External Identity and Authentication Providers For Apache HTTP Server

Jan Pazdziora
Principal Software Engineer
Identity Management Engineering, Red Hat

APACHECON EUROPE
17th November 2014
Basic Authentication

- The only authentication option in 1996 when HTTP 1.0 came out.
- To remind you what it looked (looks) like:

![Authentication Dialog]

- Status code 401 Unauthorized. It means either
  - no authentication was attempted;
  - the [login, password] pair supplied with the HTTP request in the Authorization header was wrong.
Basic Authentication: Pros

- Access protection for static content as well.
- Completely handled via HTTP server configuration.
- No logic needed in the content (in CGI).
- User identifier can be consumed in CGI scripts via `REMOTE_USER` environment variable.
  - Similar mechanisms used for other execution frameworks.
  - Or dedicated method calls (`request.getRemoteUser()`).
- Various authentication providers emerged, including databases and LDAP lookups.
Basic Authentication: Cons

- One 401 status for both "please enter login and password" and "you probably mistyped password" situations.
- Suboptimal UI in browsers: one popup window type, ending loop with Cancel, no logout (forget credentials) functionality.
- Optional authentication hard to achieve.
- Nothing beyond [login, password].
- Digest introduced by HTTP 1.1 did not address either concern.
Authentication in applications

- Basic Authentication was used heavily.
- But developers and users wanted more.
- Especially better control and user experience.
Cookie-based sessions

- Codified ex-post, based on real-life implementations in browsers.
- Originally intended for small customizations and user preferences.
- Cornerstone of authentication in today's web applications.
  - Applications handle logon form POST submissions or other authentication process, including anonymous users.
  - Applications create sessions internally, HTTP response carries Set-Cookie header with session identification.
  - Cookie sent by browser with each subsequent HTTP request in the Cookie header.
- The authentication decision has moved to applications completely.
- Applications manage their own (DB) schemas of users, groups, roles.
- Who remembers REMOTE_USER? Who needs REMOTE_USER?
GSSAPI/SPNEGO/Kerberos/Negotiate

- Server's 401 HTTP response contains WWW-Authenticate: Negotiate.
- Browser tries to get Kerberos service ticket and use the GSSAPI data in Authorization header.
- No prompting. (But no confirmation either.) Effectively, single-sign-on.
- In Apache supported by mod_auth_kerb, outside of application.
- Application might not have access to the keytab needed to verify the GSSAPI data.
- Application gets the authentication result. REMOTE_USER re-emerges.

- Cookies still useful — you want to avoid negotiate on each request.
Other mechanisms

- Other authentication mechanisms might need to use credentials and storage that HTTP server (Apache) has access to but the application does not.
  - SSL client authentication.
  - Security Assertion Markup Language (SAML).
- There can be additional checks about account's validity (PAM).
- They all might or might not be needed (supported, enabled, configured) in a particular deployment of each web application.
- Is it time to move the authentication decision back in front of the web application?
- Bring back REMOTE_USER?
## Overview of existing modules

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Apache Authentication Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Application Level</td>
<td>None</td>
</tr>
<tr>
<td>Kerberos SSO (ticket)</td>
<td>mod_auth_kerb</td>
</tr>
<tr>
<td>SAML-Based</td>
<td>mod_auth_mellon</td>
</tr>
<tr>
<td>Certificate-Based</td>
<td>mod_nss</td>
</tr>
<tr>
<td></td>
<td>mod_ssl</td>
</tr>
</tbody>
</table>
New life for GSSAPI/Kerberos

- Module `mod_auth_gssapi` by Simo Sorce.
- Replacement of `mod_auth_kerb` using only GSSAPI calls.
- Original `mod_auth_kerb` configuration:
  - LoadModule auth_kerb_module modules/mod_auth_kerb.so
  - AuthType Kerberos
  - KrbMethodNegotiate On
  - KrbMethodK5Passwd Off
  - KrbAuthRealms EXAMPLE.COM
  - Krb5KeyTab /etc/http.keytab

- With `mod_auth_gssapi`:
  - LoadModule auth_gssapi_module modules/mod_auth_gssapi.so
  - AuthType GSSAPI
  - GssapiCredStore keytab:/etc/http.keytab

- Recent MIT krb5 and Apache HTTP server 2.4 needed.
System Security Services Daemon

- Authentication and identity services on operating system level.
- Host-based access control (HBAC) when used with IPA server.

<table>
<thead>
<tr>
<th>Individual users or user groups</th>
<th>Hosts or host groups</th>
<th>Services (ssh, ftp, sudo, ...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mix them into rules</td>
</tr>
</tbody>
</table>

sssd can consult IPA to check access

- IPA is centralized identity, authentication, and authorization provider.
- Other access control schemes possible, depending on the identity source against which sssd is configured.
- Module `pam_sss.so` makes sssd services available via PAM.
PAM for Web applications

- Apache module **mod_authnz_pam**. For 2.2 and 2.4.
- PAM-based authorization of users authenticated by other modules.
- Replace requires valid-user with

  ```
  requires pam-account <PAM-service-name>
  ```

- Configure `/etc/pam.d/<PAM-service-name>`.
  - With `pam_sss.so` and `sssd` against IPA, HBAC check will be done.
  - HBAC service name has to match the PAM service name.
  - Use any service name you want: `crm-prod`, `wiki-test`, `intranet`, ...
- Especially useful for SSO that should not reach applications.
- Use as Basic Authentication provider also possible:

  ```
  AuthBasicProvider PAM
  AuthPAMService tlwiki
  ```
# PAM for applications' logon forms

- Provided by Apache server: `mod_intercept_form_submit`.

