

memcpy(), memmove(), memset(), strxfrm(), wcsxfrm(), wmemset(), wmemcpy(), wmemmove(), wcostombs(), wcsrtombs(), mbstowcs(), mbsrtowcs(), swprintf() e vsprintf().

Di seguito alcuni esempi di funzioni solitamente considerate sicure, ma utilizzate in modo errato.

Esempio:

Forma non corretta:

```
char dest[512];
char *source;
// puntatore char source manipolabile
// dall'utente
strncpy(dest, source, strlen(source));
#define LEN 5000
// LEN superiore alla capacità
// di contenimento massima di
// dest
char dest[1024];
// variabile source manipolabile
// dall'utente
char source[LEN];
memcpy(dest, source, LEN);
```

Forma corretta:

```
char dest[512];
strncpy(dest, source, sizeof(dest));
/* inserimento di NULL alla fine di
   dest
*/
...
#define LEN 1024;
char dest[1024];
// variabile source manipolabile
// dall'utente
char source[LEN];
memcpy(dest, source, LEN - 1);
/* inserimento di NULL alla fine di
   dest
*/
...
```

7.1.8.3 Bitfields

Se nel codice vengono svolte operazioni di bit shifting o si utilizzano bitfield, bisogna indicare le piattaforme con cui il codice è compatibile per mitigare problemi/errori di porting.

7.1.8.4 Macro

Se le macro sono espansive, i parametri passati non devono causare effetti collaterali.

Esempio:

Forma non corretta:

```
#define max(a, b) (a) > (b) ? (a) : (b)
risultato = max(i, j) + 3;
/*
 * tutto questo viene espanso in
 * risultato = (i) > (j) ? (i) : (j)+3;
 *
 */
```

Forma corretta:

```
#define max(a,b) ( (a) > (b) ? (a) : (b) )
```