S11/L5

BONUS 2- Analisi di un Attacco SQL Injection

Strumento di Analisi: Wireshark

Target: Database MySQL su server Ubuntu

---

1. Introduzione e Identificazione degli Attori

L'obiettivo di questa analisi è ricostruire la catena di un attacco SQL Injection tramite l'esame del traffico di rete catturato nel file `SQL_Lab.pcap`. L'attacco mira a manipolare le query di un modulo web per estrarre informazioni riservate dal database sottostante.

Dall'analisi dei flussi TCP/HTTP, sono stati identificati i seguenti attori:

- **Indirizzo IP Attaccante:** `10.0.2.4` (Sorgente delle richieste malevole).
- **Indirizzo IP Vittima:** `10.0.2.15` (Server Web/Database che elabora le query).

---

2. Ricognizione e Identificazione del Sistema

Durante la fase intermedia dell'attacco, l'aggressore ha inviato query specifiche per determinare l'ambiente tecnologico del bersaglio. Analizzando la risposta del server (flusso HTTP relativo al pacchetto 22), è stato possibile identificare:

- **Versione del Database:** `5.7.12-0ubuntu1.1`.
- **Sistema Operativo:** Distribuzione Linux Ubuntu.

---

3. Analisi della Struttura delle Tabelle

Per procedere all'infiltrazione dei dati, l'attaccante ha utilizzato il comando:

```
1' OR 1=1 UNION SELECT null, column_name FROM
INFORMATION_SCHEMA.columns WHERE table_name='users'#
```

Scopo del comando: L'aggiunta della clausola `WHERE table_name='users'` è una tecnica di raffinamento. Inizialmente, una query generica sull'intero schema produrrebbe un output eccessivamente vasto e rumoroso. Filtrando specificamente per la tabella `users`, l'attaccante ottiene esclusivamente i nomi delle colonne (come *user* e *password*), facilitando l'individuazione dei campi esatti da cui estrarre le credenziali.

---

4. Esfiltrazione dei Dati (Il "Bottino")

L'analisi si conclude con l'estrazione riuscita delle credenziali utente. Dall'output HTTP analizzato, è emerso quanto segue:

- **ID Utente Identificato:** `1337`.
- **Hash della Password (MD5):** `8d3533d75ae2c3966d7e0d4fcc69216b`.
- **Password in chiaro:** `123456`.

Nota: La decodifica dell'hash tramite tecniche di cracking (es. database di lookup come CrackStation) rivela l'utilizzo di una password estremamente debole, che rende inefficace la protezione fornita dall'hashing.

---

5. Riflessioni Finali e Prevenzione

Rischi principali delle piattaforme SQL

L'utilizzo improprio del linguaggio SQL espone le piattaforme a rischi critici di **violazione della riservatezza** (esfiltrazione di dati sensibili), **compromissione dell'integrità** (modifica non autorizzata dei dati) e **aggiramento dell'autenticazione** (accesso ad aree riservate senza credenziali valide). Un attacco riuscito può portare al controllo completo del database e delle informazioni degli utenti.

Metodi di Prevenzione

Per mitigare la minaccia di SQL Injection, è fondamentale adottare le seguenti misure di sicurezza:

1. **Query Parametrizzate (Prepared Statements):** Assicurano che l'input dell'utente sia trattato esclusivamente come dato e mai come parte del codice eseguibile della query.
2. **Validazione e Sanitizzazione dell'Input:** Implementare filtri che accettino solo dati nel formato previsto (es. solo numeri per i campi ID), scartando caratteri speciali come apici (`'`) o parole chiave SQL.
3. **Principio del Minimo Privilegio:** Configurare l'account del database utilizzato dall'applicazione web con i permessi minimi necessari, impedendo operazioni amministrative come l'eliminazione di tabelle (`DROP`)