<table>
<thead>
<tr>
<th>Module</th>
<th>Logon form submission</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Module intercepts the POST HTTP request</td>
</tr>
<tr>
<td></td>
<td>PAM auth is run with [login, password] pair (when found)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authentication passes</th>
<th>Authentication fails</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>REMOTE_USER is set to login</td>
</tr>
<tr>
<td></td>
<td>EXTERNAL_AUTH_ERROR is set to PAM message</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumes REMOTE_USER</td>
<td>Gets chance to authenticate internally</td>
</tr>
</tbody>
</table>
PAM for apps' logon forms (cont'd)

- No 401 status ever.
- Uses mod_authnz_pam internally.
- The same look of the logon screen, authenticating against central identity provider.

```xml
<Location /app/login>
  InterceptFormLogin user_fld
  InterceptFormPassword passwd_fld
  InterceptFormPAMService <PAM-service-name>
</Location>
```
# New modules

<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Apache Modules</th>
<th>Access Check</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>GSSAPI/Kerberos</td>
<td>mod_auth_kerb</td>
<td>mod_authnz_pam</td>
</tr>
<tr>
<td></td>
<td>mod_auth_gssapi</td>
<td></td>
</tr>
<tr>
<td>SAML</td>
<td>mod_auth_mellon</td>
<td></td>
</tr>
<tr>
<td>Certificate</td>
<td>mod_nss</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mod_ssl</td>
<td></td>
</tr>
<tr>
<td>Form-Based</td>
<td>mod_intercept_form_submit</td>
<td></td>
</tr>
</tbody>
</table>
Additional user information

- Web applications nowadays need more than just login name.
- Additional attributes for nice user experience, as well as authorization.
  - Email address, full name, phone number, ...
  - Group membership.
- For centrally-managed users, these should come from the central identity provider.
- Especially when applications autocreate user records.
- Module `mod_lookup_identity` uses D-Bus interface of SSSD to retrieve additional data about authenticated users.
Additional user information (cont'd)

- Proposing other environment variables beyond REMOTE_USER:
  - REMOTE_USER_EMAIL, REMOTE_USER_FULLNAME, ...
  - REMOTE_USER_GROUPS, REMOTE_USER_GROUP_N, REMOTE_USER_GROUP_1, ...

```
LookupUserAttr mail REMOTE_USER_EMAIL " "
LookupUserAttr givenname REMOTE_USER_FIRSTNAME
LookupUserAttr sn REMOTE_USER_LASTNAME

LookupUserGroupsIter REMOTE_USER_GROUP

LookupOutputGroups REMOTE_USER_GROUPS : 
```
## Module overview

<table>
<thead>
<tr>
<th>Authn Method</th>
<th>Apache Modules</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>Access Check</td>
<td>Extra User Info</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSSAPI</td>
<td>mod_auth_kerb</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mod_auth_gssapi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAML</td>
<td>mod_auth_mellon</td>
<td>mod_authnz_pam</td>
<td>mod_lookup_identity</td>
</tr>
<tr>
<td>Certificate</td>
<td>mod_nss</td>
<td></td>
<td>mod_ssl</td>
</tr>
<tr>
<td>Form</td>
<td>mod_intercept_form_submit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
External authentication in applications

- Web applications should re-learn to accept `REMOTE_USER`.
- Some changes to support the external authentication and identity are typically needed in application code.
- The reward is much richer matrix of possible deployments.
- Use of the same HBAC mechanism that enterprises use for OS.
- Already implemented:
  - Spacewalk
  - Foreman
  - ManageIQ
- Django being investigated.
Conclusion

- PAM for access to central authentication provider.
- New variables for additional REMOTE_USER_* attributes.
- Can we agree on variable names? Less work for application developers.
- By no means should applications drop their existing functionality that served them well, this is merely an additional possibility.
- Your favorite application or framework not supporting REMOTE_USER_*?
  - While we might not be able to add the feature ourselves, we will be happy to help people.
- Explore the modules, let us know what you think.
References

- www.freeipa.org/page/Web_App_Authentication
- www.freeipa.org/page/Environment_Variables#Proposed_Additional_Variables
- github.com/modauthgssapi/mod_auth_gssapi
- www.adelton.com/apache/mod_authnz_pam/
- www.adelton.com/apache/mod_intercept_form_submit/
- www.adelton.com/apache/mod_lookup_identity/
- <freeipa-users@redhat.com>
- Jan Pazdziora <jpazdziora@redhat.com>