Dream Hack

: 해커들의 놀이터, 드림핵

CONTENTS

01

Dream Hack

Dream Hack 커리큘럼대로
공부하기로 했어요.

02

Pointer

배경지식이 부족하다고
생각되어 Pointer를 공부했어요.

03

Assembly

Assembly를 복습했어요.

Dream Hack

: 해커들의 놀이터, 드림핵

드림핵 이렇게 알았어요.

System Hacking

Reverse Engineering

Web Hacking

Cryptography

Mobile Hacking

Pointer

: C언어 최종보스, 포인터

solve_me.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
  int sz = 0x30;
  char *buf = (char *)malloc(sizeof(char) * sz);
  puts("Hello World!");
  printf("Education + Hack = ?\\n");
  fgets(buf, sz, stdin);
  if (!strncmp(buf, "DreamHack", 9))
    printf("Welcome Hackers :)\\n");
  else
    printf("No No :/n");
  return 0;
}
```

Pointer

컴퓨터 프로그래밍에서 변수의 메모리 주소를 가리키는 변수에요.

즉, 어떤 메모리 주소를 가리키는 값이라고 할 수 있어요.

int num = 1;

변수이름 : num
메모리 주소 : abcedfgh

int *ptr = #

변수이름 : *ptr

Why Pointer ?

1. 메모리 관리 : 포인터는 동적 메모리 할당에 사용돼요.

2. 함수 호출 : 인자를 복사하는 데 시간이 걸리므로 대용량 데이터 구조를 함수로 전달할 때는 인수로 구조체나 배열의 포인터를 전달

```c
#include <stdio.h>
#include <stdlib.h>

int main() {

    int num = 1;
    int* ptr = &num;

    printf("num 변수의 값 : %d₩n", num);                    // num 변수   : 1

    printf("*ptr 변수의 값 : %d₩n", *ptr);                  // *ptr 변수   : 1

    printf("&num 주소의 값 : %p₩n", &num);                  // &num 주소 : 00BEFAD0

    return 0;
}
```

malloc ()

동적으로 메모리를 할당하는 함수에요.

동적 메모리 할당이란, 프로그램 실행 중에 필요한 만큼 메모리를 할당하는 거에요.
이는 프로그램이 실행되는 동안 메모리를 효율적으로 사용하기 위해 사용돼요.

```
int* ptr = malloc(sizeof(int));
```

1. 포인터 ptr 변수에 4바이트로 메모리를 동적 할당 한다.
2. "4바이트를 동적 할당 한다." -> 4바이트를 단위로 동적 할당한다는 뜻인가?
   -> 예를 들어, 입력받은 인자가 15byte 라고 하자.
   -> 4바이트씩 4번 할당되는건가..?

3. int* ptr = malloc(sizeof(int)); -> int* ptr = malloc(4); : 이렇게 표현해도 되는 것인가?

4. int* ptr = (int*)malloc(sizeof(int)) : 에서 앞에 (int*)는 무엇인가?

5. 나는 앞으로 point 와 malloc를 이렇게 쓸 것 같다.

int* ptr = malloc(4) 이렇게 쓰고 우리말로 : 포인터 변수 ptr을 4byte로 동적할당 해라.

이렇게 이해했는데 맞는건가..?

malloc ()

```
void* malloc (size_t size);
```

why malloc () ?

1. 프로그램이 실행되는 동안 필요한 만큼의 메모리를 할당할 수 있어요.
2. 할당된 메모리는 프로그램 실행 중에 변경할 수 있어요.
   따라서 프로그램이 실행 중에 필요한 메모리를 동적으로 할당할 수 있어요.
3. 동적 메모리 할당은 메모리 사용을 최적화할 수 있어요.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main() {
    int sz = 0x30;                                  // sz 변수에 48 대입
    char* buf = (char*)malloc(sizeof(char) * sz);  // sz 변수에 있는 48을 포인터 buf에 동적 할당
    puts("Hello World!");                           // Hello Wokld! 출력
    printf("Education + Hack = ?₩n");               // Education + Hack = ? 출력
    fgets(buf, sz, stdin);                          // 1. 입력 받은 값을 48byte 동적 할당을 받은 buf 변수에 저장
                                                    // 2. sz 변수의 크기 즉, 48바이트 만큼 받을 수 있음
                                                    // 3. stdin 표준 출력 스트림
    if (!strncmp(buf, "DreamHack", 9))              // 만약, buf와 DreamHack와 같으면
        printf("Welcome Hackers :)₩₩n");            // 출력
    else
        printf("No No :/n");                        // 아니면 출력
    return 0;
```

```
int main() {
    int sz = 0x30;                                          // sz 변수에 48 대입
    char* buf = (char*)malloc(sizeof(char) * sz);  // sz 변수에 있는 48을 포인터 buf에 동적 할당
 }
```

