CRYPTOGRAPHY

MESIGI3906
CERTIFICATE

JOHN CAGNOL / AMAURY DARSCH
DER CALCUL SCIENTIFIQUE / DER GÉNIE INFORMATIQUE

AGENDA

- The X.509 PKI
  - Basic principles
  - ASN.1 certificate encoding
  - Certificate contents
- Applications with OpenSSL
  - Creating a self-signed root certificate
  - Creating a certificate request
  - Signing a certificate

X.509 PKI

- In modern communication infrastructure, it is crucial to assert that a public and a private key are bound to the right entity.
- A trusted Certificate Authority (CA) is a signed binding between an authority and a key.
- By signing a certificate, an organisation asserts that a public key is bound to the right entity.
- The X.509 Public Key Infrastructure (PKI) standard defines the profiles used to manage a certificate.

CHAIN OF TRUST

- A certificate can be signed by an organization that asserts its authenticity.
- When an organization signs a certificate, it creates a chain of trust which needs to be verified.
- Since the root certificate cannot be signed, it must be self-signed and trusted as it is.
- Importing a self-signed certificate is an operation that must be carefully analyzed and certainly not done blindly.

CA HIERARCHY

- IETF hierarchy
  - IPRA : Internet Policy Registration Authority
  - PCA : Policy Certification Authority
  - CA : Certificate Authority
- VeriSign
  - Class 1 : individuals
  - Class 2 : organizations
  - Class 3 : server and software signing
  - Class 4 : online transactions between companies
  - Class 5 : governmental security

CA REVOCATION

- A certificate can be revoked by an authority when it is necessary to do so
  - Compromised private key
  - Invalid chain of trust or signing party
- Certificate revocation list
  - Published by an authority
  - Revised periodically
  - Based on certificate serial number
- Online Certificate Status Protocol (OCSP)
  - Online verification of certificate validity

CA DESCRIPTION

- A certificate is described by its contents, a signature algorithm and a signature value
- In ASN.1 notation :

```
Certificate :== SEQUENCE {
  tbsCertificate      TBSCertificate,
  signatureAlgorithm AlgorithmIdentifier,
  signatureValue      BIT STRING
}
```

ASN.1 CERTIFICATE

```
TBSCertificate :== SEQUENCE {
   version [0]           EXPLICIT Version DEFAULT v1,
   serialNumber          CertificateSerialNumber,
   signature             AlgorithmIdentifier,
   issuer                Name,
   validity              Validity,
   subject               Name,
   subjectPublicKeyInfo SubjectPublicKeyInfo
   issuerUniqueID  [1]   IMPLICIT UniqueIdentifier OPTIONAL
   subjectUniqueID [2]   IMPLICIT UniqueIdentifier OPTIONAL
   extensions      [3]   EXPLICIT Extensions OPTIONAL
}
```

CA FIELDS (I)

- Version
  - The certificate version is 1 by default
  - Must be version 3 if extensions are used
  - Most modern certificates are moving to version 3
- Serial number
  - A unique number that identifies the certificate
  - Used in the revocation list
- Signature
  - The signature algorithm used to sign the certificate

CA VERSION SIDE NOTE

Note that the version numbers are one less than the actual X.509 version because in the ASN.1 world you start counting from 0, not 1 (although it's not necessary to use sequences of integers for version numbers. X.420, for example, is under the impression that 2 is followed by 22 rather than the more generally accepted 3).

Peter Gutmann

X.509 Style Guide

CA FIELDS (II)

- Signature
  - The algorithm identifier used to sign the certificate
  - Same as in the certificate sequence
  - Why ? May be some obscure attack!
- Name
  - A relative distinguished name (RDN)
  - A unique name used to identify everything on earth, and even in the universe
  - Derived from the X.500

CA RDN EXAMPLE

- Organization
  - O=Pole Léonard de Vinci
- Organization Uniit
  - OU=Internet Certificate Traffic School
- Location
  - L=Paris
- Common Name
  - CN=PULV
- Email
  - E=pulv@devinci.fr

CA COMMON NAME

- The CA Common Name (CN) is the most important component of the RDN
  - It defines the name binding of the certificate
- For a self-signed root certificate
  - Not necessarily defined
  - CN=PULV/emailAddress=pulv@devinci.fr
- For a certificate request
  - The host or domain to bind
  - CN=esilv.devinci.fr/emailAddress=esilv@devinci.fr

CA RDN SIDE NOTES

When the X.500 revolution comes, your name will be lined up against the wall and shot

John Gilmore

They can't be read, written, assigned, or routed. Other than that, they're perfect

Marshall Rose

CA FIELDS (III)

- Validity
  - Not before and not after time define the valid time range for the certificate
  - An incredible mess when it comes to manage the seconds and the cutover time
- SubjectPublicKeyInfo
  - The certificate public key
- UniqueIdentifier
  - Only in version 2 certificate
  - Obsolete in version 3

GENERATING A ROOT CA

- A root certificate is generated from OpenSSL as a self signed certificate
  - `openssl req -new -x509 -node -out pulv.pem -keyout pulv.pem`
- The certificate can be examined with OpenSSL or loaded in a browser
  - `openssl asn1parse  -in pulv.pem`
  - `openssl x509 -text -in pulv.pem`

CONCLUSION

- The certificate is the most important component used in the chain of trust
  - It must be managed carefully
  - It must not be imported without precise verification
- There are still some issues with the extensions:
  - Alternate names can be defined
  - Spoofing is a potential problem
  - Wildcard and multiple-rdn must be regarded as suspicious

SIGNING A CA

- A certificate is generated from OpenSSL
  - `openssl req -new -node -out esilv.pem -keyout esilv.pem`
- The certificate is signe by the PULV
  - `openssl x509 -req -in esilv.pem -CA pulv.pem -Cakey pulv.pem -set_serial 1 -out pulv-esilv.pem`