Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Authentication im Web

Weiterführende Themen zu Internet- und WWW-Technologien

11.07.2011, Kai Fabian

Inhalt

1. Begriffsabgrenzung

- Authentifizierung:
  - i. e. S.: Überprüfung/Nachweis der Identität der Gegenstelle
  - i. a. S.: Überprüfung/Nachweis einer Eigenschaft d. Gegenst.
  - Beispiele:
    - Login auf einer Homepage
      - Server überprüft, ob die Gegenstelle im Besitz spezifischen Wissens ist (Benutzername, Passwort)
    - Dienstausweis, bspw. Polizei
      - Polizist weist nach, im polizeilichen Staatsdienst zu stehen (und daraus folgend Befugnisse im Sinne des Polizeigesetzes ausüben zu dürfen, vgl. nächstes)

1. Begriffsabgrenzung

- Autorisierung:
  - Bedarf meist einer vorhergehenden Authentifizierung
  - Zuweisung und/oder Überprüfung von Berechtigungen
  - Beispiele:
    - EC-Kartenbezahlung (mit Unterschrift)
    - Amtliche Fahrerlaubnis („Führerschein“)

2. HTTP „Basic Authentication“ (RFC 2617)

Client	Server
GET /intern HTTP/1.1	
	HTTP/1.1 401 Unauthorized WWW-Authenticate: Basic realm="RealmName"
GET /intern HTTP/1.1 Authorization: Basic aHR0cDovL24zcmQua2FpZmaFiaWFuLmRlLw==	
	HTTP/1.1 200 OK ~~WWW-Authenticate: Basic realm="RealmName"~~

Realm: Dafür vorgesehen, den geschützten Bereich eindeutig zu identifizieren (Caching des Passworts im Browser)

Authorization-Token: Base 64-codierter String (Bennutzername „:“ Passwort)

Wiederholtes Senden des WWW-Authenticate-Headers unnötig

3. Single Sign-on

- Ziel von Single Sign-on:
  Nutzung von einer einzigen benutzerseitigen Authentifizierung gegenüber mehreren unabhängigen Softwaresystemen
- Meist Nutzung einer dritten Partei („Authentication Provider“)

- Vorteile:
  - Benutzer muss sich nur einmal authentifizieren
  - Phishing-Attacken werden erschwert
- Nachteile:
  - Der Authentication Provider kann Dritte als den Benutzer authentifizieren
  - Verfügbarkeit des Authentication Providers muss gewährleistet sein
  - Kein verbindlicher Standard für „Single Sign-Off“

3.1. Übersicht

- Weiter verbreitete Verfahren:
  - OpenID
    - Diverse SDKs von verschiedenen Anbietern (Open Source)
  - Facebook Connect (OAuth 2.0)
    - Facebook Connect JavaScript SDK

- Weniger weit verbreitete Verfahren:
  - Liberty Alliance Project (Projekt beendet)
  - Windows Live ID (ehem. Microsoft Wallet, Microsoft Passport, .NET Passport, Microsoft Passport Network, …)
    - Windows Live ID Web Authentication SDK (ASP.NET, Java, Perl, PHP, Python Ruby)

3.2. OpenID

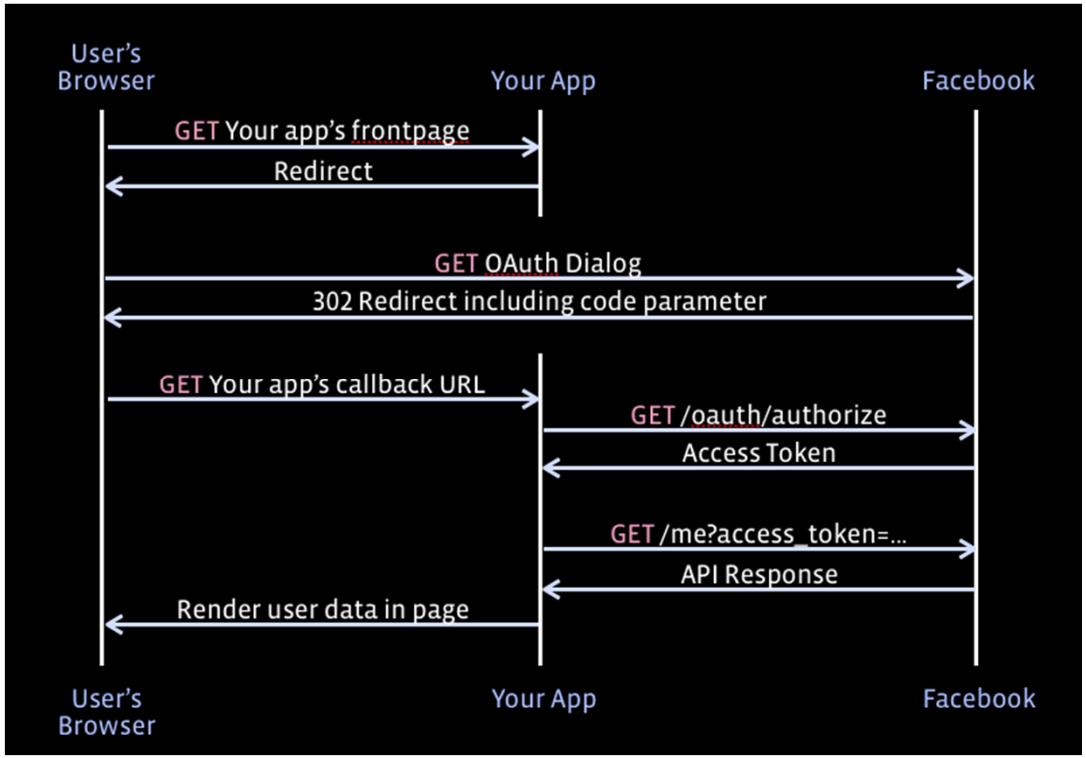
- Offener Standard für dezentralisierte Authentifizierung
- „URL based Identity“ (alternativ: XRI)
- Ablauf (OpenID 1.0):
  1. Benutzer gibt gegenüber der Gegenstelle seine OpenID an
  2. Gegenstelle lädt die Authentifizierungsseite herunter
     i. … und sucht nach der „Provider URL“ (<link>-Tag)
     ii. … und überprüft ggf. die Nutzung eines „Delegates“ (ebf. <link>-Tag)
     iii. … im Falle des „stateful logins“: Vereinbarung eines gemeinsamen Geheimnisses
  3. Benutzer wird an OpenID-Provider weitergeleitet
     i. … und prüft die Identität des Nutzers (authentifiziert ihn)
     ii. … und lässt danach den Nutzer die Gegenstelle autorisieren

