Assignment2

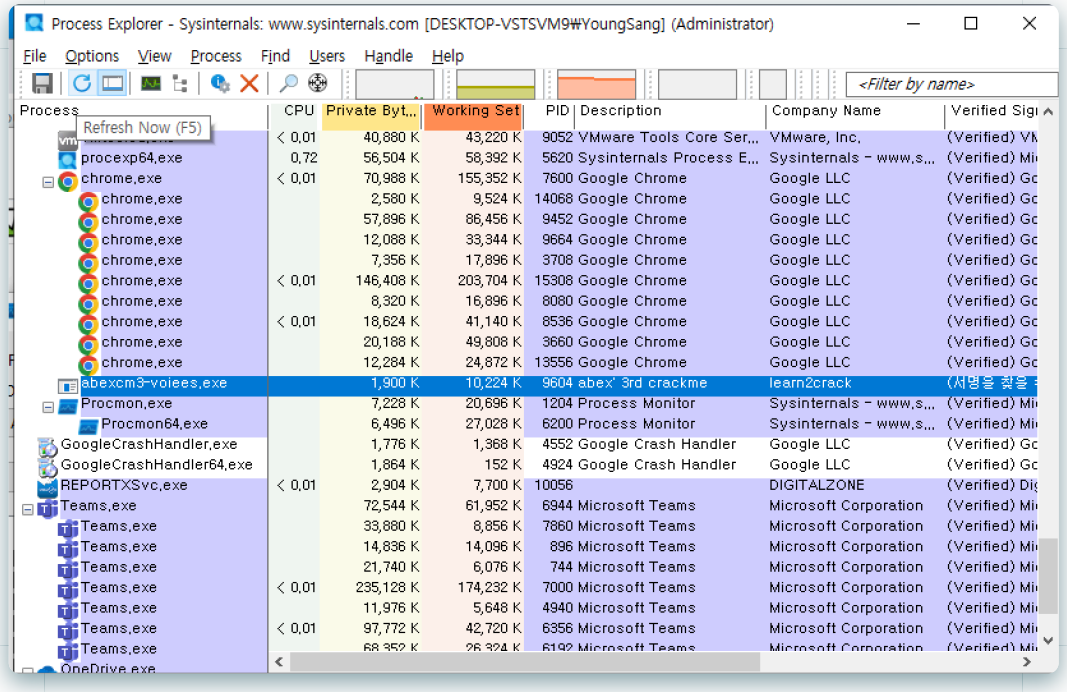
> 강민준

✔ Description.

```
abexcm3-voiee.exe를 분석 및 코드 패치 없이 keyfile을 인증
1. 실행흐름 파악
2. 분석 지점을 파악 (=> 분석 이후, 어떤 지점을 주안점으로 삼아야하는가?)
3. 코드 동적 분석 및 내부 알고리즘 파악 (=> 분석 지점에 대한 동적 분석 수행)
/**
    keyfile을 만들어내는 것, binary 수정X, 분석한 알고리즘을 파악한대로      필요한 parameter를 채워넣는 방식
**/
```

