Secure Information Sharing*
A Final Report submitted to
Network Information and Space Security Center (NISSC)
for Spring 2004 Sponsored Project

Ganesh Godavari, and Edward Chow
Department of Computer Science, University of Colorado, 1420 Austin Bluffs Parkway
Colorado Springs, Colorado 80917 USA
gkgodava@cs.uccs.edu, chow@cs.uccs.edu

1 Appendix

1.1 Installation

Currently SIS 0.1 has been tested on x86 linux. It currently uses OpenSSL 0.9.7d for cryptographic functions, Apache 1.3.31 for web services, and ModSSL 2.8.18-1.3.31 for providing SSL/TLS support in Apache. The following instructions install SIS 0.1 as well as OpenSSL 0.9.7d, Apache 1.3.31, and ModSSL 2.8.18-1.3.31.

1. download the source code of SIS 0.1 from http://blanca.uccs.edu/ infoshare/sis-0.2.tar.gz
2. download the distributions of Apache, mod_ssl and OpenSSL
   $ wget http://www.axint.net/apache/httpd/apache_1.3.31.tar.gz
   $ wget http://www.modssl.org/source/mod_ssl-2.8.18-1.3.31.tar.gz
   $ wget ftp://ftp.openssl.org/source/openssl-0.9.7d.tar.gz
   $ wget http://blanca.uccs.edu/ infofuse/src/sis-0.2.tar.gz

   if you donot have openldap download openldap rpm's from
   $ lynx ftp://fubphpc.tu-graz.ac.at/pub/redhat/9/RedHat/RPMS/openldap-devel-2.0.27-8.rpm
   $ lynx ftp://fubphpc.tu-graz.ac.at/pub/redhat/9/RedHat/RPMS/openldap-servers-2.0.27-8.rpm
   $ lynx ftp://fubphpc.tu-graz.ac.at/pub/redhat/9/RedHat/RPMS/openldap-2.0.27-8.rpm
   $ lynx ftp://fubphpc.tu-graz.ac.at/pub/redhat/9/RedHat/RPMS/openldap-clients-2.0.27-8.rpm

   install rpm's using
   $ rpm -Uvh openldap*.rpm

3. extract the distributions of Apache, mod_ssl and OpenSSL
   $ tar -xvzf apache_1.3.31.tar.gz

*Draft Report 0.1

$ tar -xvzf mod_ssl-2.8.18-1.3.31.tar.gz
$ tar -xvzf openssl-0.9.7d.tar.gz

4. Build OpenSSL
   $ cd openssl-0.9.7d
   $ ./config
   $ make
   $ cd ..

5. extract SIS module into apache
   $ tar -xvzf sis-0.2.tar.gz
   $ mv sis-0.2 apache_1.3.31/src/modules/
   $ cd ..

6. Build and install the SSL and SIS aware Apache
   $ cd mod_ssl-2.8.18-1.3.31
   $./configure
   –with-apache=../apache_1.3.31
   –with-ssl=../openssl-0.9.7d
   –prefix=/usr/local/apache
   –activate-module=src/modules/sis-0.2/mod_sisauth_ldap.c
   $ cd ..
   $ cd apache_1.3.31
   $ make
   $ make certificate
   $ make install

Now SIS 0.2 module has been created and added to apache. You need to have a section like below in the httpd.conf file:

<Directory "/usr/local/apache/htdocs/foo">
Options Indexes FollowSymLinks
AllowOverride All
LDAP_Port 389
AuthLDAPAuthoritative on
Bind_Pass "secret"
UID_Attr cn
CACertificateFile /home/gkgodava/rbac/3281/openssl/openssl-0.9.7c/3281/openssl-ac-supp/ca-bundle.crt
SISrequire "sis:1.Role:1.=Manager,sis:1.Group:1.=Info Share,sis:1.OU:1.=UCCS,"
</Directory>

where

- *AuthLDAPAuthoritative* Setting this directive to 'no' (by default it is 'yes') allows for both RBAC authorization to be performed using AC's

- *LDAP_Port* The port on LDAP server. The default and standard port number for LDAP is 389.
- *Bind_Pass* The bind password (in plain text).

- *UID_Attr* The attribute to use in LDAP search. The default LDAP attribute is 'cn'.
- *CACertificateFile* The path for the Attribute Authrority certificate.
- *SISrequire* ACL role specification (see section 4.2 for further details).

DO NOT forget to edit the above section and make sure to change the LDAP_Server to your Ldap server, Base_DN and other required attribute as well.

