RHEL 5.2 Build Notes for PKIF and PKIFXML

Install from the RPM packages available through the software installation system

- development tools group (includes autotools, etc.)
- openssl development package (for curl)
- curl development package
- gtk2 development package
- openldap development package
- NSS development package (if using system NSS)
- libtool-ltdl development package
- pcre development package (required by swig)

Build from source and install:

- boost (no test, graph or python is necessary) 1.44
- wxWidgets 2.8.11 current stable
- swig 2.0.1
- xerces version 3.1.1
- xalan (from repository https://svn.apache.org/repos/asf/xalan/c/trunk revision 1005215)
- xml security version 1.5.1

Decide where you'll be installing PKIF. For the sake of this document, it'll be /usr/local/pkif. There's nothing outside these instructions that assumes this is the location, but this is becoming the recommended way to install PKIF. By installing outside the locations where FHS-compliant RPMs will copy libraries, libtool can use rpath in the ELF libraries it produces to make life easier at runtime.

0. Download and expand the PKIF archives. Be sure to use the -a flag when invoking unzip. This fixes the line endings on most of the text files. For ease of reference in paths farther down this document, here's how I'm unpacking the files

```
mkdir pkif-build
cd pkif-build
unzip -a ../downloads/PKIF2_X_X-source.zip
mv PKIFv2_1_5-source pkifv2
unzip -a ../downloads/PKIFv2_X_X -srclib.zip
mv PKIFv2_1_5-srclib pkifv2/srclib
mv PKIFv2_1_5-srclib-nss/nss pkifv2/srclib
rm -rf PKIFv2_1_5-srclib-nss
unzip -a ../downloads/PKIFv2_X_X-PKIFXML-src.zip
mv PKIFv2_1_9-PKIFXML-src pkifv2/applications/PKIFXML
unzip -a ../downloads/JPKIF2_2_2-source.zip
mv JPKIF2_1_9-source pkifv2/applications/PKIFXML
```

At the end of this step, my tree looks like this:

```
[tester@wgbeier-rhel52test pkif-build]$ tree -d -L 3
.
`-- pkifv2
   |-- applications
   |    |-- JPKIF
   |    |-- PKIFCMS
   |    |-- PKIFResources
   |    |-- PKIFTSP
   |    |-- PKIFERS
   |    |-- PKIFSCVP
   |    |-- PKIFXML
```

```
|   |-- SWIG
|   `-- temp
|-- devel
|   |-- build_utility
|   |-- obj
|   |-- projects
|   `-- src
|-- dist
|   |-- 2003
|   `-- 2005
`-- srclib
    |-- IETFASN1Library
    |-- ObjLib
    |-- cryptopp
    |-- lib
    |-- libcurl
    |-- nss
    `-- projects
```

1. Build the dependencies. In the case of RHEL5.2, you need to build ObjLib, IETFASN1Library, cryptopp. The system nss and libcurl are fine on RHEL5.2. (Note: If for some reason the system-installed nss is insufficient, the one available for download with PKIF can be built using command below. Newer versions of NSS should work too; the one we ship is `nss-3.12.4-with-nspr-4.8`.)

```
cd pkifv2/srclib/ObjLib/
make -f Makefile.nix
cd ../IETFASN1Library
make -f Makefile.nix
```

If building nss provided with PKIF.

```
cd ../nss
perl grabbuild.pl . build release
```

{note that some error messages may go by during this step. At the end, if the script reports a successful build, the pieces of nss that PKIF needs should be in working order. The messages are generally related to test cases and the like that were removed from the archive to make the download easier and not needed for PKIF.}

```
cd ../cryptopp
make libcryptopp.a
cp libcryptopp.a ../lib
```

2. Build and install the core PKIF library along with its headers.

```
cd ../../devel
sh build_utility/autogen.sh
cd obj
 ../configure --prefix=/usr/local/pkif CFLAGS=-g CXXFLAGS=-g
make
sudo make install
sudo cp ../../srclib/nss/mozilla/dist/Linux2.6_x86_glibc_PTH_OPT.OBJ/lib/* /usr/local/pkif/lib
```

Interesting configure options:
--prefix: where make install will install to, and what libtool will use for rpath
--with-boostconf: how boost was configured when built and installed. This can be found by inspecting the name of the boost libraries.
--with-nssconfig: look at the mozilla/dist directory under srclib to find this
--with-srclib: Where the built copy of the srclib directory shown in the tree at the top of this document lives.

