CalNet InCommon-Comodo Certificate Service

Tips for other campuses
If you are contemplating the InCommon Certificate service at your campus, feel free to borrow advice from UC Berkeley on Implementing the InCommon Certificate Service.

Contents
- Server certificate articles in the Comodo KB
- General FAQ
- Departmental Certificate Administrator (DCA) FAQ
- Server certificate articles in the Comodo KB
  - SSL Trouble Shooter
  - CSR Generation
  - SSL Installation Instructions
- General FAQ
  1. Where can I learn more about this program?
     See the InCommon Certificate Service page. Note also the Support link on that page.
  2. When is this program ready for campus?
     We can issue certificates now.
  3. What is the procedure for a campus unit to acquire SSL certs?
     Send your request along with a CSR (created using a 2048-bit or larger public key), and specifying the server software type (e.g. Apache, Tomcat, Microsoft IIS 5.x+ or, if unknown or generic, OTHER), to calnet-pki@lists.b.e and CalNetOps staff will handle the request and issue the certificates. We have implemented the InCommon-Comodo CA Service's ability to delegate some PKI administration to departmental authorities. See the DCA FAQ section of this page. This distributed administrative model has been discussed and implemented in coordination with the campus CalNetIdM and developer steering general security committee.
  4. Why does my server’s Security Contact need to be involved?
     Before we have fully implemented distributed administration, we are using the Security Contact application to help to validate requests for server certificates. In light of this, you might want to forewarn the folks on your Security Contact mailing list about any pending certificate request for an IP address for which they have security notification responsibility.
  5. Does this system have the capability to do Subject Alternative Name (SAN) certificates where we can use one certificate with multiple DNS hostnames per IP address?
     Yes, the following types of certificates are supported: Comodo EV SGC SSL (EV/SAN), Comodo EV Multi Domain SSL, InCommon Wildcard SSL Certificate, InCommon SSL, InCommon Intranet SSL (secure internal servers using either a full server name or a private IP address), InCommon Unified Communications Certificate (UCC/SAN), InCommon Multi Domain SSL, Corporate Secure Email Certificate.
  6. What are the available lifetimes for certificates?
     We can issue 1-, 2-, or 3-year certificates. We will provision a 3-year certificate unless requested otherwise.
  7. How does Comodo handle certificate revocation lists (CRLs)?
     See this Comodo KB article and also note that each certificate provisioned will have a X509v3 CRL Distribution Points entry for live access to the current CRL.

The InCommon Certificate Service will expand to cover these certificate needs over time. Until then, for private, CalNetAD CA-signed certificates, please continue to use the CalNetPKI service offered by IST Enterprise Windows.
8. **What is the major difference between UCC/SAN and Multi-Domain/SAN certificates (MDC)?**

The main (and perhaps only) difference is that the MDC can have the Subject CN (or primary domain name) set to a group name: essentially a non-valid domain name. All of the requested FQDNs will appear as dnsName entries in the SubjectAltName (SAN) extension. The UCC certificate is identical in that the requested FQDNs are in the SAN field, but it also contains a valid FQDN as the CN in the Subject. Other than this, these two types of certificates appear to be functionally equivalent.

9. **How do I generate a CSR and install the signed certificate?**

For help with generating a CSR and other certificate issues, consult the Comodo Knowledge Base for your web-server type. Note that for UCC/SAN or Multi-Domain/SAN certificates the CSR you generate only needs to be for the single Common Name domain, aka the Primary Domain Name. Additional domains that you may require in the Subject Alternative Name will be added at the time of provisioning the certificate, but in any case should always be listed in your written request to calnet-pki@lists.berkley.edu or to your Departmental Certificate Administrator. Note also that you must create at least 2048-bit key pairs as in the examples listed below.

