Garantir

GaraSign HashiCorp Vault Integration

Table of Contents

Preface

Document Information

Title	GaraSign HashiCorp Vault Integration
Product Name	GaraSign
Product Version	1.9.0 & above

Trademarks

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. Without limiting the rights under the copyright reserved above, no part of this publication may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the prior written permission of Garantir.

Disclaimer

Garantir makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Garantir reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Garantir to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be without error or otherwise perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

Please send any constructive comments on the contents of this document to the following email address: support@garantir.io

Document Overview

GaraSign can integrate with multiple different HSMs and key managers and can even do so simultaneously. This document describes how to integrate GaraSign with HashiCorp's Vault as its key container, via Vault's Transit engine.

Intended Audience

All products produced by Garantir are designed to be installed, configured, operated, and maintained by personnel with the necessary knowledge, skill, training, and qualifications to safely perform their duties. This document is intended for personnel responsible for architecting, engineering, installing, configuring, operating, and/or troubleshooting enterprise signing solutions. It is assumed that the readers of this document and the users of its content are proficient with:

- Basic networking concepts
- Security concepts including, but not limited to, authentication, authorization, digital signatures, and logging
- Installing, configuring, and using HashiCorp Vault

Additionally, it is strongly recommended that readers of this document first read the GaraSign datasheet.

GaraSign Overview

GaraSign is a flexible enterprise cryptographic signing platform that carefully balances security and performance. Unlike many solutions today that require clients to upload the data to be signed to a central location, GaraSign is designed as a remote signing platform. As shown in Figure 1 below, GaraSign clients first hash the data to sign and then send the computed hash value over a secure channel to the GaraSign server for signing. Using this approach, no matter how large the data is, the data sent over the network is minimal which allows for optimum performance. Behind the GaraSign Signing Server sits one or more cryptographic tokens (e.g., HSMs, key managers, etc.) that GaraSign can offload signature processing to once the appropriate authentication and authorization checks have been performed.

By placing the GaraSign server between the client and the back end HSM, not only is performance significantly increased, other enterprise features such as advanced authentication, enterprise logging, email notifications, and more are enabled. As an additional benefit, the GaraSign server acts as a buffer between your end clients and your HSMs, which significantly reduces the integration complexities that your clients must deal with and helps to shield your cryptographic keys from attack and misuse.

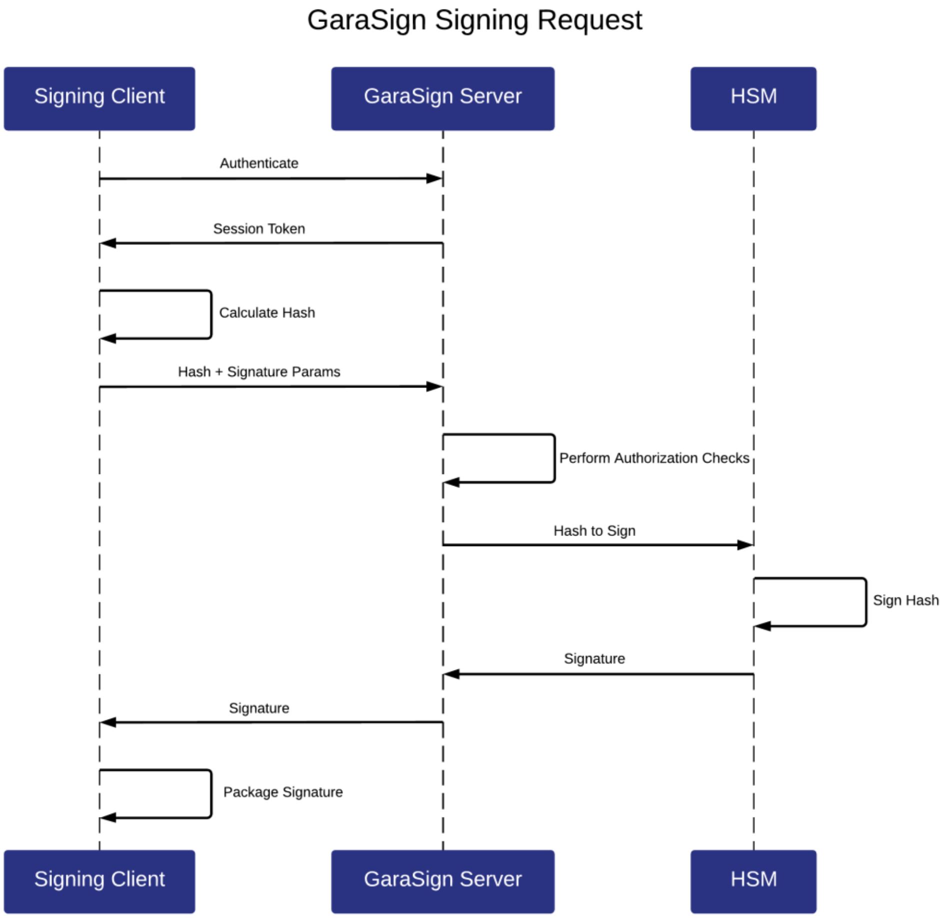


Figure 1 - GaraSign Signing Request

Note: while this document makes mention of REST servers, GaraSign is designed to sit on-premise in your network and/or in your virtual private cloud. Garantir does **not** have access to any of your infrastructure and does **not** control your cryptographic keys.

