Automated Certificate Checks

David O'Callaghan
Trinity College Dublin · Grid-Ireland CA

15th EU Grid PMA · Nicosia, Cyprus · January 2009

e-INIS
The Irish National e-Infrastructure

GFD.125 AND POLICY COMPLIANCE TESTING

Items to check for

- CA/Issuer certificate requirements
  - Name uniqueness (openssl x509 -subject)
  - RDN component ordering (openssl asn1parse -dump)
  - Version: must be v3
  - basicConstraints: critical? CA:FALSE? (openssl x509 -text)
  - keyUsage: only certSign, cRLSign?
  - CDP (if present) has http URL in URI attribute
  - Validity period: between 3 and 20 years, warn if <10
  - signatureAlgorithm: must be SHA-1
- CA/Issuer certificate other checks
  - Policy OIDs (should not be present)
  - eKU: check for weird attributes
  - Serial number: warn if 0, error if <0
- On update: Did serial number change?

Items to check for in client cert

- EE cert requirements
  - Is the namespace unique? Does it match the meta-data? (openssl x509 -subject)
  - Is the RDN ordering correct (openssl asn1parse -dump)
  - basicConstraints: critical, CA:FALSE?
  - keyUsage: no serverAuth in personal cert?
  - eKU: does it match any netscape attributes?
  - aKI: contains only keyID, error if name or serial detected
  - nsCertType: matches eKU if present
  - CDP: must be included and must return DER CRL
- EE cert other checks
  - keyUsage: no nonRepudiation &c?
  - DC naming, matching registered WHOIS contacts

Items to check for in meta-data

- crl_url is http
- crl_url yields a CRL issued by the CA
- crl_url can sustain 4Hz requests

- CRL validity is > 7 days (warn if >3days, manual if >31 days)

- email: address does not bounce

- alias is reasonable representation of CA name
- alias matches [A-Za-z][-A-Za-z0-9]+
- ca_url if present: gets CA cert, warn if missing
- url: leads to non-empty html or text page

- status: matches accreditation

Automate checks

- CA / Issuer certificates
- End-entity certificates
- Meta-data

against

Requirements and Best Practices from GFD-C.125

Certificate Checking Utility

The certificate checking utiltity should be

- Sys-admin friendly
- Use familiar scripting languages and libraries

Perl & OpenSSL

- How to describe the tests?
- How to handle the certificates?

How to decribe the tests?

- Assertions & Comparisons
- Structure for a set of tests

A portable test format?

<xml> can be <too> vebose </too></xml>

(s-exps are-nice (if (you-like (or 'lisp 'scheme))))

but...

Disadvantages

- Parser for XML/s-exps/JSON/…
- Still need to define a language for assertions and comparisons

Quick & Dirty

- Use a standard Perl test framework

  Test::Harness and Test::More
- Supports assertions and comparisons
- Get overall result for a test suite

Tied to Perl but does what we want.

How to test the certificates?

Using Perl and OpenSSL

- OpenCA
- **Crypt::OpenSSL::X509**
- Net::SSLeay

Crypt::OpenSSL::X509

by Dan Sully

- Basic access to subject, dates, pubkey, etc.
- No support for extensions, name components

