Altri problemi di overflow si manifestano a seguito di circostanze diverse e non necessariamente correlabili alla copia o allo spostamento di dati in un buffer insufficiente. Le principali problematiche di overflow oggi conosciute vengono di seguito descritte.

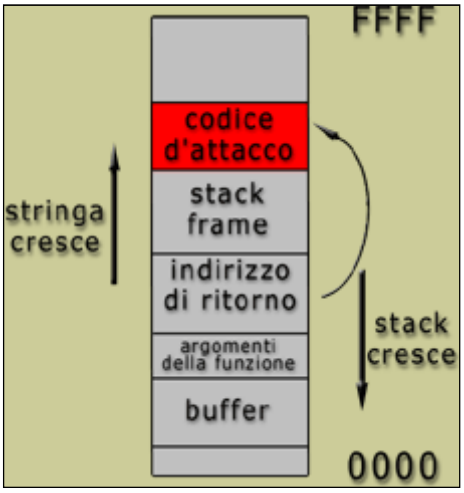
6.5.1 Stack overflow

Il principio di sfruttamento è molto semplice e si basa sulla possibilità di saturare un buffer oltre le sue reali capacità di contenimento, fino a sovrascrivere l'indirizzo di ritorno della funzione vulnerabile. L'indirizzo di ritorno è un valore posizionato nella regione di memoria stack che permette all'applicazione, al rientro della funzione chiamata, di riprendere l'esecuzione dall'istruzione immediatamente successiva. Questo valore è puntato da diversi registri, in base all'architettura hardware per la quale l'applicazione è stata compilata (ad esempio EIP su piattaforma x86 o RIP su piattaforma x64). Riuscendo a saturare un buffer oltre le sue capacità di contenimento, un aggressore ha la possibilità di sovrascrivere, con valori prettamente arbitrari, tutte le aree di memoria adiacenti, fino a giungere all'indirizzo di ritorno, facendo proseguire l'esecuzione del programma da qualsiasi indirizzo di memoria desiderato, deviando il regolare flusso esecutivo dell'applicazione.
L'esecuzione di codice malevolo attraverso uno stack overflow si sostanzia fondamentalmente in tre step:

- l'aggressore satura il buffer non soggetto a bound-checking e colloca ad un certo punto della memoria lo shellcode;
- l'aggressore sovrascrive l'indirizzo di ritorno della funzione vulnerabile con l'indirizzo in memoria in cui risiede lo shellcode;
- Dal ritorno della funzione lo shellcode viene eseguito;

Esempio:

Rappresentazione generica di uno stack overflow:



Contromisure

Il programmatore deve configurare i cicli sugli array in modo da non superare il numero di elementi previsto. Un loop per tutta la lunghezza *possibile* del buffer potrebbe attivare il codice malevolo.

6.5.2 Off-by-one/Off-by-few

Gli overflow che si manifestano nello stack sono oggi meno frequenti rispetto al passato, ma non sono del tutto scomparsi. In realtà, queste problematiche sono ancora riscontrabili nei moderni software, a causa di errate pratiche di programmazione. Gli overflow definiti Off-by-one o Off-by-few ne sono la dimostrazione palese. Rientrano in questa categoria tutti gli overflow che, al contrario degli stack overflow, permettono di eccedere solo di uno o pochi byte oltre le reali capacità di contenimento di un buffer. Questa condizione, a seconda del compilatore utilizzato, della predisposizione dei buffer e delle variabili in memoria e quindi soprattutto dell'architettura hardware su cui il software gira, può permettere ad un aggressore di alterare a