Security APIs

Motivation

Currently, EVE does not have capability to provide data security at rest. This is being designed and implemented [1]. With this, EVE will provide capabilities like file system encryption, but it is up to the EVE Controller to make use of these capabilities to to achieve the security goals. For this EVE needs to define its interface towards EVE controller, and provision a way to define security policies from the Controller. This proposal focuses more on the interaction between EVE and EVE controller(EVC) in the context of realising a use case that the user might have to secure data processed on the EVE platform.

Terminology

We introduce a few terms here for better understanding of this proposal
Edge Container Objects (ECO) - A VM or Container deployed on an EVE instance.
ECO Images - The image file for a particular VM or Container. The image that is used for deploying the VM or Container for the first time in the production environment.
Mutated ECO Image: As the ECO starts running on an EVE instance, it continuously changes its runtime state, and it starts accumulating data feeds from its external interfaces. All this is stored on its virtual disk. We call this virtual disk that contains the modified ECO state as mutated ECO image
Local File Store - Space on the permanent storage disk on EVE instance that is consumable from ECO, e.g. /persist/img . A file store need not be a secure file system.
Vault - A secure version of a Local File Store, where the files are encrypted using filesystem encryption support (e.g. fscrypt)

Sample Use Cases

Assuming that EVE provides a capability to store some files in an encrypted filesystem, we can foresee the following use cases:

a) A user might want to run the Edge Containers out of this secure file system, so that data that is stored by these Edge Containers is stored in encrypted form at rest. A user might do this is to prevent an attacker from reading the application data if the EVE node is stolen or drive is taken out.

b) A user might also wish to store sensitive parts of EVE configuration (e.g. Image data store credentials), under this secure file system, so that it stays encrypted at rest.

c) A user might also want to be able to create secure file stores and be able to associate an arbitrary Edge Container with such a secure file store.

d) A user might choose to use a separate file store for each of his Edge Containers on an EVE node - so that compromising one vault does not lead to access to data of all the Edge Containers

e) A user might want to control security policies for such file stores using user-defined policies, e.g. whether key is protected by IP fencing, TPM attestation etc. He might also decide whether key for the vault is to be provided from Controller or it can be from the TPM on the EVE platform.

Proposed Interface between EVE and EVC

We are proposing to have a user-visible construct called “Vault”. A Vault is a secure file system, protected by native file system encryption. Therefore the interface has 3 parts to it:

a) Lifecycle management of a “Vault” - CRUD (Create, Replace, Update, Delete) for Vaults

- A list of vault configuration as part of the EVE node configuration

- EVE node will post the status messages for the Vault CRUD operation results.

b) Association of Edge Containers with a Vault - To control data at rest requirements of a Edge Container

- The app instance configuration may include a reference to a defined vault.

  The Vault will be used store the mutated business sensitive information for the container.

c) Attestation of the device through PCR quote and Nonce, Geo-location, IP Address information etc.

- This will be used for remote attestation challenge/response exchange between EVC and EVE node.

  This will be done on device reboot as well as at periodic intervals, to make sure the EVE node is not compromised.

