Inhaltsverzeichnis

[Softwareanalyse und gefundene Schwachstellen 2](#_Toc503797716)

[Sicherheitsrelevante Schwachstellen 2](#_Toc503797717)

[Daten 2](#_Toc503797718)

[Logik 2](#_Toc503797719)

[Penetrationtests 2](#_Toc503797720)

[Konfiguration 2](#_Toc503797721)

[Weitere Mängel 2](#_Toc503797722)

[Verbesserungen 2](#_Toc503797723)

[Struktur 2](#_Toc503797724)

[Sicherheitsrelevante Mängel 3](#_Toc503797725)

[Weak Typing 3](#_Toc503797726)

[URL Routing 3](#_Toc503797727)

# Softwareanalyse und gefundene Schwachstellen

## Sicherheitsrelevante Schwachstellen

### Daten

#### Login

- MD5

- Ordnerstruktur

Auf der Datenbank werden die Logindaten abgespeichert. In dem SQL-Script schlosslauf.sql ist neben dem CREATE TABLE Befehl auch ein INSERT mit Standardlogindaten zu finden. Dies ist ein grosses Sicherheitsrisiko. Nun kann man argumentieren, dass das Passwort ja verschlüsselt ist. Das Problem dabei ist jedoch, dass die Verschlüsselungsmethode MD5 angewandt wurde. Mittels einer Google-Suche («md5 <Hash aus Tabelle>») fanden wir bereits auf dem obersten Link das Passwort in Klartext und konnten uns erfolgreich einloggen.

### Logik

login: MD5 🡪 Rainbowtable

beliebig oft mit gleichem usernamen einloggen 🡪 Brute Force

noch keine Registrierung möglich, nur sekretärin mit schwachem login

### Konfiguration

Ordnerstruktur nicht anzeigen

File kann einfach angezeigt werden, datenbank kann downloaded werden 🡪 inklusive logindaten!!!

PHP-Version 5.6

## Weitere Mängel

Errorseite nicht einfach fail, sondern besser login mit usernamen

login anzeige fluchtlinien

# Verbesserungen

## Struktur

* Resources Ordner hinzugefügt um eine bessere Projektstruktur zu erreichen

## Sicherheitsrelevante Mängel

### Weak Typing

* Weak typing Schwäche von PHP umgangen durch die Verwendung von === anstatt ==, wodurch bei Vergleichen kein implizierter Cast durchgeführt wird.
* Yoda Expression verwendet, um bei Vergleichen zu verhindern, dass Variablen den Wert zugewiesen bekommen was immer als „true“ ausgewertet wird, wenn ein Programmierfehler vorhanden ist.

### URL Routing

Durch die nicht vorhandene .htaccess Konfiguration waren Source Code sowie auch weitere relevante Dateien in öffentlich zugreifbaren Ordnern abgelegt worauf Angreifer durch simples abändern der URL auf die Dateien zugreifen konnten.

Um dies zu verhindern haben wir eine .htaccess Datei hinzugefügt (<https://gist.github.com/sepehr/3370984>) welche diese Schwäche umgeht indem der Zugriff auf diese Dateien verhindert wird

Reflexion

Ordnen nach OSWAB oder wie es heisst!

Penetrationtests

OWASP Zap durchführen und protokollieren