3. Build and install PKIFCMS along with its headers:
(from applications/PKIFCMS)

```
sh autogen.sh
```

```
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

4. Build and install PKIFTSP along with its headers:
(from applications/PKIFTSP)

```
sh autogen.sh
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

5. Build and install PKIFERS along with its headers:
(from applications/ PKIFERS)

```
sh autogen.sh
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

6. Build and install PKIFSCVP along with its headers:
(from applications/PKIFSCVP)

```
sh autogen.sh
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

7. Build PKIFResources along with its headers.
(from applications/PKIFResources)

```
sh autogen.sh
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

prefix, boostconf and srclib have the same meanings as for PKIF core. The build process for PKIFResources requires libltdl and all of wxwidgets. The configuration serialization portion, though, runs with only the non-gui portions of that; we just don't have its build process separated as far as we'd like to yet.

8. Build and install PKIFXML along with its headers:
(from applications/PKIFXML/devel)

```
sh autogen.sh
cd obj
```

```
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

9. Build and install JPKIF along with its headers:
(from applications/SWIG)
```
sh JPkifSwig.sh
```
(from applications/SWIG/jpkif_module)
```
sh autogen.sh
cd obj
export PKIF_HOME=/usr/local/pkif
../configure --prefix=/usr/local/pkif
make
sudo make install
```

configure options have the same meanings as for PKIF core.

Alternatively, Eric Karlson's bash script (Supplied by Eric Karlson pkif@ekarlson.net) can be used to build everything. It's available for download from

http://pkif.sourceforge.net/buildpkif.sh

Follow these steps to use buildpkif.sh:

1) Create a stock RedHat 5.2 system – the install must include all the standard GCC tools, development libraries, etc
2) Download and install the SUN JDK 5.0 (http://java.sun.com/javase/downloads/index_jdk5.jsp)
3) Create a directory named "pkif" in a suitable location on your hard drive
4) Copy the attached script into the "pkif" directory, and change JAVA_HOME to the location where java was installed.
5) Create a directory under the "pkif" directory named "downloads"
6) Place the following files into the "downloads" directory
    a. boost_1_44_0.tar.gz (http://sourceforge.net/projects/boost/files/)
    b. swig-2.0.1.tar.gz (http://sourceforge.net/projects/swig/files/)
    c. xerces-c-src_3_1_1.tar.gz (http://xerces.apache.org/xerces-c/download.cgi)
    d. Export xalan using the following comman:
        1. svn export https://svn.apache.org/repos/asf/xalan/c/trunk --revision 1005215
        2. Rename trunk to Xalan-C_1_11_0-src
        3. Create Xalan-C_1_11_0-src.tar.gz using tar czf Xalan-C_1_11_0-src.tar.gz Xalan-C_1_11_0-src
    e. xml-security-c-1.5.1.tar.gz (http://santuario.apache.org/dist/c-library/old/)
    f. wxGTK-2.8.11.tar.gz (http://sourceforge.net/projects/wxwindows/files/wxGTK/2.8.11/wxGTK-2.8.11.tar.gz/download)
    g. JPKIF2_X_X-source.zip
    h. PKIFv2_X_X-source.zip
    i. PKIFv2_X_X-srclib-nss.zip
    j. PKIFv2_X_X-srclib.zip
    k. PKIFv2_X_X-PKIFXML-src.zip
7) "cd" into the "pkif" directory
8) Add java bin directory to the path
9) Run the "buildpkif.sh" script

At this point, you should be ready to go. To try it out with one of the samples, unpack them in your home directory, build and run:
For example, from the verification sample:

