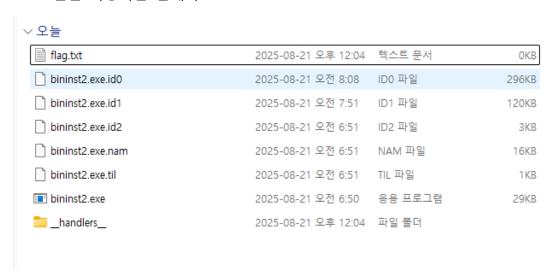


Frida 툴을 사용하는 문제다.



프로그램을 실행시키면 어떤 것도 뜨지 않고 바로 종료된다.

- \$ frida-trace -i *File* -f bininst2.exe -X KERNEL32
- -X KERNEL32 옵션을 통해 해당 동적라이브러리를 제외하고 계측을 시도한다.

```
Started tracing 576 functions. Web UI available at http://localhost:54627/
            /* TID 0x8a6c */
  1965 ms NtDeviceIoControlFile()
  1966 ms RtlDosApplyFileIsolationRedirection_Ustr()
  1966 ms RtlDosApplyFileIsolationRedirection_Ustr()
  1966 ms RtlDosApplyFileIsolationRedirection_Ustr()
  1966 ms NtQueryAttributesFile()
  1966 ms NtQueryAttributesFile()
1966 ms NtOpenFile()
1966 ms RtlDosApplyFileIsolationRedirection_Ustr()
1973 ms GetSystemTimeAsFileTime()
  1973 ms
            GetSystemTimeAsFileTime()
  1973 ms GetModuleFileNameW()
             | GetModuleFileNameW()
  1973 ms
  1973 ms AreFileApisANSI()
  1973 ms | AreFileApisANSI()
  1973 ms CreateFileA()
  1973 ms CreateFileA entering in module: KERNELBASE.dll
  1973 ms lpFileName = <Insert path here>
  1973 ms
                CreateFileA()
  1973 ms
                      CreateFileW()
                          CreateFileW()
  1973 ms
```

해당 프로세스에서는 CreatFile API를 사용한다. 또한 IpFileName에 <Insert path here>라는 문자열을 파일 경로로 넘기고 있는데, 파일이 없어서 강제종료된 것이었다.

그래서 매개변수에 flag.txt를 전달하고 파일 이름을 넘긴다.

```
8
     defineHandler({
       onEnter(log, args, state) {
9
         log('CreateFileA()');
10
         const newPath = "flag.txt";
11
12
         const buf = Memory.allocUtf8String(newPath);
13
         this.buf = buf;
14
         args[0] = buf;
15
         log('lpFileName = ' + args[0].readUtf8String());
16
       },
17
18
       onLeave(log, retval, state) {
19
       }
20
     });
21
```

KERNEL32/CreateFileA 에 flag.txt를 생성하는 코드를 작성한다.

그리고 실행했는데 flag.txt에 아무 값도 채워지지 않았다.

WriteFile 로그를 보면 0바이트를 채우는 것을 알 수 있다.

```
8
     defineHandler({
9
       onEnter(log, args, state) {
         log('WriteFile() - hFile = ' + args[0]);
10
11
         log('WriteFile() - nNumberOfBytesToWrite = ' + args[2]);
12
       },
13
14
       onLeave(log, retval, state) {
15
16
     });
```

그리고 WriteFile 두 번째 매개변수에는 어떤 값으로 세팅 되어있는지 보기 위해 log를 하나 더 추가했다.

```
defineHandler({
   onEnter(log, args, state) {
      log('WriteFile() - hFile = ', args[0]);
      log('WriteFile() - lpbuffer = ', hexdump(args[1]));
      log('WriteFile() - nNumberOfBytesToWrite = ', args[2]);
   },
   onLeave(log, retval, state) {
   }
});
```

```
WriteFile() - lpbuffer =
                                         4 5 6 7 8 9 A B C D E
140002270 63 47 6c 6a 62 30 4e 55 52 6e 74 6d 63 6a 46 6b
                                         cGljb0NURntmcjFk
140002280 59 56 39 6d 4d 48 4a 66 59 6a 46 75 58 32 6c 75 140002290 4e 58 52 79 64 57 30 7a 62 6e 51 30 64 47 6c 76
                                         YV9mMHJfYjFuX2lu
                                         NXRydW0zbnQ0dGlv
1400022a0 62 69 46 66 59 6a 49 78 59 57 56 6d 4d 7a 6c 39
                                         biFfYjIxYWVmMzl9
15 ms WriteFile() - nNumberOfBytesToWrite = 0x0
  15 ms
         | WriteFile()
```

임의의 값이 나왔고, base64로 디코딩 했더니 flag가 출력됐다.

cGljb0NURntmcjFkYV9mMHJfYjFuX2luNXRydW0zbnQ0dGlvbiFfYjIxYWVmMz19