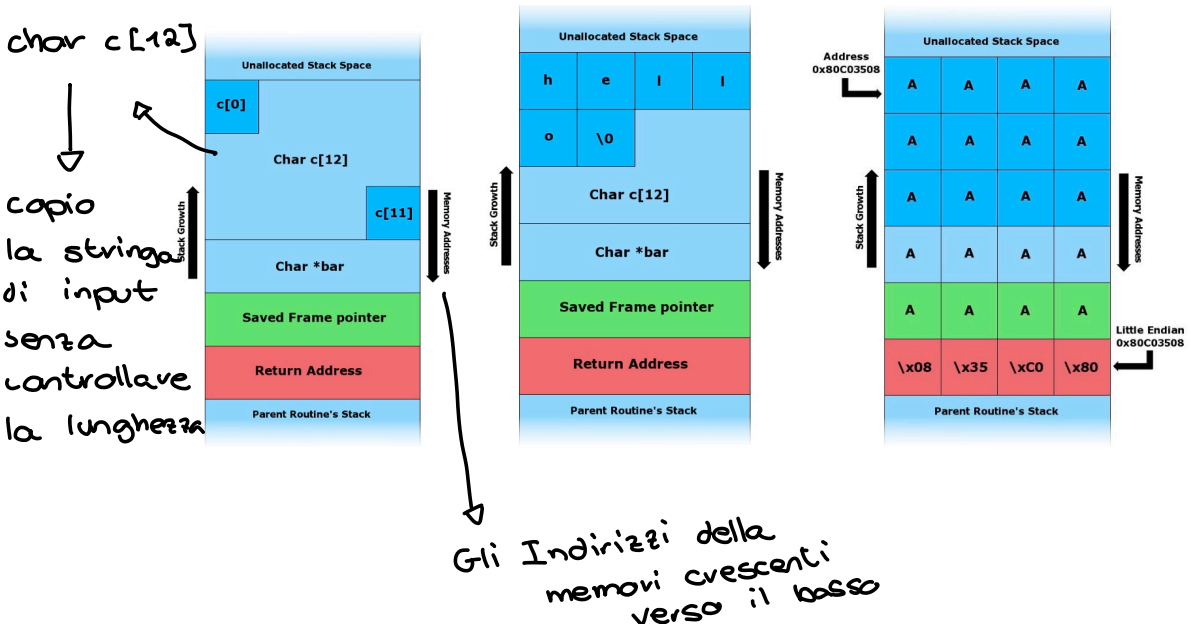
Buffer Overflow

- La vulnerabilità risiede nella manipolazione dello stack in modo da ottenere un attacco efficace all'esecuzione del programma stesso.
- La manipolazione dello stack avviene dando un input che verrà costruito ad hoc per avere sullo stack un effetto di "presa di controllo".
- La stringa di input viene passata a una funzione che copia il contenuto dentro a una variabile, senza controllarne la lunghezza.



- Supponendo che venga fatta una strcpy → Inserire una stringa di dimensione superiore a 12 bytes va a coprire indirizzi di memoria successivi

- L'obiettivo dell'attaccante è sovrascrivere il return address, con un indirizzo inserito nella stringa di input. In questo modo, effettuare una jump ad un altra

locazione di memoria con codice malevolo

- A volte il codice malevolo viene inserito nella stringa di input.

- Per risolvere → pensare a una crescita dello stack nella direzione opposta
  - ↓ utilizzate dei canary points che se sovrascritti non possono garantire la sicurezza.

- L'attacco è completo quando il codice viene iniettato ed eseguito allora si potrebbe andare a modificare quell'area di memoria in modo che non sia eseguibile bloccando il sistema.

- SQL Injection

- La situazione in cui siamo è un sistema web:
  - Utente: visitatore con browser
  - Server: elabora dati degli utenti
  - Dati utente: form, URL (&val = val)

- La vulnerabilità è associata a difetti del programma presso il server (PHP, ecc). il quale deve colloquiare con un server DB. Per questa ragione, generalmente il codice difettoso comprende interrogazioni sql.

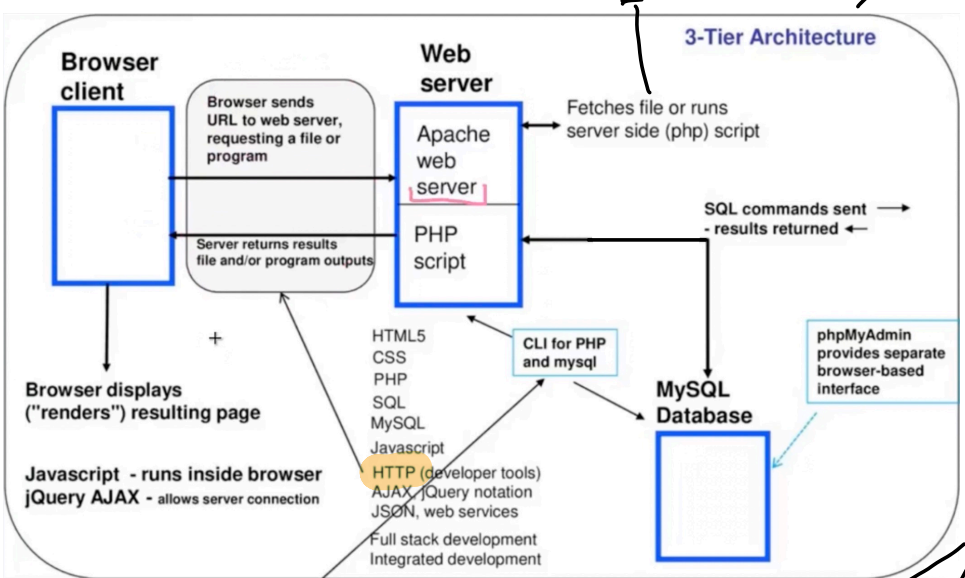
- Attacco
  - Lettura dati presenti sul DB
  - Modifica non autorizzata dei dati presenti sul DB.

Un esempio è `' OR '1' = '1'`, sostituisco il campo con una condizione vera. In questo modo la query SQL ha successo.

→ Vengono usati strumenti come SQLmap

→ Possibili difese → convertire la stringa di input dell'utente, prima di inserirlo nella query
- ↓ igienizzazione dell'input
- ↘ escape / quoting

→ Da lezione



html

① Apache web server + HTTP
- botta e risposta (semplice)
- ↓ no connessione aperta

→ la debolezza avviene quando c'è del cmd
- c'è dell'eseguibile così da iniettare

La minaccia è codice eseguibile!

posso attaccare, browser, web server e database.

Clients (attaccante) → HTTP POST request → Server (vittima)
- ↓ "form"

- Il cross-site scripting (XSS)
  - è un tipo di vulnerabilità di sicurezza che si può trovare in alcune applicazioni web.
  - ↓ iniettare script lato client di altri utenti