Assembly

: 포너블의 필수, 어셈블리어

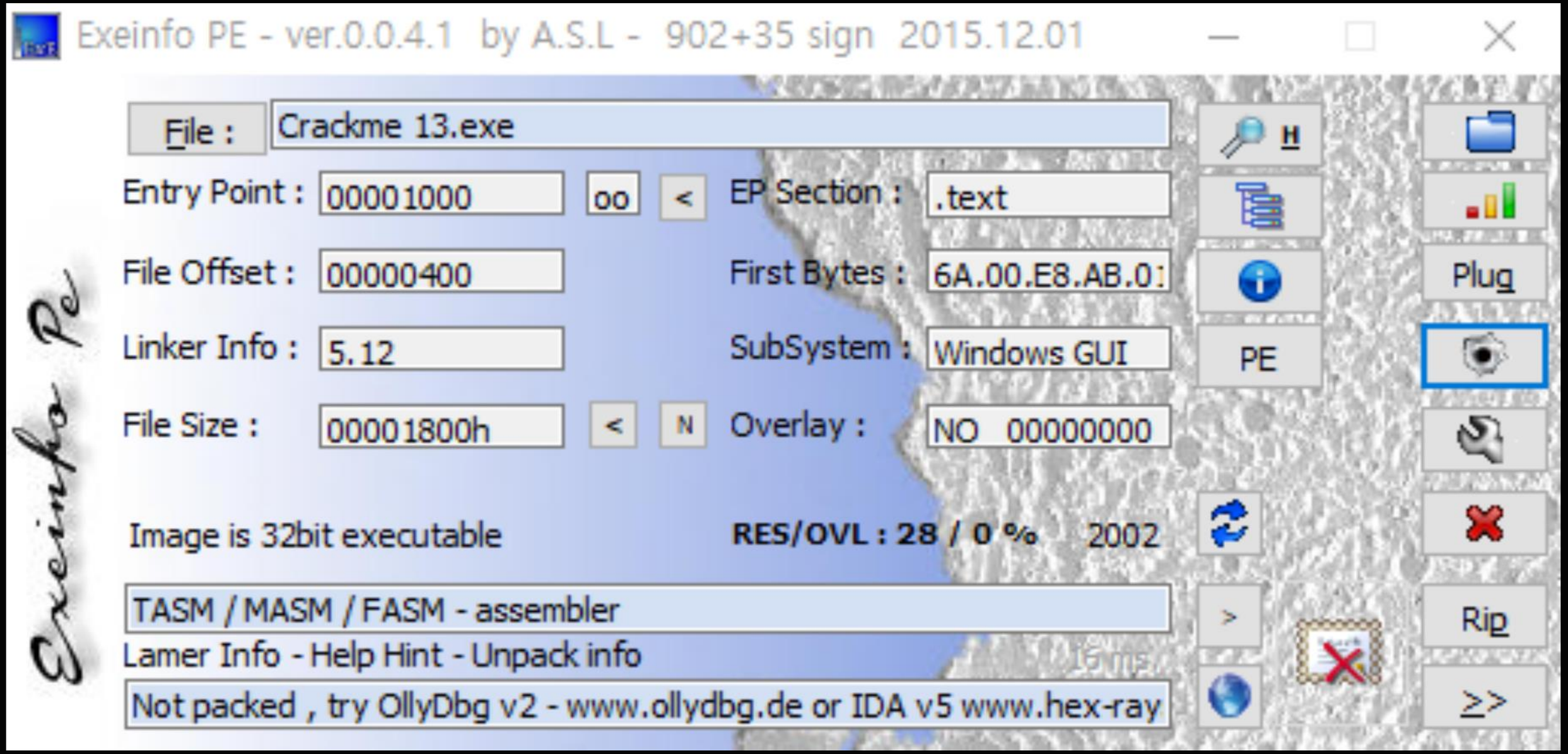
Assembly

