There were a lot of people wanting to offer encryption in OpenStack services like Nova and Swift but with no execution plans.
We decided to build the Barbican platform that would solve these problems for OpenStack.

– Jarret Raim, Rackspace

Barbican to REST API, serwis secure storage do OpenStacka służący przechowywaniu kluczy I sekretów, wykorzystujący I łączący inne pluginy a także współdziałający z Keystone.

Opensource secret storage service zaprojektowano specjalnie do użycia z Openstack, w związku z tym jest to serwis wysoko skalowalny (API są skalowalne poprzez Load Balancer a SQL DB, RabbitMQ mają opcje High Availability).
Jego przesłaniem jest to, aby nie trzymać kluczy “obok” zaszyfrowanych danych.
Najlepiej działa w wersji OS Queens. Zostało tam usuniętych wiele sporych błędów (jak live migration, brak komunikacji z Novą (naprawiono w Pike) I usprawniono transparentne szyfrowanie wolumenów.

BARBICAN nie jest wystarczająco bezpieczny do wprowadzenia go na produkcję!
(a przynajmniej taki stan rzeczy był prezentowany w roku 2017 na Openstack Summit w Bostonie)

Barbican wykorzystuje tzw. “sekrety” do działania.
Sekret to pojedyncza pozycja, np.
-Klucz prywatny,
-Hasło,
-Klucz SSH

Jak tworzyć sekrety?

```
curl -X POST -H "content-type:application/json" -H "X-Auth-Token: $TOKEN" \
-d '{"payload": "my-secret-here", "payload_content_type": "text/plain"}' \
http://localhost:9311/v1/secrets
```

co daje output w postaci:

```
{"secret_ref": "http://localhost:9311/v1/secrets/2a549393-0710-444b-8aa5-84cf0f85ea79"}
```

Powyższy link to właśnie secure link. Do niego odnosi się Barbican chcąc odebrać sekret.
Powinny być one przechowywane w osobnej bazie danych najlepiej na osobnym serwerze w odrębnej sieci.

Provides:

- ReST API for Secrets Management
- Pluggable Backends
  - Simple Crypto
  - PKCS#11 and KMIP (HSM)
- Integration with Nova, Cinder, and Swift, Neutron, Heat, and many other OS projects
- Integration with KeyStone for Auth and RBAC
- Built to Scale

Does Not Provide:

- Graphical User Interface
- Key Splitting for Secure Import/Export Plain Text Keys using multiple Key Custodians
- Generation of X.509 Certificates (Since Pike)
- Volume Encryption

Pluginy:

1. Simple Crypto - najprostszy plugin, testowany na DevStacku. Jest default'owym secret store plugin. Wszystkie sekrety są szyfrowane przez klucz AES-256-bit key encryption key (KEK) przechowywany w konfiguracji Barbicana

2. PKCS#11 + HSM – jeden z najbardziej wszechstronnych pluginów. HMS, czyli Hardware Security Module. Podczas gdy Simple Crypto używa pojedynczego klucza, PKCS#11 używa kilku. HMS przechowuje dwa Master Keys: master encryption key (MKEK) I master HMAC signing key. HMS używa tych kluczy do szyfrowania sekretów dla każdego najemcy (dobra, to dziwnie brzmi, lepiej tak: for each tenant). Zaszyfrowane wartości są przechowywane w bazie danych Barbicana. Jest to drogie rozwiązanie, lecz zapewniające znaczną poprawę bezpieczeństwa względem Simple Crypto. Jeżeli nie jest potrzebna tak wysoka gwarancja bezpieczeństwa I można pozwolić sobie na lekki spadek wydajności, alternatywą może byc zastosowanie software-based HSM.

3. PKCS#11 + software HSM – Tak samo jak hardware'owy HSM, używa kilku kluczy. Zawiera to master key I key encryption key, który jest deszyfrowany przez master key. Nie jest bezpieczny I szybki jak hardware HSM, lecz znacznie poprawia bezpieczeństwo względem Simple Crypto. W planach OpenStack Stein jest wprowadzenie testowej wersji tego rozwiązania.

