Securing Apache

Setting up Apache with ModSSL and Mod_Auth_Kerb

Mark Earnest
Senior Systems Programmer
Academic Services & Emerging Technologies
Pennsylvania State University
mxe20@psu.edu

PENNSTATE

Web Authentication

- Mod_Auth_Kerb allows your webserver to authenticate users with their PSU AccessID & password

- ModSSL provides both the encrypted channel (HTTPS) for the userID and password to go over, and identifies the server as trusted by a Certificate Authority
    - Passwords should NEVER be sent over an unencrypted protocol

PENNSTATE

Required Packages

- OpenSSL
  - Library that provides the SSL functions necessary for modSSL
  - http://openssl.org
- KerberosV5
  - Library that provides the Kerberos functions necessary for mod_auth_kerb
  - http://web.mit.edu/kerberos/www
- Apache
  - Webserver – Version 1.3.x or 2.0.x
  - http://httpd.apache.org
- ModSSL
  - http://www.modssl.org
- Mod_auth_kerb
  - http://modauthkerb.sourceforge.net

PENNSTATE

Building ModSSL

- Apache, OpenSSL, and modSSL may already be installed on your OS, check documentation.
- OpenSSL need to be built and installed. This is beyond the scope of this presentation. Refer to the INSTALL file that comes with the source.
- Configure ModSSL:

  ./configure --with-apache=/apache/source/*dir* \

  ---with-*ssl*=/openssl/source/dir \

  ---prefix=/*usr*/local/apache

- Build Apache (See Apache Documentation)
  - --enable-shared --enable-module=most

PENNSTATE

Self Signed Certs

- Generate a self signed cert with by running “make certificate” in Apache's source tree.
- Answer questions (Country, State, Organization, etc)
  - CommonName (CN) MUST be your servers fully qualified hostname
- Start Apache with “/path/to/apache/sbin/apachectl startssl” or “/path/to/apache/bin/httpd -DSSL”
- Test by pointing a web browser to https://yourhostname.psu.edu
- A selfsigned cert will produce an error message on the user's browser. For a production machine you will likely want to purchase a signed certificate from a Certificate Authority (CA) such as Verisign, RSA, or Thawte. PSU is also piloting a Certificate Authority.

PENNSTATE

CA Signed Certs

- Generating a key
  - *openssl genrsa -out yourhostname.key 1024*
  - Optionally add “-des3” if you wish to encrypt the key. NOTE: This will require you enter a password every time you start the web server.
- Generating a CSR
  - *openssl req -new -key yourhostname.key -out yourhostname.csr*
  - As before, answer all questions. CommonName MUST be the servers fully qualified hostname
- Getting the CSR Signed
  - Send CSR to the CA. The method for doing this varies from CA to CA. Refer to the CA's instructions regarding this.
- Receive the signed certificate from the CA
  - The CA will send you back a PEM encoded .crt file. This is your signed certificate.

PENNSTATE

Using your New Cert

- Edit Apache's httpd.conf file to point to your key and certificate
  - SSLCertificateFile /path/to/yourhostname.crt
  - SSLCertificateKeyFile /path/to/yourhostname.key
    - Set permissions on the key to ensure only root can read and write to it.
- Your certificate has an expiration date. Check with your CA to find out how long the certificates they issue are good for.
- You must restart Apache after changing the httpd.conf file to load the new certificate and key.
- Another way to test the SSL connection is with:
  - *openssl s_client -connect yourwebserver.psu.edu:443*
- Force connections to go over SSL (httpd) with the rewrite module:
  - *RewriteEngine On*
  - *Rewriterule ^/(.*) https://testtable5.aset.psu.edu/$1 [R]*

PENNSTATE

Kerberos

- If the Kerberos5 libraries and header files are not installed with your OS, you must download and install them. Refer to your OS's documentation or http://web.mit.edu/kerberos/www
- To configure Kerberos5 to authenticate into PSU's KDC, your /etc/krb5.conf must include the following:

[*libdefaults*]

default_realm = dce.psu.edu

[realms]

dce.psu.edu = {

kdc = f04s05.cac.psu.edu

kdc = f04s06.cac.psu.edu

kdc = *vader.offsite.psu.edu*

}

PENNSTATE

Building mod_auth_kerb

- To build mod_auth_kerb as a DSO (the reason we used –*enabled-shared* to the Apache build config), run the following command:

  */path/to/apache/sbin/apxs -c -DKRB5 -DKRB_DEF_REALM=\\\"dce.psu.edu\\\" *

  *-I/usr/kerberos/include -L/usr/kerberos/lib -lkrb5 *

  -ldl -lcom_err -lk5crypto mod_auth_kerb.c

- To build for Apache 2.0.x, you must add -DAPXS2 to the above command.
- Copy mod_auth_kerb.so to the Apache modules directory.
- Add the following to Apache's httpd.conf file:
  - LoadModule kerb_auth_module modules/mod_auth_kerb.so

PENNSTATE

Using mod_auth_kerb

- Option #1 - .htaccess File

```
AuthType KerberosV5
AuthName "Penn State Access Account Login"
KrbAuthRealm dce.psu.edu
require valid-user
```

- Option #2 – httpd.conf <directory> or <location> directive

```
<Directory /var/www/html/protected>
        AllowOverride None
        AuthType KerberosV5
        AuthName "Penn State Access Account Login"
        KrbAuthRealm dce,psu.edu
        require valid-user
</Directory>
```

PENNSTATE

General Apache Security Tips

- Use strict permissions.
  - In your apache directory, bin, conf, and log should be writable ONLY by root.
- Pay special attention to CGI scripts/programs
  - Run only trusted CGIs.
- Prevent .htaccess files from overriding your settings
  - Place “AllowOverride None” inside httpd.conf in the <Directory /> directive.
  - This prevents .htaccess files from overriding your security directives unless specifically allowed in the httpd.conf file for a specific directory.
- Remove “Indexes” from <Directory /> directive.
  - This prevents browsers from getting file listing on directories that do not have an index.html file.
- Remove “FollowSymLinks” from <Directory /> directive