Virtual HSM Implementation in OpenVZ Containers

Kartashov D., Krinkin K.

St-Petersburg Academic University of Russian Academy of Sciences

14/11/2013

Motivation & requirements

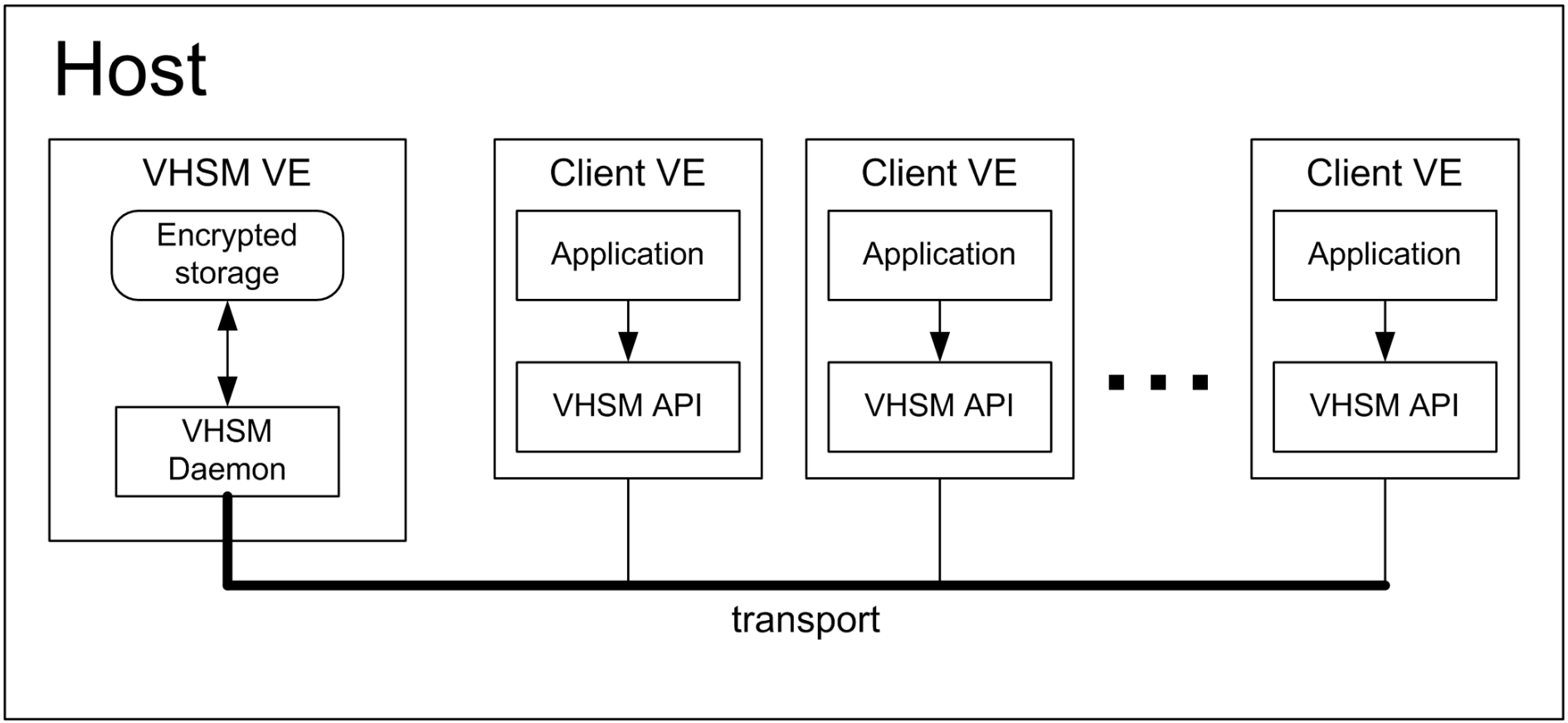
Motivation

- Virtual hosting service providers need to protect sensitive customers' data.
- HSMs meet these needs but their costs are high.
- It's desirable to have a virtual device that works like the HSM.

