Technische Projektdokumentation „EncryptedChat“ – Projekt Modul 151

Von Pedro de Souza Winkler und Paul Stonjek

Registrierung und Login

Über die Startseite der Anwendung (http://localhost/EncryptedChat/Site/index.php) sind Registrierungs- und Login-Formular verfügbar. Im Registrierungsformular müssen neben einem gewünschten Username und Passwort auch eine gültige Email-Adresse, Vorname und Nachname angegeben werden, sowie ein optionales Profilbild. Im Login kann man sich unter Angabe des Username und Passwort einloggen. Nach dem Einloggen werden oben rechts das Profilphoto und der Username angezeigt.

User Input Validation

Alle User Inputs bei der Registrierung werden clientseitig geprüft, auf Länge, Format (bei Email), und enthaltene Zeichen bei Username und Passwort. Zur Prüfung werden HTML5-interne „Input“-Attribute verwendet. Bei ungenügender oder ungültiger Eingabe werden entsprechende Meldungen ausgegeben.

Zudem werden alle Eingaben serverseitig geprüft, bevor sie in die Datenbank eingefügt werden. Bei ungültiger oder ungenügender Eingabe werden detailliertere Fehlermeldungen an den Benutzer ausgegeben.

Passwort-Sicherheit

Passwörter der Nutzer werden nicht im Klartext in der Tabelle gespeichert, sondern als salted Hashes abgelegt. Zur Überprüfung der Passworter wird password_verify() eingesetzt.

Speicherung und Abruf der Daten

Zur Interaktion mit der Datenbank wurde das Profil „WebServerRoot“ angelegt, welches nur über die Privilegien SELECT, INSERT und UPDATE verfügt und nur auf die EncryptedChat- Datenbank zugreifen darf.

Alle Nutzerdaten werden als Varchars entsprechender Größe gespeichert. Die Größen sind so gewählt, dass die Vorgaben in der Input-Validierung eingehalten werden. Für das Passwort sind 255 Zeichen gewählt, um auch für zukünftige Hashprozesse kompatibel zu sein.

Alle Nutzereingaben werden vor der Eingabe in die Datenbank zum Schutz vor Script-Injections auf HTML special chars geprüft, und zum Schutz vor SQL-Injections werden nur Prepared Statements zum Umgang mit Nutzereingaben verwendet.

Session-Sicherheit

Mit jedem Aufruf der Seite, und damit auch nach jeder Nutzerdaten-Eingabe, wird die Session-ID des eingeloggten Benutzers aktualisiert, um Session-Hijacking und Session Fixation vorzubeugen.

Änderung von Passwort und Username

Eingeloggte Nutzer können mit einem Klick auf Ihren Username oben rechts ihren Username oder ihr Passwort ändern. Hierbei kommen dieselben Methoden und Sicherheitsvorkehrungen wie bei Registrierung und Login zum Einsatz, wie Client- und Serverseitige Validierung, Session-Regeneration, Passwort-Hashing, etc.

Messaging

Nach dem Einloggen kann ein Nutzer auf die „Chat“-Seite zugreifen, welche über einen Link oben rechts auf der Startseite verfügbar ist. Im Chat sieht der Benutzer auf der linken Seite eine Auflistung aller auf dem Server registrierten Benutzer, und kann einen aus der Liste auswählen, um die Messaging-Funktion zu aktivieren.

Nach der Auswahl werden alle Nachrichten zwischen dem eingeloggten und dem ausgewählten Benutzer angezeigt. Der Benutzer kann die Liste raufscrollen, um auch ältere Nachrichten einzusehen. Vom Nutzer verfasste Nachrichten verfügen über „Edit“- und „Delete“-Buttons, mit denen die Nachrichten entsprechend bearbeitet oder gelöscht werden können.

Über die das Eingabefeld unten kann der Benutzer eigene Nachrichten an den gewählten Gesprächspartner verfassen und versenden.

Senden und Lesen

Nach Auswahl eines Gesprächspartners, sowie alle 5 Sekunden und nach dem Senden, Editieren oder Löschen einer Nachricht, werden die angezeigten Nachrichten aktualisiert. Hierbei wird die Session-ID des eingeloggten Benutzers verwendet, um auf serverseite die user-ID festzustellen, und aus den gespeicherten Nachrichten werden alle Nachrichten zwischen dem eingeloggten und dem ausgewählten Benutzer geladen.

Die Abfrage läuft asynchron über AJAX-calls, wodurch die Seite nicht neu geladen werden muss.

Bearbeiten und Löschen

Bearbeiten und Löschen von Nachrichten funktioniert über Buttons, welche an den Nachrichtenboxen angebracht sind. Über einen AJAX-Call wird die API aufgerufen, welche die Nachricht als gelöscht markiert für DELETE, oder die neue Eingabe validiert und einfügt für EDIT. Wie bei der Registrierung werden auch hier HTML special chars ersetzt und alle Datenbankeingaben über Prepared Statements gehandhabt.

Speichern der Nachrichten

Alle Nachrichten werden in einer eigenen Tabelle gespeichert. Für jede Nachricht wird eine ID, der Inhalt, die IDs der beteiligten Nutzer, und das Sendedatum gespeichert, sowie ein Bit, das markiert, ob eine Nachricht editiert oder gelöscht wurde.

Der Inhalt jeder Nachricht wird als Text in latin1_swedish_ci-Collation gespeichert. Sämtliche HTML special chars werden beim Einfügen in die Datenbank entschärft, um Script Injection zu verhindern.