hochschule coburg university of applied sciences

Fakultät
Elektrotechnik und Informatik

PROJEKTARBEIT

Snap2Pass

Verfasser:	Michael Koch, B.Sc. und Christian Schelter, B.Sc.
Fach:	Identitätsmanagement
Betreuender Dozent:	Prof. Dr. Thomas Wieland
Abgabetermin:	28.06.2012

Inhaltsverzeichnis Seite

1. Einführung

1.1. Themenstellung

Es soll ein Dienst für die Authentifizierung gemäß dem Snap2Pass-Konzept[1] erstellt werden. Der Nutzer bekommt bei der Anmeldung an eine Website einen QR-Code angezeigt. Diesen liest er mit seinem Mobilgerät aus und meldet sich mit dem Mobilgerät beim Server an. Diese Anmeldung ist dann auch für den Browser gültig, so dass dort ein Zugang ohne Benutzerdateneingabe gewährt wird.

1.2. Beschreibung des Verfahrens

1.2.1. Traditionelle formularbasierte Anmeldung

Die Authentifizierung eines Benutzers gegenüber einer Webapplikation erfolgt meistens über ein Eingabeformular der jeweiligen Anwendung, in der der Benutzer seinen Benutzernamen und sein Passwort eintragen muss. Mit dem Abschicken des Formulars werden diese Daten normalerweise über eine HTTP-POST-Anfrage an eine Seite der Webapplikation übermittelt. Da HTTP von sich aus keine verschlüsselte Datenübertragung anbietet, ist es zwingend erforderlich, HTTPS einzusetzen. Die Webapplikation prüft nach irgendeinem Verfahren, ob die übermittelten Anmeldedaten gültig waren und merkt sich z.B. in einer Sitzungsvariablen, dass sich der Benutzer bereits angemeldet hat. Waren die Anmeldedaten ungültig, so wird der Zugriff verwehrt und der Benutzer gelangt wieder auf die Anmeldeseite oder auf eine Fehlerseite.
Auch das hier beschriebene Verfahren erlaubt eine lokale Formularanmeldung als Fallback-Lösung, falls der Benutzer über kein mobiles Endgerät verfügen sollte oder JavaScript im Browser deaktiviert sein sollte.

1.2.2. Snap2Pass

Der Grundgedanke der formularbasierten Authentifizierung gilt auch für den hier vorgestellten Snap2Pass-Ansatz, allerdings erfolgt eine Aufspaltung des Datenflusses in zwei getrennte Kommunikationskanäle. Es soll vermieden werden, dass der Benutzer an einem unsicheren PC direkt seine Anmeldedaten eingeben muss, um eine Webapplikation nutzen zu können, da diese Daten z.B. mit Hilfe von Keyloggern mitgeschnitten werden könnten. Deutlich sicherer wäre es, wenn man die Übermittlung der Anmeldedaten über ein vertrauenswürdiges eigenes Gerät (z.B. ein Smartphone oder ein Tablett) vornehmen könnte, jedoch die Applikation weiterhin am PC bedienen könnte. Snap2Pass verfolgt

[1] Beschreibung des Snap2Pass-Konzepts gemäß: Dodson et al.; Stanford University; Snap2Pass: Consumer-Friendly Challenge-Response Authentication with a Phone; http://prpl.stanford.edu/papers/soups10j.pdf; Stand: 20.06.2012

