Reversing Advanced

강사. 유효곤

(ugonfor@gmail.com)

Contents

문제해설

Garbage

Kaboom!

문제해설

Garbage

- File Size 부족 (끝이 망가짐)
- UPX 언패킹
- 고치고 나면 main함수를 확인 가능
- 오른쪽과 같이 MEMORY[*]로 나타나는 것들은 어셈블리를 확인해보면 call에 해당하는 것을 알 수 있음
- Module Name이 없어서 정보를 load 못하여 생긴 것

```
v14[1] = 874199833;
v14[2] = 1042484251;
v14[3] = 1108412467;
v14[4] = 1931350585;
sub_401000(v14, 20, v11, 0);
v3 = MEMORY[0x12418](v9[0], 0x40000000, 2, 0, 2, 128, 0);
sub_401045(v9);
if ( v3 != -1 )
{
  v8 = 0;
  sub_401000(v12, 61, v10, v4);
  MEMORY[0x123F8](v3, v9[0], 61, &v8, 0);
  sub_401045(v9);
  MEMORY[0x12426](v3);
  sub_401000(v14, 20, v11, v5);
  MEMORY[0x12442](0, 0, v9[0], 0, 0, 0);
  sub_401045(v9);
}
v6 = MEMORY[0x123E4](-1);
MEMORY[0x12404](v6);
return 0;
}
```

garbage.exe.bin

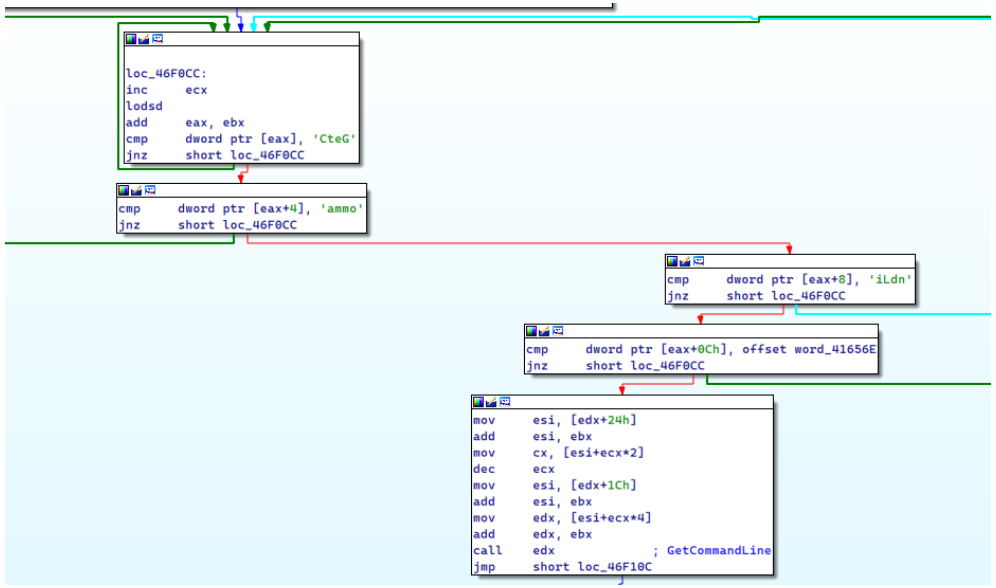
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00011634	N/A	00011494	00011498	0001149C	000114A0	000114A4
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
	66	00000000	00000000	00000000	00012434	0000D000
	1	00000000	00000000	00000000	00012452	0000D10C

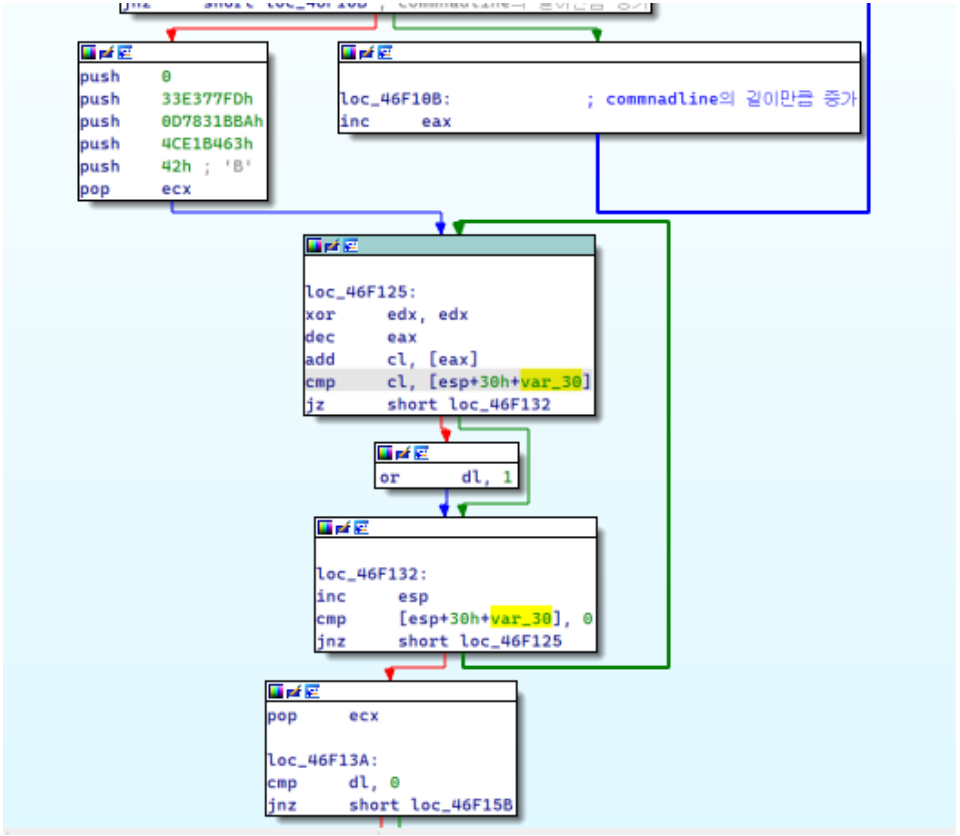
OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
N/A	000123E4	0000	GetCurrentProcess
N/A	000123F8	0000	WriteFile
N/A	00012404	0000	TerminateProcess
N/A	00012418	0000	CreateFileA
N/A	00012426	0000	CloseHandle
N/A	000128B0	0000	WriteConsoleW
N/A	000128A2	0000	CreateFileW