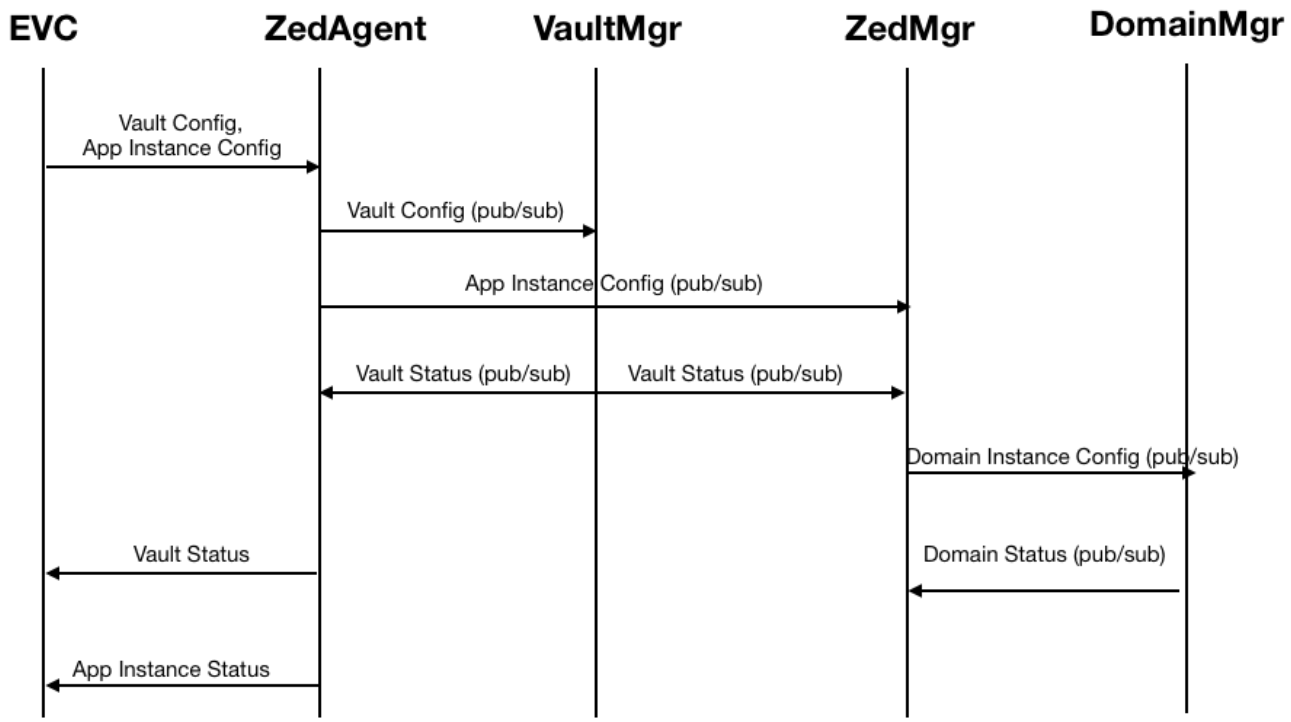
Same API that carries other config, handled by zedAgent

Vault related configuration would be pushed along with other config (by /api/v1/eddgedev/config), and parsed by zedagent. Zedagent would interact with Vault Manager service for implementing file system encryption requirements. Any file system interaction to setup/unlock the vault directory will have to be done by Vault manager according to the security config received, and then signal others that vault directory is now ready for use. Zedmanager will synchronise with Vault Manager to make sure the Vault is ready to use before any edge container that needs this vault is started by domain manager. Other services can listen to Vault Manager to perform any task they need to do on top of the Vault directory(Currently only zedagent and zedmanager).

Presence / absence of a Vault configuration below will implicitly drive creation/retainment or, deletion for the Vault.

Component Interaction

The following diagrams describe component level interaction, to handle the above config items:



Break-up of the proposed Vault Config

- Identity of the Vault
- Security Policy for the Vault
- Key Information for the Vault

Vault Identifier

UUID - Unique Id generated by EVC for the Vault

Version - For handling message schema change in future

Name - User provided name string for the Vault

Vault Security Policy

Data handling policy will define operational mode of the vault:

- Lock
- Unlock
- Change Key

Key Information

If controller is configured to use EVC generated keys for the Vault, this section will carry the key information to be used for the associated vault

Fscrypt provides a way to change the master key associated with an encrypted folder, without re-encrypting the contents. This is possible due to the protectors and policies constructs used by fscrypt (master key protects the protector, and protector in turn protects the final key used for encryption). Please see here for more details.

We can use this fscrypt feature to periodically rotate the master keys used for a given vault. The key rotation policy will be in the controller and will not be intimated to EVE. For a key rotation scheme, a maximum of two keys will be intimated to the EVE node. Controller will store and publish, the last published key along with the most current key. This will cover cases, when the EVE node is not able to communicate with controller. If there is no key rotation configured, both old and new key fields in the configuration will be the same.

