Additional Technical Documentation

Kiosk and the QuickBooks Web Connector

Cornell University Chemistry Stockroom

John Sammis, CEO & Project Manager
Alex Michaluk, Primary Developer
GORGES

John Guttridge, President & Senior Technical Consultant
Nicole Tedeyan, Project Manager
Brightworks Computer Consulting

Version: September 6, 2013

Table of Contents

CUWebAuth Documentation

General Documentation

Here is Cornell's CUWebAuth documentation page:

https://confluence.cornell.edu/x/OgGw
CUWebAuth Documentation > CUWebAuth 2.0

This next page, one level deeper, may be what you specifically need to get started:

https://confluence.cornell.edu/x/ZpPAB
CUWebAuth Documentation > CUWebAuth 2.0 > Integrating CUWebAuth With Your Application

The home page above states, "Users of CUWebAuth should join CUWA-ANNOUNCE-L to pick up announcements for new releases and security issues. You are also invited to join CUWA-L, a discussion list for CUWebAuth administration issues." Here's more about those lists:

https://confluence.cornell.edu/x/DIQ_BQ

As an FYI, here's CIT's "About" pages on CUWebLogin, the user's side of the authentication, I believe:
http://www.it.cornell.edu/services/cuweblogin/about.cfm

If you need to contact someone at CIT to discuss CUWebAuth at a technical level, email <idmgmt@cornell.edu>. For instance, you may need some test NetIDs.

PRE BUILT BINARIES

https://confluence.cornell.edu/display/CUWAL/Installing+Prebuilt+Binaries

Issues with Logout

Logging out of a Cornell CUWebAuth session is unfortunately not supported (see:https://confluence.cornell.edu/display/CUWAL/FAQ). Further research into Kerberos in general revealed that this is a design flaw.

Alex and I hacked together a solution so we can truly guarantee a CUWebAuth logout for the CUChemLab kiosks. We found out that there are FIVE unique cookies that identify web server and CUWebAuth sessions. However from the domain ktest.stockroom.chem.cornell.edu, we only had permission to delete FOUR of the cookies. To remove the fifth cookie, we had to do it from a domain that included ".login.cornell.edu".

One solution would be to have a domain such as ktest.stockroom.chem.login.cornell.edu formally registered with a DNS (domain name service) record. However that would involve many permissions from the Cornell IT administrators, and there was no guarantee of approval.

Since the stockroom kiosks are dedicated machines, the solution is to add this domain to the local "hosts" file that tricks these computers into thinking that "ktest.stockroom.chem.login.cornell.edu" is a real domain and points to the stockroom server.

Here is the single line that must be added to the stockroom kiosk computers "hosts" file:

128.253.34.142 ktest.stockroom.chem.login.cornell.edu

This solution has been tested on several GORGES computers, and the "logout" link works perfectly. On my Windows development computer, the hosts file is located here:

C:\Windows\System32\Drivers\etc\hosts

LAMP Server Administration

Server Information

domain: ktest.stockroom.chem.cornell.edu
IP addr: 128.253.34.142

Install Packages

// update packages
yum update
// install supporting packages, including Apache and PHP
yum install lynx wget httpd php mysql mysql-server php-mysql mod_ssl mod_auth_kerb openssh krb5-workstation krb5-appl-clients openssh-clients
yum install php-soap php-xml
// start and configure MySQL
service mysqld start
chkconfig mysqld on
mysql_secure_installation
(enter new root password; remove anonymous user; remove test DB; disallow remote login)
// start Apache service
service httpd start
chkconfig httpd on
// allow HTTP access through firewall
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
// test in web browser: http://ktest.stockroom.chem.cornell.edu
cd /var/www/html
vi index.php
Test
// test in browser: http://ktest.stockroom.chem.cornell.edu

