dec [eax+00001154] |

isaac-ng_rebirth.exe+D11C0:
010511B6 - 89 98 640B0000 - mov [eax+00000B64],ebx
010511BC - 8B 44 24 0C - mov eax,[esp+0C]
010511C0 - FF 88 54110000 - dec [eax+00001154] <<
010511C6 - A1 A85A3101 - mov eax,[isaac-ng_rebirth.exe+395AA8]
010511CB - 83 B8 D8CD1000 02 - cmp dword ptr [eax+0010CDD8],02

EAX=11105108
EBX=00000000
ECX=D8581B28
EDX=00000000

Replace

Show disassembler

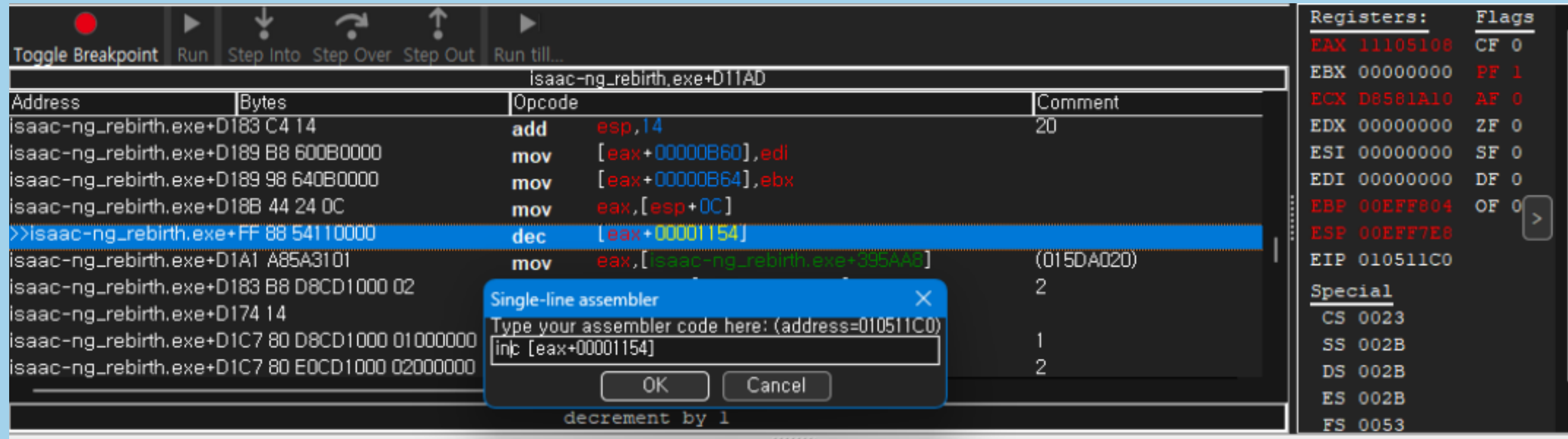
Add to the codelist

More information
decrement by 1

Stop

폭탄을 사용할 때 해당 메모리에 접근/수정 하는 코드를 찾아준다.

05 | 아이작 : 코드 인젝션



해당 코드를 아예 패치해서 사용하는 것도 좋지만..