10. **What information needs to be included in the CSR for a SAN certificate?**

Optionally in the CSR itself, but required in the requesting e-mail, please list the primary Subject CN (fully-qualified DNS name, FQDN) required, and any additional CNs (as FQDNs) to be added to the SAN field of the provisioned certificate. For example, the request might be:

   ```
   Please provision a Multi-Domain/SAN certificate as follows:

   myhost.berkeley.edu (primary), myhost-b1.berkeley.edu, myhost-b2.berkeley.edu
   using the included CSR.
   ```

11. **How can I validate that my certificate is correctly installed on my server?**

In addition to using validation web sites such as the COMODO SSL Analyzer, you can use the OpenSSL tool, s_client as follows, for example:

   ```
   openssl s_client -host somehost.berkeley.edu -port 443 -showcerts -verify 3
   ```

12. **How can I create a CSR with a SAN field?**

Note that having the SAN field defined in the CSR is not a requirement, but this can be submitted if desired. For example, with the Java 7 keytool you can use the following syntax with the BASH shell on RHEL:

   ```
   export JAVA_HOME=/opt/jdk1.7.0
   sudo ${JAVA_HOME}/bin/keytool -genkeypair -alias tomcat \
   -keyalg RSA -keysize 2048 \
   -dname "CN=myhost.berkeley.edu, \n   OU=MyDept, \n   O=University of California at Berkeley, \n   L=Berkeley, \n   S=California, C=US" \
   -ext "SAN=DNS:myhost.berkeley.edu,DNS:myhost-b1.berkeley.edu,DNS:myhost-b2.berkeley.edu" \
   -keystore /etc/tomcat6/tomcat6_keystore.jks
   sudo ${JAVA_HOME}/bin/keytool -certreq -alias tomcat \
   -file /etc/tomcat6/certs/myhost.csr -sigalg SHA256withRSA \
   -ext "SAN=DNS:myhost.berkeley.edu,DNS:myhost-b1.berkeley.edu,DNS:myhost-b2.berkeley.edu" \
   -keystore /etc/tomcat6/tomcat6_keystore.jks
   ```

13. **What about doing that using OpenSSL?**

Sure, thanks to Jim Blair, see this example using Python (also available as an attachment):

   ```python
   ```
#!/usr/bin/python

# Generate a key, self-signed certificate, and certificate request.
# Usage: gencert hostname [hostname...]  
# When more than one hostname is provided, a SAN (Subject Alternate Name) 
# certificate and request are generated. The first hostname is used as the primary CN for the request.
# Author: James E. Blair <jeblair@berkeley.edu>  2010-06-18 
# With help from this thread: 
# http://www.mail-archive.com/openssl-users@openssl.org/msg47641.html

import os,sys
import subprocess
import tempfile

OPENSSL_CNF=""
[ req ]
default_bits  = 2048
default_md   = sha256
distinguished_name = req_distinguished_name
prompt = no
%(req)s

[ req_distinguished_name ]
C=US
ST=California
L=Berkeley
O=University of California at Berkeley
OU=IST
%(cn)s

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[ v3_ca ]
supportKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
basicConstraints = CA:true
subjectAltName = @alt_names

[ alt_names ]
%(alt)s
"

SAN_REQ = ""
x509_extensions = v3_ca # The extentions to add to the self signed
cert
req_extensions = v3_req  # The extensions to add to a certificate request
**

def run(args):
  p = subprocess.Popen(args,
                     stdin=subprocess.PIPE,
                     stdout=subprocess.PIPE,
                     stderr=subprocess.STDOUT,
                     close_fds=True)
  p.stdin.close()
  while True:
    o = p.stdout.read(1)
    if not o: break
    sys.stdout.write(o)
    sys.stdout.flush()
  r = p.wait()
  if r:
    raise Exception('Error running %s' % args)

