Web Single Sign-On: OpenID, Shibboleth, and friends

COSC412
Learning objectives

• Discuss risks from managing password databases

• Understand how web-based single sign on (SSO) systems use cryptography

• Describe the different participants in web SSO systems and how they interact

• Contrast the different features and limits of SSO systems
Motivation for Web-based SSO

• Avoid users authenticating multiple times
  • (Similar in motivation to Kerberos)

• … both within and across services
  • Contrast to Kerberos: independent organisations

• User convenience and security are coupled
  • A need for multiple passwords tends to lead to shared passwords or poor password choices…
# Passwords in the wild: Adobe-fail

(October 2013)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Count</th>
<th>Plaintext</th>
<th>Ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1911938</td>
<td>123456</td>
<td>EQ7fI6pT7i/Q=</td>
</tr>
<tr>
<td>2</td>
<td>446162</td>
<td>123456789</td>
<td>j9p+HwtWWT86aMjgZFLzYg==</td>
</tr>
<tr>
<td>3</td>
<td>345834</td>
<td>password</td>
<td>L8qbAD3jl3jioxG6CatHBw==</td>
</tr>
<tr>
<td>4</td>
<td>211659</td>
<td>adobe123</td>
<td>BB4e6X+b2xLioxG6CatHBw==</td>
</tr>
<tr>
<td>5</td>
<td>201580</td>
<td>12345678</td>
<td>j9p+HwtWWT/ioxG6CatHBw==</td>
</tr>
<tr>
<td>6</td>
<td>130832</td>
<td>qwerty</td>
<td>5djv7ZCI2ws=</td>
</tr>
<tr>
<td>7</td>
<td>124253</td>
<td>12345676</td>
<td>dQi0asWPYyQ=</td>
</tr>
<tr>
<td>8</td>
<td>113884</td>
<td>111111</td>
<td>7LqYzKVeq8I=</td>
</tr>
<tr>
<td>9</td>
<td>83411</td>
<td>photoshop</td>
<td>PMDTbP0LZxu03SwrFUvYGA==</td>
</tr>
<tr>
<td>10</td>
<td>82694</td>
<td>123123</td>
<td>e6MPXQ5G6a8=</td>
</tr>
<tr>
<td>11</td>
<td>76910</td>
<td>1234567890</td>
<td>j9p+HwtWWT8/HeZN+3oiCQ==</td>
</tr>
<tr>
<td>12</td>
<td>76186</td>
<td>000000</td>
<td>diQ+ie23vAA=</td>
</tr>
<tr>
<td>13</td>
<td>70791</td>
<td>abc123</td>
<td>kCcUSCmonEA=</td>
</tr>
<tr>
<td>14</td>
<td>61453</td>
<td>1234</td>
<td>ukxzEcXU6Pw=</td>
</tr>
<tr>
<td>15</td>
<td>56744</td>
<td>adobe1</td>
<td>5wEAInH22i4=</td>
</tr>
<tr>
<td>16</td>
<td>54651</td>
<td>macromedia</td>
<td>WqflwJFYW3+PszVFZo1Ggg==</td>
</tr>
<tr>
<td>17</td>
<td>48850</td>
<td>azerty</td>
<td>hjAYsdUA4+k=</td>
</tr>
<tr>
<td>18</td>
<td>47142</td>
<td>iloveyou</td>
<td>rpkvF+oZzQvioxG6CatHBw==</td>
</tr>
<tr>
<td>19</td>
<td>44281</td>
<td>aaaaaaa</td>
<td>xz6PlEGr6g=</td>
</tr>
</tbody>
</table>
So what went wrong at Adobe?

• Well, “everything”, really…
  • 130,324,429 passwords in one leaked database
  • symmetric key cryptography used instead of hashing
  • ECB mode used—CTR would have worked nicely!
  • same key for every password

• Users’ password hints were stored in plaintext
  • That’s bad if the password hint is the password!
  • (… needless to say, that’s a terrible hint)
Security in practice with passwords

• Password changing policies: often hopeless
  • Requirements for changing passwords over time
  • Requirements for password content

• Requiring regular password change has risks:
  • Users more likely to parameterise their passwords
  • Users choose easier to remember passwords…

• Silly requirements and restrictions—key point: entropy
XKCD #936 sums it up very well

Correct horse battery staple

- Through 20 years of effort, we’ve successfully trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess.
Password managers

• Why is using a password manager good?
  • Secure passwords can be used: not even chosen by users!
  • Can use independent passwords per account

• Can go back to a single master password

• However there are some downsides:
  • You’re not remembering the password yourself
  • Synchronisation across devices?
Cryptography in MacOS Keychain

• While OS-specific, Apple got this one right
  • Has been included since MacOS 8.6!
  • Fairly recently gained iCloud Keychain (MobileMe?)

