Virtual HSM implementation in OpenVZ containers

Dmitriy Kartashov

Saint-Petersburg Academic University of RAS

2014

Introduction

Hardware Security Module (HSM) – external pluggable device that stores data in the internal memory and performs cryptographic operations on that data.

Motivation

- some host providers offer facilities to improve the security of sensitive data;
- it's achieved by using hardware security modules;
- maintaining these devices is expensive for customers.

Aim

- we want to develop a solution which security is comparable to HSM, but utilization and maintenance costs are much lower.

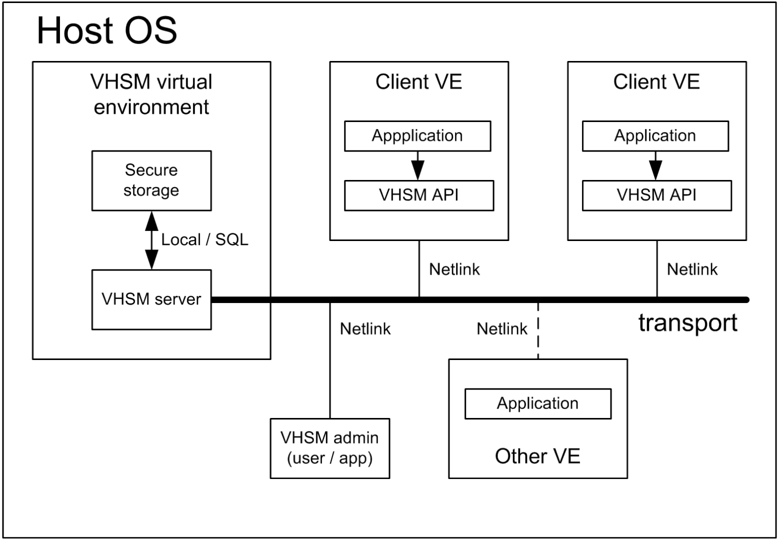
Idea of Virtual HSM

- Store the sensitive data and operate on them in one environment and process the results of cryptographic operations in the other.
- Runtime environments are represented by virtual containers.
- Client application cannot access the secret data directly — this is achieved by OS mechanisms.

Alternative solutions

- OpenDNSSEC SoftHSM or any other software token — secure storage implementation with PKCS#11 API.
  Disadvantages:
  - cryptographic operations are performed in a client application environment;
  - non-scalability.
- Trusted Virtual Securty Module (TvSM) — security module that uses Java VM as isolated environment.
  Disadvantages:
  - non-standard API;
  - can't be used by host providers;

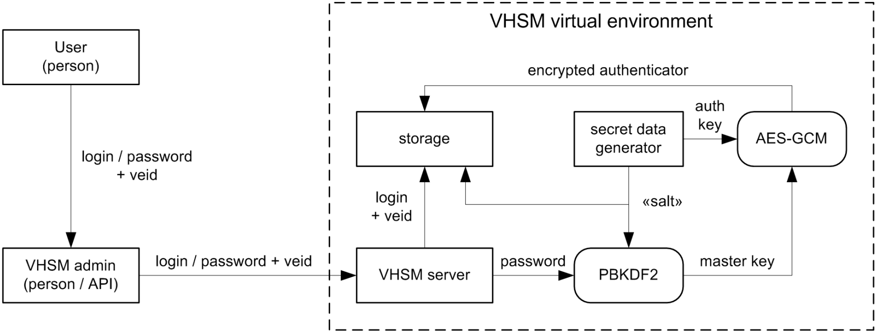
Virtual HSM architecture



Virtual HSM components

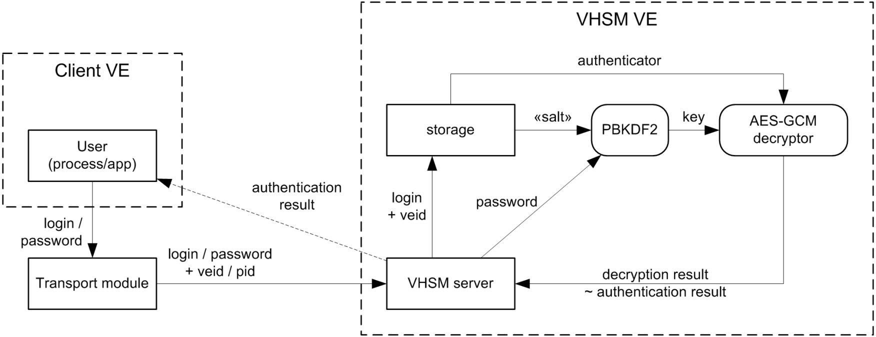
- **VHSM server**
  - authentication;
  - performs cryptographic operations on secret data;
- **Secure storage**
  - keeps encrypted user data;
- **Transport**
  - data exchange between client and server virtual environments;
  - container identification;
- **VHSM API library**
  - interaction with VHSM from client environment;

