Leveraging Guest Accounts for Ubiquitous Web Sign-On System Acceptance
Web Single Sign-On

- CoSign
- CAS
- PubCookie

Sign-on at one web site, but get access to a wide variety of web sites. A few reasons why WebSSO is a good idea:
1) Security: all user credentials go to a well known URL & applications don’t ever get user credentials.
2) User convenience: users don’t have to remember or enter their credentials repeatedly.
3) Less work: If the central vs local split is done well, both the central group & the application admin have less work.
Our old solution (Cookie Server) wasn’t a WebSSO. It wasn’t as secure, and actually made more work for both central & application admins. It also wasn’t convenient for the users. Since it was not yet ported to modern systems, and there were some known vulnerabilities, we recommended to our Provost’s Office that it be replaced. At its peak, it protected 20–30 applications, and there were many other applications implementing local authentication.

As part of the University’s “IT Commons” effort, we invited anyone with interest to opt-in to the requirements gathering & design process.
CoSign Basic Features

• Compartmentalized Security
• Kerberos V
• Proxy Kerberos Tickets
  - http://filedrawers.org
• High Availability
• Global Logout

CoSign’s architecture permits ANY web server to participate, without requiring that the web server first pass a security audit. FileDrawers is our web-enable AFS file manager. It has permission to obtain proxied kerberos tickets from CoSign. CoSign implements high availability by replicating the logged in and registered state information between CoSign servers. CoSign allows the user to logout of one web site, and be logged out of all CoSign protected web sites.
CoSign Extended Features

- Centralized Guest Accounts
- Proxy CoSign Cookies
- BasicAuth
- Re-Authentication
- X.509
- Multi-Factor

Added in support of University of Edinburgh’s deployment of uPortal. Like Proxy Kerberos Tickets, except the second tier service would be a CoSign-protected service instead of a Kerberos protected service.

Support for BasicAuth allows any authN that apache supports -- including other WebSSOs.

Financial Operations wanted re-authentication.

The CoSign CGI can translation x.509 subject DNs to user IDs.

3rd draft of the multi-factor spec is available on our web site.
As security conscious admins, it was initially shocking to visit a service that we had not yet logged into, and have it display private information.
Influential Factors

• Users start to expect Single Sign-On
• Smaller footprint than old solution
• Support, support, support

In contrast to security conscious admins, users hardly noticed when Single Sign-On worked, but frequently noticed when they visited an application that didn’t participate. Webmasters find running a CoSign protected web site less work than our old solution or “rolling their own.”

One of the requirements we heard was that admins sometimes needed help to install the cosign filter, so the programming staff made frequent site visits.
Plateau

- All the easy transitions are done.
- IT Leaders are excited, but...
- Sysadmins say, “CoSign doesn’t work for us.”
- Why not?

Around 80 applications using cosign.
Barriers

- Faculty Collaborate Outside of UMich
- Departments support collaboration with guest access
- Departmental support of guest accounts is resource intensive

1) A “roll your own” solution costs a lot of staff time to setup / develop / maintain.
2) Guest accounts cost a lot of departmental staff time to support, e.g., provisioning, password resets, etc.
We asked the members of the AuthN/Z Working Group to review our plans. We came up with a continuum of accounts, from completely unaffiliated and unlikely to ever be affiliated; through visiting scholars, doctors in rotation, internships, clerkships; to obviously affiliated UMich People.

One suggestion was to just give everyone an ID. But, this would cost us the strong identity assertions that we currently make, e.g., show up in the Accounts Office with two forms of government issued ID.
Given that the ostensible purpose of the effort was to facilitate faculty collaboration, why not share authentication with our “peer” institutions -- one university’s guests might be another university’s Faculty.
Shibboleth

- Federated administration
  - agreements between institutions
- Access control based on attributes
  - not necessarily identity
- Restricts access to participating institutions

Shib is all about agreeing on identity assertions, which tend to be strong, like uniqnames. Roles based, presumes that applications can be written with roles in mind, rather than simple access control lists.
We could advocate for Shibboleth on campus, helping departmental admins make their application Shibboleth Targets.
However, a lot of the people who had access were from non-participating institutions, which was a show-stopper for shib.
Eventually, we conceptualized the problem thusly. Treating everyone as guests allows UMich to not burden the home institution into order to collaborate with one of our faculty.
The power of weak assertions is easy of use. While University of Michigan might not make a strong assertion about the identity of one of these guests, the member of the institution is probably quite confident that the email address belongs to the person they are trying to collaborate with.
We were shocked to get a proposal from our Student Billing department suggesting the use of guest accounts for paying the University. However, their logic was unassailable: authorization would be controlled by the student, validated each term, and granted access only to pay bills.
Usage

- January 2005
  - 32,109 requests
  - 26,428 accounts
- October 2005
  - 68,688 requests
  - 56,318 accounts

Largest .edu is MSU with 506.
Vast majority of .coms from Yahoo, Hotmail, AOL, GMail, etc.
In toto, 9000 unique domains.
Support Costs

- Several “one off” feature requests
- Assemble focus groups
- How do we reduce support costs even more?

After the initial deploy of “Friend”, began to get several feature requests, mostly focused on further reducing support costs.
The first three features are directly focused on further reducing departmental support costs of guest accounts. Having invitations filtered as SPAM was the only common question handled by central support. The University’s security team suggested that we include some phishing “counter measures” in the invitation code, to prevent Friend from being used nefariously.
The portion of Friend tied to CoSign is quite small. Any WebSSO ought to be able to use this code base.
Future Direction

- Debrief Friend v2
- CoSign as a Shibboleth target
- Inter-institutional WebSSO

This gets us back toward more strongly authenticating our guests.
“If you build it, they will come.”
...and any infrastructure you put in its way is an impediment to collaboration.