Plugin zewnętrzny KMIP **–** tak samo jak PKCS#11, używa HSM. Róznica polega na tym, że sekrety są przechowywana bezpośrednio na HSM, co zapewnia lepsze bezpieczeństwo kosztem potencjalnego ograniczenia wydajności i skalowalności

Barbican sam w sobie nie zapewnia szyfrowania wolumenów!
Używa LUKS, czyli szyfrowania Linuxa o którym poniżej:

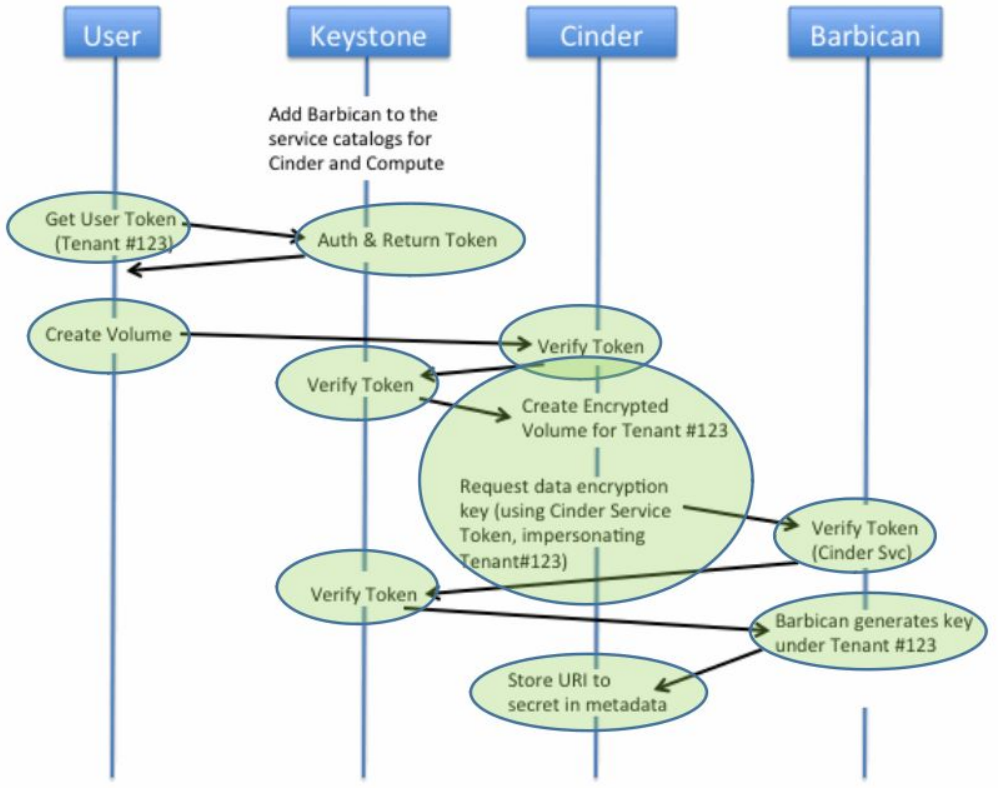
- LUKS - Linux Unified Key Setup
  - Allows for multiple user keys or passwords per volume
    - Master Key always stays the same
  - Supports CPU Hardware Acceleration (it's fast!)
  - Uses CryptSetup and DM-CRYPT
    - Decrypts full volume to a Local Block Device
    - Protects iSCSI attached volumes
    - Can also protect ephemeral storage if using LVM
- Queens = QEMU Native LUKS support
  - QEMU 2.6 and LibVirt 2.2 introduce native LUKS support

Barbican zawsze używa klucza Master Key

Deszyfrowanie wolumenów odbywa się na Hypervisorze, więc nie jest potrzebny host agent; działa na każdym systemie operacyjnym, działa z bootowalnymi wolumenami.
Ogólnie integracja serwisów Nova I Cinder zapewnia “Transparent volume encryption”.
Każdy wolumen jest chroniony swoim własnym kluczem