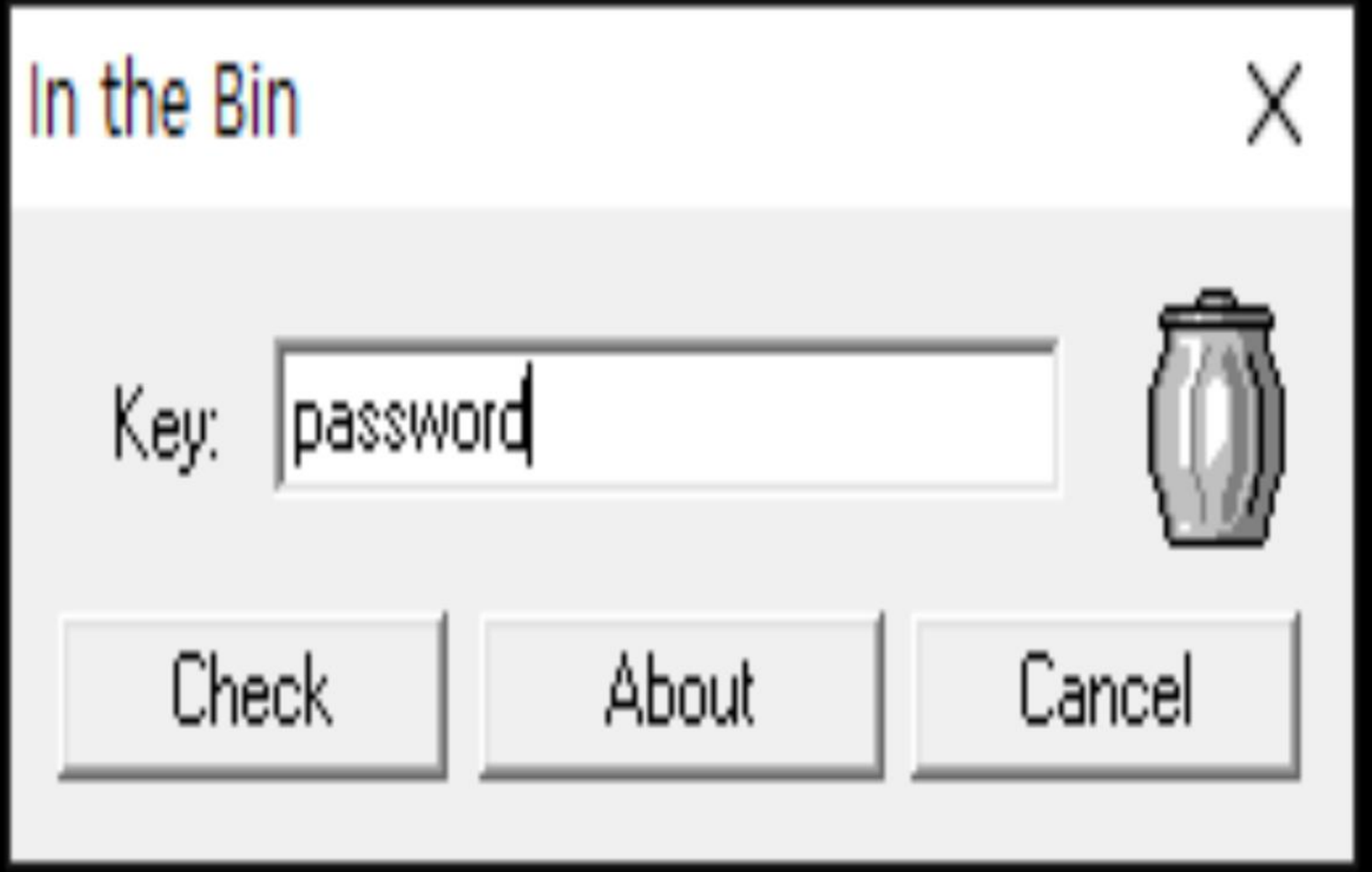
컴퓨터 구조와 연관된 저급 언어로, 기계어를 대신하여
사람이 읽고 쓰기 쉽도록 만들어진 언어에요.