문제해설

Kaboom!

- UPX를 풀고나서 한참동안 분석해도 Flag를 못 찾는 것을 확인했어야 함.
- 이후 UPX를 풀기전에 분석을 해보면,

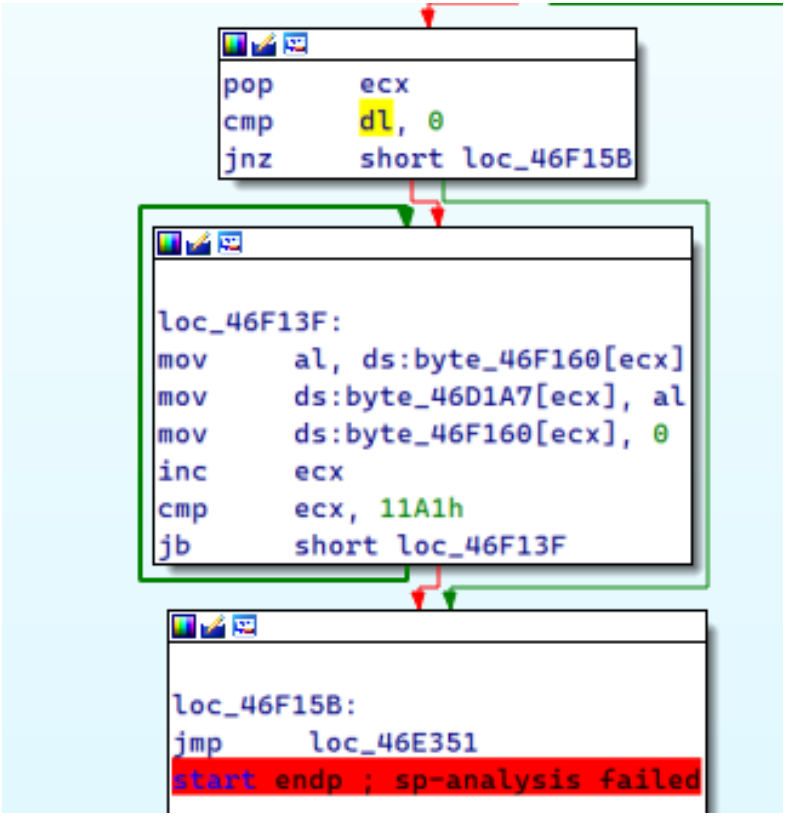




문제해설

Kaboom!

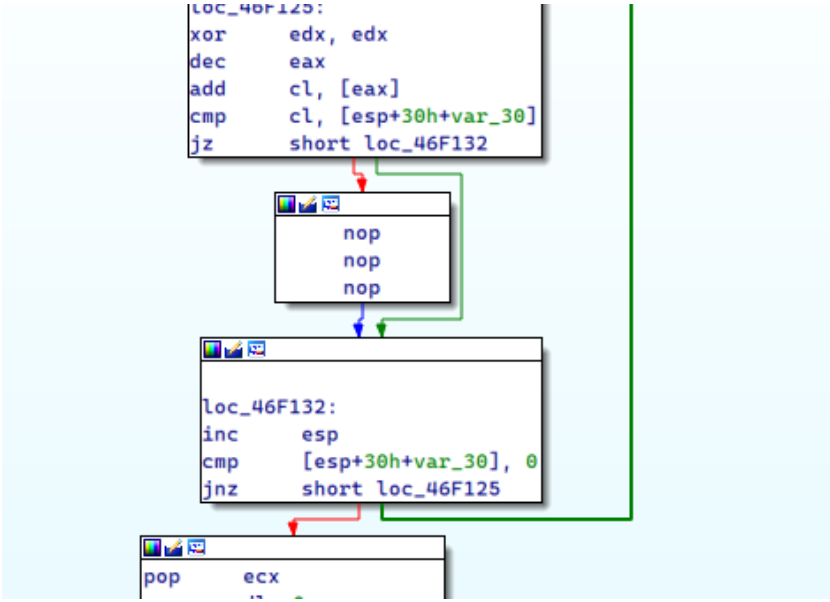
- UPX를 풀고나서 한참동안 분석해도 Flag를 못 찾는 것을 확인했어야 함.
- 이후 UPX를 풀기전에 분석을 해보면,
- or dl, 1에서 dl의 값을 cmp와 비교해서 1로 바꿔주고 있으니, dl의 값을 0으로 하고나서 진행하면, dl에 따라서 code flow가 바뀐다.



