LB 183 Dokumentation

# Handlungsziel 1:

## Aktuelle Bedrohungen erkennen und erläutern

Um die aktuellen Bedrohungen erkenne zu können, kann man sich an die OWASP Top 10 wenden. OWASP aktualisiert ihre Bedrohungsliste alle 4 Jahre was sie sehr aktuell und nützlich macht. Auf Ihrer Seite findet man folgende Abbildung:



OWASP überprüft Seiten mithilfe von anderen Leuten und wertet die Daten dann aus. Was noch erwähnenswert ist, ist das OWASP mit der Datensammlung für die OWASP Top 10 schon begonnen hat. Bis Dezember 2024 werden Daten gesammelt.

## Informationen beschaffen

Um sich vor möglichen Sicherheitslücken schützen zu können, muss man sein Projekt gründlich geplant haben. Hier kann man sich wieder an die OWASP Top 10 orientieren.

OWASP bietet zu allen Sicherheitslücken in ihrer Liste eine Übersicht, Beschreibung, Massnahmen und sogar Beispielsangriffe. Darunter findet man noch Referenzen und eine Liste von Common Weakness Enumeration (CWE) welche zu dieser Sicherheitslücke gehören.

OWASP bietet zusätzlich noch einen Cheatsheat an: <https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html>

Neben OWASP existieren noch viele weitere Seiten, welche Informationen zur Erkennung und Bekämpfung von Sicherheitslücken bieten.

## Auswirkungen darstellen

Um die möglichen Auswirkungen anzuzeigen, nehme ich SQL injection als Beispiel.   
(Diese Sicherheitslücke werde ich später in der Insecure App auch beheben)

### Schutzziele

Die Schutzziele sind in drei Kategorien unterteil: Vertraulichkeit, Integrität und Verfügbarkeit.

Beurteilen wir, wie stark die Schutzziele von einer SQL injection betroffen wären. 0 steht für gar nicht betroffen, 1 steht für teilweise betroffen und 2 steht für kritisch betroffen.

|  |  |  |  |
| --- | --- | --- | --- |
| Szenario | Vertraulichkeit | Integrität (Daten) | Verfügbarkeit |
| Angreifer gibt Liste von Produkten aus | 0 | 0 | 0 |
| Angreifer gibt Liste von User aus | 2 | 0 | 0 |
| Angreifer meldet sich als Admin an | 1 (2) | 2 | 1 |
| Angreifer löscht Tabellen oder sogar DB | 0 | 2 | 2 |

Angreifer gibt Liste von Produkten aus:

* Vertraulichkeit: Solange die Produkte keine sensiblen Informationen beinhalten, ist die Vertraulichkeit nicht betroffen.
* Integrität: Der Angreifer weiss das SQL injections möglich sind, was die Integrität der Daten in Frage stellt.
* Verfügbarkeit: Die Verfügbarkeit der Produkte ist (noch) nicht betroffen

Zwar werden alle Schutzziele als null gewertet, doch der Angreifer weiss nun, das SQL injections möglich sind und er wird dies höchstwahrscheinlich ausnutzen.

Angreifer gibt Liste von User aus:

* Vertraulichkeit: Alle Informationen zum Benutzer wurden preisgelegt, die Vertraulichkeit ist stark betroffen.
* Integrität: Die Daten wurden (noch) nicht verändert, die Gefahr besteht trotzdem.
* Verfügbarkeit: Die Benutzer sind weiterhin verfügbar.

Angreifer meldet sich als Admin an:

* Vertraulichkeit: Je nach Recht des Admins ist die Vertraulichkeit leicht oder stark betroffen.
* Integrität: Als Admin kann der Angreifer höchstwahrscheinlich Änderungen vornehmen und somit daten bearbeiten und löschen.
* Verfügbarkeit: Sollte der Admin Daten löschen können, so ist die Verfügbarkeit betroffen

Angreifer löscht Tabellen oder sogar DB:

* Vertraulichkeit: Der Angreifer löscht die Tabellen, somit gehen die Daten verloren.
* Integrität: Die Daten werden gelöscht, die Integrität ist stark betroffen
* Verfügbarkeit: Die Daten oder vielleicht sogar die Datenbank wird gelöscht, die Verfügbarkeit ist stark betroffen.

Zusätzlich gibt es noch sekundäre Schutzziele. Ich stelle eines der 4 Szenarios dar.

Angreifer meldet sich als Admin an

* Authentizität: Nicht mehr garantiert. Bei einem Admin könnte es sich um einen Angreifer handeln, der sich als Admin ausgibt.
* Nachweisbarkeit: Sind Logging und Triggers implementiert, kann nachgewiesen werden, welcher Admin kompromittiert wurde. Ansonsten nicht betroffen.
* Nicht-Abstreitbarkeit: Sind Logging und Trigger korrekt implementiert, können Operationen nicht abgestritten werden.
* Anonymität: In diesem Fall nicht relevant. (Person, welche Handlungen ausführt, soll nicht identifiziert werden können)

### Risiko

Bewerten wir nun das Risiko einer SQL injection   
(Diese Tabelle habe ich von PR\_183\_Grundbegriffe kopiert)

Injection befindet sich auf den Top 3 (OWASP Top 10), was die Eintrittswahrscheinlichkeit sehr hoch macht.

Der Schaden kann verheerend sein (Ausgabe von persönlichen Daten, CRUD-Operationen)

In diesem Fall soll man das Risiko vermeiden

Niedrig Hoch

Schaden (Impact)

Niedrig Hoch

Eintrittswahrscheinlichkeit  
(Likelihood)

Risiko reduzieren

Risiko reduzieren

Risiko vermeiden

Risiko akzeptieren

## Angriffsfläche

Die Angriffsfläche bezeichnet die Punkte, welcher ein Angreifer ausnutzen kann, um in das System einzudringen. Diese Fläche möchte man so klein wie möglich halten. In Bezug auf Injections müsste man nur so viele User eingaben ermöglichen wie nur nötig. Oder Usereingaben mit vordefinierten Antworten ersetzen.

# Handlungsziel 2

Ich werde die Sicherheitslücke «SQL-Injections» genauer betrachten. Ich werde zeigen, wie man die Ursachen erkennen kann und welche Gegenmassnahmen man vornehmen kann.

## Sicherheitslücke und deren Ursache erkennen

Wie schon erwähnt, werde ich SQL-Injection behandeln. Doch wie kann meine diese Sicherheitslücke erkennen?

Eine Methode ist, selbst probieren. Bei der Insecure app muss man sich mit einem Benutzer anmelden. Testen wir es aus.