• Keychain provides single DB for secrets
  • Is used for macOS user login
  • Browsers: Safari, Chrome
  • Wi-Fi passwords
  • Apps such as svn, alpine, certificates, etc.
MacOS Keychain Access
MacOS multiple Keychains

- Previous screenshot shows multiple keychains
  - Collect passwords of different sensitivity
  - Different “lock after inactivity / sleep” settings
MacOS iCloud Keychain

- iCloud keychain: “magical” sync. across iOS & macOS

- Keychain sync involves a “circle of trust” with asymmetric crypto initiated from first device
  - iCloud gets to see the public key(s) only
  - Streams password update on demand to other devices
  - Apple does not get to decrypt the passwords

- Keychain backup is different…
Under the hood: MacOS Keychains

- MacOS provides the `security` command
- `security` will pop up GUI elements if it needs to

```bash
oucs1324 → ~ security find-generic-password -l 'Dummy item' -w dme-srg.keychain
not a good password
```

![Keychain Access](image-url)
We can explore Keychain items

- Retrieve attributes other than the password:

```
oucs1324 → ~ security find-generic-password -l 'Dummy item' dme-srg.keychain
keychain: "~/Users/dme/Library/Keychains/dme-srg.keychain"
class: "genp"
attributes:
  0x00000007 <blob>="Dummy item"
  0x00000008 <blob>==<NULL>
  "acct"<blob>="dme"
  "cdat"<timedate>=0x32303134303832383031343634315A00  "20140828014641Z\000"
  ...
  "mdat"<timedate>=0x32303134303832383031343634315A00  "20140828014641Z\000"
  "nega"<sint32>==<NULL>
  "prot"<blob>==<NULL>
  "scrp"<sint32>==<NULL>
  "svce"<blob>="Dummy item"
  "type"<uint32>==<NULL>
```

- Encryption? **macOS**: DES3?  **iOS**: AES-256
Other password managers

• Integrated into browsers: Chrome / Firefox

• Java has `java.security.KeyStore`
  • Like Keychain, this stores certificates, etc., too
  • Helped Java provide consistent operation in the bad-days of USA encryption export restrictions

• Projects such as LastPass, 1Password, etc.
• … but let’s return to **not** using passwords!
OpenID

• First appeared around 2005 as a LiveJournal feature
  • Then called ‘Yadis’: Yet Another Distributed Identity System
  • Brad Fitzpatrick was developer of both

• Idea is that your FOAF description would indicate your chosen identity server
  • (Friend of a friend—social network graphs)
  • Identity server digitally signs that you are who you say you are
OpenID focus

- OpenID is very much user-driven
  - Individuals are able to change their ID server

- Also very ‘netizen’-focused
  - “Who you are” might be entirely blog-based
  - … it’s important that you own your blog posts,
  - but doesn’t matter who you are in the real world
  - Provides features for users who own domains…
How OpenID works

• Another case of "triangular" authentication

• Key parties involved:
  • OP: OpenID Provider
  • RP: Relying Party
  • User agent: operated by an end user

1. Request
2. Who are you?
   Get your OP to tell me
3. Give me a note for RP about who I am
4. Here’s your note for RP
5. Request
   May establish shared secret

RP (the service you want to access)

OP (e.g. Google)
OpenID 2.0: authenticating to a site

- User navigates to protected resource
  - Identity URL requested
  - `mod_auth_opendid` is generating this page on our Apache server

- User enters their ID URL
  - e.g. `dme26.livejournal.com`
OpenID module redirects to OP (1)

- Client's user agent establishes secure connection to OP

- OP can authenticate the user
  - ... or continue an existing session

- **Crucial**: OP checks that user is satisfied the RP should receive that user's details
  - Not just authentication
  - Attributes can be passed to the RP
OpenID module redirects to OP (2)

Identity Validation

Another site on the web wants to validate your LiveJournal identity. No information will be shared with them that isn't already public in your profile, only that you're who you've already told them you are (if you told them).

The address wanting permission is:

http://localhost:8080/4-OpenId-Auth/

Do you want to pass your identity to them?

Yes; just this time.  Yes; always.  No.
RP OpenID code sent login info (1)

• RP’s OpenID software has state or is stateless

• Implementation with state:
  • A shared secret already negotiated with OP
  • The dme26.livejournal.com interaction has state:
    • see openid.assoc_handle within OP/RP interaction

• Stateless implementation:
  • An additional confirmation interaction is required with the OP
    in order to verify the OP’s statement
RP OpenID code sent login info (2)

### Authentication via OpenID

<table>
<thead>
<tr>
<th>PHP Version 5.3.10-1ubuntu3.24</th>
</tr>
</thead>
</table>

| System | Linux precise 3.2.0-23-generic #36-Ubuntu SMP Tue Apr 10 20:39:51 UTC 2012 x86_64 |
| Build Date | Aug 1 2016 20:14:32 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php5/apache2 |
| Loaded Configuration File | /etc/php5/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php5/apache2/conf.d |
| Additional .ini files parsed | /etc/php5/apache2/conf.d/phpd.ini |
| PHP API | 20090626 |
| PHP Extension | 20090626 |
| Zend Extension | 20090626 |
| Zend Extension Build | AP20090626,NTS |
| PHP Extension Build | AP20090626,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| IPv6 Support | enabled |
What's happening on the network?