VHSM server — registration



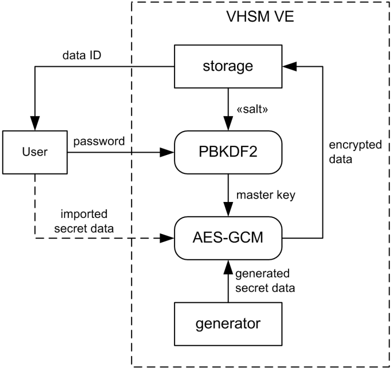
- registration via VHSM API by admin-user;
- the master key used for user data encryption is generated by the PBKDF2 from the user password;
- 256-bit authentication key encrypted with master-key in GCM mode is generated through the registration process;

VHSM server — authentication



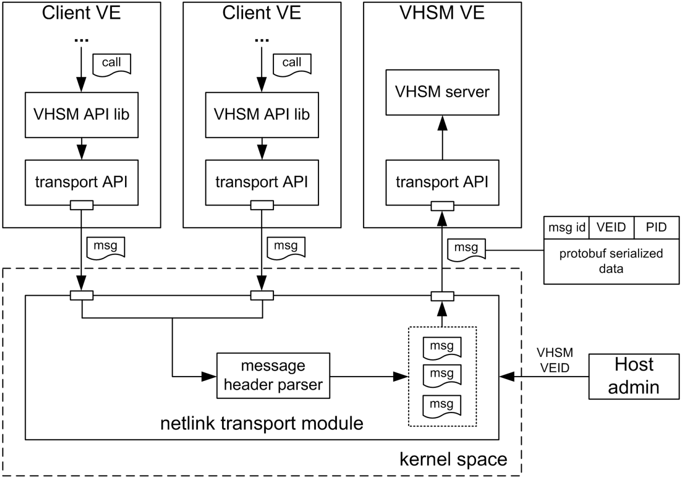
- user login and password + container id;
- auth key decrpyption success grants access to the VHSM;

Secure storage



- SQL database;
- encryption — AES-GCM with user master key;
- secret data are accessed by id;

Transport

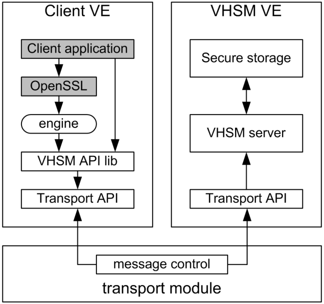


- high-level communication protocol — Protocol Buffers;
- inter-container communication — Netlink;

Client virtual environment

Part of PKCS#11 API:

- session management, user authentication;
- key management: import, generation, destroying;
- digital signature (HMAC-SHA1), encryption (AES-GCM);
- user management: creation, modification, destroying;



Threat analysis

Confidentiality

- reading secret user data from the client application memory
  - secret data are processed in the isolated and trusted environment only;
- user data disclosure due to database leakage
  - secret data are stored in the encrypted form, the encryption key is not stored in the persistent storage and derived from the user password;
- direct DB access / SQL-injection
  - the database is stored in isolated environment; SQL prepared statements usage.

Threat analysis

Privileges escalation / Accessibility

- sending of ill-formed messages
    - transport module checks the message header. Attacks on protobuf parser are difficult because of fixed message structure;
- DoS-attack by calling API functions or sending messages frequently
    - currently no protection is implemented;

Conclusion

Virtual HSM is one of possible implementations of the software HSM where logical execution environments are separated and isolated.

Advantages:

- host-providers don't require additional resources to maintain this solution;
- scalability is limited only by hardware resources.

Disadvantages:

- less secure than a real HSM;
- poor performance due to lack of hardware acceleration.

Links:

- repository: `http://git.openvz.org/?p=vhsm`
- wiki: `http://openvz.org/Virtual_HSM`

Thank You