WideChar String vs MultiByte String

취약점분석_이태규

문제 제기 :
소스상에서
StringCbPrintfW(string_buf,sizeof(string_buf),L"동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세");
StringCbPrintfW(string_buf,sizeof(string_buf),L"All work and no play makes jack a dull boy.");
StringCbPrintfA(string_bufa,sizeof(string_bufa),"동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세");
StringCbPrintfA(string_bufa,sizeof(string_bufa),"All work and no play makes jack a dull boy.");

이렇게 큰 차이가 없는 StringCbPrintf (W, A) ... (L, ' ') "~~" 의 두 형태의 출력함수가
--> 冒t?샜? 1퍁눗픱? 획t밝? 蹉캐]퐜 l l w o r k a n d n o p l a y m동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세All work and no play makes jack a dull boy.
결과적으로는 엄청난 차이를 갖는다는 것을 알 수 있다.

Q) 이런 문제가 왜 생기며 해결방법을 제시하라.
A) 텍스트에는 수많은 인코딩 방법이 존재한다. 그 중 WideChar String(유니코드) 와 MultiByte String 의 차이 때문에 위와 같은 문제가 발생한다. 한 개의 Text파일은 한 개의 인코딩 방식을 가지고 텍스트를 저장한다. 따라서 한 텍스트 문서에 두 가지 방식으로 인코딩된 문자를 동시에 사용하면 어떤 하나는 필연적으로 우리가 알아볼 수 없는 형태로 출력된다. 따라서 이를 해결하기 위해 한 텍스트 파일에는 한 개의 인코딩만을 사용하여 문자들을 저장하고 그 인코딩 방식을 알려주는 BOM(Byte Order Mark) 문자를 사용한다.

Encoding	Representation
UTF-8	EF BB BF
UTF-16 빅 엔디안	FE FF
UTF-16 리틀 엔디안	FF FE
UTF-32 빅 엔디안	00 00 FE FF
UTF-32 리틀 엔디안	FF FE 00 00
SCSU	0E FE FF
UTF-EBCDIC	DD 73 66 73
BOCU-1	FB EE 28

표 1 : 유니코드 별 BOM값