05 | 아이작 : 코드 인젝션

```
[ENABLE]
//code from here to '[DISABLE]' will be used to enable the cheat
alloc(newmem2,2048)
label(returnhere2)
label(originalcode2)
label(exit2)

newmem2: //this is allocated memory, you have read,write,execute access
//place your code here
inc [eax+00001154]
originalcode2:
//dec [eax+00001154]

exit2:
jmp returnhere2

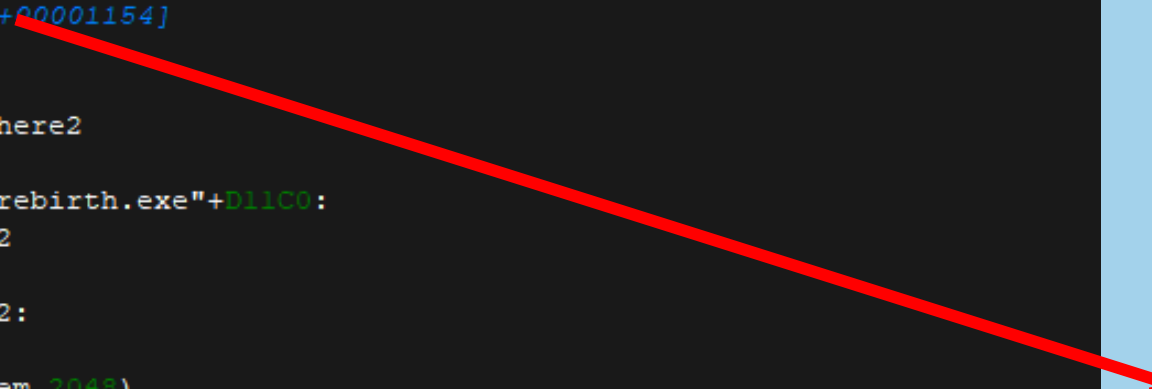
"isaac-ng_rebirth.exe"+D11C0:
jmp newmem2
nop
returnhere2:

alloc(newmem,2048)
label(returnhere)
label(originalcode)
label(exit)

newmem: //this is allocated memory, you have read,write,execute access
//place your code here
j1 isaac-ng_rebirth.exe+D11E8
originalcode:
//jng isaac-ng_rebirth.exe+D11E8

exit:
jmp returnhere

"isaac-ng_rebirth.exe"+D1117:
jmp newmem
nop
returnhere:
```



```
inc [eax+00001154]
originalcode2:
//dec [eax+00001154]
```

lua script 를 이용하여 코드 인젝션을 On/Off 할 수 있도록 구현



05 | 아이작 : 코드 인젝션

| 18 | 1C | 20 | 24 | 28 | 2C | 30 | 34 | 89ABCDEF0123456789ABCDEF01234567 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------------------------------|
| FFFFFFFF | 1166DE00 | 64616548 | 6E776F44 | 38320000 | 435F322E | 00000008 | 0000000F | . f.HeadDown..28.2_C..... |
| FFFFFFFF | 0000000E | 0000000E | 00000000 | 00000001 | 00000001 | 00000000 | 00000000 | |
| 00000000 | 00000019 | 00000000 | 00000000 | 00000000 | 00000000 | 00000001 | 00000000 | |
| 00000003 | 11660000 | 00000000 | 00000000 | 015DF9FC | 00000001 | 00000000 | 00000000 |f.....]..... |
| 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | |
| 00000000 | 00000000 | 00000000 | 00000000 | FFFFFFFF | 0000000A | 3F99999A | 00000000 | ?.... |
| 0000090A | 40600006 | C1BE0000 | 00000000 | 00000000 | 20202020 | 00000000 | 00000000 |`€..... |
| 3F800000 | 3F800000 | 3F800000 | 3F800000 | 00000000 | 00000000 | 00000000 | 00000000 | .. ?.. ?.. ?.. ?..... |
| 00000000 | 00000000 | 00000000 | 3F800000 | 3F800000 | 3F800000 | 3F800000 | 00000000 | ?.. ?.. ?.. ?.... |
| 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 3F800000 | 3F800000 | ?.. ? |
| 00000000 | 15EE2888 | 15EE2888 | 15EE2894 | 74696100 | 11105108 | 00000000 | 00000000 | (. (. (..ait.Q..... |
| 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | |
| 00000000 | 00000000 | 00000000 | 00000000 | 3F800000 | 00000000 | 00000001 | 00000000 | ?..... |
| 00000001 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | |
| 00000001 | 1166E301 | 00000000 | 00000003 | 00000003 | FFFFFFFF | 00000000 | 00000000 | f..... |
| 00000000 | BF800000 | 00000000 | 00000000 | 3F800000 | 00000000 | 00000000 | 00000000 | ?..... |
| 3FEB16B0 | BC6EFC43 | 4064A436 | 00000000 | 00000000 | 00000000 | 00000009 | 00000000 | . ?C n 6 d@..... |
| 0000001A | 00000001 | 428C1E60 | 438BCBDD | C1000000 | 00000000 | 14F31684 | 00000400 |`. B C... .. |

캐릭터의 데미지, 공격 속도, 이동속도 등 대부분의 스테이터스가 하나의 구조체로 묶여있다.