문제해설

Kaboom!

- UPX를 풀고나서 한참동안 분석해도 Flag를 못 찾는 것을 확인했어야 함.
- 이후 UPX를 풀기전에 분석을 해보면,
- or dl, 1에서 dl의 값을 cmp와 비교해서 1로 바꿔주고 있으니, dl의 값을 0으로 하고나서 진행하면, dl에 따라서 code flow가 바뀐다.



문제해설

Kaboom!

- UPX
  확인
- 이후
- or dl
  으니,
  code

IDA View-EIP | Pseudocode-A

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    int v3; // eax
4    int v4; // eax
5
6    if ( argc < 2
7      || (sub_402DF1((char *)argv[1], "defuse"), v3)
8      || (sub_401523("INS{"), sub_4011AE("INS{", "INS{GG EZ clap PogU 5Head B) Kreygasm <3<3}", v4)) )
9    {
10     sub_401393("%s", 68);
11     return 1;
12   }
13   else
14   {
15     sub_401393("Congrats! The flag is %s\n", 84);
16     return 0;
17   }
18 }
```

UNKNOWN | main:18 (406B91) | (Synchronized with EIP, IDA View-EIP)

문제해설

Kaboom!

- UPX를 풀고나서 한참동안 분석해도 Flag를 못 찾는 것을 확인했어야 함.
- 이후 UPX를 풀기전에 분석을 해보면,
- or dl, 1에서 dl의 값을 cmp와 비교해서 1로 바꿔주고 있으니, dl의 값을 0으로 하고나서 진행하면, dl에 따라서 code flow가 바뀐다.
- 만약 그럼에도 패치를 통해 해결 했을 때, 잘 안되었다면?
  - => 입력값을 가지고 xor하는 등의 로직이 있을거라 예상되어 공격자가 요구하는 입력값을 찾아 주어야 함.

문제해설

Kaboom!

- 흥미로운 점!
- 이 문제를 어떻게 만들었을 까요??
- https://github.com/Insomnihack/Teaser-2020/tree/master/kaboom

문제해설

Kaboom!

- 왼쪽 소스코드에서 FLAG부분
  - 실제 FLAG랑 youtube랑 두가지로 컴파일
  - 이후, UPX packing을 통해 각각의 파일 패킹 데이터 가져옴

Source code

The original source code is very short and simple:

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

#define DEFUSE "defuse"
#define FLAG   "https://www.youtube.com/watch?v=oGJr5N2lgsQ"
#define CANARY "INS{"
#define KABOOM "KABOOM!\n"

int main(int argc, char *argv[])
{

    if (   (argc < 2)
        || (strcmp(argv[1], DEFUSE) != 0)
        || (memcmp(CANARY, FLAG, strlen(CANARY)) != 0)
       )
    {
        printf("%s", KABOOM);
        return EXIT_FAILURE;
    }

    printf("Congrats! The flag is %s\n", FLAG);
    return EXIT_SUCCESS;
}
```

It just checks that the first command line argument is "defuse" and that the flag starts with the correct format `INS{`. If one of these conditions is false, it prints `KABOOM!`, otherwise it prints the flag. In this case, the second condition is always false so it always explodes.

문제해설

Kaboom!

- UPX2 Section의 크기를 키워서 코드를 삽입함.
  - Section의 사이즈를 키워줘야 함 + code를 삽입해서 실행할 것이기에 exec 권한 추가

원본

kaboom_original.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000240	00000248	0000024C	00000250	00000254	00000258	0000025C	00000260	00000262	00000264
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	0004F000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	0001E000	00050000	0001DC00	00000400	00000000	00000000	0000	0000	E0000040
UPX2	00001000	0006E000	00000200	0001E000	00000000	00000000	0000	0000	C0000040

수정본

kaboom_original.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000240	00000248	0000024C	00000250	00000254	00000258	0000025C	00000260	00000262	00000264
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	0004F000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	0001F000	00050000	0001E600	00000400	00000000	00000000	0000	0000	E0000040
UPX2	00002000	0006F000	00001300	0001EA00	00000000	00000000	0000	0000	E0000040

문제해설

Kaboom!