Requirements

- basic HSM functionality;
- based on OpenVZ;
- uses Netlink for transport;

Project architecture



Main components

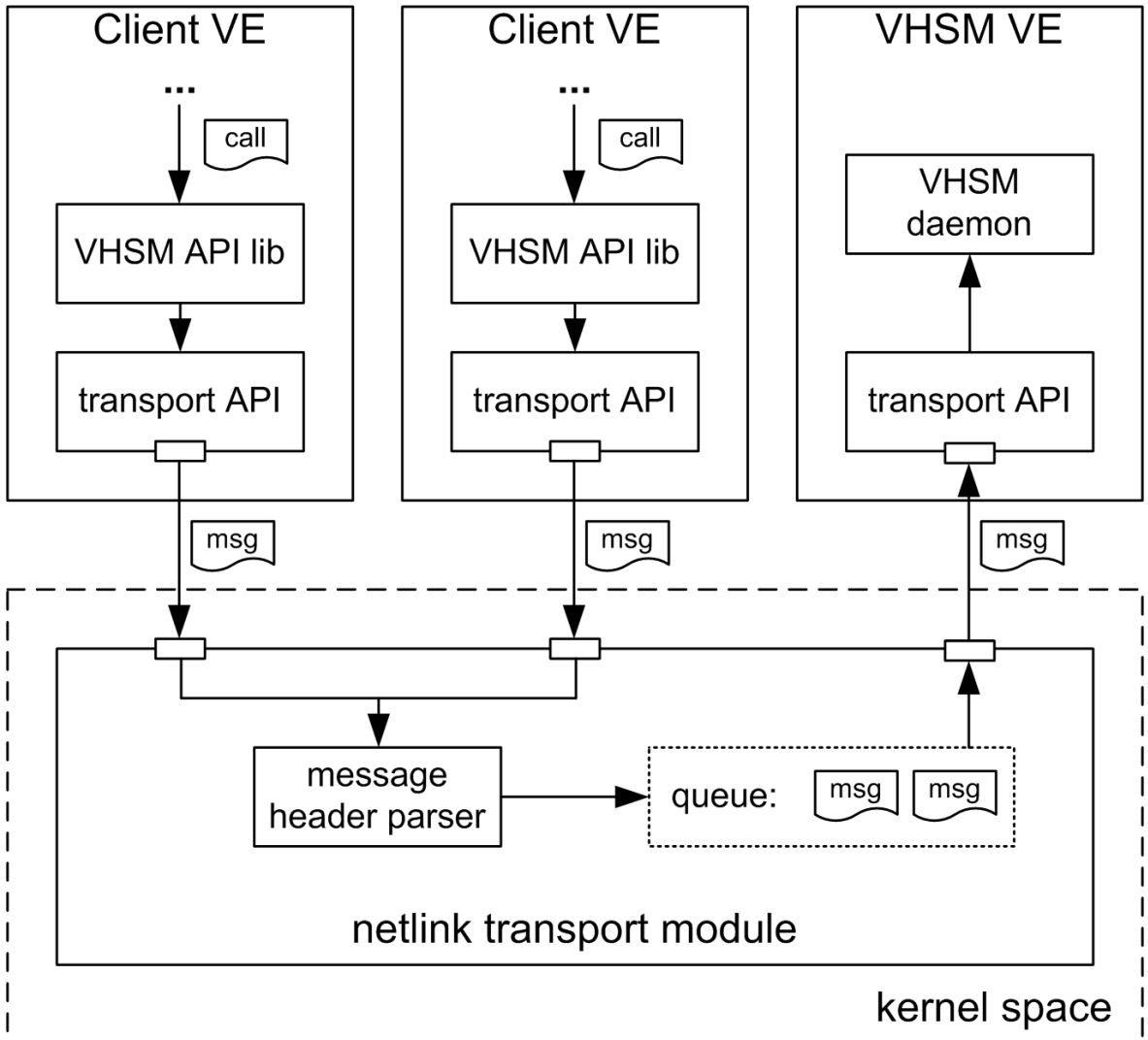
- **VHSM server**
  - authentication;
  - cryptographic operations on secret data;
- **Encrypted storage**
  - sensitive user data storage;
- **Transport**
  - message transfer;
  - container identification;
- **VHSM API**
  - wraps the VHSM transport;
- **OpenSSL engine**
  - interface between the VHSM API and user application;

VHSM server & encrypted storage

- access:
  - user login and password;
  - the master user data encryption key based on user password is generated by the function PBKDF2;
- authentication:
  - 256-bit authentication key encrypted with master-key in GCM mode;
- cryptographic function computation:
  - accessing secret data using id;
  - only result of the operation is returned to the user;
- secret data storage:
  - an SQLite database stores user data encrypted with the master-key by AES-GCM;

Transport

- protocol — `Google Protobuf`
- netlink based implemetation



VHSM API

- session management
  - initiate/terminate session;
  - user authentication;
- key management
  - import;
  - generation;
  - deletion;
- hashing and MAC
  - standard functions: `init`, `update`, `final`

OpenSSL engine

OpenSSL engine can be used to delegate cryptographic functions to VHSM

The hashing algorithm is changed in the current implementation, so the default OpenSSL functions for HMAC can be used.