Association of Edge Container with the Vault

App Instance configuration will carry this information - Whether the App is protected by End-to-End Security, and if yes, what is the Vault to associate this App Instance with. Zedmanager will consume this configuration, and co-ordinate between Vault manager and Domain Manager to make sure the required Vault is ready before launch of the User Application.

Components Interacting with RW Partition on EVE

Component	Directory/File	Comments	Contains Sensitive Data?
Domain Mgr	/persist/img /persist/rkt	for storing the mutable ECO disk images	Yes
Downloader	/persist/downloads	for downloading Edge Container Images	No
Verifier	/persist/downloads	for verifying integrity of downloaded images	No
ZedAgent	/persist/config	for storing EVE device configuration	Yes
TPM Mgr	/persist/config/tpm_in_use	for marking TPM mode of operation	No
device-steps.sh	/persist/IMGA, /persist/IMGB	for storing image specific logs, info	No
Network Interface Manager (NIM)	/persist/status	for storing DevicePortConfigList	No

Providing Security By Default

While the interface described provides way for a user to create and manage his own “Local File Stores”, and configure policies for it (like storage limit, encryption enablement, key rotation frequency) and associate them with his ECO Images (e.g. ECO 1 to use Local File Store A, ECO 2 and 3 to use Local File Store B etc), what might be easier for the user is to have some Vaults created by default by EVE, and thus user might need to do nothing to secure his ECO instances, and it is enabled by default for a user who does not know/care/want to control Vaults at a much granular level.

Therefore it is proposed that, we create a couple of Vaults by default:
a) A Vault to store EVE device configuration (for EVE host OS consumption) - let's call it Config Vault - to store sensitive parts of EVE device configuration (e.g. S3 credentials)

b) A Vault to store ECO related files (for ECO consumption) - let’s call it Image Vault - to store and launch mutated ECO images

Even though these vaults are created by default, a User (if he wants) can change the policies associated with these Vaults, through the interface specified in this proposal, like he would do for any user-created Vaults.

Attestation Challenge by EVC

This is to challenge EVE to provide a requested information, to prove EVE's software/physical location states are untampered. On successful response, further config updates will have Vault section with appropriate Vault config like keys. On failing to provide a satisfiable response, EVC will not send the vault configuration to EVE, and will keep sending Attestation Challenge in place of Vault configuration. Attestation Challenge can be:
a) PCR quote with nonce included

b) Geo location along with the IP address

Attestation Challenge will be handled by TPM manager, after zedagent publishes the config to TPM Manager. Details about attestation are outside the scope of this document. What concerns here is the fact that, based on attestation outcome, EVC may not (based on user configured policies) reveal the Vault Key, by not sending any Vault config to EVE.

Security Threats Addressed

Security Threat Scenario	TPM Key	Controller Key	Controller Key Rotation	Key from TPM + Controller	TPM + Controller Key with Attestation	TPM + Controller Key with Attestation, with Key Rotation
Storage drive is taken out and inserted into another system/PC to read the data from the SSD directly using offline crypto tools	Protected	Protected	Protected	Protected	Protected	Protected
Storage drive is taken out and inserted into another system/PC to read the data, by spoofing the Device Identity and talking to Controller	Protected	Not Protected	Not Protected (on non-TPM devices)	Protected	Protected	Protected
EVE device is taken out, and booted up in another location to access its data, but the theft has been detected	Not Protected	Protected	Protected	Protected	Protected	Protected
EVE device is taken out, and booted up in another location to access its data, but no knowledge of it being stolen	Not Protected	Not Protected	Protected	Not Protected	Protected (Using Geo Fencing)	Protected (Using Geo Fencing)
EVE device is not taken out, but some other malware is loaded on the system, and is used to get access from remote to access the information	Not Protected	Not Protected	Not Protected	Not Protected	Protected (PCR value change detection)	Protected(PCR Value change detection)
Brute force attack for Key identification	Not Protected	Not Protected	Protected	Not Protected	Not Protected	Protected

References

1. https://wiki.lfedge.org/display/EVE/Encrypting+Sensitive+Information+at+Rest+at+the+Edge
2. The pull request corresponding to this proposal: https://github.com/lf-edge/eve/pull/186