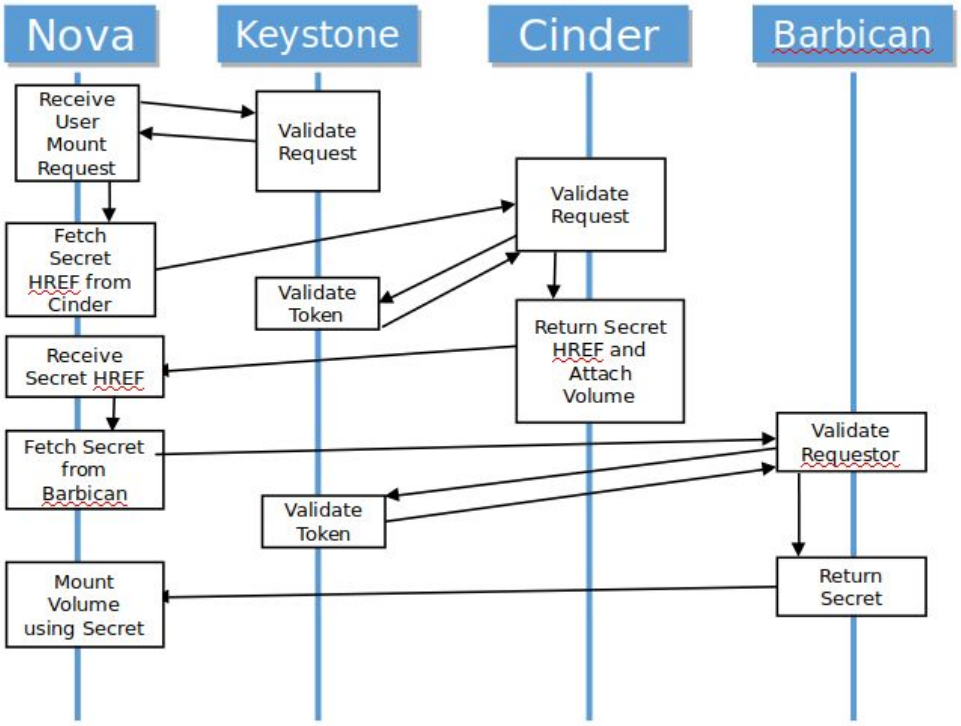
UWAGA!
Barbican nie zapewnia ochrony poszczególnych plików. Jeżeli ktoś zdobędzie dostęp do systemu, nie będzie widział żadnych zabezpieczeń I pliki będą dla niego otwarte.

Workflow Barbicana:



1. Użytkownik dostaje token od Keystone
2. Użytkownik zleca Cinderowi utworzenie wolumenu z użyciem tego tokenu
3. Cinder weryfikuje token użytkownika od Keystone’a
4. Cinder odpytuje Barbicana z klucza
5. Barbican sprawdza token od Cindera z tokenem od Keystone’a
6. Barbican tworzy sekret I zwraca secure link do Cindera
7. Cinder przechowuje secure link (secret HREF) w metadanych wolumenu

Schemat montowania zaszyfrowanego wolumenu



Zgodnie z opinią krążącą w internetach, dokumentacja Barbicana jest nieintuicyjna, informacje są porozrzucane I mało użyteczne, jednak deweloperzy są bardzo pomocni I z ich wsparciem można zainstalować Barbicana.

Potencjalnie przydatne linki (od, moim zdaniem, najlepszych):

Prezentacja Barbicana (skąd brałem obrazy) z Openstack Summit 2018
Wytłumaczenie zasady działania, co jest plusem zastosowań Barbicana a co minusem, jakie błędy mogą się pojawić podczas instalacji. Wydaje się najlepszym źródłem wiedzy, choć mocno teoretyczna
https://www.youtube.com/watch?v=pWu0fDgBbk4
Link do demo (prezentacja powyzej, od 24:56)
demo
GitHub gościa wyżej, instrukcja instalacji I pliki konfiguracyjne
https://github.com/dwannamaker-onr/openstack-queens-barbican-guide

Trochę o API: szablony, templatki I przykłady requestów. Można szybko przelecieć, ale jak coś by było potrzebne, to wiadomo gdzie szukać
https://www.youtube.com/watch?v=zVQc053falE

Pluginy I opcje deploymentu
https://opensource.com/article/18/10/are-your-secrets-secure

Dokumentacja RedHat do instalacji Barbicana (wydaje się dość szybka oczywiście o ile nie będzie pluło błędami)
https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/13/html/manage_secrets_with_openstack_key_manager/installing_barbican

Cała dokumentacja Barbicana. Tak jak pisałem, mocno postrzępiona I rozwalona na wiele części.
https://wiki.openstack.org/wiki/Barbican/Documentation

Spis komend do cli Barbicana
https://docs.openstack.org/python-barbicanclient/stein/cli/details.html

Porównanie Vault I Barbicana (zasada działania, co kiedy lepiej użyć)
https://www.youtube.com/watch?v=4wTJFGfRyCc

Baza Openstackowa
https://docs.openstack.org/barbican/latest/

Deploy zaszyfrowanego wolumenu na queens. Jest jeszcze pt.1 gdzie stawia controller
https://www.youtube.com/watch?v=mXXjVGdyUXY&t=295s