3.2. OpenID

4. Der OpenID-Provider leitet den Benutzer zur Gegenstelle zurück
   a) „stateful“/„smart“: Die Gegenstelle prüft, ob der OpenID-Provider das gemeinsame Geheimnis geschickt hat
   b) „stateless“/„dumb“: Die Gegenstelle fragt eigenständig beim OpenID-Provider an, ob die übermittelte Authentifizierung korrekt ist
5. (optional:) Die Gegenstelle fordert Benutzerinformationen beim OpenID-Provider an

3.3. OAuth (am Beispiel von Facebook)

- Auch bekannt als ehem. „Facebook Connect“
- Unterstützt Client- und Serverseitigen Ablauf
- Verwendet OAuth 2.0
- Facebook-Spezifika:
  - „App“ muss registriert werden
  - „App“ erhält Zugang zu Facebook-Funkt. über „Graph API“, z. B. öffentliche Facebook-Profildaten



Bildquelle: https://developers.facebook.com/docs/authentication/

3.3. OAuth (am Beispiel von Facebook)

Hasso Plattner Institut

```php
<?php

function print_action($action) {
	printf("Es folgt nun: %s", $action);
}

print_action("DEMO");

?>
```

4. Janrain Engage

- Janrain (http://www.janrain.com/) ist kommerzieller Anbieter von Authentifizierungs-SDKs (u. A. für OpenID)
- Schwerpunkt auf Social Web-Systeme
- „Janrain Capture" sammelt Profildaten verbundener Social Network-Accounts
- „Janrain Engage": Auch als ‚Social Login' bezeichnet
  - Sign-on SDK für:
    - Facebook, Google, Twitter
    - Paypal, Yahoo, OpenID
    - Windows Live ID, mySpace, AOL
    - LinkedIn, Symantec PIP, myOpenID
  - Freie und kostenpflichtige Lizenzen

4. Janrain Engage

- Janrain Engage: ehemals „Janrain RPX“
- „Basic“-Lizenz (kostenlos)
  - ohne Support
  - Max. 2.500 verschiedene Benutzer/Jahr
  - Maximal 6 „Authentication Providers“
  - Sammeln von grundlegenden Profilinformationen
    - Name, Geschlecht, E-Mail-Adresse, Alter, Wohnort, …
  - Design kann nicht angepasst werden
  - „Social Sharing“: z. B. Hinterlassen von Pinnwand-Einträgen
  - Plugins für einige verbreitete Systeme (Wordpress, Drupal, …)
  - Maximal ein Administratorbenutzer

4. Janrain Engage

```php
<?php

function print_action_again($action) {
	printf("Es folgt nun erneut: %s", $action);
}

print_action_again("DEMO");

?>
```

5. Quellen

- Zu 1.:
  - http://de.wikipedia.org/wiki/Authentifizierung
  - http://de.wikipedia.org/wiki/Autorisierung
- Zu 2.:
  - http://tools.ietf.org/html/rfc2617
- Zu 3.:
  - https://en.wikipedia.org/wiki/Single_sign-on
  - http://openid.net/specs/openid-authentication-1_1.html
  - http://openid.net/specs/openid-authentication-2_0.html
  - http://developers.facebook.com/docs/guides/web/
  - http://developers.facebook.com/docs/authentication/
- Zu 4.:
  - http://www.janrain.com/products/engage

Fragen?

Vielen Dank für eure Aufmerksamkeit!