2 Administration Tool

Administration tool is used to simplify the process of creating, managing PKC and AC certificates. it uses OpenSSL 0.9.7d for generation of certificates

1. download the source code of Admin tool from http://blanca.uccs.edu/ infoshare/sis-admin-0.1.tar.gz

2. extract SIS Admin tool
   $ tar -xvzf sis-admin-0.1.tar.gz
   $ Make

3. create CA certificate using the command (make sure to back up the cacert.pem and cakey.pem)
   $ sh admin.sh ca

4. create User certificates using the command
   $ sh admin.sh pkc <username>
   certificates and key are created with the username e.g. <username>-cert.pem (certificate); <username>-key (key).
   also browser<username>.p12 will be created this can be used for storing the certificate in the web browser of the user.

5. create attribute certificates using the command
   $ sh admin.sh ac <issuer-certficate><Rbac-policy-file ><outputfilename >
   creates both 'User Role specification' and 'Delegated Role Specification' AC depending on the issuer.

3 Ldap Server

Ldap server is used to store PKC and AC certificates generated using the Administration tool. OpenLdap server cannot store attribute certficates. Follow the below instructions on how to store attribute certificates in OpenLdap server.

1. download the patch for open-ldap-2.0.27 from http://blanca.uccs.edu/ infoshare/patchfileldap
2. apply the patch to add attribute certificate schema
   $ patch -p0 <patchfileldap

3. configure slapd.conf. below is a snippet of the configuration file
   **********Snippet Of slapd.conf **********
   database ldbm
   suffix "dc=UCCS,dc=edu"
   rootdn "cn=manager,dc=UCCS,dc=edu"
   rootpw secret
   index cn,sn,st pres,eq,sub

4. start ldap server
   service ldap start

5. add an entry to the ldap server
   sample ldif file can be downloaded from http://blanca.uccs.edu/ infoshare/sample.ldif

we are not able to hold the AC as an object in Ldap server. so make sure you can specify the whole path of the AC's location.

4 Demo

The demo below assumes that apache webserver and ldap server are running on the same machine.

1. Generate Certification Authority for PKI and AA

   $ sh admin.sh ca
   generating CA
   Generating a 1024 bit RSA private key
   .....++++++
   .....++++++
   writing new private key to 'private/cakey.pem'
   Enter PEM pass phrase: ——————————>Enter some password - A
   Verifying - Enter PEM pass phrase: ——————————>repeat the password - A
   ——
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.
   ——
   Country Name (2 letter code) [US]:
   State or Province Name (full name) [Colorado]:
   Locality Name (eg, city) [Colorado Springs]:
   Organization Name (eg, company) [University of Colorado at Colorado Springs]:
   Organizational Unit Name (eg, section) [Info Share]:
   Common Name (eg, YOUR name) [Ganesh k Godavari]:Admin Tool
   Email Address [gkgodava@ncdcrx1.uccs.edu]:admintool@uccs.edu

Enter pass phrase for private/cakey.pem.enc:————————>repeat the password - A
writing RSA key
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'AAcakey.pem'
Enter PEM pass phrase: ————————————>Enter some password - B
Verifying - Enter PEM pass phrase: ————————————>repeat the password - B
—
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
—
Country Name (2 letter code) [US]:
State or Province Name (full name) [Colorado]:
Locality Name (eg, city) [Colorado Springs]:
Organization Name (eg, company) [University of Colorado at Colorado Springs]:
Organizational Unit Name (eg, section) [Info Share]:
Common Name (eg, YOUR name) [Ganesh k Godavari]:Attribute Authority
Email Address [gkgodava@ncdcrx1.uccs.edu]:aaAdmin@ncdcrx1.uccs.edu
Enter pass phrase for AAcakey.pem.enc: ————————————>repeat the password - B
writing RSA key
generating CA done make sure to backup the following files ./cacert.pem, private/cakey.pem, ./AAcacert.pem, and ./AAcakey.pem

password are used to protect your files from unauthorized viewing of data. Passwords are generally forgotten over a period of time, so i am removing the passwords for private key files of both PKI's and AA's.

