Verschlüsselung mit Java

Passwort Safe

Beschreibung

Dieses Dokument beinhaltet die Aufgabe ein Passwort Safe zu erstellen und dabei die Sicherheitsvorkehrungen, die im Unterricht besprochen wurden einzusetzen.

Inhaltsverzeichnis

1 Projekt Passwort-Safe

1.1 Passwort-Safe App mit Java-Verschlüsselung, Spring Boot und React Frontend

Für die Speicherung von Passwörtern soll ein Passwort-Safe als Web-Anwendung programmiert werden, wie Sie sie wahrscheinlich bereits nutzen, um Ihre Passwörter an einem sicheren Ort abzulegen. Solche Passwort-Manager oder Passwort-Safes gibt es mittlerweile viele.

Hier sind ein paar Beispiele:
https://keepass.info, https://www.ubs.com/ch/de/private/digital-banking/private/safe.html

Die besten Passwort-Manager 2020: https://www.software-experte.com/passwortmanager

Die Web-Anwendung soll mit einem Front- und Backend umgesetzt werden. Hierfür kommen React im Frontend und Spring-Boot im Backend zum Einsatz. Andere Frameworks müssen mit der Lehrperson besprochen und von ihr abgesegnet werden.

1.2 Funktionalität der Anwendung

Die Anwendung soll eine Liste von Webseiten und Programmen, sowie den dazugehörenden Zugangsdaten wie Benutzernamen und Passwörter verschlüsselt speichern und auf Verlangen zeigen. Die Daten sollen in der Ablage strukturiert abgespeichert werden können. Eine mögliche Struktur könnte beispielsweise die Felder wie URL, Benutzername, Passwort, Bemerkungen, E-Mail, etc. beinhalten und in Rubriken unterteilt werden.

Die Listen mit den Zugangsdaten sollen auch bearbeitet werden können. Dabei sollen Einträge gelöscht, neue hinzugefügt und bestehende verändert werden können.

Die gespeicherten Zugangsdaten dürfen auch in Rubriken unterteilt werden. Beispielsweise könnte eine Rubrik „Privates“ und eine „Geschäft“ heissen oder "Games", "Hobbies", etc.

Die Anwendung soll mit einem uns bekannten Java-Verschlüsselungsverfahren die gespeicherten Zugangsdaten verschlüsseln. Welches Verfahren Sie dabei einsetzen spielt dabei keine Rolle. Für diese Aufgabe ist Ihnen die Wahl des Verfahrens freigestellt. Wichtig sind selbstverständlich die Wahl des richtigen Verfahrens, die korrekte Anwendung und der Umgang mit dem/den Schlüssel(n).

1.3 Authentisierung und Autorisierung

Der Zugang erfolgt über ein zentrales Login Fenster im Frontend. Darüber kann sich ein Benutzer anmelden und sieht danach seine gespeicherten Zugangsdaten.

Beispiel Vorgehen:
- Nach erfolgter Authentisierung entschlüsselt die Anwendung die Zugangsdaten, lädt diese in den Speicher und hält diese bereit, um sie für die Anzeige auszuliefern.
- Die Benutzerinnen und Benutzer können fest im Programm/Datenbank programmiert werden. Auf eine Registrierung neuer Benutzer wird in dieser Aufgabe verzichtet.
- Wie gesagt, soll als Verschlüsselungsverfahren ein Symmetrisches- und/oder ein Asymmetrisches-Verfahren eingesetzt werden. Beispielsweise AES, Blowfish, DES, RSA, etc.

1.4 Datenbank oder Mockup

Sie dürfen selbst entscheiden, ob Sie die Datenbankschicht mit einer echten Datenbank oder mit einem Mockup realisieren möchten. Eine 2-Schichten-Architektur drängt sich allerdings in beiden Fällen auf.

1.5 Master Passwort ändern

- Sie können ein Fenster vorsehen, über welches ein Benutzer/-in sein Passwort ändern kann.

1.6 Sicherheitsrelevante Punkte

Die Anwendung soll sicher programmiert werden. Hierfür sollen die folgenden Punkte beachtet werden, sowie auch die nachstehend aufgelisteten Owasp Top Ten Risks.

Die folgende Liste zeigt eine nicht abschliessende Liste von sicherheitsrelevanten Punkten.

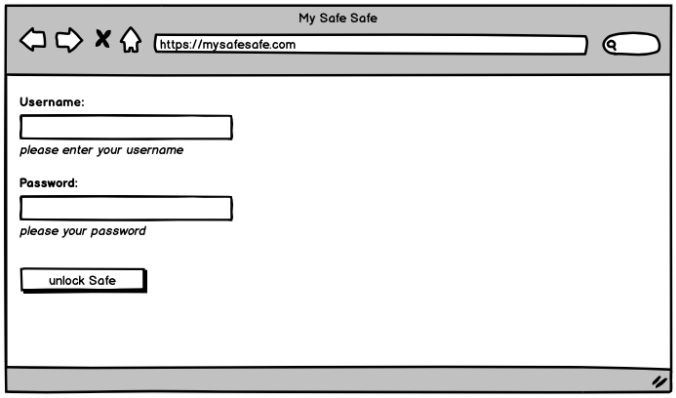
- Passwörter nicht im Klartext abspeichern
- Passwortfeld verwenden (*****)
- Passwortgüte überprüfen (Länge und Komplexität)
- Evtl. Pepper und/oder Salt anwenden
- Wahl des Verfahrens und der Algorithmen
- Handling der Zugangsdaten, Sicherheit Front- und Backend
- Benutzerinnen und Benutzer haben nur Zugriff auf die eigene Passwortliste.