genau diesen Ansatz. Das Problem an dieser Stelle ist jedoch, eine Verbindung zwischen der lokalen PC Sitzung und der Sitzung zur Anmeldeseite am Handy herzustellen. Hierzu stellt die Webapplikation einen QR-Code bereit, der außer der Adresse der Webapplikation auch noch die Sitzungsnummer der lokalen PC Sitzung enthält. Der Benutzer fotografiert diesen QR-Code mit seinem mobilen Endgerät ab und gelangt über seinen mobilen Browser auf die Anmeldeseite der Webapplikation. Dort gibt er seine Anmeldetaten ein, welche über das mobile Netz des Internetproviders an die Webapplikation übermittelt werden. Auch hier erfolgt die Übermittlung über das HTTPS Protokoll. Anschließend prüft die Webapplikation die Gültigkeit der Anmeldedaten und vermerkt dies in der Sitzung. Während der kompletten Anmeldephase prüft die Anmeldeseite am PC über zyklische AJAX Requests, ob sich der Benutzer bereits angemeldet hat und leitet den Benutzer z.B. auf die Startseite weiter, sobald in der Sitzung vermerkt wurde, dass sich der Benutzer erfolgreich authentisiert hat. Abschließend wird die Identifikationsnummer der Sitzung neu generiert, sodass Hijacking Angriffe deutlich erschwert werden.

2. Beschreibung der Software

2.1. Aufbau

Die Software ist modular aufgebaut, das heißt, dass die Authentifizierungsmethode, das Verfahren zur Verschlüsselung der eingegebenen Passwörter, sowie die abschließende Aktion beliebig durch den Betreiber ausgetauscht werden können. Außerdem ist es möglich, jeweils eigene neue Module zu entwickeln.

2.1.1. Password-Modul

Jedes Password-Modul stellt eine Möglichkeit bereit, einen Algorithmus auf das eingegebene Passwort anzuwenden. Beispielsweise ist es möglich Passwörter im Klartext, oder als Hash weiter zu verarbeiten. Zudem bietet dieses Konzept die Möglichkeit, neue Module zu programmieren, welche andere Algorithmen verwenden, beispielsweise um in einem Hash-Verfahren ein Salt einzufügen. In der Konfigurationsdatei besteht die Möglichkeit, das zu verwendende Password-Modul auszutauschen.

Bisher implementierte Password-Module:

- **Empty**
  Anstelle des Passworts wird eine leere Zeichenfolge weitergegeben.
- **Plain**
  Das Passwort wird 1:1 weitergegeben.
- **MD5**
  Es wird nur der MD5-Hash des Passworts weitergegeben.
- **SHA1**
  Es wird nur der SHA1-Hash des Passworts weitergegeben.

2.1.2. Auth-Module

Jedes Auth-Modul stellt eine Validierungsmöglichkeit für die eingegebenen Logindaten bereit. Das zu verwendende Modul kann in der Konfigurationsdatei ausgewählt werden.
Zudem ist es möglich, neue Auth-Module zu entwickeln, um zum Beispiel Vergleiche auf Basis von Daten aus einer Datenbank, oder eine Authentifizierung über einen RADIUS-Server zu realisieren.
Bisher implementierte Auth-Module:

- **Debug**
  Es findet ein Vergleich auf konstante fest einprogrammierte Werte statt.
- **Bypass**
  Die Anmeldetaten werden 1:1 weitergegeben, allerdings muss sowohl ein Benutzername, als auch ein Passwort angegeben werden.
- **LDAP**
  Zur Überprüfung der Anmeldetaten wird ein LDAP Server herangezogen, dessen Adresse in einer gesonderten Konfigurationsdatei angegeben werden kann.

2.1.3. Action-Module

Das Action-Modul stellt die Abschlussaktion der Snap2Pass Authentifizierung dar. Hier kann für jede Aktion, die im Anschluss durchgeführt werden soll, beispielsweise die Anmeldung eines Benutzers in einer Datenbank oder die Protokollierung von Anmeldevorgängen, in einem neuen Modul realisiert werden. Die Auswahl des Moduls erfolgt, wie bei allen anderen Modulen, in der Konfigurationsdatei.

Bisher implementierte Action-Module:

- **Debug**
  Es werden am PC die Anmeldedaten angezeigt.
- **Redirect**
  Der Benutzer wird über ein Redirect zu einer anderen Seite weitergeleitet. Die Anmeldedaten werden typischerweise in der Sitzung abgelegt und somit anderen Seiten bereitgestellt.
- **Forward**
  Der Benutzer wird über ein Forward zu einer anderen Seite weitergeleitet. Die Anmeldedaten werden typischerweise über POST-Variablen zur gewünschten Seite übermittelt.