2. Generate pkc for user gkgodava, edwardchow, and apache

$ sh admin.sh pkc gkgodava
genererating user PKI certificate
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'pkireq.pem'
—
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
—
Country Name (2 letter code) [US]:
State or Province Name (full name) [Colorado]:
Locality Name (eg, city) [Colorado Springs]:
Organization Name (eg, company) [University of Colorado at Colorado Springs]: dc=uccs,dc=edu————————>should be the values specified in the suffix field of the ldap server - A
Organizational Unit Name (eg, section) [Info Share]:
Common Name (eg, YOUR name) [Ganesh k Godavari]:

Email Address [gkgodava@uccs.edu]: ————————>should be username@<ldapservername> - A
Getting request Private Key
Generating certificate request
Using configuration from openssl.cnf

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'US'
stateOrProvinceName :PRINTABLE:'Colorado'
localityName :PRINTABLE:'Colorado Springs'
organizationName :PRINTABLE:'dc=uccs,edu'
organizationalUnitName:PRINTABLE:'Info Share'
commonName :PRINTABLE:'Ganesh k Godavari'
emailAddress :IA5STRING:'gkgodava@uccs.edu'
Certificate is to be certified until Jul 6 19:02:31 2005 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Enter Export Password:————————————>Enter some password - C
Verifying - Enter Export Password: ————————————>Repeat the password - C
...done

Password C is used to encrypt files in pkcs#12 format. The files are avialable are stored in the directory '/usr/local/apache/bin/P12'. One can use secure or non-secure ftp to connect to the server and download the file. illustrated below is an example of storing certificates in Internet Explorer.

Repeat the above procedure for edwardchow and webserver. Can use the default value except for Common Name (cn) and email address for the the other users.

(a) Open Internet Explorer and choose Tools>Internet Options, then click the Content tab. IE's Content tab is displayed in the Internet Options dialog.

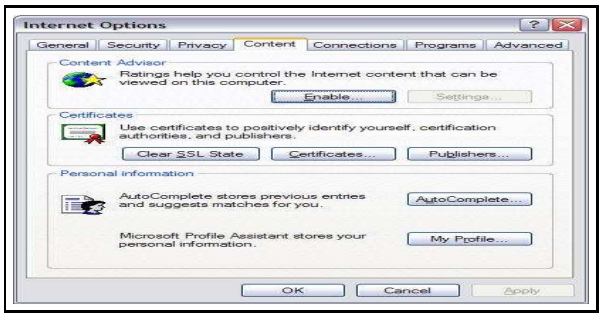

Figure 1: Content Tab

(b) Click the Certificates button. The Certificate Manager dialog opens with the "Personal" tab selected by default.

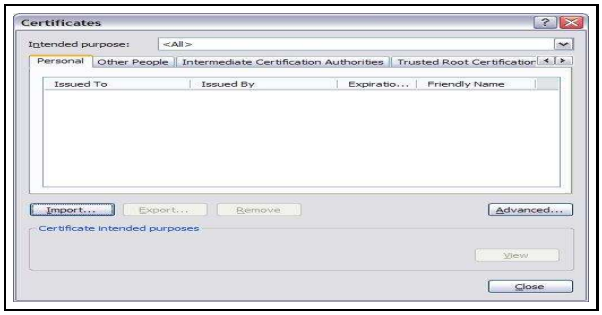

Figure 2: Certificate Tab

(c) Click the Import button inside the Personal tab

Figure 3: Import Prompt

(d) Click Next button, click on the Browse button and specify the file location. Make sure to set 'files of type' to 'Personal Information Exchange'

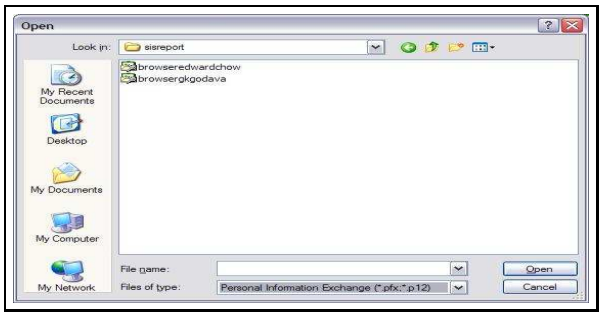

Figure 4: select the file to import

(e) Click Next button, enter 'password - C' in the 'Password:' textbox, and click Next button, set 'certificate store' to Personal

Figure 5: Certificate Store

(f) Click Next button and press Finish button.