opcode operand1, operand2

add apple banana

Crack Me 13


Exeinfo PE - ver.0.0.4.1 by A.S.L - 902+35 sign 2015.12.01
File :
Crackme 13.exe
Entry Point :
00001000
oo
<
EP Section :
.text
File Offset :
00000400
First Bytes :
6A.00.E8.AB.01
Linker Info :
5.12
SubSystem :
Windows GUI
File Size :
00001800h
<
N
Overlay :
NO 00000000
Image is 32bit executable
RES/OVL : 28 / 0 %
2002
TASM / MASM / FASM - assembler
Lamer Info - Help Hint - Unpack info
Not packed , try OllyDbg v2 - www.ollydbg.de or IDA v5 www.hex-ray
H
PE
Plug
Rip
>
>>


In the Bin
Key:
password
Check
About
Cancel

첫 번째 해결 방법

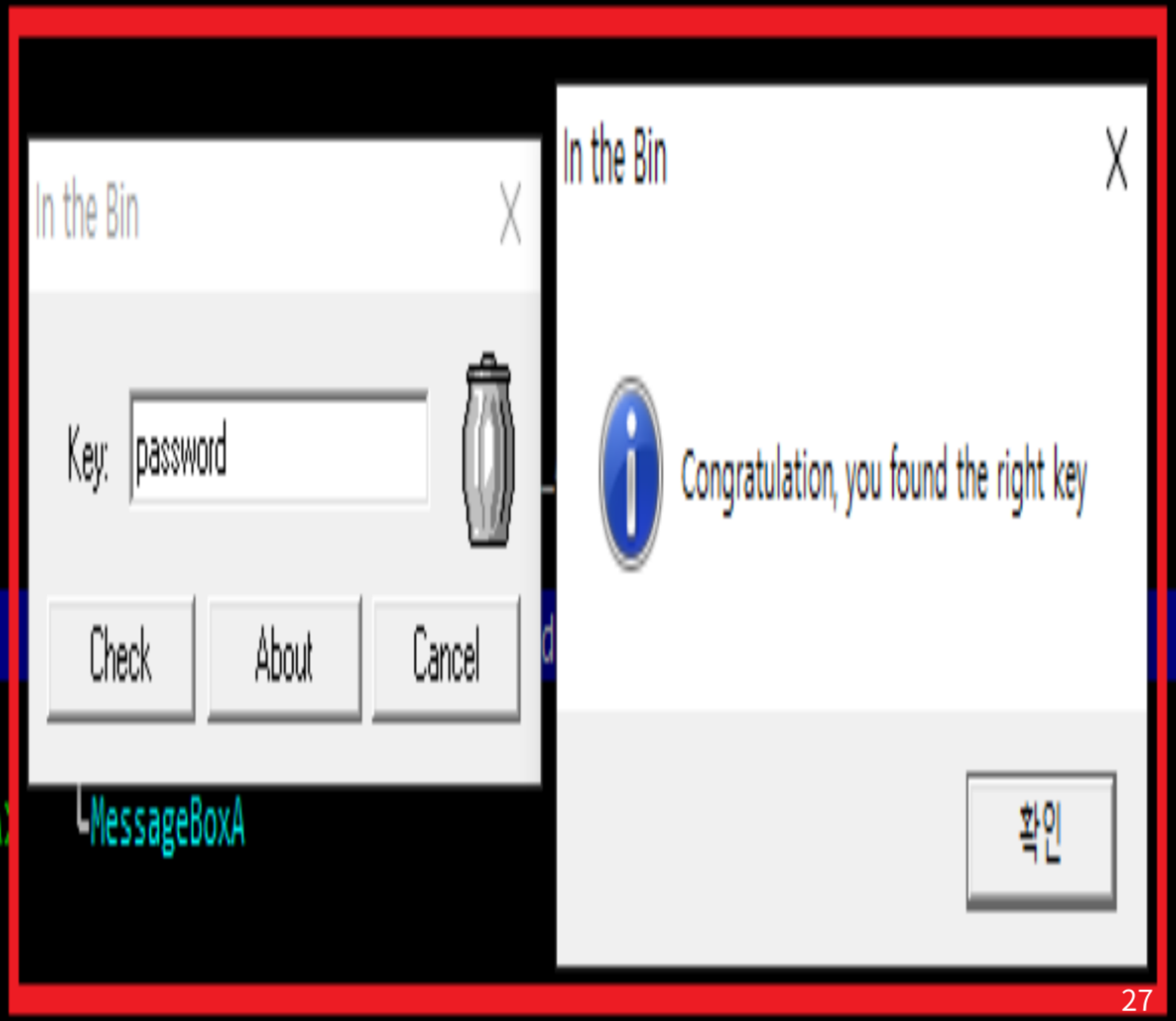
[코드 수정]

```
00401073   . E8 97000000 CALL Crackme_.0040110F
00401078   . 83C6 04     ADD ESI,4
0040107B   ^EB EB        JMP SHORT Crackme_.00401068
0040107D   > 3D BF5620FA CMP EAX,FA2056BF
00401082     75 14       JNZ SHORT Crackme_.00401098
00401084   . 6A 40       PUSH 40
00401086   . 68 30354000 PUSH Crackme_.00403530
0040108B   . 68 3B354000 PUSH Crackme_.0040353B
00401090   . FF75 08     PUSH DWORD PTR SS:[EBP+8]
00401093   . E8 02010000 CALL <JMP.&USER32.MessageBoxA>
00401098   > EB 6C       JMP SHORT Crackme_.00401106
```

```
Style = MB_OK|MB_ICONASTERISK|MB_APPLMODAL
Title = "In the Bin"
Text = "Congratulation, you found the right key"
hOwner
MessageBoxA
```

```
00401078   . 83C6 04     ADD ESI,4
0040107B   ^EB EB        JMP SHORT Crackme_.00401068
0040107D   > 3D BF96287A CMP EAX,7A2896BF
00401082     EB 00       JMP SHORT Crackme_.00401084
00401084   . 6A 40       PUSH 40                          Style = MB_OK|MB_ICONASTERISK|MB_APPLMODAL
