Dokumentation

Inhalt

[Die Applikation 3](#_Toc501958242)

[Sicherheitsanalyse 4](#_Toc501958243)

[Lösungen 4](#_Toc501958244)

# Die Applikation

Unsere Applikation

# Sicherheitsanalyse

|  |  |  |  |
| --- | --- | --- | --- |
| **Schwachstelle** | **Gefahr** | **Klassifizierung** | **Gegenmassnahme** |
| Passwort in DB | Passwort wurde nicht gehasht:  Leichte Beute für Hacker, welche in die DB eindringen | Kann gravierende Folgen haben | Passworthash mit Salt erzeugen. |
| Login | SQL-Injection möglich:  Benutzer kann über Passwortfeld sich selbst zum Admin machen, Passwörter von anderen Benutzern ändern und Benutzer löschen. | Kann teilweise gravierende Folgen haben | Prepared Statements  Serverseitige Feldüberprüfung vor dem Ausführen des SQL-Syntax |
| Suche | SQL-Injection möglich:  Benutzer kann über Suchfeld sich selbst zum Admin machen, Passwörter von anderen Benutzern ändern und Benutzer löschen. | Kann teilweise gravierende Folgen haben | Prepared Statements  Serverseitige Feldüberprüfung vor dem Ausführen des SQL-Syntax |
| Edit | SQL-Injection und XSS möglich:  Kann über die URL, sowie über die Eingabefelder schadhaften SQL-Code einschleusen.  Kann in den Eingabefeldern schadhaften JavaScript-Code einschleusen, welcher dann auch in der DB gespeichert wird. | Kann gravierende Folgen haben |  |
| Keine Serverseitige Feldüberprüfung | Somit kann auch schadhafter Code in die Datenbank kommen:  XSS im Edit speichert den schadhaften Code in die Datenbank und führt diesem immer beim Neuladen der Seite aus. | Kann gravierende Folgen haben |  |
| Sensible Eingabefelder nicht genug gesichert. | Passwortfeld wird im Klartext angezeigt. | Kann teilweise gravierende Folgen haben |  |

# Lösungen