2.1.4. Konfiguration

Die Konfigurationsdatei enthält alle Einstellungen der Snap2Pass Anwendung. Hier ist es möglich, intern verwendete Variablennamen zu verändern und die zu verwendende Module auszuwählen.

2.1.5. Messages

Die Anmeldeseiten für PCs und mobile Endgeräte besitzen eine feste Grundstruktur, um u.a. eine korrekte Darstellung des QR-Codes zu garantieren. Sämtliche Beschriftungen von Elementen, sowie die Bereiche oben, unten und links der Authentifizierungsmaske, können beliebig mit HTML gefüllt werden.

2.2. Einsatzgebiete

Snap2Pass kann in der aktuellen Version nur für PHP-Anwendungen, welche auf demselben Server liegen, verwendet werden, da die Kommunikation ausschließlich über die von PHP bereitgestellte Session erfolgt.

2.3. Benötigte Software

Zum Betrieb von Snap2Pass ist PHP in Version 5 oder höher erforderlich. Eine Datenbank oder Schreibberechtigungen auf das Dateisystem sind nicht notwendig.
Auf der Client-Seite setzt Snap2Pass einen Browser, mit aktivierter JavaScript und Cookie Unterstützung voraus. Ist diese Voraussetzung nicht gegeben, kann sich der Benutzer trotzdem formularbasiert anmelden, da diese Methode weder auf JavaScript noch Cookies angewiesen ist. Für die Anmeldung über QR-Code wird außerdem ein Handy benötigt,

welches QR-Codes lesen und die darin kodierte URL in einem Browser darstellen kann. Auf der mobilen Seite wird kein JavaScript oder Cookies benötigt.

3. Probleme

Das Konzept ist anfällig für Browser-Plugins, die den QR-Code manipulieren und so einen Man-in-the-Middle-Angriff erlauben. Daher sollte der Benutzer stets die URL im mobilen Browser überprüfen.

HTTP allein ist immer unsicher, da keine Verschlüsselung bereitgestellt wird. Abhilfe schafft die zwingend notwendige Verwendung von HTTPS, jedoch schrecken Self-Signed Zertifikate Benutzer ab oder sie könnten vom Browser je nach Konfiguration grundsätzlich abgelehnt werden.

Wirkliche Sicherheit gäbe es nur bei beidseitiger Authentifizierung via TLS, was aber in der Praxis kaum möglich ist.

