

Q: What's the problem ?

```
#include <stdio.h>

int main()
{
    char ch = 0xFF;

    printf( "%02X\n", ch );
    return 0;
}
```

預期結果：

FF

實際結果：

FFFFFFFF

發生什麼事情了？

Ans:

此處的問題在於「轉型」的問題，從「char」轉至「int」

```
printf( "%02X\n", ch );
```

等效於

```
printf( "%02X\n", (int)ch );
```

而 ch 值為 0xFF，以 step by step 的方式來說明結果：

- 1) 在 Visual C++ 中 char 是指 signed char，因為具有 2 補數的表式方式
- 2) 因為 ch 的值為 0xFF，MSB 為 1，所以其值為負值，經過 2 補數的計算，其值為 -1
- 3) 從 char 轉至 int 時，會做所謂的「sign extension」，-1 從原本為 0xFF 轉為 0xFFFFFFFF
- 4) printf 中的 "%02X"，指的是「以 16 進位來顯示型別為 unsigned int 的數值，而最小寬度為 2，小於 2 則補 0」
- 5) 因此，顯示結果為 FFFFFFFF

解決的方式，可以將

```
char ch = 0xFF;
```

改為

```
unsigned char ch = 0xFF;
```

或在

```
printf( "%02X\n", ch );
```

強迫它轉型

```
printf( "%02X\n", (unsigned int)(unsigned char)ch );
```

會造成結果非預期，主要是因為 sign extension 所造成的錯誤，此種問題稱為「Sign extension error」。

Reference

- Sign extension
http://en.wikipedia.org/wiki/Sign_extension
- Sign extension error
http://www.owasp.org/index.php/Sign_extension_error

Written By James Liang @ 12/29/2006