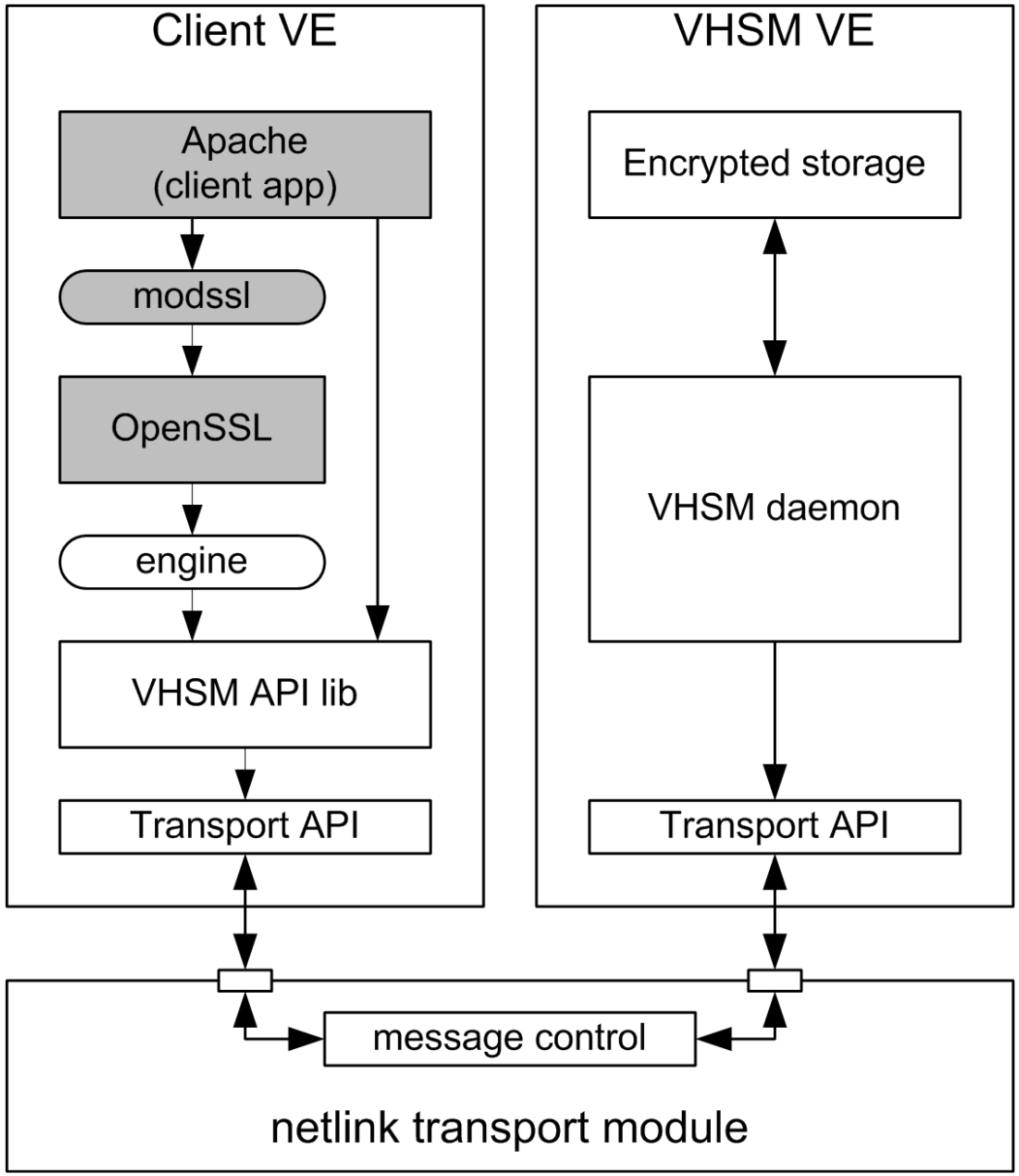
Cons:

- the engine implementation is based on the current OpenSSL implementation;
- configuration files are not secure.

Pros:

- end users can easily integrate VHSM support into their application.

Usage example



Summary

Under development:

- user roles and access levels;
- pluggable authentication module with VHSM;
- support for other virtual environments;

Refs:

- repository:
  `http://git.openvz.org/?p=vhsm`
- bugtracker:
  `http://dev.osll.ru/projects/vhsm`