Owasp Top Ten Risks

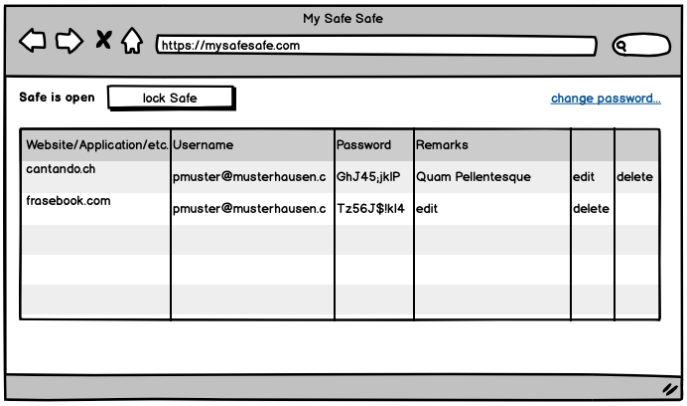
- Broken Access Control
- Injection
- Cross-Site-Scripting
- Cryptographic failures
- Insecure Design
- Identification and Authentication failures

1.7 Muster Bildschirm Skizzen / Vorschläge

Beispiel für eine mögliche Umsetzung des Einstiegsfensters für die Anmeldung und um den Safe zu öffnen.



Bei geöffnetem Safe könnte das Fenster dann wie folgt aussehen. Machen Sie sich weiterführende Gedanken dazu und machen Sie es besser.



2 Form und Rahmenbedingungen

Einzel oder Partnerarbeit

BBWRL

2.1.1 Zeitplan für das Projekt

11.12.2024	Projektstart, Def. Partner, Grundkonzept, Anwendung (Git-Repo-Link)	(2 Lektionen)
18.12.2024	Zwischenabgabe 1 Anwendung (Git) und Reflexion	(2 Lektionen)
08.01.2025	Zwischenabgabe 2 Anwendung (Git) und Reflexion	(2 Lektionen)
15.01.2025	Endabgabe	(2 Lektionen)

2.2 Grundkonzept

Schreiben Sie in einer Datei mit dem Namen **Grundkonzept.md** Ihr Grundkonzept auf und speichern Sie die Datei im Git-Repository. Wenn Ihre Anwendung auf einer SOA-Architektur basiert, speichern Sie die Datei im Repository des Backends.
Die beiden Partnerinnen und Partner machen sich einzeln und zusammen Gedanken zu den Sicherheitsaspekten der Applikation. Daraus sind ihre Überlegungen zum gewählten Grundkonzept und dem Aufbau der Applikation ersichtlich.

→ Grundkonzept.md

2.3 Softwareablage

Das Projekt ist mit und in einem Softwareverwaltungssystem zu führen (Git). Die beiden Partnerinnen und Partner, wie auch die Lehrperson haben Zugriff auf das Repository.
Die Entwicklung ist von beiden Partnerinnen und Partner durchzuführen. Diese checken ihren Beitrag zur Erweiterung der Applikation regelmässig ein. Die Aufgaben sind so zu planen, dass beide Partner am Projekt arbeiten können. Eine komplexe Branch Struktur wird aber nicht erwartet.

→ Lehrperson einladen und Zugriffsrechte einstellen (BBWRL)

2.4 Technische Rahmenbedingungen

- Spring-Boot-Thymeleaf (Serverbasiertes Rendering) oder
- React Frontend und Spring-Boot-Backend.
- Owasp Top Ten Risks

→ im Grundkonzept aufführen

2.5 Dokumentation in der Form einer Reflexion

Die Dokumentation kann durchaus auch weiterführende Gedanken zu sicherheitsrelevanten Punkten beinhalten an welche die Gruppe gedacht hat, aber die Zeit für die Umsetzung nicht reichte.

Die Dokumentation ist als Markdown Datei im Git-Repository mit dem Namen **Dokumentation.md** zu führen. Die beiden Gruppenpartner bearbeiten diese gleichermassen.
Umfang: vergleichbar mit maximal zwei A4 Seiten.

→ Dokumentation.md

2.5.1 Was ist Markdown?

Markdown ist eine einfache Auszeichnungssprache, bei der, ähnlich wie bei dem bekannten Wiki-Text, Dokumente mit Markdown über spezielle Sonderzeichen formatiert werden können. Markdown kann mit jedem simplen Text-Editor oder mit speziellen Editoren erstellt werden.
https://de.wikipedia.org/wiki/Markdown
https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

2.6 Bewertungskriterien

2 Punkte	Form und Zusammenarbeit in der Gruppe
2 Punkte	Einhaltung der Termine
4 Punkte	Dokumentation und Reflexion
10 Punkte	Lock bzw. Verarbeitung und Sicherheit
10 Punkte	Unlock-Fester mit Struktur und Inhalt (Login)
4 Punkte	Benutzername und Passwort (Anmeldung) richtig verarbeitet
4 Punkte	Speicher und Verarbeitung der Zugangsdaten (Login)
2 Punkte	Passwortgüte überprüfen (Länge und Komplexität)
2 Punkte	Wahl und Einsatz des Verfahrens
4 Punkte	Sicherheitsaspekte der Applikation allgemein

44 Punkte Total