Installing and Configuring Webauth

Stephen Quinney
<stephen.quinney@oucs.ox.ac.uk>

Systems Development and Support
Computing Services

Background – Basic Auth

- Simplest authentication scheme for web services is *HTTP Basic Authentication.*
  - Client requests protected resource and server responds by requesting authorization.
  - Client resends request with header encoding the username and password.
- Security issues with this though.
  - Username and password must be sent with every request as HTTP is stateless.

Background - Cookies

- Along with a response to an HTTP request the server can send a piece of state information that the client stores – a *cookie.*
- The cookie is then sent by the client along with any future requests.
- This can be used to avoid verifying username and password with every request.

Single Sign-On

- SSO systems consist of several components:
  - client (web browser)
  - application server
  - login server
  - external authentication service

Login Server

- Trusted central authentication service
- Interacts directly with the users
- Verifies username and password with backend authentication services
- Issues cookies to provide SSO functionality
- Provides authentication information to the application server

Application Server

- Enforces authentication.
- Redirects users who are not authenticated to the login server.
- Verifies authentication information from login server.
- Issues cookies to maintain application sessions.
- Provides authentication information to the applications.

Benefits of SSO

- Passwords are only sent to the central login server over SSL.
- Users only need to authenticate once per session.
- Leverages existing authentication system.
- Works with all modern browsers.

Stanford Webauth

- The login server issues two cookies:
  - Cookie given for access to the application server is a Kerberos service ticket.
  - Cookie shared between login server and web browser is a Kerberos ticket-granting ticket.

Installing Webauth

Full details are available from:

http://webauthv2.stanford.edu/

and:

http://www.oucs.ox.ac.uk/webauth/

Software Requirements

- You will need:
  - Apache2 (2.0.43 or better)
  - OpenSSL (0.9.7)
  - MIT Kerberos v5 (1.2.1 or better)
  - cURL (7.10 or better)

Debian Packages

- For Sarge, edit your /etc/apt/sources.list so it includes:
  - deb http://archives.eyrie.org/debian sarge main non-free contrib
- For etch and sid, the packages are now in the main archive.
- Remember to run “apt-get update”
- Install libapache2-webauth and krb5-user.

SSL

- Resources protected by Webauth require SSL (https).
- May need to generate a certificate:
  - In Debian use apache2-ssl-certificate
- Ensure Apache2 is listening on port 443.
- Enable and configure ssl as needed.
- Create a VirtualHost for port 443.

Kerberos

```
[libdefaults]
    default_realm = OX.AC.UK

[realms]
OX.AC.UK = {
        kdc = kdc0.ox.ac.uk
        kdc = kdc1.ox.ac.uk
        kdc = kdc2.ox.ac.uk
        admin_server = kdc-admin.ox.ac.uk
}

[domain_realm]
        .ox.ac.uk = OX.AC.UK
        ox.ac.uk  = OX.AC.UK
```

Kerberos

- Contact **support@sysdev.oucs.ox.ac.uk** to acquire a webauth Kerberos principal.
- Generate a keytab:

```
# kadmin -p username/itss

kadmin: ktadd -k /etc/webauth/keytab
  webauth/hostname@unit.ox.ac.uk@OX.AC.UK

kadmin: quit

# chown www-data /etc/webauth/keytab

# chmod 0600 /etc/webauth/keytab
```

Configure Apache2

```
LoadModule webauth_module /usr/lib/apache2/modules/mod_webauth.so

# Set the locations for various Webauth related files

WebAuthKeyring /var/lib/webauth/keyring
WebAuthKeytab  /etc/webauth/keytab
WebAuthServiceTokenCache /var/lib/webauth/service_token_cache
WebAuthCredCacheDir /var/lib/webauth/cred_cache

# Point to the Oxford Webauth service

WebAuthLoginURL “https://webauth.ox.ac.uk/login/”
WebAuthWebKdcURL “https://webauth.ox.ac.uk:8443/webkdc-service/”
WebAuthWebKdcPrincipal service/webkdc

# If you are having trouble:

#WebAuthDebug on
```

Enable Webauth protection

```
<Location /private>
    WebAuthExtraRedirect on
    AuthType WebAuth
    require valid-user
</Location>
```

Per-directory Access Control

Allow .htaccess:

```
AllowOverride AuthConfig
```

put into your .htaccess:

```
AuthType WebAuth
require user jdoe
```

Group Access Control

```
AuthType WebAuth
AuthGroupFile /web/groups
require group admin
```

The group file would contain:

```
admin: bob joe anne
```

Alternate access methods

```
AuthType WebAuth
require user jdoe
order deny,allow
deny from all
allow from ox.ac.uk
satisfy any
```

Useful Links

OUCS Webauth documentation:

http://www.oucs.ox.ac.uk/webauth/

Primary documentation:

http://webauthv3.stanford.edu

For Kerberos principals and Webauth help:

support@sysdev.oucs.ox.ac.uk