CUWebAuth Installation and Configuration

// get 32-bit or 64-bit status
uname -a
(displays ".i686" for 32-bit or x86_64 for 64-bit)
// get Apache version
yum info httpd
(says that Apache is version 2.2.15)
// download RedHat 6 64-bit system apache (2.2.15) mod for CUAuthWeb
// https://cuwabuild.cit.cornell.edu/hudson/job/cuwal2-distro/label=RedHat6-64bit/lastSuccessfulBuild/artifact/cuwal-2.2.1.210/apache/.libs/mod_cuwebauth.so
cd /root
wget https://cuwabuild.cit.cornell.edu/hudson/job/cuwal2-distro/label=RedHat6-64bit/lastSuccessfulBuild/artifact/cuwal-2.2.1.210/apache/.libs/mod_cuwebauth.so
mv mod_cuwebauth.so /etc/httpd/modules/
chmod a+x /etc/httpd/modules/mod_cuwebauth.so
vi /etc/httpd/conf/httpd.conf
(append)
LoadModule cuwebauth_module modules/mod_cuwebauth.so
// Kerberos file
cp -p /etc/krb5.conf /etc/krb5-ORIG.conf
vi /etc/krb5.conf
[libdefaults]
default_realm = CIT.CORNELL.EDU

```
  kdc_timesync = 1
  ccache_type = 4
 [realms]
  CIT.CORNELL.EDU = {
   kdc = kerberos.login.cornell.edu:88
   kdc = kerberos2.login.cornell.edu:88
   admin_server = kerberos.login.cornell.edu:749
   default_domain = cit.cornell.edu
  }
  CORNELL.EDU = {
   kdc = ad1.cornell.edu:88
   kdc = ad2.cornell.edu:88
   kdc = ad3.cornell.edu:88
   kdc = ad4.cornell.edu:88
   kdc = ad5.cornell.edu:88
   kdc = ad6.cornell.edu:88
   default_domain = cornell.edu
  }
  GUEST.CORNELL.EDU = {
   kdc = obsidian1.cit.cornell.edu:88
   kdc = obsidian2.cit.cornell.edu:88
   admin_server = obsidian1.cit.cornell.edu:749
   default_domain = guest.cornell.edu
  }
  GUESTX.CORNELL.EDU = {
   kdc = jade.cit.cornell.edu:88
   admin_server = jade.cit.cornell.edu:749
   default_domain = guestx.cornell.edu
  }
 [domain_realm]
  cornell.edu = CIT.CORNELL.EDU
  .cornell.edu = CIT.CORNELL.EDU
  .mail.cornell.edu = CIT.CORNELL.EDU
//  added an unused development file /etc/krb5.conf-DEVELOPMENT on server
//  get Kerberos 5 keytab file
//  requested from http://serviceidmanager.cit.cornell.edu (must be CU staff)
//  contacted Oliver for keytab file
//  file: https.kiosk.stockroom.chem.cornell.edu.keytab
//  url: https://kiosk.stockroom.chem.cornell.edu
# mv ~/https.kiosk.stockroom.chem.cornell.edu.keytab /etc/httpd/conf/
# chown apache:apache /etc/httpd/conf/https.kiosk.stockroom.chem.cornell.edu.keytab
# chmod 600 /etc/httpd/conf/https.kiosk.stockroom.chem.cornell.edu.keytab
# vi /etc/httpd/conf/httpd.conf
 (append)
 CUWAKeytab /etc/httpd/conf/https.kiosk.stockroom.chem.cornell.edu.keytab
 CUWAKerberosPrincipal https/kiosk.stockroom.chem.cornell.edu@CIT.CORNELL.EDU
 CUWAWebLoginURL  "https://web1.login.cornell.edu https://web2.login.cornell.edu https://web3.login.cornell.eduhttps://web4.
login.cornell.edu"
 <Directory "/var/www/html">
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
```

```
   AuthName CORNELL
   AuthType all
   require valid-user
  </Directory>
  LogLevel debug
  ServerName ktest.stockroom.chem.cornell.edu
  RequestHeader unset REMOTE_USER early
# service httpd restart
```

