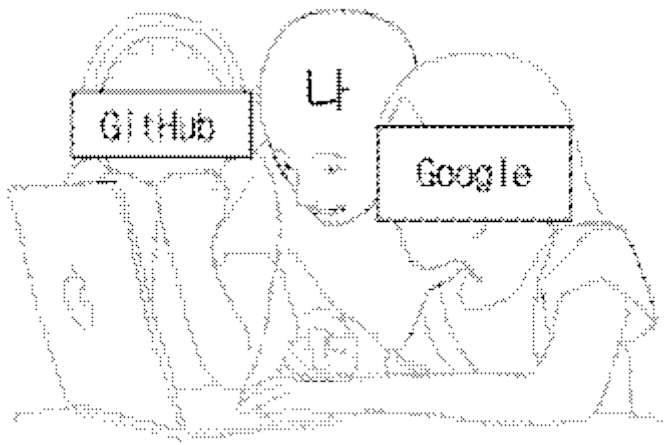
PCAP CODE, 관심분야

2021-07-27
차윤지

목차

pcap

코드 리뷰



관심 분야
찾아보기

리버싱

pcap 코드리뷰

```
1   #include <pcap.h>
2   #include <stdio.h>
3   //#include <stdbool.h>
4
5   #define true 1
6   #define false 0
7   #define NULL 0
8
9   #include<net/ethernet.h>
10  #include<netinet/ip.h>
11  #include<netinet/tcp.h>
12
13
14
```

<pcap.h> = 패킷 관련 핸들링을 위해 추가

<stdio.h> = printf를 사용하기 위해 추가

<stdbool.h> = 컴파일 오류로 주석처리 후
5~7줄에 별도 정의

```c
#include <pcap.h>
#include <stdio.h>
//#include <stdbool.h>

#define true 1
#define false 0
#define NULL 0

#include<net/ethernet.h>
#include<netinet/ip.h>
#include<netinet/tcp.h>


```

```c
    struct ether_header *eth_h;
    struct ip *ip_h;
    struct tcphdr *tcp_h;
```

```
15  void usage() {
16      printf("syntax: pcap-test <interface>\n");
17      printf("sample: pcap-test wlan0\n");
18  }
19
20  typedef struct {
21      char* dev_;
22  } Param;
23
24  Param param  = {
25      .dev_ = NULL
26  };
27
28  int parse(Param* param, int argc, char* argv[]) {
29      if (argc != 2) {
30          usage();
31          return false;
32      }
33      param->dev_ = argv[1];
34      return true;
35  }
36
```

= 사용법을 알려줍니다.

= 매개변수 구조체 정의입니다.

```
37  int main(int argc, char* argv[]) {
38      if (!parse(&param, argc, argv))
39          return -1;
40
41      char errbuf[PCAP_ERRBUF_SIZE];
42
43      pcap_t* pcap = pcap_open_live(param.dev_, BUFSIZ, 1, 1000, errbuf);
44
45      if (pcap == NULL) {
46          fprintf(stderr, "pcap_open_live(%s) return null - %s\n", param.dev_, errbuf);
47          return -1;
48      }
49
50      struct pcap_pkthdr* header;
51      const u_char* packet;
52
53      struct ether_header *eth_h;
54      struct ip *ip_h;
55      struct tcphdr *tcp_h;
56
```

= 강제종료

```
37  int main(int argc, char* argv[]) {
38      if (!parse(&param, argc, argv))
39          return -1;
40
41      char errbuf[PCAP_ERRBUF_SIZE];
42
43      pcap_t* pcap = pcap_open_live(param.dev_, BUFSIZ, 1, 1000, errbuf);
44
45      if (pcap == NULL) {
46          fprintf(stderr, "pcap_open_live(%s) return null - %s\n", param.dev_, errbuf);
47          return -1;
48      }
49
50      struct pcap_pkthdr* header;
51      const u_char* packet;
52
53      struct ether_header *eth_h;
54      struct ip *ip_h;
55      struct tcphdr *tcp_h;
56
```

