SQL Injections

Renny

Ulisse

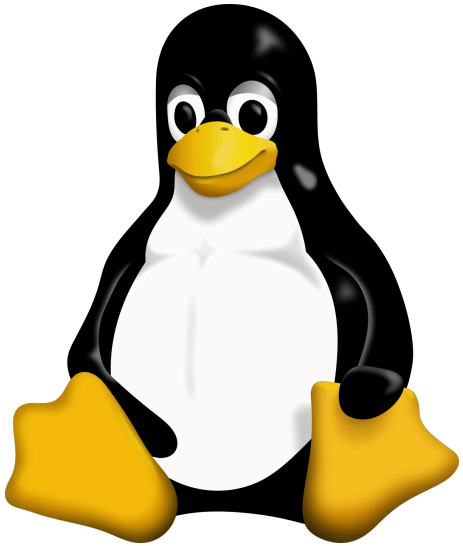
13/12/2024

Preambolo

Cosa ci serve:

- basi di web security (`requests`)
- conoscenza di base di **SQL** (ma anche no)
- creatività

Questa lezione si può seguire da Windows o Linux (ma sempre meglio installare Linux 🙏).



Non vi ricordate / non sapete SQL?

Hint : guardate qui

(manco io me lo ricordo)

A cosa mi serve SQL?

Alcuni di voi avranno visto **SQL** in sé per sé, ma come lo integro in un'applicazione e per cosa lo uso?

Use cases ovvi:

- conservare dati strutturati:
  - credenziali applicazioni web 🔥?
  - dati personali di utenti?

Chiaramente per un attaccante un database è sempre una superficie d'attacco notevole.

Come interagisce un’applicazione con il DB?

App <=> DB

In qualche modo la mia applicazione farà **query** al database sottostante.

Possiamo immaginarci qualcosa del genere:

```
user_input = ...  \# prendiamo input dall'utente
query = "SELECT \* FROM users WHERE id = " + user_input
cursor.execute(query)
```

Come al solito, il primo approccio che prendiamo è spesso sbagliato. Come posso rompere un codice del genere?

Se avete voglia di testare un po' (e dovreste), provate questo sito.

Cosa non va?

Come lo rompo?

Guardando qualsiasi applicativo conviene sempre ragionare partendo da dove abbiamo **controllo**, ovvero il nostro input. Dove andrà a finire ciò che mandiamo? Che tipo di restrizioni sono imposte su ciò che mandiamo?

Ragionando su questi binari, ci rendiamo presto conto che:

- Possiamo mandare quello che ci pare.
- Andrà a finire direttamente nella query.

Nulla a questo punto ci vieta di mandare **comandi**, ovvero parole che SQL interpreta come codice: possiamo iniettare (injection) **SQL** a nostro piacimento!

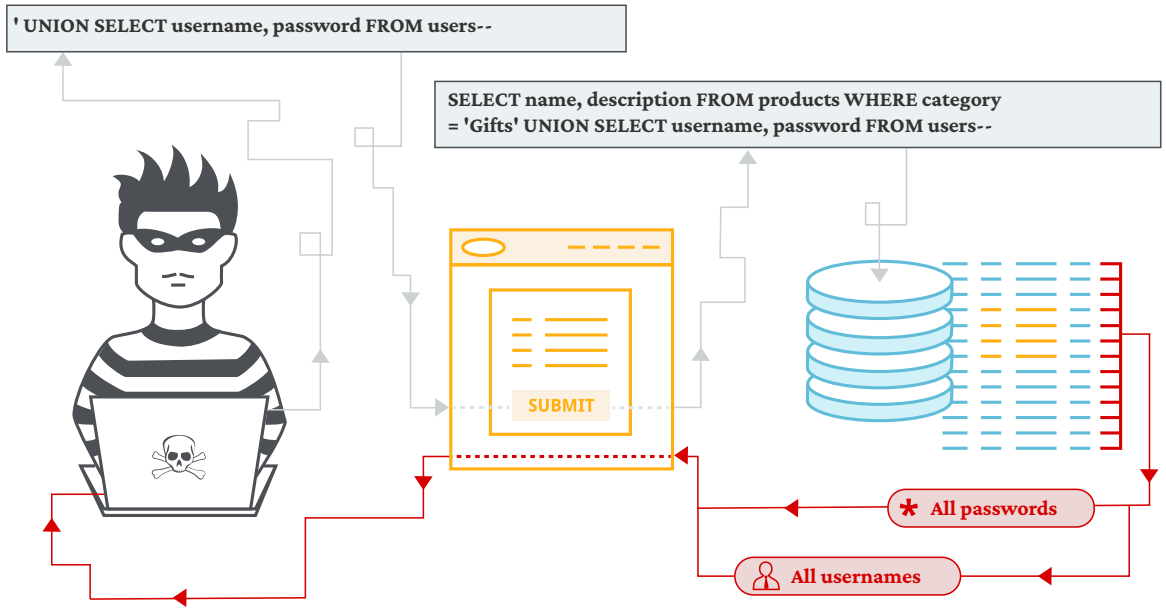
Abbiamo una SQL injection, e questo vuol dire che siamo liberi di leggere e scrivere nel DB, in modo (spesso) arbitrario.