- Entrypoint를 바꿔줌
- CFF Explorer에서는 upx0 section의 RAW Size 정보가 없기에 upx0이라 나오지만, 실제로 section을 확인해보면 UPX2에 해당함. (보통 upx패킹은 upx0에서 시작)

AddressOfEntryPoint	00000120	Dword	0006D930	UPX0

kaboom_original.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000240	00000248	0000024C	00000250	00000254	00000258	0000025C	00000260	00000262	00000264
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	0004F000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	0001F000	00050000	0001E600	00000400	00000000	00000000	0000	0000	E0000040
UPX2	00002000	0006F000	00001300	0001EA00	00000000	00000000	0000	0000	E0000040

문제해설

Kaboom!

- Entrypoint를 바꿔줌
- CFF Explorer에서는 upx0 section의 RAW Size 정보가 없기에 upx0이라 나오지만, 실제로 section을 확인해보면 UPX2에 해당함. (보통 upx패킹은 upx0에서 시작)

AddressOfEntryPoint	00000120	Dword	0006D930	UPX0

kaboom_original.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
00000240	00000248	0000024C	00000250	00000254	00000258	0000025C	00000260	00000262	00000264
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
UPX0	0004F000	00001000	00000000	00000400	00000000	00000000	0000	0000	E0000080
UPX1	0001F000	00050000	0001E600	00000400	00000000	00000000	0000	0000	E0000040
UPX2	00002000	0006F000	00001300	0001EA00	00000000	00000000	0000	0000	E0000040

문제해설

Kaboom!

- UPX2 Section에 추가적인 루틴 삽입 (입력값 check하는 루틴)
- https://github.com/Insomnihack/Teaser-2020/blob/master/kaboom/kaboom_patch.py
- 만약에, 틀리면 decode: 를 지나침. 맞으면 decode 실행
- 이후, jumpToOEP (어셈블리로 쉘코드 만들고, 삽입해주었음)

```
81
82  pop ecx
83  cmp dl,0
84  jne jumpToOEP        # If the test fails, do nothing and return to OEP, get JEBAITED
85
86  decode:               # If you did not get JEBAITED, decode the original binary
87  mov al, byte ptr [originalBinary + ecx]
88  mov byte ptr [{} + ecx], al
89  mov byte ptr [originalBinary + ecx], 0x0
90  inc ecx
91  cmp ecx, {}
92  jb decode
93
94  jumpToOEP:
95  jmp {}
96
```

문제해설

Kaboom!

- 보통 UPX 자체 프로그램이 패킹, 언패킹을 할 때는 실행해서 하는 것이 아니라 정적으로 하기 때문에, 원래 있던 데이터보다 뒤쪽에 있는 데이터는 무시됨. (문제에서 upx unpack을 하면 youtube link로 나타났던 이유)
- 이런 식으로 비슷하게 Pyinstaller로 설치된 파일에 백도어 심는 것도 가능하겠죠?

Dll Injection

DLL이 뭔가요?

프로세스는 어떻게 DLL을 load할까요?