= 에러 정보를 담을 버퍼를 준비

```
37  int main(int argc, char* argv[]) {
38      if (!parse(&param, argc, argv))
39          return -1;
40
41      char errbuf[PCAP_ERRBUF_SIZE];
42
43      pcap_t* pcap = pcap_open_live(param.dev_, BUFSIZ, 1, 1000, errbuf);
44
45      if (pcap == NULL) {
46          fprintf(stderr, "pcap_open_live(%s) return null - %s\n", param.dev_, errbuf);
47          return -1;
48      }
49
50      struct pcap_pkthdr* header;
51      const u_char* packet;
52
53      struct ether_header *eth_h;
54      struct ip *ip_h;
55      struct tcphdr *tcp_h;
56
```

= 강제종료2

```
37  int main(int argc, char* argv[]) {
38      if (!parse(&param, argc, argv))
39          return -1;
40
41      char errbuf[PCAP_ERRBUF_SIZE];
42
43      pcap_t* pcap = pcap_open_live(param.dev_, BUFSIZ, 1, 1000, errbuf);
44
45      if (pcap == NULL) {
46          fprintf(stderr, "pcap_open_live(%s) return null - %s\n", param.dev_, errbuf);
47          return -1;
48      }
49
50      struct pcap_pkthdr* header;
51      const u_char* packet;
52
53      struct ether_header *eth_h;
54      struct ip *ip_h;
55      struct tcphdr *tcp_h;
56
```

```
61  while (true) {
62
63      int res = pcap_next_ex(pcap, &header, &packet);
64      if (res == 0) continue;
65      if (res == PCAP_ERROR || res == PCAP_ERROR_BREAK) {
66          printf("pcap_next_ex return %d(%s)\n", res, pcap_geterr(pcap));
67          break;
68      }
69
70      eth_h = (struct ether_header *)packet;
71      ip_h = (struct ip *) (packet + sizeof(struct ether_header));
72      tcp_h = (struct tcphdr *) (packet + sizeof(struct  ether_header) + sizeof(struct ip));
73      const u_char *payload = (const u_char *)(packet + sizeof (struct ether_header) + sizeof (struct ip) + sizeof (struct tcphdr));
```

