При динамично разполагане на памет може да се появи препалване на буфера, информация бива записвана извън границите на буфера което предоставя възможност на недоброжелател да вмъкне код с който да промени поведението на програмата. Това може да се осъществи с техниките Unlink и Frontlink.

Пример за Unlink macro:

#define unlink(p, BK, FD) {

FD = P->fd;

BK = P->bk;

FD-> bk =BK;

BK->fd = FD;

}

Пример за код за Buffer Overflow:

intmain(intargc, char \*argv[]) {

char \*first, \*second, \*third;

first = malloc(666);

second = malloc(12);

third = malloc(12);

strcpy(first, argv[1]);

free(first);

free(second);

free(third);

return(0);

}

Командата strcpy() е без граници което допуска препълване на буфера. free(first); освобождава първия блок памет.free(second); ако паметта е свободна, я слепва към първия блок, за да провери дали паметта е свободна командата free() проверява PREV\_INUSE бит на третия блок.

Недоброжелателят може да запише информация в адресното пространство на 2рия блок примерно -4, по този начин третия блок изглежда да е 4 байта преди 2рия блок. free() операцията се извършва и се слепва 2рия блок към първия блок, така вече пойтерите на първия блок слочат към недоброжелателен код.