Extending Crypt::OpenSSL::X509

Added support

- get extensions by OID or name
- get name components
- check object type

Putting it all together

DEMO

GFD-C.125 Test Suite

```
##
## Test Suite for CA Certificates under GFD.C.125 "Grid Certificate Profile"
##
## David O'Callaghan <david.ocallaghan@cs.tcd.ie>
## 2009-01-09
## VERSION 0.1.2
##
## TODO: implementation is incomplete
##

# Pre-requisites
use CheckCertsTest;

for my $certfile(@certlist) {
    ok(my $x509 = Crypt::OpenSSL::X509->new_from_file($certfile), "new_from_file $certfile");
    diag "\n\n * * * \nCert Subject: ", $x509->subject, "\n";
    ok($x509->subject, "Subject: " . $x509->subject);

    # Cert version 2.1
    cmp_ok($x509->version, '==', 2, 'version 2.1');

    # Serial & Message Digest 2.2
    like($x509->serial, qr/[a-fA-F0-9:]+/, 'Serial number format 2.2');
    unlike($x509->sig_alg_name, '/md5/i', 'Message digest MUST NOT be md5 in new CA certs 2.2');
    like($x509->sig_alg_name, '/sha-?1/i', 'Message digest SHOULD be sha-1 2.2');

    # Issuer and Subject names 2.3
    my $subject_name = $x509->subject_name();
    ok($subject_name->has_entry('CN'), 'DN has CN 2.3');

    my $entries = $subject_name->entries();
    for my $entry (@$entries) {
        ok($entry->is_printableString(), $entry->type() . ' is printableString 2.3') or diag("Name
component ", $entry->as_string(), " SHOULD be printableString")
            unless $entry->type() eq "DC";
    }
```

```
#2.3.1 CN should be descriptive
#2.3.2 should consist of "DC", "C", "ST", "L", "O", "OU" and "CN"
#2.3.2 if using DC, DCs should be at start
if($subject_name->has_entry('DC')) {
    my $loc = $subject_name->get_index_by_type('DC');
    is($loc, 0, "DC at start 2.3.2");
    my $oldloc = $loc;
    while($subject_name->has_entry('DC', $oldloc)) {
        my $loc = $subject_name->get_index_by_type('DC', $oldloc);
        next if $oldloc == $loc;
        is($loc, $oldloc+1, "Multiple DCs at start 2.3.2");
        $oldloc = $loc;
    }
}
#2.3.2 DN should have 0
#2.3.2 C should match issuer
#isa($x509->subject, "ASN1_SEQUENCE", 'DN is ASN1 SEQUENCE 2.3');

ok(not($subject_name->has_long_entry('serialNumber')), 'DN does not have serialNumber 2.3.3');
ok(not($subject_name->has_long_entry('emailAddress')), 'DN does not have emailAddress 2.3.4');
ok(not($subject_name->has_entry('UID')), 'DN does not have UID 2.3.5');
ok(not($subject_name->has_oid_entry('0.9.2342.19200300.100.1.1')), 'DN does not have userID
.5');

# Extensions in CA certificates 2.4
my $exts = $x509->extensions_by_name();
ok($$exts{'basicConstraints'}, 'Has basicConstraints 2.4.1');
# TODO is($$exts{'basicConstraints}->value(), "CA: TRUE", 'basicConstraints CA: TRUE 2.4.1');
ok($$exts{'basicConstraints'}->critical(), 'basicConstraints critical 2.4.1') or
    diag("basicConstraints SHOULD be marked critical in CA certificates");
ok($$exts{'keyUsage'}, 'Has keyUsage 2.4.2');
ok($$exts{'keyUsage'}->is_critical(), 'keyUsage critical 2.4.2') or
    diag("keyUsage SHOULD be marked critical in CA certificates");
```

```
# extendedKeyUsage 2.4.3
ok(!($$exts{'extendedKeyUsage'}), "No extendedKeyUsage 2.4.3") or
    diag("CA certificates SHOULD NOT include extendedKeyUsage extension");
if($$exts{'extendedKeyUsage'}) {
    ok(not($$exts{'extendedKeyUsage'}->critical()), "extendedKeyUsage not critical 2.4.3") or
        diag("extendedKeyUsage MUST NOT be marked critical in CA certificates");
}
# nsCertType, nsComment, nsPolicyURL, nsRevocationURL 2.4.4
foreach my $ext ("nsCertType", "nsComment", "nsPolicyURL", "nsRevocationURL"){
    ok(not($$exts{$ext}), "No $ext 2.4.4") or
        diag("CA certificates SHOULD NOT include $ext extension");
    if($$exts{$ext}) {
        ok(!($$exts{$ext}->critical()), "$ext not critical 2.4.4") or
            diag("$ext MUST NOT be marked critical in CA certificates");
    }
}
# TODO If nsCertType is used, it MUST be consistent with the keyUsage extension.

# certificatePolicies 2.4.5
if($$exts{'certificatePolicies'}) {
    ok(not($$exts{'certificatePolicies'}->critical()), "certificatePolicies not critical
5") or diag("certificatePolicies extension SHOULD NOT be marked critical if present");
}

# TODO cRLDistributionPoints 2.4.6
if($$exts{'crlDistributionPoints'}) {
    # DO
}
# Authority and Subject Key Identifier 2.4.7
```

To be continued...

Items to check for

- CA/Issuer certificate requirements
  - Name uniqueness (openssl x509 -subject)
  - RDN component ordering (openssl asn1parse -dump)
  - ~~Version: must be v3~~
  - ~~basicConstraints: critical?~~ CA:FALSE? (openssl x509 -text)
  - keyUsage: only certSign, cRLSign?
  - CDP (if present) has http URL in URI attribute
  - Validity period: between 3 and 20 years, warn if <10
  - ~~signatureAlgorithm: must be SHA-1~~
- CA/Issuer certificate other checks
  - Policy OIDs (should not be present)
  - eKU: check for weird attributes
  - ~~Serial number: warn if 0, error if <0~~
- On update: Did serial number change?

Results

- CA certs with MD5 signatures (9)
- KeyUsage missing (2) or not marked critical (14)
- BasicConstraints not marked critical (7)
- Certs with emailAddress in DN (12)
- NsCertType (33) and nsComment (23)

To Do

- Enhance Crypt::OpenSSL::X509 further
- Finish the GFD-C.125 test suite
- Package nicely

Ideas

- A new test suite for sanity / vulnerability checks
  - Exponents, Known-weak keys: See Jim B's RAT talk
  - Unaccredited higher-level CAs
- Gather useful tests / scripts from CAs

Links

https://www.eugridpma.org/cgi-bin/cvsweb.cgi/util/checkcerts/

david.ocallaghan@cs.tcd.ie