✔ 분석 과정

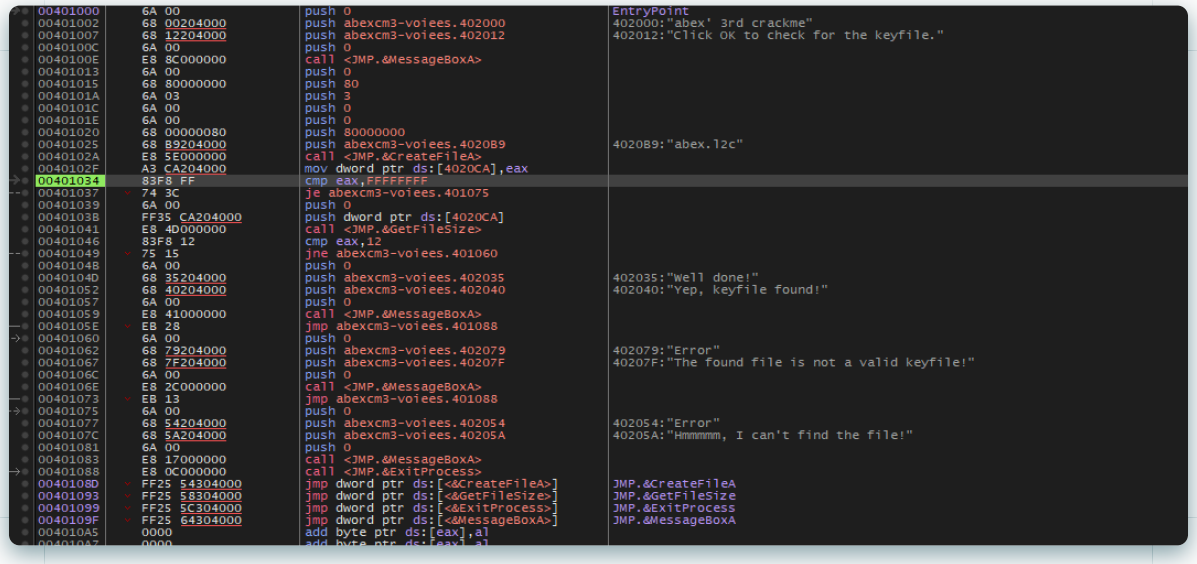
1. 프로세스가 실행될 때, 단일 모듈인지 확인하기 위하여 `Process Explorer` 를 이용한다.



`abexcm3-voiee.exe` 는 단일 모듈의 실행 파일인 것을 확인해볼 수 있다.

---

2. xdbg를 이용하여 실행흐름을 분석한다.



위와 같이, `CreateFileA()` 의 인자값으로 {"abex.l2c", 80000000, 0, 0, 3, 80, 0}을 넘겨 특정 operation을 수행한 다음, 리턴 값과 0xFFFFFFFF를 비교한다. 만약 같다면, Error 문으로 분기하게 된다. 코드 패치 없이 해당 분기문으로 분기하지 않기 위해서 `CreateFileA()` 의 명세를 살펴보아야한다. CreateFileA 명세

C++ Copy

```
HANDLE CreateFileA(
  [in]           LPCSTR                lpFileName,
  [in]           DWORD                 dwDesiredAccess,
  [in]           DWORD                 dwShareMode,
  [in, optional] LPSECURITY_ATTRIBUTES lpSecurityAttributes,
  [in]           DWORD                 dwCreationDisposition,
  [in]           DWORD                 dwFlagsAndAttributes,
  [in, optional] HANDLE                hTemplateFile
);
```

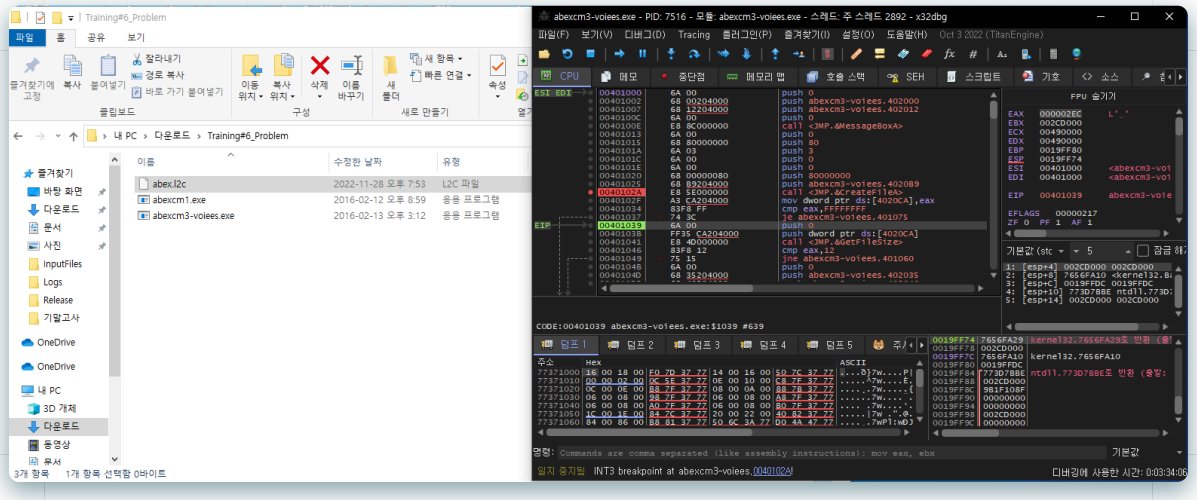
OPEN_EXISTING 3	Opens a file or device, only if it exists. If the specified file or device does not exist, the function fails and the last-error code is set to **ERROR_FILE_NOT_FOUND** (2). For more information about devices, see the Remarks section.