0. Change PKIF_HOME in the makefile to /usr/local/pkif
1. make
2. Make sure PKIF and xml-security-c are in the library search path.
3. ./simpleVerify <signed.xml should show output like:

```
PKIF is trusting:
********************************************************************************
*****

simpleVerificationSample
Current time: 20081020221550Z
Validation time of interest: 20081020221550Z
Root certificate information
    Subject: cn=Trust Anchor,o=Test Certificates,c=US
    Issuer: cn=Trust Anchor,o=Test Certificates,c=US
    Serial number: 0x01
Target certificate information
    Subject: cn=Valid EE Certificate Test1,o=Test Certificates,c=US
    Issuer: cn=Good CA,o=Test Certificates,c=US
    Serial number: 0x01

********************************************************************************
*****
--------------------------------------------------------------------------------
Printing information from path validation results:
--------------------------------------------------------------------------------
- Path successfully validated
- Basic checks successfully performed
- Cert signatures successfully verified
- Most severe revocation status: NOT REVOKED
- Explicit policy indicator: FALSE
- Authority constrained policy table
    + Row: 0
        * 2.5.29.32.0
        * 2.16.840.1.101.3.2.1.48.1
        * 2.16.840.1.101.3.2.1.48.1
- User constrained policy set
    + 2.16.840.1.101.3.2.1.48.1
- Authority constrained policy set
    + 2.16.840.1.101.3.2.1.48.1

--------------------------------------------------------------------------------
Printing information from certificate path
--------------------------------------------------------------------------------
- Discovered 1 total paths.
- Discovered 0 paths that failed basic validation checks.
- Returned 1 paths for external inspection.
- Dumping certificate path and certificate status info
    +  Certificate #1
        * Issuer DN    : cn=Trust Anchor,o=Test Certificates,c=US
        * Serial Number    : 0x02
        * Subject DN    : cn=Good CA,o=Test Certificates,c=US
        * Not Before    : 20010419145720Z
        * Not After    : 20110419145720Z
            + Source: LOCAL
            + Builder score: 11425
            + Is not a trust anchor
            + diagnostic code: 0 : Success
            + Basic checks successfully performed
            + Signature successfully verified
```

```
        + Revocation status: NOT REVOKED
        + Revocation source #1
           - Revocation source error code: 0 : Success
           - Revocation source type: 1
           - Revocation source status: NOT_REVOKED
           - CRL #1
              + CRL issuer: cn=Trust Anchor,o=Test Certificates,c=US
              + thisUpdate: 20010419145720Z
              + nextUpdate: 20110419145720Z
   +  Certificate #2
      * Issuer DN    : cn=Good CA,o=Test Certificates,c=US
      * Serial Number    : 0x01
      * Subject DN    : cn=Valid EE Certificate Test1,o=Test Certificates,c=US
      * Not Before    : 20010419145720Z
      * Not After    : 20110419145720Z
         + Source: LOCAL
         + Builder score: 0
         + Is not a trust anchor
         + diagnostic code: 0 : Success
         + Basic checks successfully performed
         + Signature successfully verified
         + Revocation status: NOT REVOKED
         + Revocation source #1
            - Revocation source error code: 0 : Success
            - Revocation source type: 1
            - Revocation source status: NOT_REVOKED
            - CRL #1
               + CRL issuer: cn=Good CA,o=Test Certificates,c=US
               + thisUpdate: 20010419145720Z
               + nextUpdate: 20110419145720Z

Signature verified.
<PurchaseOrder>
   <Company>Widgets.Org</Company>
   <Product>A large widget</Product>
   <Amount>$16.50</Amount>
   <Due>16 January 2010</Due>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
         <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <ds:Reference URI="#xpointer(/)">
            <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
               <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315#WithComments"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>Pxb61Src1V56jb4lUA2nqz0WfWA=</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>

<ds:SignatureValue>m1a+Env/Y/Hzqg2sMOkQj8Rjdf1wBSprgwM1Hon+FDFWNcS1f3i6JKxJPTHJiMRJmI
0bPzo9J/XE
Clul00wCXC3ZWaESUww80MclRAl2ztgO9GaxX2v4GOvin+V715ee0m5lSQ0ZdX+kjbTo3m+fjpv0
+Mx+DyxNcaEnxk8XkaY=</ds:SignatureValue>
   <ds:KeyInfo>
<ds:X509Data>
```