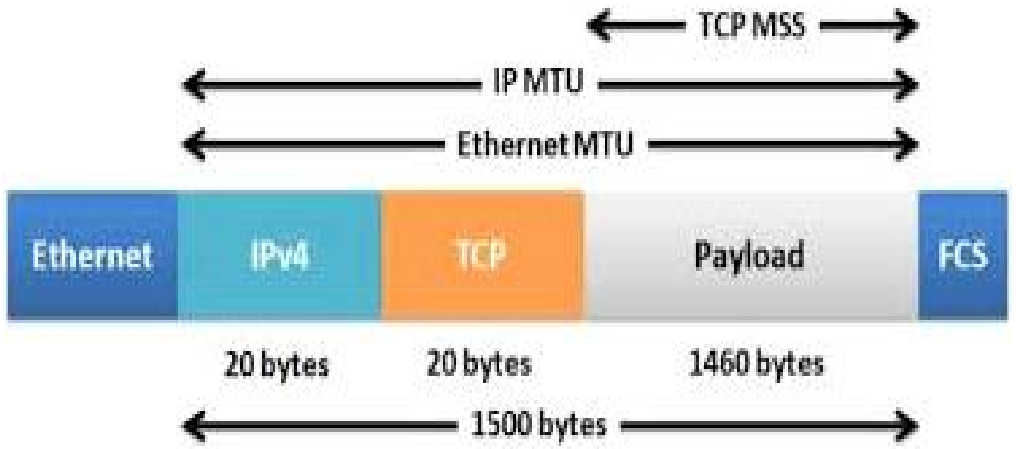
63번 줄= 패킷을 수집

64번 줄= 수집된 패킷이 없다면 while문의 처음으로 돌아간다

65번 줄= 만약 패킷 수집 중 오류 발생

66번 줄 ~ 67번 줄= 오류 정보 출력 후 프로그램 종료

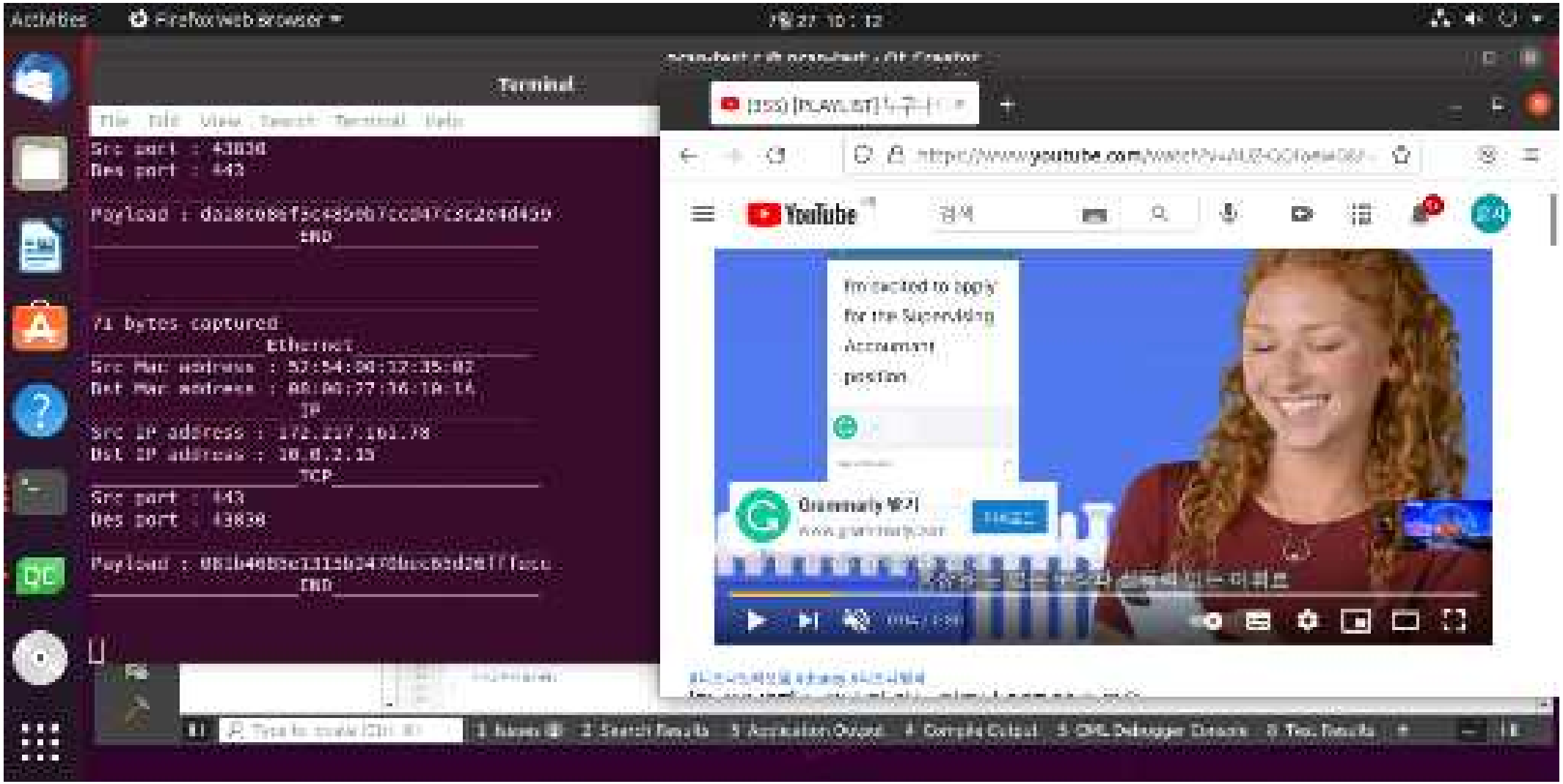
```
61    while (true) {
62
63        int res = pcap_next_ex(pcap, &header, &packet);
64        if (res == 0) continue;
65        if (res == PCAP_ERROR || res == PCAP_ERROR_BREAK) {
66            printf("pcap_next_ex return %d(%s)\n", res, pcap_geterr(pcap));
67            break;
68        }
69
70        eth_h = (struct ether_header *)packet;
71        ip_h = (struct ip *) (packet + sizeof(struct ether_header));
72        tcp_h = (struct tcphdr *) (packet + sizeof(struct  ether_header) + sizeof(struct ip));
73        const u_char *payload = (const u_char *)(packet + sizeof (struct ether_header) + sizeof (struct ip) + sizeof (struct tcphdr));
```



```
        printf("_____Ethernet_____\n");
        printf("Src Mac address : %02X:%02X:%02X:%02X:%02X:%02X\n", eth_h->ether_shost[0], eth_h->ether_shost[1], eth_h->ether_shost[2], eth_h->ether_shost[3], eth_h->ether_shost[4], eth_h->ether_shost[5]);
        printf("Dst Mac address : %02X:%02X:%02X:%02X:%02X:%02X\n", eth_h->ether_dhost[0], eth_h->ether_dhost[1], eth_h->ether_dhost[2], eth_h->ether_dhost[3], eth_h->ether_dhost[4], eth_h->ether_dhost[5]);
        printf("_____IP_____\n");
        printf("Src IP address : %s\n", inet_ntoa(ip_h->ip_src));
        printf("Dst IP address : %s\n", inet_ntoa(ip_h->ip_dst));
        printf("_____TCP_____\n");
        printf("Src port : %d\n", ntohs(tcp_h->th_sport));
        printf("Des port : %d\n", ntohs(tcp_h->th_dport));
        printf("_____\n");
        printf("Payload : ");
        for (int i=0; i<16; i++)
        {
            printf("%02x", payload[i]);
        }
        printf("\n");
        printf("_____\n");
        printf("Nu bytes captured:%d", header->caplen);
        printf("_____END_____\n");
        printf("\n\n");


    }


    pcap_close(pcap);
}
```