Supported Configurations

From a cryptographic and API perspective, GaraSign makes use of the standard Vault Transit Engine which is available as part of the Open Source Vault. However, customers may wish to use the Enterprise version of Vault for better High Availability, integration of Vault with an HSM, and support. Please see your Vault documentation for more information on Enterprise vs Open Source features.

Supported Keys and Algorithms

GaraSign supports signing data with RSA and Elliptic Curve keys. While GaraSign does not impose any restrictions on the key size or curve type, the following table provides the list of keys that are officially supported and tested with each GaraSign release:

Key Type	Size/Curve
RSA	2048, 3072, 4096
Elliptic Curve	NIST P-256, NIST P-384, NIST P-521

Clients can sign data using any of the key types listed above with any of the following hash algorithms:

- MD5*
- SHA-1
- SHA-224 (SHA-2)
- SHA-256 (SHA-2)
- SHA-384 (SHA-2)
- SHA-512 (SHA-2)
- SHA3-224*
- SHA3-256*
- SHA3-384*
- SHA3-512*

*Only supported by exportable keys. Note: GaraSign never exports keys to the signing clients, they can only ever be exported to the Signing and Administrative servers, if configured to do so. See FAQ for more information.

HashiCorp Vault Integration

Integrating GaraSign with the HashiCorp Vault is done by performing the following steps:

1. Open the firewall so that all GaraSign Administration and Signing server can communicate to Vault over the required port (default: 8200)
2. Retrieve the Vault token for GaraSign to use to make use of the desired Vault Transit engine
3. Install the GaraSign software for each node in your GaraSign cluster
4. Start (or restart) the Tomcat instances on the Signing and Administration servers
5. From the GaraSign Administrative Console, create a Key Container of type Vault

Details for steps 1 and 2 can be found in your HashiCorp Vault documentation.

Details for step 3 can be found in your GaraSign documentation, although this is typically handled by your GaraSign professional services personnel.

The rest of this section focuses on step 5.

Create HashiCorp Vault Key Container

Follow the steps in your GaraSign Admin User Guide documentation to launch the GaraSign Administrative Console and login. Once logged in, execute the following steps:

1. From the *Main Menu* select *Key Management*, *Key Container Management* and then *Create Key Container*.

2. Give the key container a name. Note: this name must be globally unique amongst all key containers in your GaraSign deployment.