<ds:X509Certificate>MIICajCCAdOgAwIBAgIBATANBgkqhkiG9w0BAQUFADA7MQswCQYDVQQGEwJVUzE
aMBgGA1UEChMRVGVzdCBDZXJ0aWZpY2F0ZXMxEDAOBgNVBAMTB0dvb2QgQ0EwHhcNMDEwNDE5MTQ
1NzIwWhcNMTEwNDE5MTQ1NzIwWjBOMQswCQYDVQQGEwJVUzEaMBgGA1UEChMRVGVzdCBDZXJ0aWZ
pY2F0ZXMxIzAhBgNVBAMTGlZhbGlkIEVFIENlcnRpZmljYXRlIFRlc3QxMIGfMA0GCSqGSIb3DQEBAQUAA4G
NADCBiQKBgQCtpKu/a6Co7KcKOymboEA+MmgoryXHT1dxExmQ1lO7yah2L8j8RG6ox5Tr37TV8Y21ti3Mo
pcFH+iXDSX31fixsYCZkcpjMI4kbjXmjGOeFKu1vnbBmcb5JBISiUeg22tIRFoJ4zThi3GLVecGijyOVReA5LiPy
mEKG7fAB3241wIDAQABo2swaTAfBgNVHSMEGDAWgBS3LqaCy8LIvKh7J0TXNTPfmhWUxzAdBgNVHQ4E
FgQUOsyUZQyFqTzB4K9RMyoUSI+ekVswDgYDVR0PAQH/BAQDAgTwMBcGA1UdIAQQMA4wDAYKYIZIAW
UDAgEwATANBgkqhkiG9w0BAQUFAAOBgQCkaGfCqYi0681n9Dit36lg3U/9gTZoNqPMaAaLUQV3Crzxx2MG
InhTyKchYydbV8HD89N2jzzYq7J2KM/ZEAfjskCdsj1SiMNkbYZe3rZZOldrPCGFgzUGTNakQxkpxU5j7plivQic
/OZ7+mMTi0fnjGRi9M+aa744VmH6FgCt1w==</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</PurchaseOrder>

Similarly, ./simpleVerify <invalid.xml should show:
PKIF is rejecting:
****
*****

simpleVerificationSample
Current time: 20081020221834Z
Validation time of interest: 20081020221834Z
Root certificate information
    Subject: cn=Trust Anchor,o=Test Certificates,c=US
    Issuer: cn=Trust Anchor,o=Test Certificates,c=US
    Serial number: 0x01
Target certificate information
    Subject: cn=Invalid EE Signature Test3,o=Test Certificates,c=US
    Issuer: cn=Good CA,o=Test Certificates,c=US
    Serial number: 0x02

****
*****

Printing information from path validation results:

- Path not successfully validated
- Diagnostic code: 3006 -> PATH_SIGNATURE_VERIFICATION_FAILED
- Basic checks successfully performed
- Cert signatures not successfully verified
- Most severe revocation status: NOT CHECKED
- Explicit policy indicator: FALSE
- Authority constrained policy table
    + Row: 0
        * 2.5.29.32.0
        * 2.16.840.1.101.3.2.1.48.1
        * 2.16.840.1.101.3.2.1.48.1
- User constrained policy set
    + 2.16.840.1.101.3.2.1.48.1
- Authority constrained policy set
    + 2.16.840.1.101.3.2.1.48.1
- Certificate with following information failed with diagnostic code 3006 :
PATH_SIGNATURE_VERIFICATION_FAILED
    + Issuer DN   : cn=Good CA,o=Test Certificates,c=US
    + Serial Number   : 0x02
    + Subject DN   : cn=Invalid EE Signature Test3,o=Test Certificates,c=US
    + Not Before   : 20010419145720Z
    + Not After   : 20110419145720Z

```
-------------------------------------------------------------------------------
Printing information from certificate path
-------------------------------------------------------------------------------
- Discovered 1 total paths.
- Discovered 0 paths that failed basic validation checks.
- Returned 1 paths for external inspection.
- Dumping certificate path and certificate status info
   +  Certificate #1
      * Issuer DN    : cn=Trust Anchor,o=Test Certificates,c=US
      * Serial Number    : 0x02
      * Subject DN    : cn=Good CA,o=Test Certificates,c=US
      * Not Before    : 20010419145720Z
      * Not After    : 20110419145720Z
         + Source: LOCAL
         + Builder score: 11425
         + Is not a trust anchor
         + diagnostic code: 0 : Success
         + Basic checks successfully performed
         + Signature successfully verified
         + Revocation status: NOT CHECKED
   +  Certificate #2
      * Issuer DN    : cn=Good CA,o=Test Certificates,c=US
      * Serial Number    : 0x02
      * Subject DN    : cn=Invalid EE Signature Test3,o=Test Certificates,c=US
      * Not Before    : 20010419145720Z
      * Not After    : 20110419145720Z
         + Source: LOCAL
         + Builder score: 0
         + Is not a trust anchor
         + diagnostic code: 3006 : PATH_SIGNATURE_VERIFICATION_FAILED
         + Basic checks successfully performed
         + Signature not successfully verified
         + Revocation status: NOT CHECKED

An error occurred during verification.
   Message: DSIGSignature::verify() - no verification key loaded and cannot determine from
KeyInfoResolver
```