Activities
Firefox Web Browser
Terminal
File Edit View Search Terminal Help
Src port : 43838
Des port : 443
Payload : da28c0867fcc4b5b67ccb47c5c2e4d459
END
71 bytes captured
Ethernet
Src Mac address : 57:54:00:12:35:02
Dst Mac address : 08:00:27:16:19:14
IP
Src IP address : 172.217.161.78
Dst IP address : 10.0.2.15
TCP
Src port : 443
Des port : 43838
Payload : 0816d4b5c13135b2470bcc65d2617fccc
END
YouTube
I'm excited to apply for the Succeeding Accountant position
Grammarly

리버싱이란?

소프트웨어 분야에서 해당 프로그램의 구조, 기능, 동작 등의 원리를
역으로 따라가며 이해하고 분석

부족한 부분이나 추가 되었으면 하는 새로운 기능 등을 추가

리버싱 방법

정적 분석 = 파일의 겉모습을 관찰하여 분석하는 방법(파일 실행X)
파일의 종류, 크기, 헤더정보, 등록 정보 등의 내용을 확인할 수 있다.

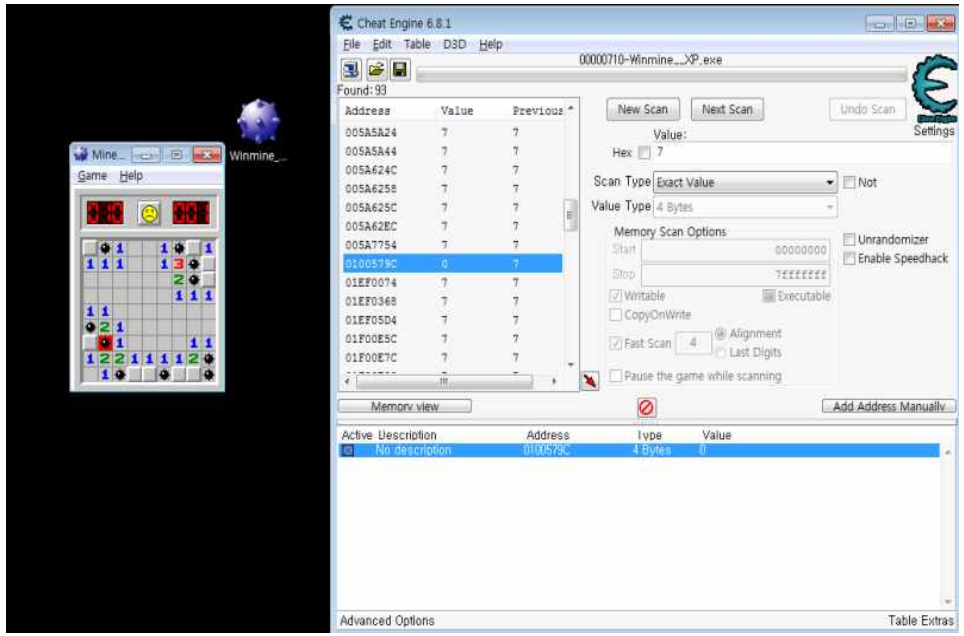
동적 분석 = 파일을 직접 실행시켜 그 행위를 분석하고, 디버깅을 통하여
코드의 흐름과 메모리의 상태 등을 자세히 살펴보는 방법