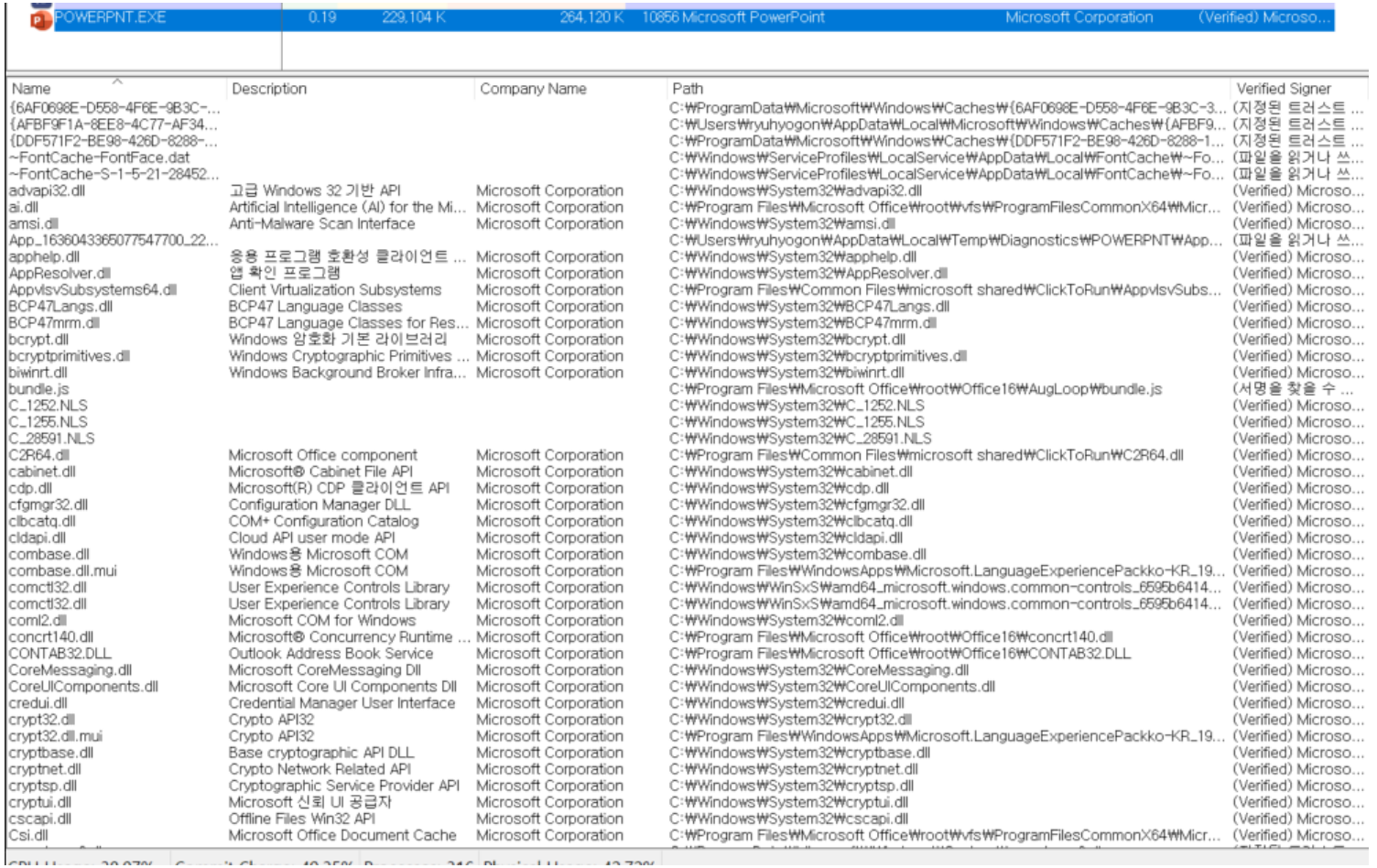
DLL injection이 뭔가요?

DLL injection이 어떤 과정을 통해 이루어지나요?

Dll Injection

DLL이 뭔가요?

- 동적 라이브러리 파일
- 리눅스에서 .so파일에 대응 되는 것으로..
- 프로세스에 포함되지 않은 함수나 코드, 즉 외부에서 선언한 함수를 가져다 쓰기 위해서 사용하는 파일
- Export Function을 제공하는 파일
- 여러 프로세스가 share하는 파일



POWERPNT.EXE 0.19 229,104 K 264,120 K 10856 Microsoft PowerPoint Microsoft Corporation (Verified) Microso...