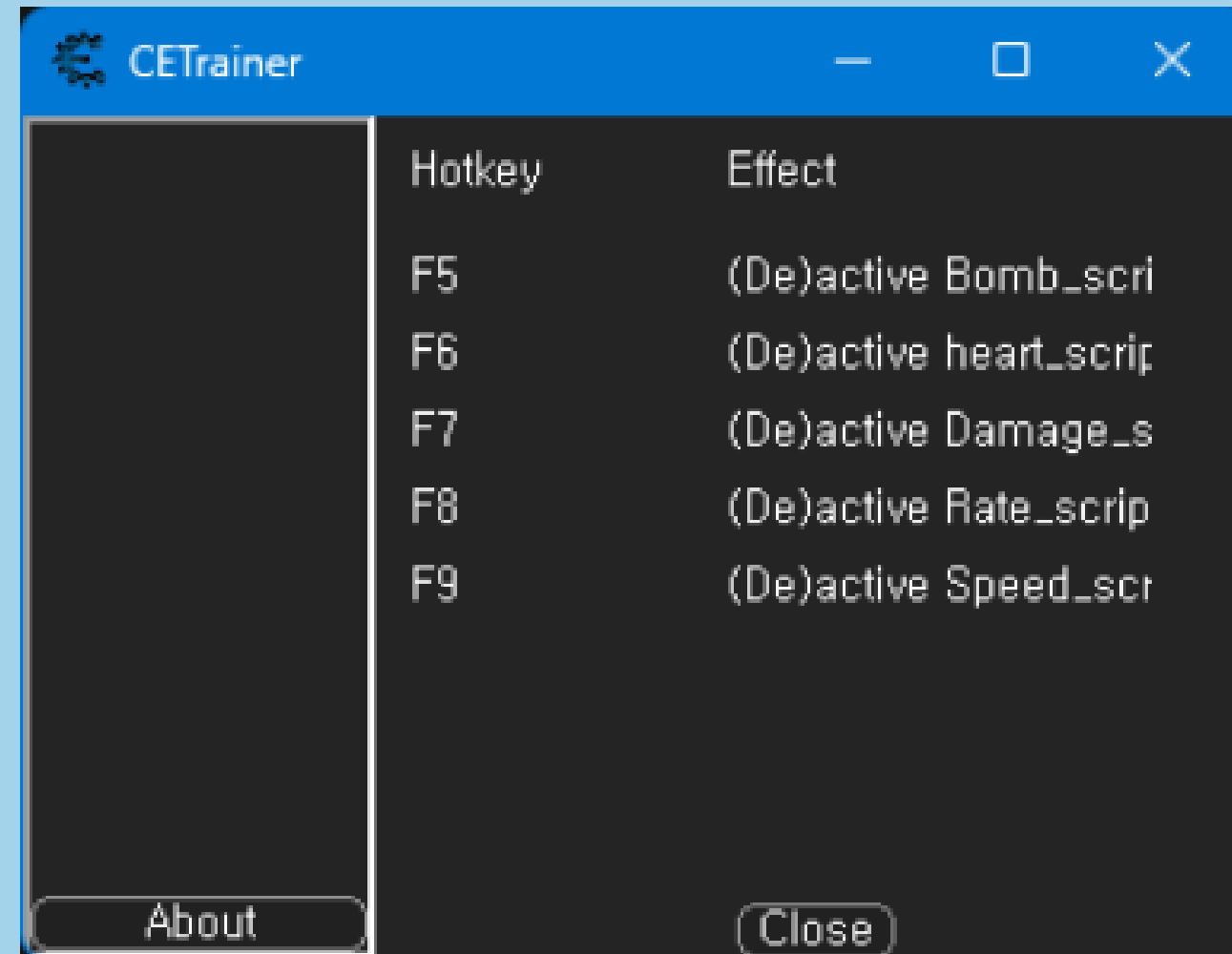
05 | 아이작 : 코드 인젝션

```
Damage : +0xbc  
shotspeed : +0xb0  
Rate : +0xac  
Speed : +0x188  
Luck : +0x18c  
Coin : +0x40  
Bomb : +0x30  
Key : +0x34  
Heart : +0x20  
Total Heart : +0x10
```

직접 추출해낸 값들의 오프셋들은 이렇하다.



06 | 아이작 : 치트 실행기



치트엔진에서 제공해주는 Trainer 생성기를 통해 치트 실행기 생성



06 | 아이작 : 치트 실행기



치트엔진에서 제공해주는 Trainer 생성기를 통해 치트 실행기 생성

