APNIC Resource Certification Trials

Routing SIG

APNIC 21

Outline

- Current Goals
- Methodology
- OpenSSL issues/overview
- OpenSSL Configuration
- Example: Certificate Creation
- Example: Resource signing
- Example: Validation
- Current Status
- Next Steps

(current, short term) Goals

- Develop useful demonstrator code to release into the community
  - Use F/OSS, avoid (re)implementation where possible
- Develop code viable for use in APNIC services
  - Perl, under mod_perl/apache2
  - Exploit existing coding methodologies
    - APNIC Forms/Framework
    - Emerging use of XML, REST, AJAX

Methodology

- Dual approach. Bootstrap tests with shell, perl command line
- Explore APNIC Framework extensions to support infrastructure
  - Leverage experience in bootstrap phase
  - CPAN library provides Convert::ASN1
    - Perl hash of ASN.1 code
    - Initial X509 module available
      - Poorly documented
  - Target OpenSSL for CA/PKI functions

OpenSSL Issues/Overview

- Openssl command tool provides
  - Signing (SHA1) Basic CA functions (request, sign, crl)
  - Verification
    - With caveats: unknown extensions problem(s)
  - PEM <->DER conversion
  - Debug opportunities
  - Perl abstractions available
- Caveats
  - Much functionality is un- or poorly documented
  - C-code internals complex

OpenSSL Configuration

- openssl.cnf
  - Textfile, 'MSDOS' model of

```
[BLOCK-NAME]
var=val
```

- Can call command with 'extra' .cnf file
  - Decided to construct RFC3779 assertions as external .cnf file, invoke in certificate sign
  - Encoded as O.I.D.s.t.r.i.n.g:{flags}:DER:HE:EX:ST:RI:NG

Example: RFC3779 in openssl .cnf

ASN 17814

Ipv4 203.160.32.0/20

encodes as: (with other RFC3779 data)

```
[ APNIC_CERT_EXTENSIONS ]
basicConstraints = CA:true
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
keyUsage=critical,digitalSignature,keyCertSign,cRLSign

2.5.29.32=critical,DER:30:0C:30:0A:06:08:2B:06:01:05:05:07:0E:02
1.3.6.1.5.5.7.1.7=critical,DER:30:0F:30:0D:04:03:00:01:01:30:06:03:04:04:C
   B:A0:20
1.3.6.1.5.5.7.1.8=critical,DER:30:08:A0:06:30:04:02:02:45:96
```

In Perl...

- main::(./makecertasn.pl:324): print STDOUT printcnf('2.5.29.32', $pdu);

```
  DB<2> x $pdu
"0\cL0\cJ\cF\cH+\cF\cA\cE\cE\cG\cN\cB"
```

- DB<4> x @tmp

```
0  ARRAY(0x85622d8)
   0  HASH(0x84ded50)
      'policyIdentifier' =>
  '1.3.6.1.5.5.7.14.2'
  DB<5>
```

Example: Openssl cert creation

```
# openssl ca -verbose -batch \
 -config openssl.cnf \
 -extfile APNIC-CA.cnf -extensions \
  APNIC_CERT_EXTENSIONS \
 -in $@ -out ${@:.csr=.cert}
```

- Normal Openssl cert req as -in
- PEM encoded cert as -out.
- Keeps 'CA' config clean
- Change APNIC-CA.cnf per req.

Example: resource signing

- Uses SHA1(256) sign function in OpenSSL
  - Detached signature
  - Avoid ROA ASN.1 encoding for now
  - Can sign any text (RPSL object)
- Verify problems
  - Signing pubkey **has** to be in .DER format
    - Extraction from certificate is clumsy
    - (can provide service on web, or when issued)

Current Status

- Initial test certificates generated from APNIC 'super' cert covering all APNIC managed space
  - 8k .DER file (10k PEM)
  - 1085 signings in <30min
  - Short lifetimes
  - Explicit 'non production' CA DN
  - Cert Dns based on fec0 name model
  - Private key, openssl .cnf files made available for downstream test
    - Can make daughter certs, ROA etc

Framework status

- Basic cert handling processes
  - Fetch cert, examine cert
  - Request cert
  - Sign cert
- Coding using Perl openssl API

Next Steps

- Test BBN Certs (Charles Gardiner)
- Web based 'be an ISP' demonstrator
  - Daughter certs, ROAs,
    - Check, validate revoke
    - Good to be able to test bogus certs
- Finalize timeline to release certs to membership, functionality in MyAPNIC
  - Subject to APNIC Framework completion