Name	Description	Company Name	Path	Verified Signer
{6AF0698E-D558-4F6E-9B3C-...			C:₩ProgramData₩Microsoft₩Windows₩Caches₩{6AF0698E-D558-4F6E-9B3C-3...	(지정된 트러스트 ...
{AFBF9F1A-8EE8-4C77-AF34...			C:₩Users₩ryuhyogon₩AppData₩Local₩Microsoft₩Windows₩Caches₩{AFBF9...	(지정된 트러스트 ...
{DDF571F2-BE98-426D-8288-...			C:₩ProgramData₩Microsoft₩Windows₩Caches₩{DDF571F2-BE98-426D-8288-1...	(지정된 트러스트 ...
~FontCache-FontFace.dat			C:₩Windows₩ServiceProfiles₩LocalService₩AppData₩Local₩FontCache₩~Fo...	(파일을 읽거나 쓰...
~FontCache-S-1-5-21-28452...			C:₩Windows₩ServiceProfiles₩LocalService₩AppData₩Local₩FontCache₩~Fo...	(파일을 읽거나 쓰...
advapi32.dll	고급 Windows 32 기반 API	Microsoft Corporation	C:₩Windows₩System32₩advapi32.dll	(Verified) Microso...
ai.dll	Artificial Intelligence (AI) for the Mi...	Microsoft Corporation	C:₩Program Files₩Microsoft Office₩root₩vfs₩ProgramFilesCommonX64₩Micr...	(Verified) Microso...
amsi.dll	Anti-Malware Scan Interface	Microsoft Corporation	C:₩Windows₩System32₩amsi.dll	(Verified) Microso...
App_16360433650775477000_22...			C:₩Users₩ryuhyogon₩AppData₩Local₩Temp₩Diagnostics₩POWERPNT₩App...	(파일을 읽거나 쓰...
apphelp.dll	응용 프로그램 호환성 클라이언트 ...	Microsoft Corporation	C:₩Windows₩System32₩apphelp.dll	(Verified) Microso...
AppResolver.dll	앱 확인 프로그램	Microsoft Corporation	C:₩Windows₩System32₩AppResolver.dll	(Verified) Microso...
AppvIsvSubsystems64.dll	Client Virtualization Subsystems	Microsoft Corporation	C:₩Program Files₩Common Files₩microsoft shared₩ClickToRun₩AppvIsvSubs...	(Verified) Microso...
BCP47Langs.dll	BCP47 Language Classes	Microsoft Corporation	C:₩Windows₩System32₩BCP47Langs.dll	(Verified) Microso...
BCP47mrm.dll	BCP47 Language Classes for Res...	Microsoft Corporation	C:₩Windows₩System32₩BCP47mrm.dll	(Verified) Microso...
bcrypt.dll	Windows 암호화 기본 라이브러리	Microsoft Corporation	C:₩Windows₩System32₩bcrypt.dll	(Verified) Microso...
bcryptprimitives.dll	Windows Cryptographic Primitives ...	Microsoft Corporation	C:₩Windows₩System32₩bcryptprimitives.dll	(Verified) Microso...
biwinrt.dll	Windows Background Broker Infra...	Microsoft Corporation	C:₩Windows₩System32₩biwinrt.dll	(Verified) Microso...
bundle.js			C:₩Program Files₩Microsoft Office₩root₩Office16₩AugLoop₩bundle.js	(서명을 찾을 수 ...
C_1252.NLS			C:₩Windows₩System32₩C_1252.NLS	(Verified) Microso...
C_1255.NLS			C:₩Windows₩System32₩C_1255.NLS	(Verified) Microso...
C_28591.NLS			C:₩Windows₩System32₩C_28591.NLS	(Verified) Microso...
C2R64.dll	Microsoft Office component	Microsoft Corporation	C:₩Program Files₩Common Files₩microsoft shared₩ClickToRun₩C2R64.dll	(Verified) Microso...
cabinet.dll	Microsoft® Cabinet File API	Microsoft Corporation	C:₩Windows₩System32₩cabinet.dll	(Verified) Microso...
cdp.dll	Microsoft(R) CDP 클라이언트 API	Microsoft Corporation	C:₩Windows₩System32₩cdp.dll	(Verified) Microso...
cfgmgr32.dll	Configuration Manager DLL	Microsoft Corporation	C:₩Windows₩System32₩cfgmgr32.dll	(Verified) Microso...
clbcatq.dll	COM+ Configuration Catalog	Microsoft Corporation	C:₩Windows₩System32₩clbcatq.dll	(Verified) Microso...
cldapi.dll	Cloud API user mode API	Microsoft Corporation	C:₩Windows₩System32₩cldapi.dll	(Verified) Microso...
combase.dll	Windows용 Microsoft COM	Microsoft Corporation	C:₩Windows₩System32₩combase.dll	(Verified) Microso...
combase.dll.mui	Windows용 Microsoft COM	Microsoft Corporation	C:₩Program Files₩WindowsApps₩Microsoft.LanguageExperiencePackko-KR_19...	(Verified) Microso...
comctl32.dll	User Experience Controls Library	Microsoft Corporation	C:₩Windows₩WinSxS₩amd64_microsoft.windows.common-controls_6595b6414...	(Verified) Microso...
comctl32.dll	User Experience Controls Library	Microsoft Corporation	C:₩Windows₩WinSxS₩amd64_microsoft.windows.common-controls_6595b6414...	(Verified) Microso...
coml2.dll	Microsoft COM for Windows	Microsoft Corporation	C:₩Windows₩System32₩coml2.dll	(Verified) Microso...
concrt140.dll	Microsoft® Concurrency Runtime ...	Microsoft Corporation	C:₩Program Files₩Microsoft Office₩root₩Office16₩concrt140.dll	(Verified) Microso...
CONTAB32.DLL	Outlook Address Book Service	Microsoft Corporation	C:₩Program Files₩Microsoft Office₩root₩Office16₩CONTAB32.DLL	(Verified) Microso...
CoreMessaging.dll	Microsoft CoreMessaging Dll	Microsoft Corporation	C:₩Windows₩System32₩CoreMessaging.dll	(Verified) Microso...
CoreUIComponents.dll	Microsoft Core UI Components Dll	Microsoft Corporation	C:₩Windows₩System32₩CoreUIComponents.dll	(Verified) Microso...
credui.dll	Credential Manager User Interface	Microsoft Corporation	C:₩Windows₩System32₩credui.dll	(Verified) Microso...
crypt32.dll	Crypto API32	Microsoft Corporation	C:₩Windows₩System32₩crypt32.dll	(Verified) Microso...
crypt32.dll.mui	Crypto API32	Microsoft Corporation	C:₩Program Files₩WindowsApps₩Microsoft.LanguageExperiencePackko-KR_19...	(Verified) Microso...
cryptbase.dll	Base cryptographic API DLL	Microsoft Corporation	C:₩Windows₩System32₩cryptbase.dll	(Verified) Microso...
cryptnet.dll	Crypto Network Related API	Microsoft Corporation	C:₩Windows₩System32₩cryptnet.dll	(Verified) Microso...
cryptsp.dll	Cryptographic Service Provider API	Microsoft Corporation	C:₩Windows₩System32₩cryptsp.dll	(Verified) Microso...
cryptui.dll	Microsoft 신뢰 UI 공급자	Microsoft Corporation	C:₩Windows₩System32₩cryptui.dll	(Verified) Microso...
cscapi.dll	Offline Files Win32 API	Microsoft Corporation	C:₩Windows₩System32₩cscapi.dll	(Verified) Microso...
Csi.dll	Microsoft Office Document Cache	Microsoft Corporation	C:₩Program Files₩Microsoft Office₩root₩vfs₩ProgramFilesCommonX64₩Micr...	(Verified) Microso...