- I gave ID URL to VM’s Apache:

- Our RP (mod_auth_openid) talks to the OP

- An XRDS document may lead to OP URLs
  - (eXtensible Resource Descriptor Sequence)
Interactions with our VM and livejournal.com

# my browser’s form submission
GET /4-OpenId-Auth/?openid_identifier=dme26.livejournal.com HTTP/1.1
# RP’s OpenID module's request to OP
GET / HTTP/1.1
Host: dme26.livejournal.com
...
# request to OP (over HTTPS)
# response from OP is 200 OK (over HTTPS)
# response from OP to UA: redirect to auth
HTTP/1.1 302 Found
# https://www.livejournal.com/openid/server.bml?openid.assoc_handle=1536...
...
# (My UA interacts with livejournal, which our VM does not see)
...
# my browser is redirected here by OP
GET /4-OpenId-Auth/?&modauthopenid.nonce=bhP098xs6n&openid.mode=id_res&...
# request from RP to OP over HTTPS
# response from OP to RP over HTTPS
# request from RP to OP over HTTPS
# response from OP to RP is 200 OK
# response to my client: sets cookie in user agent and redirects
# my browser makes another request with the open_id_session_id cookie set
GET /4-OpenId-Auth/
# data from server ...
Another web-SSO

• Shibboleth: web-based SSO based on SAML
  • (Security Assertion Markup Language)

• Compared to OpenID, more ‘institutional’
  • You don’t get to move your ID provider

• Very popular across academic organisations
  • They require sign-on to external services (library)
  • Often more ‘loosely coupled’ than corporates
How Shibboleth works

• This looks familiar … quite similar to OpenID
  • WAYF: Where Are You From service
  • SP: Service Provider
  • IdP: Identity Provider
  • User agent

1. Request
2. Who are you?
   Get your IdP to tell me
3. I need to find my IdP
4. Look over there...
5. Give me a note for SP
6. Here’s your note for SP
7. Request
   7b. SP may ask IdP for user attributes

WAYF (e.g. Tuakiri)
IdP (e.g. Otago)
Tuakiri: NZ Access Federation

- Facilities at any member’s site can be set up to be accessed by any other member’s users

Universities
- Auckland University of Technology
- Lincoln University
- Massey University
- The University of Auckland
- University of Canterbury
- University of Otago
- University of Waikato
- Victoria University of Wellington

CRIs
- AgResearch
- Environmental Science & Research
- GNS Science
- Landcare Research New Zealand
- Plant and Food Research
- NIWA
- SCION
WAYF: Where Are You From?

• Tuakiri: heavily involved setting up federation
  • Beyond accessing others’ hosted services, need federation for shared services
  • e.g., New Zealand eScience Infrastructure (NeSI)

• Technically, Tuakiri provides the WAYF service
  • As shown earlier, switches user through to IdP
Example: accessing a web service

- I want to retrieve some research data
- Stored on a NeSI (prototype) service: http://df.bestgrid.org/BeSTGRID/home/S3DR/datasets/20130816-Dunedin/GpsLog.txt
- The above data is protected but front page isn’t:
Tuakiri

- Can set up a default IdP (home organisation)
- WAYF will then redirect without user input
Otago IdP

• The IdP is a server running at Otago for us
• Users have control over attribute release
User agent returns to SP

• If all is well, original request will be honoured
  • There is usually a requirement for time synchronisation

• Various ‘sessions’ can be established:
  • SP will likely initiate some type of web session
  • Shibboleth may refresh SP session without authentication
  • IdP may allow Shibboleth to refresh without authentication

• Compared to OpenID, Shibboleth links to your real world identity more directly
But... who am I?

• Shibboleth's institutional focus may not reflect reality effectively in corner cases

• Principals may have multiple affiliations, e.g.,
  • Primarily, I’m eyeda59p@otago.ac.nz
  • However I’ve also been deye490@auckland.ac.nz
  • Further afield, I’ve been dme26@cam.ac.uk too

• So at times I had three Shibboleth IdPs (!)
  • ...and they didn’t know that I’m the same person
In summary

• If stuck with passwords, use a key-store

• Web Single Sign On systems offer increased convenience and security for web users
  • Typically have a similar distributed structure

• As an overgeneralisation:
  • OpenID focuses on users’ online identities
  • Shibboleth focus: users’ roles in organisations