명세를 읽어보면, `CreateFileA()` 의 첫번째 인자인 "abex.l2c" 인 파일에 대해서 OPEN_EXISTING 모드로 open한다는 것을 확인할 수 있다. 또한, 해당 함수가 실패할 경우, `INVALID_HANDLE_VALUE` 값을 반환한다는 것을 알 수 있다. "abex.l2c"파일이 존재하지 않을 경우, 해당 함수가 실패한다는 것을 확인할 수 있었는데 이 때의 반환 값은 0xFFFFFFFF 이다.

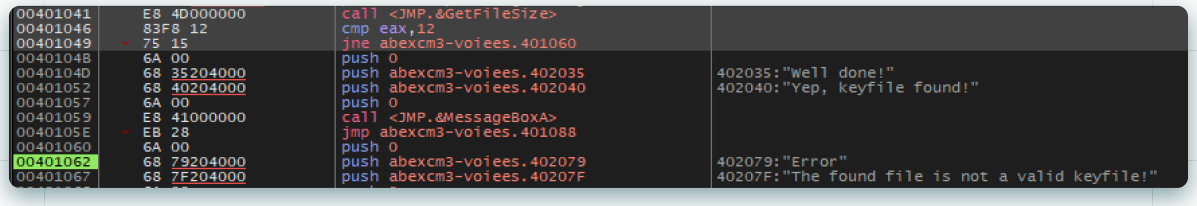
---

3. `CreateFileA()` 함수가 실패하지 않도록 적절하게 인자를 채워준다.

`CreateFileA()` 은 WIN32 API로서 시스템콜을 내부적으로 호출한다. "abex.l2c" 파일을 적절하게 생성해준다면 아래와 같이 원하는 분기 내부로 들어올 수 있게된다.



그럼에도 불구하고, 이번에는 `The found file is not a valid keyfile!` 라는 결과를 확인할 수 있었다. 해당 실패 분기로 분기하게 된 원인을 살펴보자.



`GetFileSize()` 함수의 반환 값이 0x12이 아니기 때문에, Error로 분기한 것을 확인할 수 있다.

`GetFileSize()` 의 명세를 확인하여, 반환 값이 0x12가 되도록 만들어보자. GetFileSIze 명세

Return value

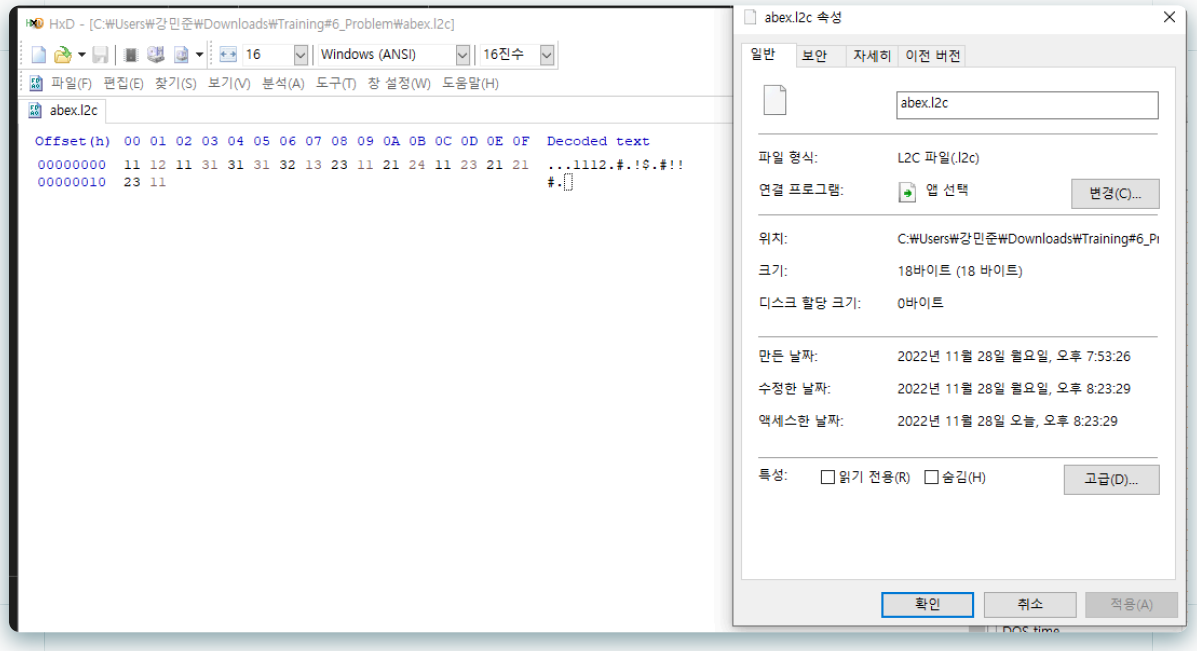
If the function succeeds, the return value is the low-order doubleword of the file size, and, if *lpFileSizeHigh* is non-**NULL**, the function puts the high-order doubleword of the file size into the variable pointed to by that parameter.