3. Generate UserRoleSpecifcation Attribute Certificates
   Now let us create Attribute certificates for user edwardchow, and gkgodava with the roles Manager and Assist. Manager in the Group 'Info Share' of the organizational unit (OU) UCCS. Creating Attribute Certificate for user 'edwardchow'

   $ sh admin.sh ac AAcacert.pem attrfile.txt edwardchowUserRoleSpecification.pem
   AC Version= 2,
   AC serial 14
   20040706201743ZValid not before:
   20050706201743ZValid not after:
   AC Issuer: /C=US/ST=Colorado/L=Colorado Springs/O=University of Colorado at Colorado Springs/OU=Info Share/CN=Attribute Authorrity
   AC Holder: /C=US/ST=Colorado/L=Colorado Springs/O=University of Colorado at Colorado Springs/OU=Info Share/CN=Attribute Authorrity
   Attributes present: 1
   Attribute Number: 0
   Attribute NID: 650 , Name: Our example OID
   First attribute field holds:
   <?xml version="1.0" encoding="utf-8" standalone="yes"?>
   <!–======= SIS rbac example =======–>

<sis>
<Role>Manager</Role >
<Group>Info Share</Group >
<OU>UCCS</OU >
</sis>

_____

Signature Algorithm: md5WithRSAEncryption
68:72:2d:9f:41:f7:7e:70:2e:2f:94:11:93:6f:10:06:74:af:
5a:f7:b2:90:a6:1c:ac:bf:2c:6a:13:08:db:99:c2:38:5b:3a:
81:0e:00:22:8a:ae:e3:41:47:0a:ea:89:ce:5c:22:04:0b:ca:
78:b3:bb:a3:67:d1:ea:69:65:0e:24:a2:79:c6:da:3f:f2:9a:
ea:5e:bb:1c:1e:2b:2f:52:ef:38:ac:d5:b1:55:77:ee:94:99:
59:7d:4c:2b:bf:c1:5d:12:b1:d9:5c:b3:8f:1b:77:ab:5b:17:
2c:b4:11:1a:1b:c8:31:b4:94:e2:61:8a:0e:2c:6e:30:f5:63:
cb:05

Creating Attribute Certificate for user 'gkgodava'

$ sh admin.sh ac AAcacert.pem attrfile.txt gkgodavaUserRoleSpecification.pem
AC Version= 2,
AC serial 15
20040706202007ZValid not before:
20050706202007ZValid not after:
AC Issuer: /C=US/ST=Colorado/L=Colorado Springs/O=University of Colorado at Colorado Springs/OU=Info Share/CN=Attribute Authorrity
AC Holder: /C=US/ST=Colorado/L=Colorado Springs/O=University of Colorado at Colorado Springs/OU=Info Share/CN=Attribute Authorrity
Attributes present: 1
Attribute Number: 0
Attribute NID: 650 , Name: Our example OID
First attribute field holds:
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!–======= SIS rbac example =======–>
<sis>
<Role>Assist. Manager</Role >
<Group>Info Share</Group >
<OU>UCCS</OU >
</sis>

_____

Signature Algorithm: md5WithRSAEncryption
05:a5:5a:83:b5:c0:f2:1c:54:28:c4:b2:79:5b:76:61:62:c0:
c3:5c:76:af:56:00:ef:77:48:b1:ce:22:02:f2:b7:dd:4f:d0:
c2:e6:3e:dd:6a:14:d4:9a:5f:13:f9:33:db:e1:98:9a:c2:7c:
3d:bc:34:a0:b7:f3:bf:4d:14:99:bb:07:82:70:45:c8:30:a9:
ef:7f:af:c2:98:02:e7:51:50:44:b9:c9:9d:c2:ff:8d:a7:46:
24:6a:c2:28:3c:ed:64:04:aa:57:67:dc:7e:1b:35:29:4d:ed:
0c:13:75:ca:6c:0d:13:0e:a9:5a:69:89:24:6a:9d:af:02:cb:

f0:1b

4. Ldap Server
   Now we need to add information to the ldap server for SIS based Access Control.

   (a) Download OpenLdap server configuration file
   $ wget http://blanca.uccs.edu/ infoshare/demoSlapd.conf
   $ mv demoSlapd.conf /etc/openldap/
   $ mkdir /var/lib/ldap/UCCS1/
   $ chown -R ldap /var/lib/ldap/UCCS1
   $ service ldap restart

   (b) Adding user information to OpenLdap Server
   $ wget http://blanca.uccs.edu/ infoshare/demoLdap.lidf
   $ ldapadd -xv -D "cn=manager,dc=</hostname>,dc=edu" -W -f entries.ldif1 -h 127.0.0.1
   for more information on ldap commands please refer to OpenLdap documentation

   (c) checking for succesful storage of information
   Login to any Linux machine with ldap client running on it. type the following command $ ldapsearch -vLx -h </hostname>-b "ou=Info Share,o=University of Colorado at Colorado Springs" "(objectclass=*)"
   you must see information about Edward Chow, Ganesh Godavari displayed. now you have succesfully stored information in your ldap server.

