Technische Hochschule Rosenheim
Fachbereich Informatik
Prof. Dr. R. Hüttl

IT-Security

Übung 6

In dieser Übung betrachten wir das Problem Passwörter abzuspeichern. Neben der Diskussion über die verscheidenen Schritte zur Abspeicherung von Passwörtern implementieren wir zwei Varianten in Java.

Aufgabe 1: Awareness Training Teil 2

https://training.is-fox.de/fh-rosenheim-informatik

Wir machen die folgende Module von dem Awareness-Test aus Übung 1

- Passwörter
- Passwort Managent
- Multifaktor-Authentifizierung
- Sensible Informationen

Aufgabe 2: Fragen

Beantworten sie folgende Fragen:

- Was ist das Problem beim Abspeichern eines Passwortes im Plaintext?
- Welche Probleme gibt es beim verschlüsselten Abspeichern des Passwortes?
- Welche Vorteile bietet das Abspeichern des Passwortes mittels einer Hashfunktion? Welche Attacken machen diese Verfahren unsicher?
- Wie funktionieren Hash Look Up Tables and Dictionary Attacks?
- Wie funktionieren Rainbowtables?
- Wie funktionieren „gesalzene“ Passwörter? Warum bieten sie zusätzliche Sicherheit?
- Wie kann man Passwörter die mit einem Hash und einem Salt abgespeichert werden weiter verbessern?

Hinweise finden sie unter https://happycoding.io/tutorials/java-server/secure-password-storage

Aufgabe 3: Sichere Abspeicherung von Passwörter mittels Hashing

Gegeben ist eine Klasse ***PasswordHashing*** mit einer Reihe von statischen Methoden die für das Hashing von Passwörter erforderlich sind. Damit kann man z.B. Benutzerpasswörter sicher abspeichern und beim Login wieder verifizieren.

Ergänzen sie Klasse an den markierten Stellen (TODO) um Passwörter in zwei Verfahren zu speichern und verifizieren

- Passwort Hash mit SHA512 und salt
- Passwort Hash mit PBKDF2 und salt

Schreiben sie einen Unit-Test der für Passwörter mit beiden Verfahren ein Hash berechnet und den berechneten Hash mit dem Passwort verifiziert.

Hinweise:

- Für Passwort-Hashing mit SHA512 können sie die Klasse ***java.security.MessageDigest*** verwenden.
- Für Passwort-Hashing mit PBKDF2 können sie die Klassen ***javax.crypto.spec.PBEKeySpec*** und ***javax.crypto.SecretKeyFactory*** verwenden.