FFIO, MMIO 속도비교 및 5GB 이상의 파일 MMIO 이용하기

취약점분석 이태규

MMIO 기본 개념 : 파일을 메모리 상에 올려서 빠른 속도로 File IO를 수행하는 것.

FFIO vs MMIO : 파일크기 500MB로 복사하는 시간 비교

Debug 모드에서 FFIO가 MMIO보다 빠른 것을 알 수 있는데 이는 복사할 때 FFIO는 버퍼의 크기가 4096, 즉 한번에 4096 바이트씩 읽고 쓰는 반면 MMIO는 1 바이트씩 읽고 쓰기 때문에 느려지는 것이다.
Release 모드에서 FFIO보다 MMIO가 더 빠른 것을 관측할 수 있었는데 스크린 샷을 찍기 위해 실행하자 갑자기 FFIO, MMIO 모두 0초가 뜨며 복사가 되지 않았다. Release 모드의 특성상 디버깅을 해볼 수 없었고 의문스럽지만 방법이 없는 것 같다.

Over 4GB ? : 파일의 크기가 4GB가 넘어가면 메모리상에 파일을 올릴 수 없게 되므로 따로 처리가 필요하다. 이를 위해 MapViewOfFile 함수의 아규먼트들의 의미를 파악해야 한다.

MapViewOfFile은 프로세스의 주소 공간에 파일을 매핑하기 위해서 두 가지 추가적인 정보가 필요하다. 파일의 어디부터 매핑할 것인가?, 파일의 얼마만큼을 매핑할 것인가? 여기서 어디부터 에 해당하는 것이 FileOffsetHigh와 FileOffsetLow가 되는 것이고, 얼마만큼 에 해당하는 것이 NumberOfBytesToMap이 되는 것이다.

이를 이용해 파일을 각각 적당한 크기인 500mb로 나누어 메모리에서 Page In, Out 이 일어나 듯이 조각조각 파일을 메모리에 올려서 복사를 하면 4GB가 넘는 파일도 MMIO를 통해 복사할 수 있다.

// 복사를 할 때 파일 핸들러를 다루기 복잡한 면이 조금은 존재한다. 하지만 어차피 복사란 개념이 같은 용량, 데이터의 파일을 하나 더 생성하는 것이므로, 같은 크기의 파일을 생성한다. -> 내부의 데이터 값을 원본의 값으로 덮어 씌운다. 의 순으로 복사를 하면 좀 더 쉽게 파일 핸들러에 대한 고민없이 복사를 수행할 수 있다.