PASSTORE: safe certs & password management for Grid hosts

Stefano Dal Pra, Marco Verlato
INFN
stefano.dalpra@[pd|cnaf].infn.it
marco.verlato@pd.infn.it

www.eu-egee.org

Information Society and Media

Summary

- What Sanctorum is
- Features outline
- Sanctorum's data
- Features: operational details
- Security considerations
- Install,setup,populating
- current limits

What Sanctorum is (and isn't)

- **A solution for comfortable, safe and secure digital host's certificate and password management**
  - **SAFE**: prevents human errors
  - **SECURE**: reduces risk of unauthorized access to private keys
  - **COMFORTABLE**: certificate management tasks can be automatized at most
    - manual intervention needed to add or remove hosts in sanctorum's repository
- **Not intended for personal certificates management.**

Why Sanctorum?

- **Security reasons:** The (Italian) Certification Authority (CA) recommends to maintain two backup copies for key/cert pairs on two distinct devices not network reachable.
  - Should an attacker gain access to a repository of private keys all certificates would be compromised and should be revoked.
  - No one but the owner should be able to access its private key
- **Preventing mistakes:** manually creating p. keys may expose the operator to subtle problems or mistakes (mismatching key/cert pairs, world readable key...)
  - The operator has direct access to the private keys repository
  - Unsafe practices are usually more comfortable than secure. Sanctorum attempts to be comfortable & secure at once.

Requirements

- **Hardware requirements**
  - Stand alone dedicated host
  - Raid1 disk
  - Network card (disabled)
  - KVM (recommended)
- **SW requirements**
  - Iptables
  - Openssl
  - MySQL
  - Python + modules (mysql, expect)
- **Security recommendation**
  - Only direct console access or through KVM like systems when administrative intervention is needed.

Features Outline

Sanctorum provides manual tools for:

- **Host password management**
  - display password for a given user@host:<service>
    - Example: nicedb@db-01:mysql
    - Default is root@host:ssh
  - Add/Update passwords
    - Root passwords can be updated on the remote host
      - *uses pexpect*
    - Wildcard allowed (Ex: ui-* would match hosts ui-01, ui-02 and so forth)
    - Old and new password are requested to the operator before changing (according to common passwd command)
    - Passwords are typed with noecho mode
- **Manual mode is strictly interactive by design**
  - Reason: operator cannot dump lots of info at once

Features Outline

Manual tools for Certificate management

- crtreq: Creates a certificate requests
- putreq: copies a certificate request to *user@host*:path/
  - Asks password for *user@host*, then scp the .req file
  - *user@host* is in charge to ask RA for the new certificate
- getcrt: gets a certificate from *user@host*:cert_filename and adds it to the repository (only if it matches priv. key)
- Putcrt: deploys private key and certificate to the owner host
  - Sanctorum only asks for hostname
- getkey: import a pre-existing private key
- lcrt: Displays cert status for host[s] (wildcard allowed)

Features Outline (2)

- **“automatic” mode:**
  - Expiring certs are found by scanning the repository
    - (from the cert itself: openssl x509 … -enddate)
  - Cert renewal requests are created and mailed to one or more operators
    - backup of “old” key/cert is kept
  - The operator signs it and fwd to local Registration Authority
    - mandatory manual step
  - Sanctorums checks the CA website trying to download newly released certs (Italian CA only, at present).
    - Upon success:
      - *adds the cert in the db (verify that it matches its key)*
      - *Deploy key/cert pair on the remote host*
      - *Notify operator by mail*
    - Upon failure at some step:
      - *Warns operator by mail*

Where are Sanctorum's data?

- **All data are stored in a MySQL db**
  - Sensible ones are Cyphered with sql simmetric encryption AES_ENCRYPT(<string>,'<cypher_key>')
    - Depending on setup, the cypher_key is asked at each operation or it is written in a “hidden file”
  - DB on RAID1 disk
  - Hostname, username, service in cleartext

Sanctorum Security

- **Sanctorum's security depends on setup and configuration; may range from “quite poor” to “high” depending on setup.**
- **“Strongest” (paranoid like):**
  - Network card disabled, connected with another host who masquerades sanctorum
  - Iptables fw, no rules, policy DROP (no traffic allowed)
  - Direct access only through console or KVM
- **Fine, thanks; how does it work then?**
  - When a connection is needed, sanctorum configures and activates the network card
  - Then it activates the proper fw rule to accept outgoing connection request to the target
  - Then the needed transfer happens
  - Then network down and fw rule deactivated
  - Never network reachable. (only act as client)