if __name__=='__main__':
  names = sys.argv[1:]
  if not names:
    print "Usage: gencert hostname [hostname...]"
    print
    print "  Please provide at least one hostname on the command line."
    print "  Multiple hostnames may be provided to generate a SAN request."
    print
    sys.exit(1)
  params = dict(req='', dn='', alt='')
  if len(names)>1:
    # SAN
    san_names = ''
    for i, x in enumerate(names):
      san_names += "DNS.%s = %s\n" % (i, x)
    params['req']=SAN_REQ
    params['alt']=san_names
    sanfn = '-san'
  else:
    sanfn = ''
  params['cn']='CN=%s\n' % names[0]
  keyfile = '/etc/pki/tls/private/%s.key' % (names[0], sanfn)
  crtfile = '/etc/pki/tls/certs/%s.cert' % (names[0], sanfn)
  csrfile = '/etc/pki/tls/certs/%s.csr' % (names[0], sanfn)
  (fh, cnffile) = tempfile.mkstemp()
  os.write(fh, OPENSSL_CNF%params)
  os.close(fh)

  if os.path.exists(crtfile):
print "Certificate file exists, aborting"
print "  ", crtfile
sys.exit(1)

if os.path.exists(csrfile):
    print "Certificate request file exists, aborting"
    print "  ", csrfile
    sys.exit(1)

if os.path.exists(keyfile):
    print "Key file exists, skipping key generation"
else:
    run(['openssl', 'genrsa', '-out', keyfile, '2048'])
    os.chmod(keyfile, 0400)
    run(['openssl', 'req', '-config', cnffile, '-new', '-nodes', '-key', keyfile, '-out', csrfile])
    run(['openssl', 'req', '-config', cnffile, '-new', '-nodes', '-key', keyfile, '-out', csrfile, '-x509'])
    run(['openssl', 'req', '-in', csrfile, '-text'])
14. I'm new to dealing with X.509 certificates, CSRs and all of this. Would you walk me through the basic steps necessary to generate a keypair and CSR using the gencert script and install, let's say, an InCommon SSL certificate for an Apache HTTP Server (https) on RHEL 6 using this service?

Sure, see this Extended example page for a step-by-step description of the process for generating and installing an InCommon SSL certificate.

15. How about some help with IIS servers and X.509 certificates?

Comodo does provide some KB articles dealing with several versions of IIS for CSR generation and installation of the certificates. See, for example, CSR Generation: Microsoft IIS 7.x, Certificate Installation: Microsoft IIS 7.x, and Root and Intermediate Certificate installation via MMC. See also this solution (#2) for a problem with importing certs for use with IIS 7.

16. What about other DNS domains such as anyplace.org? Can you issue certificates for such domains?

The CalNet InCommon-Comodo CA is currently registered to issue certificates for the berkeley.edu domain and its DNS subdomains plus a few other domains that InCommon has approved following our request for authorization to issue certificates on behalf of the domain. We can request to add any other DNS domains which we control or own, and for which we can provide to InCommon: (1) evidence of ownership and (2) proof of control of the DNS domain in question. For DNS domains that we do not own, this CalNet InCommon-Comodo CA will not apply so standard certificate requesting procedures with an external CA will be necessary.

17. What is the cost to the campus unit, if any?

There is no direct cost to campus units as UC Berkeley has paid the InCommon-Comodo CA institutional fee.

18. The new SHA-2 certificates have different chain (intermediate) certificates. What do I need to know about these?

These new intermediate certificates come in two options: a 3-certificate chain and a 4-certificate chain, the latter for older browsers and other clients which may not have built-in trust for the newer Root CA certificate which is part of the 3-certificate chain. More details about the new chain certificates from InCommon and Comodo.

Departmental Certificate Administrator (DCA) FAQ

1. What is a DCA?

This is the local UC Berkeley campus name for what is referred to as a DRAO (Department Registration Authority Officer) in the InCommon documentation.

2. What is expected of a DCA?

- The primary responsibility that a DCA has when issuing or renewing a certificate is to verify that requests for certificates are legitimate. If the DCA does not personally know the person making the certificate request and their business need for the certificate, due diligence would be expected in tracking down a responsible person within the DCA's unit who can vouch for the legitimacy of the request.
- Keeping a record of requests and their confirmations, e.g. an e-mail log for each request, for at least three years to allow for auditing of past transactions would also be expected of the DCA.
- Another requirement is to learn to use the InCommon CSM administrative tool for managing certificates as documented in the InCommon CA CSM RAO Admin Guide.