4. Weiterentwicklung

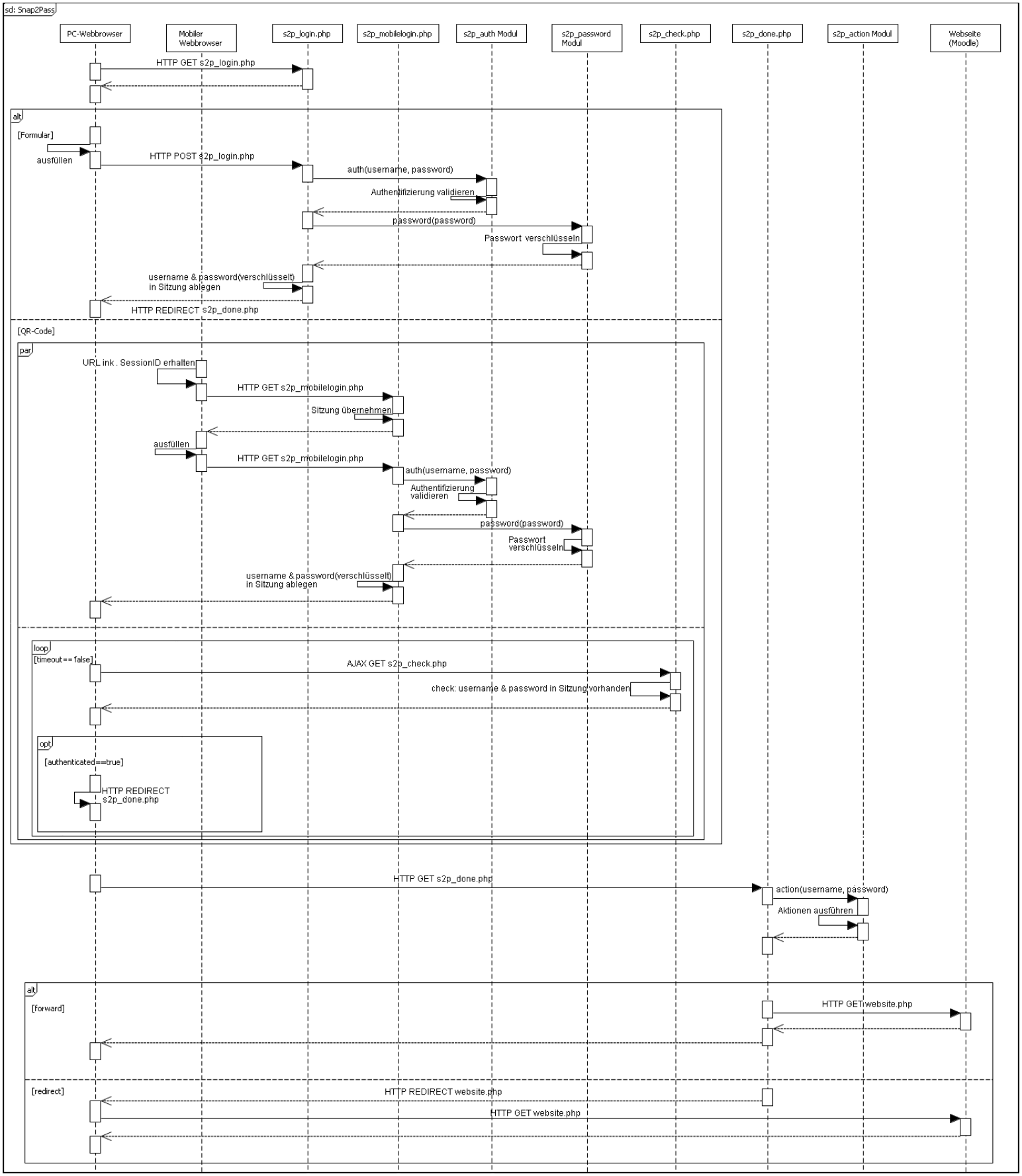
In der originalen Veröffentlichung des Verfahrens wurde Snap2Pass mit OpenID kombiniert, um eine OpenID-Authentifizierung mit den Mitteln von Snap2Pass zu realisieren. Die hier vorgestellte Softwarearchitektur sollte sich nahtlos in den Anmeldeprozess eines PHP-basierten OpenID-Providers (z.B. phpMyOpenID) einbinden lassen.

5. Fazit

Snap2Pass ist eine sichere Methode, um sich an einem nicht vertrauenswürdigen PC bei Webseiten anzumelden. Hierdurch wird während des Anmeldeprozesses das Password-Sniffing deutlich erschwert und die Verwendung eines unsicheren PCs deutlich sicherer. Leider bietet auch dieses Verfahren keinen kompletten Schutz, lediglich die beidseitige Authentifizierung mittels TLS würde die noch bestehenden Schwachstellen beseitigen und einen Man-in-the-Middle-Angriff unterbinden.
Voraussetzungen für die Nutzung von der hier vorgestellten Snap2Pass-Implementierung, sind zum einen der Besitz eines Handys, welches QR-Codes lesen und die darin enthaltene URL in einem Browser darstellen kann und zum anderen die Unterstützung von JavaScript im Browser des nicht vertrauenswürdigen PCs.

Anlagen

Anlage 1: Sequenzdiagramm



Ehrenwörtliche Erklärung

Wir versichern hiermit, dass wir unsere Projektarbeit mit dem Thema

Snap2Pass

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben.

Ort, Datum

Michael Koch, B.Sc.

Christian Schelter, B.Sc.