LessonsLearned

100+ websites

In maintaining an application which actively queries hundreds of different websites, several practices which require configuration changes have discovered.

User Agent

Several websites responded with 500 status code when presented with the default User-Agent header. One website sent a 200 status code but the html content of the page was truncated with "500 server error" For maximum compatibility, use a standard web browser user-agent string.

http.useragent = Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1

Cookie Policies

Very few websites support anything other than base Netscape cookies.

http.protocol.cookie-policy = compatibility

Cookie Header

Although some websites support multiple Cookie headers, many do not. The documentation for http.protocol.single-cookie-header is misleading. This parameter determines how Cookie headers are sent in the request. Multiple Set-Cookie headers are always supported.

http.protocol.single-cookie-header = true

Post/Redirect/Get

Post redirecting to Get turns out to be a common practice. Contrary to the RFC 2616 recommendation, this practice relies on the "broken" behavior of major web browsers. The query portion for the GET comes strictly from the Location header returned with the 302 response to the POST.

Certificates

The certificate database ($JAVA_HOME/lib/security/cacerts) in the standard java distribution contains one third of the root certificates that are present in Firefox or Internet Explorer. The following script can update the cacerts database.

```
#/bin/sh
# How to use this script
# 1.  Create temp dir
# 2. In firefox, select Tools/Options/Advanced/Encryption/View Certificates
# 3. In the Certificate Manager Dialog, Authorities tab; select all certificates and press Export…
# 4. Select OK and rename files as necessary
# 5. Add all firefox trusted authorities not already in cacerts by running this file in the temp directory

rm log
# copy the current certificate database
cp "$JAVA_HOME/jre/lib/security/cacerts" .
# Determine current list of authorities
keytool -list -v -keystore cacerts -storepass changeit |grep "Issuer:"|sort > before

for file in *.crt; do
  alias=${file%%.crt}
  logfile="${alias}".log
  echo "================== ${alias} ==================" > "${logfile}"
  echo "==printcert" >> "${logfile}"
  keytool -printcert -file "${file}" -keystore cacerts -storepass changeit >> "${logfile}" 2>&1
  owner=$(grep "Owner:" "${logfile}" | sed -e "s/Owner: //")
  issuer=$(grep "Issuer:" "${logfile}" | sed -e "s/Issuer: //")
  # is this a root certificate?
  if [ "${owner}" = "${issuer}" ] ; then
    # determine an alias
    for (( i= 0; i<10; i= i+1 )) ; do
      echo "==list ${alias}" >> "${logfile}"
      if ( keytool -list -keystore cacerts -storepass changeit -alias "${alias}" >> "${logfile}" 2>&1 ) then
        alias="${file%%.crt}(${i})"
        continue;
      fi
      break;
    done
    # import the key
    echo "==import" >> "${logfile}"
    keytool -import -file "${file}" -keystore cacerts -storepass changeit -alias ${alias} >> "${logfile}" 2>&1
<<response
yes
yes
response
    # delete any key which is a duplicate
    if ( grep "Certificate already exists in keystore under alias" "${logfile}" > /dev/null ); then
      echo "==delete" >> "${logfile}"
      keytool -delete -keystore cacerts -storepass changeit -alias ${alias} >> "${logfile}" 2>&1
    fi
  fi
  cat "${logfile}" >> log
done
#Determine new list of authorities
keytool -list -v -keystore cacerts -storepass changeit |grep "Issuer:"|sort > after
#Determine change list of authorities
diff before after
```