3. For *Key Container Type* choose *Vault.*

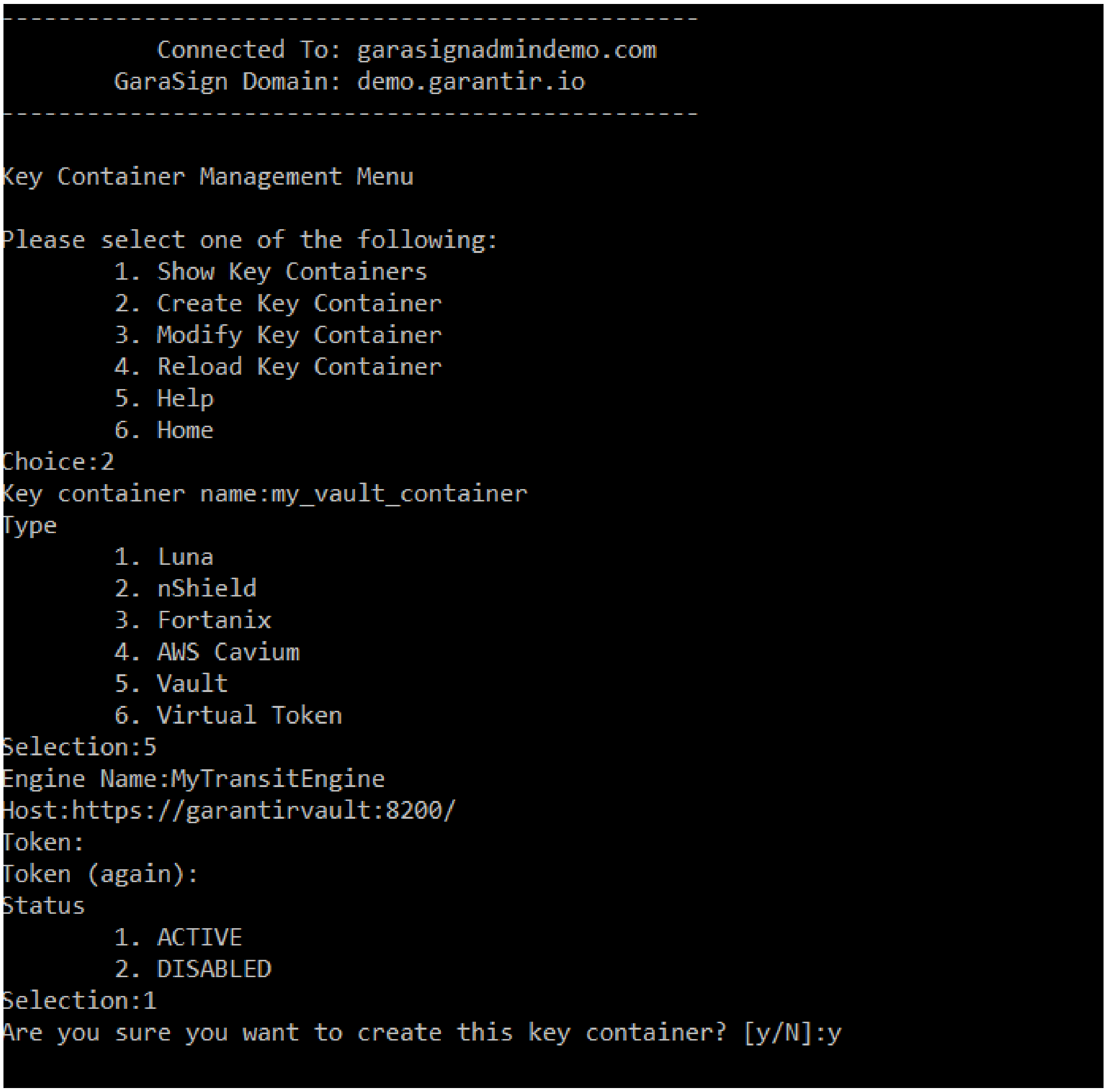
4. Enter the name of the Transit engine you wish to use.

5. Enter the URL to your Vault instance. Note: GaraSign does full certificate chain and hostname validation.

6. Enter the token for GaraSign to use to connect to Vault with. This token should have permissions on the chosen Transit engine to read, list, and create keys as well as to sign and decrypt data with those keys.

7. Choose whether this key container is to be Active or Disabled. In most scenarios the container should be Active. Please see your GaraSign Administrative User Guide for more information.

8. At the confirmation prompt please check that the information you provided is accurate. If it is, type *y* and then press Enter. Otherwise, just press Enter to cancel. Once confirmed, the process may take several moments to connect to your HashiCorp Vault cluster. Please be patient.



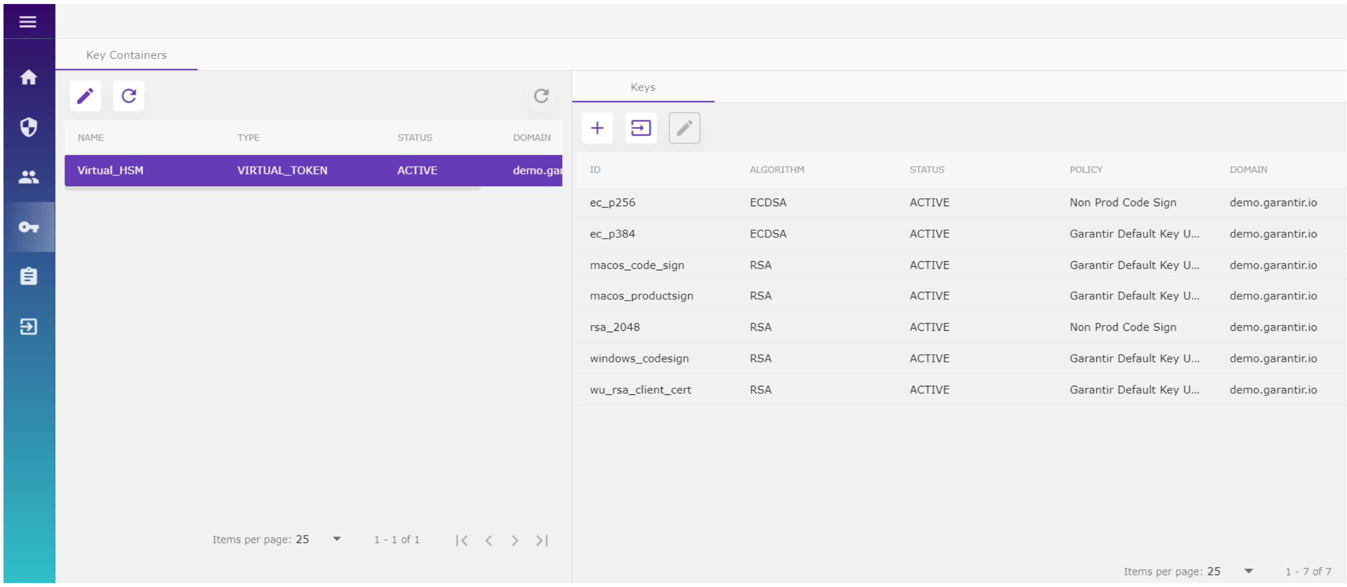
```
------------------------------------------------
          Connected To: garasignadmindemo.com
       GaraSign Domain: demo.garantir.io
------------------------------------------------

Key Container Management Menu

Please select one of the following:
        1. Show Key Containers
        2. Create Key Container
        3. Modify Key Container
        4. Reload Key Container
        5. Help
        6. Home
Choice:2
Key container name:my_vault_container
Type
        1. Luna
        2. nShield
        3. Fortanix
        4. AWS Cavium
        5. Vault
        6. Virtual Token
Selection:5
Engine Name:MyTransitEngine
Host:https://garantirvault:8200/
Token:
Token (again):
Status
        1. ACTIVE
        2. DISABLED
Selection:1
Are you sure you want to create this key container? [y/N]:y
```