Dll Injection

프로세스는 어떻게 DLL을 load할까요?

- dllmain.cpp

```
// dllmain.cpp : Defines the entry point for the DLL application.
#include "stdafx.h"


extern "C" __declspec(dllexport)int AddInt(int a,int b)
{
  return a + b;
}

extern "C" __declspec(dllexport)int MultInt(int a,int b)
{
  return a * b;
}
```

Dll Injection

프로세스는 어떻게 DLL을 load할까요?

- call_dll.cpp

```
#include <iostream>
#include "stdafx.h"

typedef int(*pDLLFunction)(int, int);
pDLLFunction pFunction = NULL;

int main()
{
    int a = 5;
    int b = 10;
    HMODULE hMod = NULL;
    hMod = LoadLibraryA("my_dll.dll");
    if (hMod == NULL)
    {
        printf("DLL Load Failed.\n");
        return 0;
    }
    pFunction = (pDLLFunction)GetProcAddress(hMod, "AddInt");
    printf("sum = %d \n", pFunction(a, b));
    pFunction = (pDLLFunction)GetProcAddress(hMod, "MultInt");
    printf("mult = %d \n", pFunction(a, b));

}
```

Dll Injection

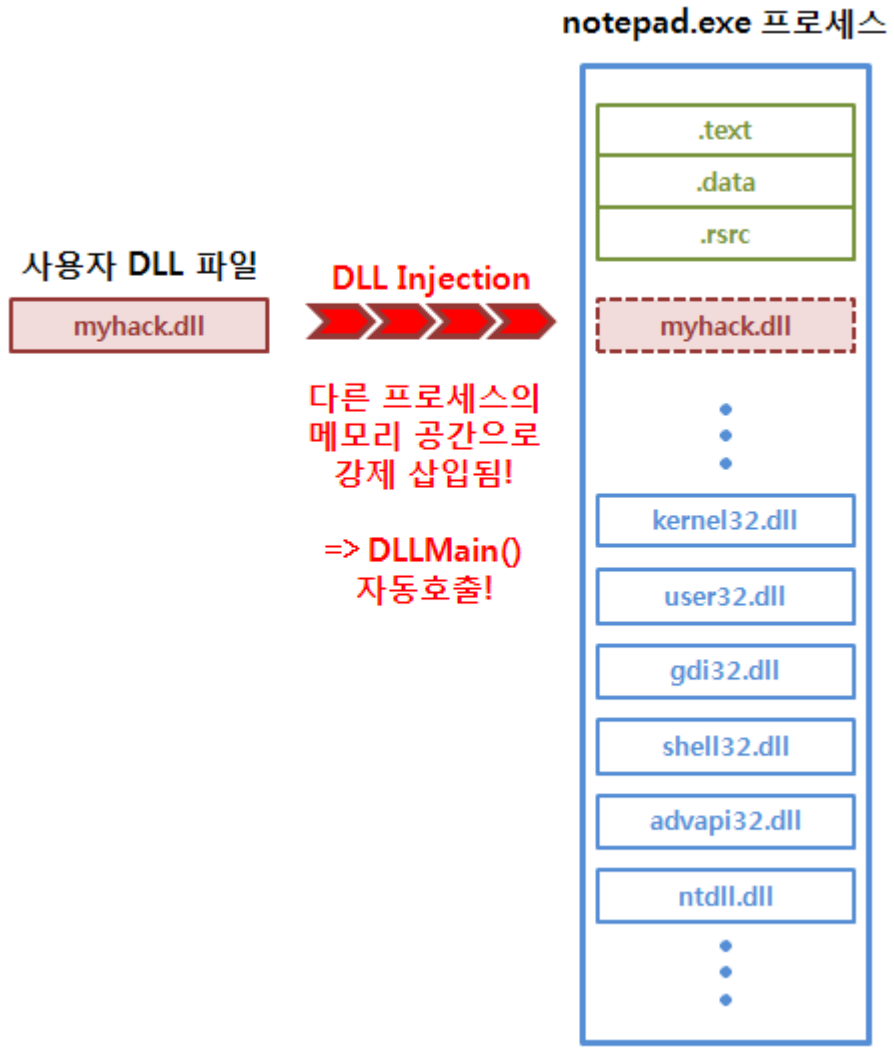
프로세스는 어떻게 DLL을 load할까요?

- 직접 VS로 빌드해서 확인해보기

Dll Injection

DLL injection이 뭔가요?

- 특별한 방법을 사용하여 내가 만든 DLL을 다른 Process에서 Load 하여 DllMain에서 필요한 기능을 실행하는 것
- 정상적으로 DLL을 Load했으니 해당 Process의 메모리에 대한 접근 권한을 갖기때문에 여러가지 동작이 가능하다.