00401086   . 68 30354000 PUSH Crackme_.00403530            Title = "In the Bin"
0040108B   . 68 3B354000 PUSH Crackme_.0040353B            Text = "Congratulation, you found the right key"
00401090   . FF75 08     PUSH DWORD PTR SS:[EBP+8]        hOwner
00401093   . E8 02010000 CALL <JMP.&USER32.MessageBoxA>    MessageBoxA
```

```
00401073   . E8 97000000 CALL Crackme_.0040110F
00401078   . 83C6 04     ADD ESI,4
0040107B   ^EB EB        JMP SHORT Crackme_.00401068
0040107D   > 3D BF96287A CMP EAX,7A2896BF
00401082     EB 00       JMP SHORT Crackme_.00401084
00401084   . 6A 40       PUSH 40
00401086   . 68 30354000 PUSH Crackme_.00403530
0040108B   . 68 3B354000 PUSH Crackme_.0040353B
00401090   . FF75 08     PUSH DWORD PTR SS:[EBP+8]
00401093   . E8 02010000 CALL <JMP.&USER32.MessageBoxA>
00401098   > EB 6C       JMP SHORT Crackme_.00401106
0040109A   . EB 61       JMP SHORT Crackme_.004010FD
```



두 번째 해결 방법

[KEY 값 찾기]


00401073 . E8 97000000 CALL Crackme_.0040110F
00401078 . 83C6 04 ADD ESI,4
0040107D > 3D BF96287A CMP EAX,7A2896BF
00401084 . 6A 40 PUSH 40
00401086 . 68 30354000 PUSH Crackme_.00403530
0040108B . 68 3B354000 PUSH Crackme_.0040353B
00401090 . FF75 08 PUSH DWORD PTR SS:[EBP+8]
00401093 . E8 02010000 CALL <JMP.&USER32.MessageBoxA>
Style = MB_OK|MB_ICONASTERISK|MB_APPLMODAL
Title = "In the Bin"
Text = "Congratulation, you found the right key"
hOwner
MessageBoxA


In the Bin
Key:
2049480383
Check
About
Cancel


In the Bin
Congratulation, you found the right key
확인

보안 쉽게 보지 말고
기본에 충실하자