3. What are some policies and best practices for a DCA?

- Do not issue wildcard certs without asking for a review by the CalNetIdM team.

We will consult with IS&P to make sure that there is a good reason for using a wildcard cert vs. using a SAN certificate. For customers with a handful of certs, Multi-Domain/SAN certs might be the best approach. For services with multiple services on many clustered hosts, wildcard certs might work best, though there is more risk that way if the private key is compromised.

4. Some tips for generating CSRs

- For hosts that may have domain components that start with a number, for example, host.1918.berkeley.edu, the Java keytool may complain when generating keypairs. Use instead OpenSSL-based techniques.

5. How many DCAs should a department have?

This will vary depending on the volume of requests for certificates or renewals. If a unit has a request volume that would impact business needs were the primary DCA not available to fulfill these requests, having a designated backup DCA would be appropriate.

6. How can I sign up to become a DCA?

If you are interested in performing the DCA function for your unit, please forward your request along with the contact information for a
7. **How are DNS domains assigned to a DCA?**
   When you apply for becoming a DCA, please also list the DNS domains and hostnames for which you would like to be responsible for issuing certificates. It is possible to request additional domains via the InCommon Admin tools, but the initial setup will be smoother if we can provision most of these up front. Examples of UC Berkeley DNS domains and hostnames you might request are: *.mysubdom.berkeley.edu, myhost1.berkeley.edu, *.mysubdom.1918.berkeley.edu, myshost2.1918.berkeley.edu, etc. The wildcard names represent subdomains which you can claim as being responsible for the identity of all of the hosts.

8. **How do I use the InCommon Certificate Manager to request a new hostname or domain for my department?**
   Starting at the **Settings** tab, select the **Domains** menu item. Now click the **Add** button in the **Controls** column for your department. Finally, click the **Add** button to request a new hostname or domain to be added to the list for your department (the name appears in red text while pending approval). This request will generate an e-mail notice to the appropriate administrator for approval. When the approval step has been completed you will be able to provision certificates for the newly delegated domain.

9. **What to put in the External requester field?**
   This depends on whether you want the person listed receiving the notices generated at the various stages of certificate provisioning.

10. **What is Certificate Discovery?**
    This feature allows you to set up a scan of a subset of the network to create an inventory of certificates and their expiration dates. Be sure to create a **Discovery Scan Summary** notification before running the scan to ensure that the report is delivered correctly.

11. **What is the IP address used for the Certificate Discovery feature?**
    The discovery scans come from one IP address (91.199.212.132), which is secure.comodo.net.

12. **Is there an API that I might use to automate some tasks?**
    In the articles found at Comodo CCM/CSM API Documentation, Comodo has documented the APIs for the Comodo Certificate Manager (CSM) which underlies the InCommon Certificate Service Manager (CSM) web application used by the InCommon Certificate Service.

13. **We need to submit an emergency certificate request this weekend. What is the turnaround from Comodo on weekends?**
    It can take up to 24 hours. If you need expedited issuance of a certificate, send the order number with a note to PartnerValidation.

14. **What effect does the Server Software option (which shows in the Request New SSL Certificate form) have on the issued certificate?**
    One known effect is to escape certain special characters. For example, in the address fields, the string #1500 becomes \#1500 when selecting Microsoft IIS 5.x and later as the Server Software type.

15. **How can I renew a cert for a new term and update it to use an SHA-2 signature and, optionally, replace the key (via the CSR) at the same time?**
    Try this procedure in the InCommonCM app:
    
    1. In the list of certificates, select a certificate to update and hit the **Renew** button. Answer OK to the prompt presented.
    2. Select the newly created entry (which has the **Requested** status) and click the **Edit** button.
    3. Change the **Type** to one of those with (SHA-2) in the profile name.
    4. Change other details such as the **Term** and **CSR**, if desired.
    5. Click OK and approve the edited request.

---

**Terms of usage**

**Terms and Conditions**
Your use of InCommon/Comodo certificates shall be governed by the terms of InCommon's Relying Party Agreement.