5. Apache Server
   We need to configure Apache webserver for SIS based Access Control.

   (a) Download Apache configuration file httpd.conf file
   $ wget http://blanca.uccs.edu/ infoshare/demohttpd.conf
   $ edit the configuration file and add in server name for 'machine name' in the demohttpd.conf file
   $ mv demohttpd.conf /usr/local/apache/conf/httpd.conf
   $ edit the configuration file and set 'SSLVerifyClient' to require, and 'SSLVerifyDepth' to 1

   (b) Configure Apache to use the certificates generated by admin tool
   $ cp /usr/local/apache/bin/P12/cacert.pem /usr/local/apache/conf/ssl.crt/cacert.pem
   $ cp /usr/local/apache/bin/P12/apache-key.pem /usr/local/apache/conf/ssl.key/server.key
   $ cp /usr/local/apache/bin/P12/apache-cert.pem /usr/local/apache/conf/ssl.crt/server.crt

   (c) adding a test directory
   $ wget http://blanca.uccs.edu/ infoshare/demoDirectory.tar.gz
   $ tar -xvzf demoDirectory.tar.gz $ mv demoInfoShare /usr/local/apache/
   $ chmod -R 755 /usr/local/apache

   (d) Start Apache webserver
   $ /usr/local/apache/bin/apachectl startssl
   startssl is equivalent to starting apachectl -k start -DSSL. One way to overcome using '-DSSL' parameter is by editing the httpd.conf to remove the <IfDefine>section so that SSL will always be available.

6. Demonstration

   On your machine add the following information in your host file. DNS Queries will be resolved locally when added to the host file.
   Note: linux: /etc/hosts and Windows: windows-install-path\system32\drivers\etc\hosts

   128.198.61.19 sis-newjersey.csnet.uccs.edu sis-newjersey
   128.198.61.17 sis-connecticut.csnet.uccs.edu sis-connecticut
   128.198.61.15 sis-canada.csnet.uccs.edu sis-canada
   128.198.61.13 sis-nissc.csnet.uccs.edu sis-nissc

   Download the certificates from `http://blanca.uccs.edu/~infoshare/src/export\_certs\_demo.zip`

   Install the certificates into your favorite local browser

   Test by clicking on the links below

   Test1 Objective: test if the user can retrieve files and post files. retrieve operations in HTTP represent read permission, while post operation in HTTP represents write permission Two users alpha-sis-nissc and beta-sis-nissc try to access information from sis-connecticut.csnet.uccs.edu, later has the write permissions and both have read permissions. alpha-sis-nissc access information from sis-connecticut.csnet.uccs.edu
   √ `https://sis-connecticut.csnet.uccs.edu/level1/review.txt`
   the user can retrieve the file review.txt using HTTP-GET
   × `https://sis-connecticut.csnet.uccs.edu/level1/upload.html`
   the user cannot upload files using the HTTP-POST
   beta-sis-nissc access information from sis-connecticut.csnet.uccs.edu
   √ `https://sis-connecticut.csnet.uccs.edu/level1/review.txt`
   √ `https://sis-connecticut.csnet.uccs.edu/level1/upload.html`

   The user beta-sis-nissc has read and write permissions, so he can succesfully retreive and upload files.

   Test2 Objective: test if the user can retrieve files from multiple agencies. Using the role access defined by the organization in accordance to its organizational structure. User beta-sis-connecticut tries to access information from two agencies sis-nissc.csnet.uccs.edu and sis-canada.csnet.uccs.edu. beta-sis-connecticut has been given level2 directory access at sis-nissc.csnet.uccs.edu and level3 directory access at sis-canada.csnet.uccs.edu.
   √ `https://sis-nissc.csnet.uccs.edu/level2/review.txt`
   × `https://sis-canada.csnet.uccs.edu/level2/review.txt`
   √ `https://sis-canada.csnet.uccs.edu/level3/review.txt`

   Test3 Objective: test if the user can retrieve files from his/her agency using the role access defined by his/her organization in accordance to its organizational structure. User epsilon-sis-newjersey tries to access information from his organization sis-newjersey.csnet.uccs.edu
   √ `https://sis-newjersey.csnet.uccs.edu/level4/review.txt`

   NOTE √ succesful access × denied/forbidden access