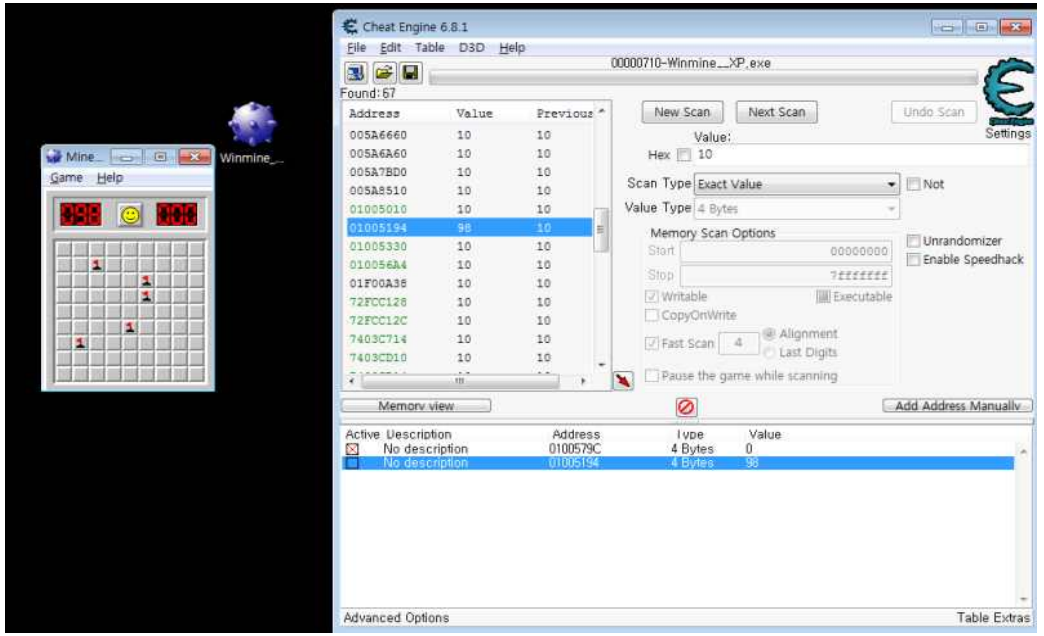
치트엔진

윈도우에서 실행되는 싱글 플레이어 게임 및 일반 응용 프로그램을 디버깅하는 데
도움이 되는 유용한 도구가 포함되어 있는 오픈 소스 도구

-> 치트 엔진을 사용하면 동적 분석을 진행하며
게임 내의 변수 값을 원하는 대로 조작할 수 있다.