Dll Injection

DLL injection은 어떤 과정을 통해 이뤄지나요?

- code 폴더의 my_dll.cpp와 dll_injection_sample.cpp 확인
- dll_injection()함수...
  - OpenProcess를 통해서 Process의 handler를 가져옴
  - VirtualAllocEx를 통해, Process에 메모리 할당.
  - WriteProcessMemory를 통해 load될 DLL의 경로를 대상 Process 메모리에 입력
  - CreateRemoteThread를 이용하여, LoadLibraryW를 실행

Dll Injection

DLL injection이 뭔가요?

- 실제로 실행해보고, ProcessExplorer를 통해 확인해보기

Libhook

리눅스에는 관련된 기능이 있을까?

Hooking with library on linux

Hooking on Linux

리눅스에는 관련된 기능이 있을까?

- Linux의 경우, shared library injection... 이 유행하지는 않는다.
  - CreateRemoteThread와 비슷한 함수가 Linux에는 존재하지 않아서 ptrace를 사용해야 한다.
  - 또한, dll injection처럼 dllMain에 해당하는 함수가 dlopen인데, 이는 libdl에 존재하여 libdl을 사용하는 프로세스만 적용이 가능하다.
- 보통은 LD_PRELOAD를 통해서 타 라이브러리보다 내 라이브러리가 먼저 로드되게 만들어 hooking + code injectio을 진행한다.

Hooking on Linux

Hooking with library on linux

- https://github.com/ugonfor/RaceGuard-Demo 의 예제를 참고

```c
#define _GNU_SOURCE
#include<stdio.h>
#include<dlfcn.h>
#include<sys/types.h>
#include<string.h>
#include<unistd.h>

#define MAX 0x100
extern char * __progname;
char cache[MAX] = {0};

// mktemp
static char *(*hook_mktemp)(char *__template) = NULL;

char* mktemp(char *__template){
    if (hook_mktemp == NULL) hook_mktemp    = dlsym(RTLD_NEXT, "mktemp");
    printf("[!] mktemp(%s) = ", __template);
    char* hook_ret                          = hook_mktemp(__template);
    printf("%s\n", hook_ret);
    return hook_ret;
}

void __attribute__ ((constructor)) before_load(void)
{
    if (hook_mktemp == NULL) hook_mktemp    = dlsym(RTLD_NEXT, "mktemp");
}
```

Hooking on Linux

Hooking with library on linux

- void __attribute__ ((constructor)) before_load(void)
- 위 함수에 해당하는 부분이 일반적으로 librar가 load되기 전에 실행된다는 특징을 가지고 있다.

과제

지뢰찾기 후킹

- Dll injection을 통하여 지뢰찾기를 클리어하라
- 자신이 원하는 입맛대로 dll injection을 만들어서 지뢰찾기 핵을 만들어라.
- 웃는 얼굴 click시 바로 모든 지뢰가 보이게 되는 것은 무조건 구현

- 메일 제목: [Injection][디미고]본인이름
  - ex) [Injection][디미고]유효곤
- 내용 : github repository 주소를 포함할 것
- git : VisualStudio 프로젝트 파일을 통째로 업로드하여 바로 빌드 할 수 있게 업로드할 것
- README 추가하여 실행방법, 구현 방법에 대한 간단한 설명

과제

지뢰찾기 후킹

- Dll injection을 통하여 지뢰찾기를 클리어하라

```
.text:01001FA6                cmp     dword_1005148, edi ; jumptable 01001F75 case 513
.text:01001FAC                jnz     short loc_1001F84
.text:01001FAE                push    [ebp+lParam]
.text:01001FB1                call    sub_100140C     ; click 시 호출되는 함수
.text:01001FB6                test    eax, eax
.text:01001FB8                jnz     loc_1001C5E
.text:01001FBE                test    byte ptr dword_1005000, bl
.text:01001FC4                jz      def_1001F75     ; jumptable 01001F75 default case, cases
.text:01001FCA                mov     eax, [ebp+wParam]
.text:01001FCD                and     al, 6
```

```
.text:01002F80
.text:01002F80 ; 지뢰를 다 보여주는 함수
.text:01002F80
.text:01002F80 ; int __stdcall sub_1002F80(char)
.text:01002F80 sub_1002F80     proc near               ; CODE XREF: sub_100347C+2F↓p
.text:01002F80
.text:01002F80 arg_0           = byte ptr  4
.text:01002F80
.text:01002F80                 mov     eax, dword_1005338
.text:01002F85                 cmp     eax, 1
```

reference

- https://blog.naver.com/PostView.nhn?blogId=tipsware&logNo=221359282016&parentCategoryNo=&categoryNo=83&viewDate=&isShowPopularPosts=true&from=search
- https://github.com/OpenSecurityResearch/dllinjector
- https://wendys.tistory.com/23
- https://hackersstudy.tistory.com/75
- https://liveyourit.tistory.com/114