Un esempio più complesso (e utile)

Una questione di controllo

Spesso abbiamo meno **controllo** sul nostro input: o è filtrato o è piazzato in modo sconveniente se vogliamo rompere l'applicazione

Provate a risolvere questa challenge di OliCyber (hint hint: **control characters**).



Definizione: Injection

Quando un programma interpreta il nostro input, che dovrebbe essere letterale ed inerte, come codice. Spesso richiede un qualche **escape character** o **sequence**.

CODE
INJECTION

Tangente: come si fa un’app sicura?

E come dovrei fare?

Ci sono modi e modi di fare query:

- I modi giusti e sicuri 🤦:
  - Query parametrizzate: uso dei placeholder per rappresentare i miei valori nella query.

```
cursor.execute("SELECT \* FROM users WHERE id = ?", (user_id,))
```

- ORM (Object Relational Mapping): Uso librerie per astrarre le query (SQLAlchemy).

Ricordatevi che in questo modo vi state semplicemente affidando al codice di altre persone, che spesso è ugualmente rotto, però meglio di niente.

E se non vedo tutto l'output?

Blind injections

È molto raro che i server connettano direttamente gli utenti al DB senza fare qualche altro calcolo in mezzo.

Forse l'**injection** è in un endpoint che restituisce solo un valore numerico oppure booleano per la logica dell'applicazione.

O ancora, immaginiamoci di avere un **injection** che ci permette di sovvertire un endpoint per vedere i dettagli di un utente, in modo da poter vedere anche la sua password: magari la query che viene eseguita è giusta, ma il server prova a interpretarla secondo il formato che si aspetta (senza la password) e crasha, senza darci altre informazioni.

Ci troviamo davanti a un esempio di **oracolo**.

Definizione: Oracolo

~~L'oracolo (dal latino oraculum) è un essere o un ente considerato fonte di saggi consigli o di profezie, un'autorità infallibile, solitamente di natura spirituale.~~

In cybersecurity, una funzione che ci permette di interrogare un programma ricevendo una risposta booleana (sì o no).

Spesso nel mondo reale la risposta non è diretta, ma deriva dalla presenza o meno di un errore: **error-based oracle**.

Come si chiede un nome a un oracolo?

Il tuo nome comincia con A?

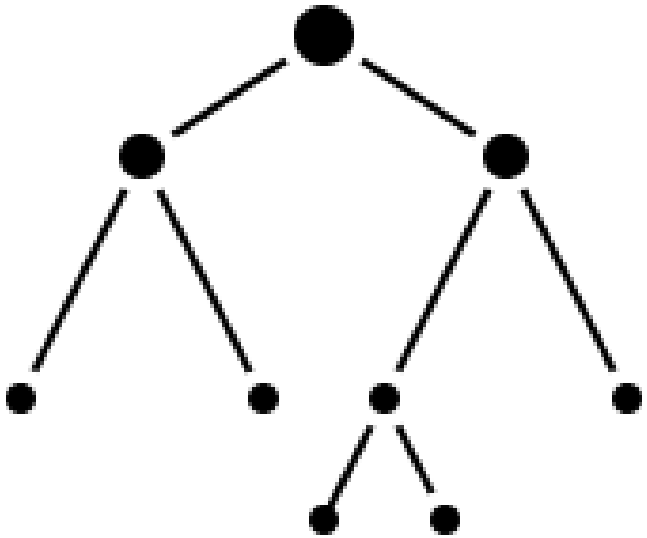
Noi vogliamo una stringa, ma l'oracolo ci dà solo sì e no! -> `divide et impera`.

Chiediamo un carattere alla volta, testando con tutto l'alfabeto: `la prima lettera è [A-Z]?`.

Dopo qualche richiesta avremo il primo carattere, e procediamo così fino alla fine.

Si può fare con meno richeste?

Lasciato come esercizio al lettore => basi di algoritmica



Una challenge ad oracolo

Familiarizzate con le challenge ad oracolo risolvendo questa !

E se l’output non c’è?

And now what?

A volte il server non ha bisogno di mandarci nulla indietro nel suo uso inteso (es. aggiornare il proprio username). Come facciamo?

Forti del fatto che la giusta combinazione di 0 e 1 può rappresentare tutto il rappresentabile, non ci rimane che trovare un modo nuovo di ricavarli.

In diversi dialetti di **SQL** abbiamo delle simpatiche funzioni, che possiamo usare in congiunzione ad `IF` per ottenere oracoli:

- `SLEEP`: blocca il DB per x secondi -> se la risposta del server è lenta, abbiamo un 1; altrimenti 0.
- Funzioni di read di risorse: alcuni server **SQL** permettono la lettura di risorse, sia locali che remote (protocolli HTTP o FTP) -> secondo voi come derivo un oracolo?

Se siete coraggiosi provate questa challenge!

⚠ WARNING ⚠

⚠️ WARNING ⚠️

Non usate tool già fatti come `sqlmap`; non imparerete molto e di sicuro non sono furbi quanto voi (**aka** non funzionano).