사용 예시



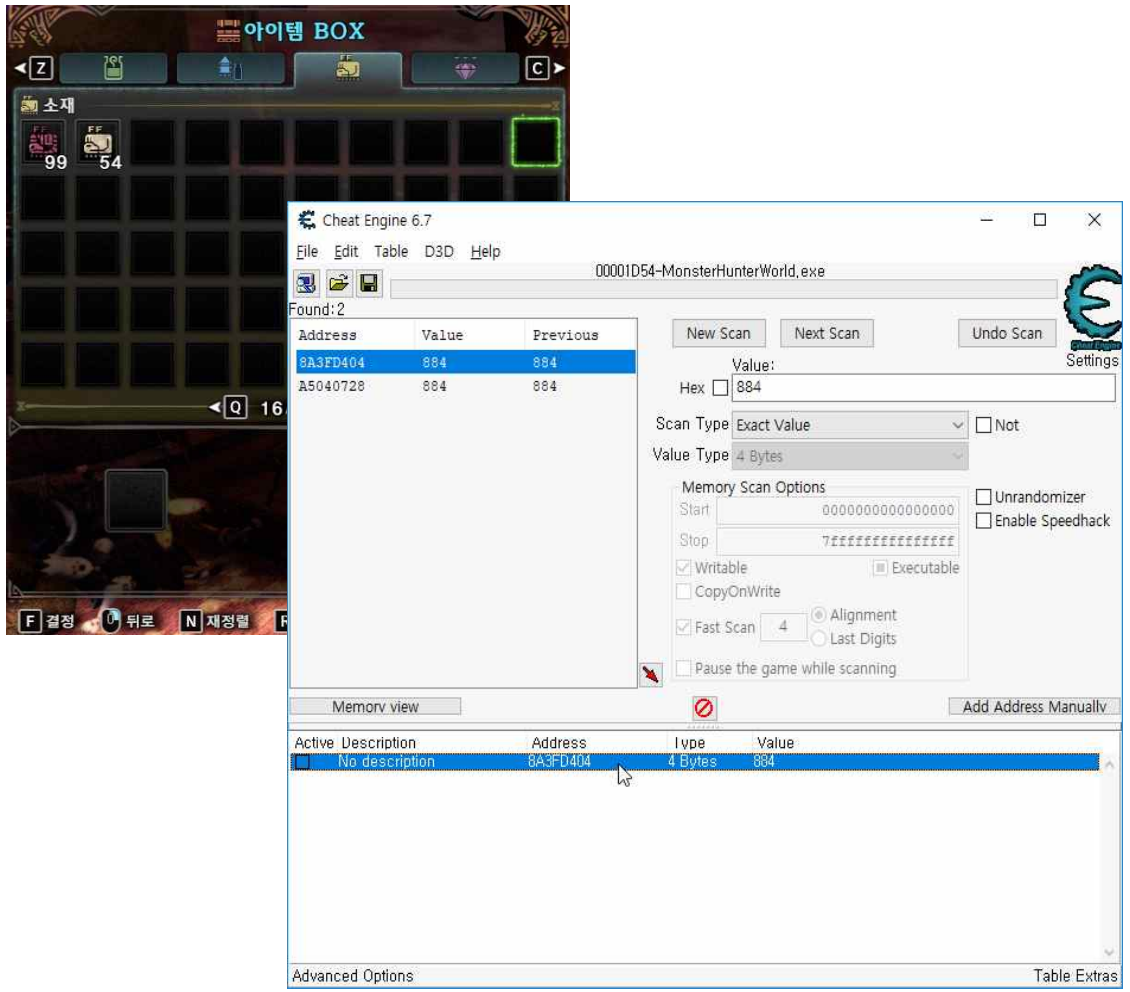


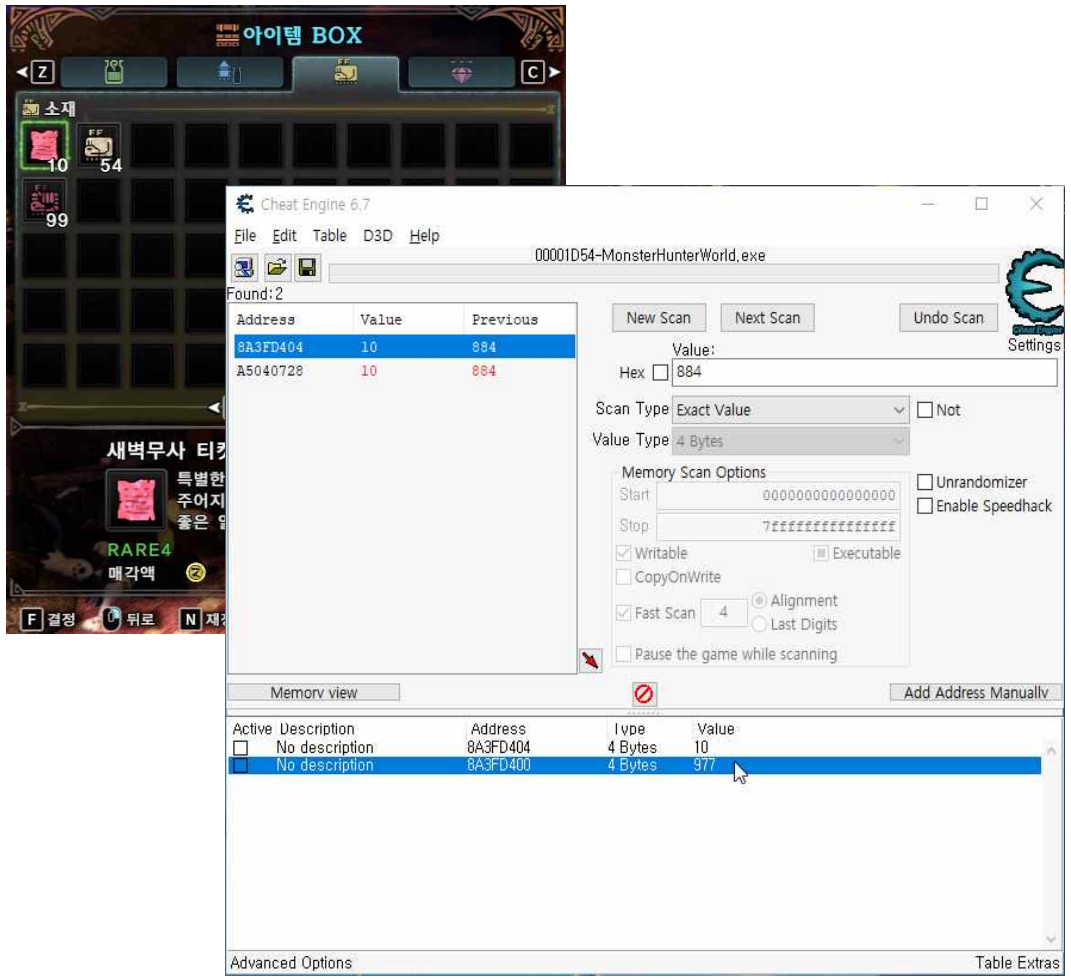
출처=
https://m.blog.naver.com/sosbaoxbak/221305871374

관심분야 찾아보기

리버싱

사용 예시


아이템 BOX
소재
99
54
Q
16
F 결정
뒤로
N 재정렬
Cheat Engine 6.7
File Edit Table D3D Help
00001D54-MonsterHunterWorld.exe
Found: 2
Address Value Previous
8A3FD404 884 884
A5040728 884 884
New Scan
Next Scan
Undo Scan
Settings
Value:
Hex
884
Scan Type Exact Value
Not
Value Type 4 Bytes
Memory Scan Options
Start 0000000000000000
Stop 7fffffffffffffff
Writable
Executable
CopyOnWrite
Fast Scan 4
Alignment
Last Digits
Pause the game while scanning
Unrandomizer
Enable Speedhack
Memory view
Add Address Manually
Active Description Address Type Value
No description 8A3FD404 4 Bytes 884
Advanced Options
Table Extras


아이템 BOX
소재
10
54
99
새벽무사 티켓
RARE4
매각액
F 결정
뒤로
Cheat Engine 6.7
File Edit Table D3D Help
00001D54-MonsterHunterWorld.exe
Found: 2
Address Value Previous
8A3FD404 10 884
A5040728 10 884
New Scan
Next Scan
Undo Scan
Settings
Value:
Hex
884
Scan Type Exact Value
Not
Value Type 4 Bytes
Memory Scan Options
Start 0000000000000000
Stop 7fffffffffffffff
Writable
Executable
CopyOnWrite
Fast Scan 4
Alignment
Last Digits
Pause the game while scanning
Unrandomizer
Enable Speedhack
Memory view
Add Address Manually
Active Description Address Type Value
No description 8A3FD404 4 Bytes 10
No description 8A3FD400 4 Bytes 977
Advanced Options
Table Extras

계획

Beacon Flooding

과제 진행

리버싱

올리 디버거 등을 이용해
코드 분석

Thank You