Sanctorum Security (2)

- **Apart for adding/removing hosts, operations can go on with no manual intervention, by configuring cron to run**
  - autorenew.py
    - creates needed cert requests and mail them to operator
  - auto_crt_upd.py
    - Downloads new certs from CA website
    - Stores them in the db (only if it matches its private key)
    - deploys them to the remote hosts
- **Problem: Download from CA website only works with Italian CA**
  - reason: no uniform way to get certs from different CA
- **Possible workaround:**
  - Retrieve certs from mailserver (not implemented).

Installing it

- Download source from:

http://forge.cnaf.infn.it/frs/?group_id=17

- Untar the archive and follows INSTALL
- Sketch:

```
mysql> create database sanctorum;
mysql> grant all on sanctorum.* to 'oper'@'localhost'
    identified by 'seckey';
mysql> flush privileges;
mysql -h localhost -u oper sanctorum -p <
    passtore_schema.sql
```

- Then edit sanctorum.conf and create .sanctorum

Configuring

- **Edit sanctorum.conf**

```
domain=cr.cnaf.infn.it
dbuser=oper
dbname=sanctorum
admin_email=user@pd.infn.it,list@pd.infn.it
#certs Time To Live
crt_TTL=20
crt_path=/root/
```

- **Edit .sanctorum**
  - Write in it the mysql db password ('seckey' here)
- **Edit net.conf**
  - Syntax matters: it is a python dictionary

Does it work?

$ ./sanctorum.py

-h prints this help

-a add a password

-g get a password

-u update a password

-l lists known hosts

-d delete an entry

[SNIP]

-crtreq Creates host.cr.cnaf.infn.it.req and host.cnaf.infn.it.key

-putreq scp host.cr.cnaf.infn.it.req user@userpc.cnaf.infn.it:

-getcrt scp crt given by CA from user@host:filename

-putcrt scp host.cr.cnaf.infn.it.[crt|key] root@host.cnaf.infn.it:/etc/grid-security/host[cert|key].pem

-lcrt lists crt infos per host[s]

-getkey scp .key from user@host:filename (use only to import an elsewhere created key!)

- **Yes, It does!**

Populating Sanctorum

- If you have a directory with certs and keys you can import them using **importcert.py**
- **python importcert.py mycrtdir/**

Cert found for argus.cr.cnaf.infn.it

key found for argus.cr.cnaf.infn.it […]

adding host,passwd,crt,key for argus.cr.cnaf.infn.it

[…]

- **Only valid cert/key couples are added. Filenames are ignored.**
- **IMPORTANT: if the host is new to sanctorum, it is added first with a fake root password assigned to it. Change it manually as needed**

Current limits and weak points

- **All hosts must be in the same domain**
- **Db encryption key and mysql user's password are the same, written into .sanctorum**
  - .sanctorum file can miss, but then sanctorum keeps asking for the secret key at each run.
  - Workaround: let sanctorum run as a distinct user. The operator runs sanctorum as sudoer and cannot access .sanctorum file
- **Automatic mode does not drive firewall and up/down network**
  - Solution: add a script to turn on network & fw before running and turn it off after.
- **Sanctorum uses scp and needs to know root passwd of remote hosts**

Key points

- **Passtore is (or can be) NEVER network reachable**
  - Network gets activated for the strictly needed time only
  - Connection initiated ONLY from sanctorum to target host
- **User cannot (directly) take a copy of a private key**
  - When copying to Grid host passtore knows remote password
- **Common errors are prevented:**
  - Mixed key/certs pair
    - Only matching certs are inserted in the db
  - copy key/cert on the wrong host
    - A key can only be copied on the right host
  - Wrong permissions on hostkey.pem,hostcert.pem
    - Correct permissions are set

If interested...

- **See https://forge.cnaf.infn.it/projects/passtore/**
  - Svn source repository
  - Write to stefano.dalpra@cnaf.infn.it or marco.verlato@pd.infn.it

Thanks for attention