Introduzione alla cyberminaccia Iniezione SQL

L'iniezione SQL rappresenta una delle minacce web più diffuse. E' anche una delle minacce con maggiori conseguenze sui dati del nostro database.

L'iniezione SQL può avvenire a causa di un codice server (ASP.NET, Java, PHP e altri) che non **sanitizza** a sufficienza l'input dell'utente, come quello che avviene quando un utente inserisce i dati di login, o in modulo di contatto, ad esempio.

Attraverso un attacco SQL, un cyberavversario può:

- Accedere ad **informazioni confidenziali**, siano queste pagine web protette da password o altri dati presenti nel database
- Effettuare un **login al sistema senza credenziali**, come utente normale o come amministratore
- **Modificare** o anche **cancellare** alcune tavole del database o interi database
- Inserire nel database **codice malevolo** per il controllo del server da remoto

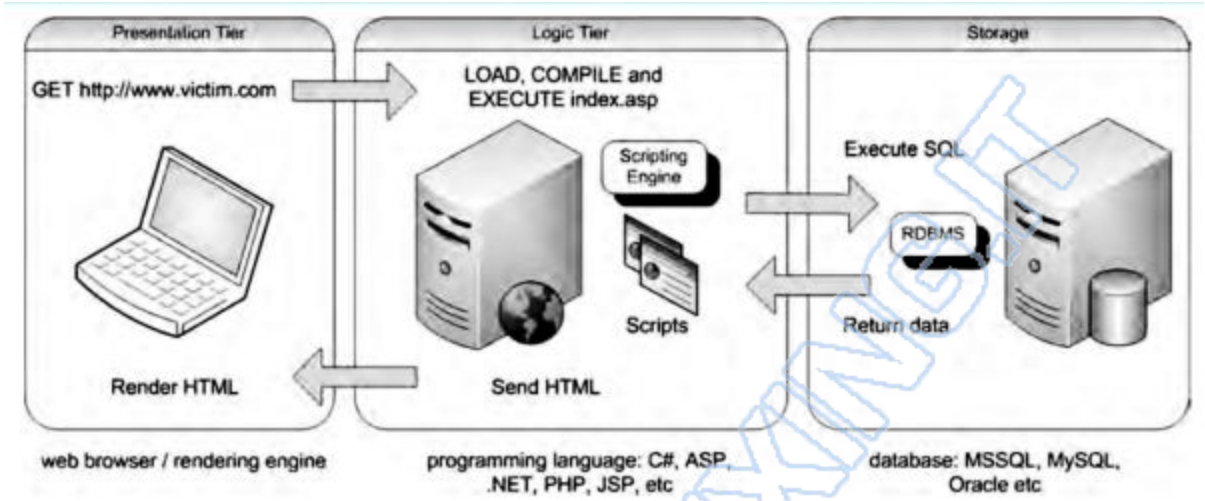
L'iniezione SQL è una delle vulnerabilità note da più tempo, ma anche una delle più temute e più complesse da controllare.

Per risolvere un'iniezione SQL, è necessario effettuare una procedura di pentesting:

- Testare errori del database da iniezioni SQL controllate
- Audit del codice server
- Fixing delle query tramite sanitizzazione dell'input utente
- Testing di verifica

Dettagli tecnici

Le applicazioni Web di oggi possono essere molto sofisticate e tecnicamente complesse. Spaziano dai siti web dinamici ai portali, dagli e-commerce alle extranet B2B. Alla base di queste applicazioni c'è spesso un'architettura illustrata qui di seguito:



Il browser web si interfaccia con un application server in grado di gestire la richiesta HTTP, estrarre le informazioni dal database e generare la pagina HTML per il client.

Attraverso la richiesta HTTP il client può trovare l'informazione che cerca. Prendiamo ad esempio una classica richiesta web,

https://www.sito.it?pag=10

Attraverso questa richiesta, il client vuole ottenere un pagina del sito, in particolare la pagina con ID=10

Il server interno estrae il parametro e formula una query verso il database, che può assumere la forma seguente:

SELECT titolo, testo_documento FROM pagine_articoli WHERE id = 10

Una volta ritrovato il testo dal database, il server procede all'impaginazione e generazione della pagina html.

Richiesta di un cyberavversario

Immaginiamo che il nostro sito contenga delle risorse confidenziali, accessibili solo attraverso una password.

La richiesta potrebbe avere questa forma:

http://www.sito.it?pagina_cliente=5&passwd=123456

In questo caso la richiesta verso il database sarà la seguente:
SELECT documento from documenti_clienti WHERE id=5 and password=123456

Il cyberavversario potrebbe essere interessato a recuperare il documento con id=5. Non conoscendo la password, potrebbe inserire una sequenza di caratteri per modificare la query database e quindi ovviare al controllo della password.

SELECT documento from documenti_clienti WHERE id=5 and password=' or …

Questo esempio è solo uno dei possibili accessi che un cyberavversario può effettuare per compromettere i dati e il sistema informatico di un target.

Il linguaggio SQL per le query database è un linguaggio intrinsecamente insicuro, capace di fornire l'accesso privilegiato a personale non autorizzato.

Per evitare le iniezioni sql sono necessari gli step qui di seguito:

- Analisi query SQL e analisi motore sql usato dal database
- Riscrittura delle query con sanitizzazione dell'input utente (filtraggio dei caratteri e analisi sintattica)
- Testing