Self-Signed Certificate

```
# cd /etc/httpd/
# mkdir ssl
# cd ssl
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout stockroom.key -out stockroom.crt
  (enter US, New York, Ithaca, CUCHemLab, stockroom, ktest.stockroom.chem.cornell.edu)
# cd /etc/httpd/conf.d/
# cp -p ssl.conf ssl.conf-ORIG
# vi ssl.conf
  (change to:)
  SSLCertificateFile /etc/httpd/conf/ssl/stockroom.crt
  SSLCertificateKeyFile /etc/httpd/conf/ssl/stockroom.key
# cd /etc/httpd/
# mkdir ssl
# cd ssl
# cp /etc/pki/tls/openssl.conf stockroom.cnf
# cat >> stockroom.cnf
  [ req_ext ]
  subjectAltName = @alt_names
  [ alt_names ]
  DNS.1 = kiosk.stockroom.chem.login.cornell.edu
  DNS.2 = ktest.stockroom.chem.cornell.edu
  DNS.3 = ktest.stockroom.chem.login.cornell.edu
  # openssl req -x509 -nodes -config stockroom.cnf -extensions req_ext -days 3650 -newkey rsa:2048 -keyout
stockroom.key -out stockroom.crt
  (enter US, New York, Ithaca, Cornell University, Chemistry Stockroom, kiosk.stockroom.chem.cornell.edu,
chemstockroom@cornell.edu)
# openssl x509 -in stockroom.crt -noout -text
  (verify Signature Alternative Names)
# cd /etc/httpd/conf.d/
# cp -p ssl.conf ssl.conf-ORIG
# vi ssl.conf
  (change to:)
  SSLCertificateFile /etc/httpd/conf/ssl/stockroom.crt
  SSLCertificateKeyFile /etc/httpd/conf/ssl/stockroom.key
```

Apache SSL Forwarding

```
# vi /etc/httpd/conf/httpd.conf
<Directory "/var/www/html">
  RewriteEngine on
  RewriteCond %{SERVER_PORT} !443
  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [R,L]
</Directory>
```

Mail Services

yum install sendmail mailx
service sendmail start
chkconfig sendmail on
mail mclark@gorges.us (for testing)
cd /etc/mail/
cp -p sendmail.cf sendmail.cf-2013-06-10
vi sendmail.cf
  (change from DS to DSappsmtp.mail.cornell.edu)
mail -s "test" mclark@gorges.us
  (view /var/log/maillog to see if Relay worked)
vi /etc/php.ini
  (change sendmail_path to:)
  sendmail_path = /usr/sbin/sendmail -t -i -r chemstockroom@cornell.edu
# service httpd restart

Testing

vi /var/www/html/index.php
  (do a var_dump of $_SERVER in a simple HTML page)
// restart browser, enter url:
https://kiosk.stockroom.chem.cornell.edu
// choose to accept certificate
// notice that browser is automatically forwarded to https://web1.login.cornell.edu
// enter NetID and password
// web browser should be automatically relocated to:
https://kiosk.stockroom.chem.cornell.edu
// debug messages are in /var/log/httpd/error.log

Sub-folder solution

Rasmus recommended only adding CUAuthWeb restrictions to a sub-folder, and within this sub-folder the $_SERVER['CUWA_*'] values are copied to session variables and then forwarded to a higher folder level. For security reasons, it is best to hash these values when stored in cookies or session variables.

CRON JOB

See ActiveCollab file named CRON: check-last-run.sh
Place into /etc/cron.hourly/check-last-run.sh

Kuali & Account Number Formatting

The Kuali account in the customer name field is stored with the following format:

IT_1234567_*****6545**********

Key points:

1. The account number is separated from the rest of the Kuali account string with underscores.
2. Placeholders are asterisks, not the number 0.
    a. Note that according to the Kuali documentation (http://www.dfa.cornell.edu/kfs/coa/accountstring.cfm), some fields can be *up to* the specified number of alphanumeric characters. Placeholders are appended to the end of the field if the field does not take up all of the allowed alphanumeric characters.

In summary:

Field	Characters	Example
Chart	2 exactly	IT
Account	7 exactly	975H7K9
Sub-account	Optional, up to 5	456
Object	Required, 6545	6545
Sub-object	Optional, up to 3	
Project code	Optional, up to 10	PROJECT
Org Ref ID	Optional, up to 8	ORGREF1

With the example shown above, the Kuali account string that is passed to QB would be:
IT_975H7K9_456**6545*** PROJECT*** ORGREF1*

Administrative User Interface

https://kiosk.stockroom.chem.cornell.edu/admin.php
usr: cuchem