Unifying access to cryptographic objects in GNU/Linux

Katholieke Universiteit Leuven

Nikos MavrogiannopoulosMavrogianno
nmav@gnutls.org

Outline

- Cryptographic tokens and modules
- Cryptographic objects
- Access to objects and issues
- Access to modules and issues

Cryptographic tokens

- Various shapes/interfaces
- Can be in software (gnome-keyring)
- Accessed in a common way -

  PKCS #11

Cryptographic tokens

- Contain objects
  - Cryptographic keys (RSA, DSA private keys)
  - Corresponding certificates (X.509)
  - Trusted certificates
- Accessed through PKCS #11 modules

Cryptographic modules (PKCS #11)

- Shared libraries providing a consistent API to access tokens and objects
- Usually reside in /usr/lib/pkcs11/

Accessing objects

Accessing objects

- Cryptographic applications:
  - Ask for “key” and “certificate” files
  - Have special options to specify objects in tokens
  - Sometimes slot number might be required

Accessing objects: Problems

- Objects are referenced in a way that is unique per application
- Accessing objects in a token, is usually a non-trivial procedure and application specific

Accessing objects: Requirements

- What is required to uniquely identify an object?
    - Object ID
    - Object type
    - Token ID
    - (Module via which it is accessed)

Access to objects

- pkcs11-helper (openvpn):
  - Has a PKCS #11 ID:
    "EnterSafe/PKCS\x2315/3075211616010310/Nikos/32F153F3E37990B08624141077CA5DEC2D15FAED"
- Openssl:
  - Opensc pkcs11-engine: some id
  - Oracle pkcs11 engine: PKCS11-URLs

Unification: PKCS #11 URLs

- A uniform way described in draft-pechanec-pkcs11uri-03
  - Can be used to describe a token
  - Can be used to describe an object
- Example:

```
pkcs11:token=mytoken;manufacturer=SnakeOil;
    model=1.0;object=my-certificate;objecttype=cert;
    id=%69%95%3E;
```

Unification: PKCS #11 URLs

- Advantages:
  - Can specify all PKCS #11 objects and tokens
  - Can be used to share objects between any applications
  - Does not cope with slots
  - Can be used in command line – in a backwards compatible way

Unification: PKCS #11 URLs

- Example of extending the “key file” and “certificate file” command line options:
  - gnutls-cli --x509keyfile pkcs11:... --x509certfile pkcs11:...

Access to modules

Access to modules

- No system-wide way
  - Typically via a command line argument:

    `--pkcs11-providers /usr/lib/pkcs11/opensc-pkcs11.so`
- PKCS #11 has issues when multiple users use a module

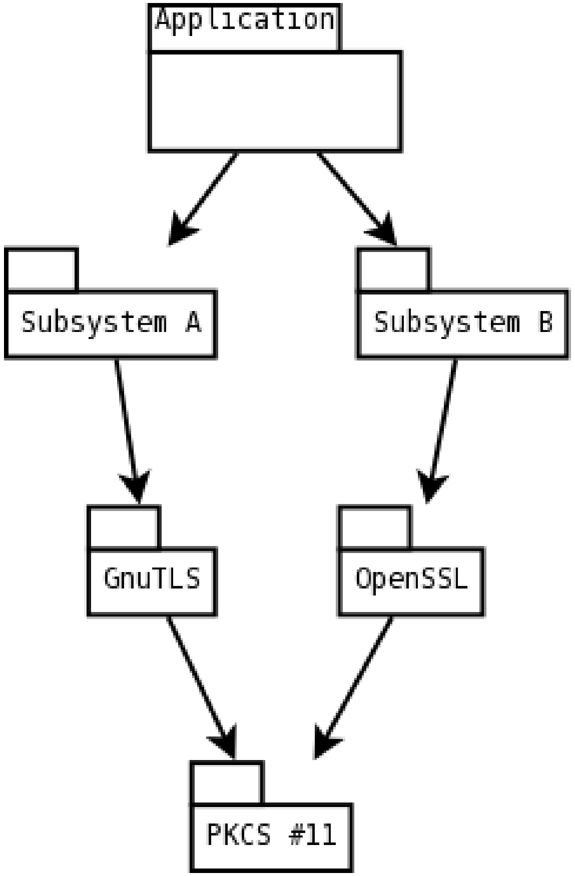
Access to modules: system-wide

- Proposed FHS: All modules in /usr/lib/pkcs11/
  - Unfortunately there lie testing modules as well or variants of modules
  - onepin-opensc-pkcs11 and opensc-pkcs11 that give duplicate objects sharing the same URLs
- GnuTLS: /etc/gnutls/pkcs11.conf

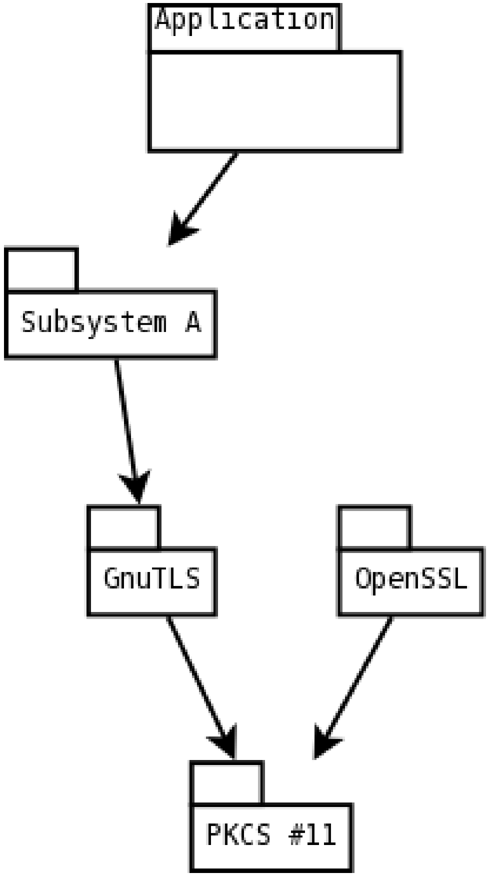
Access to modules

- We need a system-wide way to specify modules to load for all applications to share the same objects

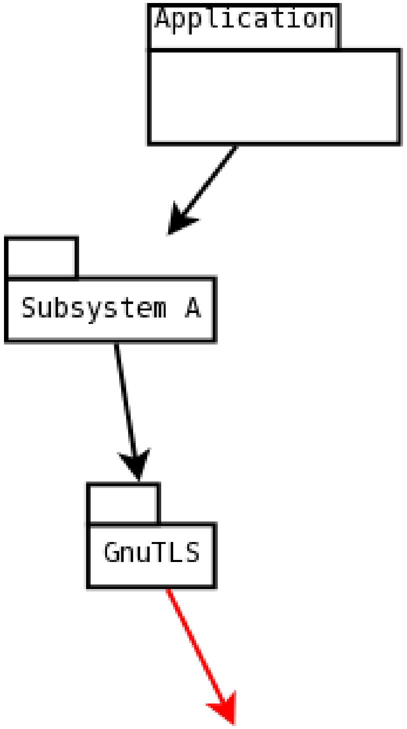
Access to modules: multiple-users



Access to modules: multiple-users



Access to modules: multiple-users



Access to modules: multiple-users

- We need a way for PKCS #11 modules to be accessible by multiple users (libraries)
  - Ongoing work of Stef Walter in p11-kit.

Open issues for unification

- Access to objects
  - Common way to specify objects
- Access to modules
  - Common configuration file
  - Multiple access to the module

Questions?