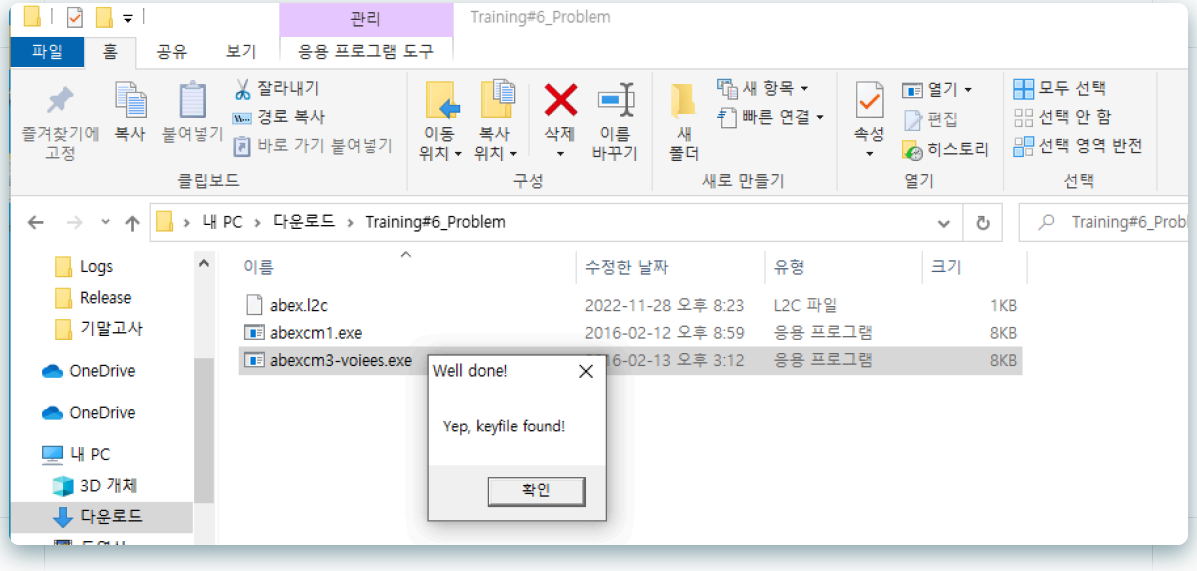
명세에 따라, DWORD(32bit 아키텍처 기준, 4바이트)의 형태로 실제 파일 크기를 반환하는 것을 확인할 수 있다. 이와 같은 사실을 통해서, "abex.l2c" 파일의 크기를 0x12바이트로 설정해주면 우리가 원하는 실행흐름인 `0x0040104B` ~ `0x0040105E` 순으로 수행할 수 있다.

---

4. 마무리

`HxD` 를 이용하여 해당 파일의 바이너리를 $12_{16}(=18_{10})$바이트 크기로 설정해준다.





성공적으로 원하는 결과를 얻을 수 있었다.