Figure 2 - Key Container Setup

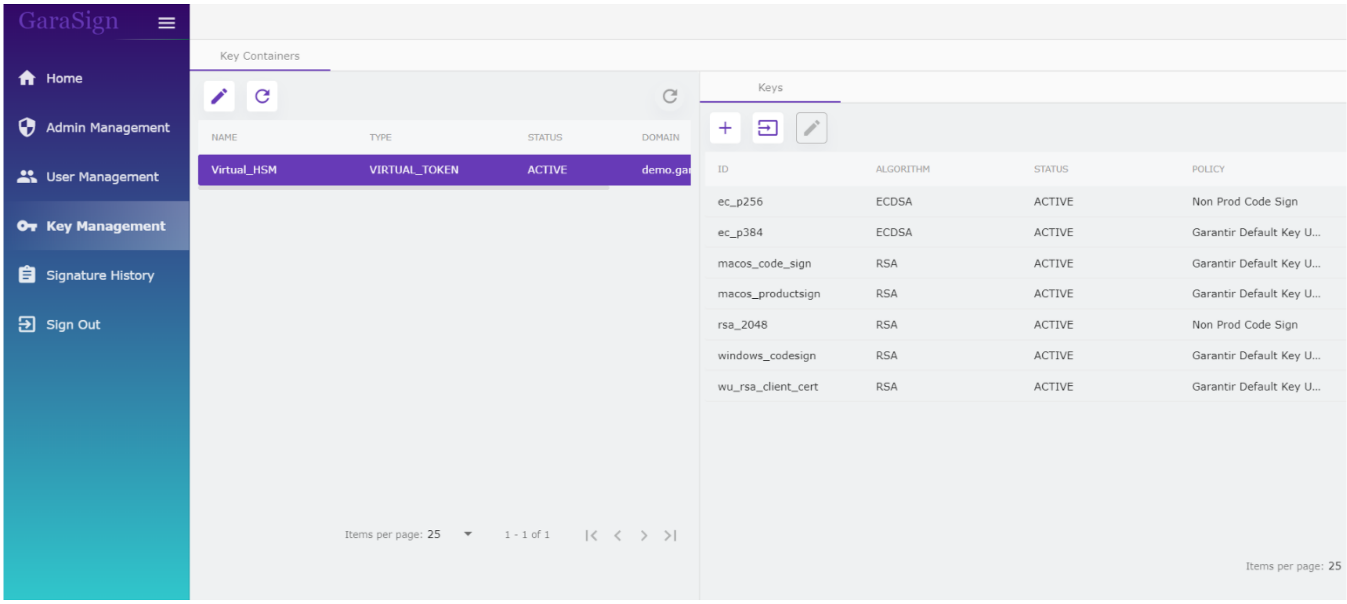
Once complete, the Vault instance can be used like any other key container in GaraSign. You can now use HashiCorp's Vault with GaraSign to create keys, generate certificate signing requests (CSRs), import certificates, sign and decrypt data, and more. Additionally, further administration of your key container can be done from the more user-friendly web UI, as shown below. For more information on how to use GaraSign for key management, please see your GaraSign Administrative User Guide.



NAME	TYPE	STATUS	DOMAIN
Virtual_HSM	VIRTUAL_TOKEN	ACTIVE	demo.gar

ID	ALGORITHM	STATUS	POLICY	DOMAIN
ec_p256	ECDSA	ACTIVE	Non Prod Code Sign	demo.garantir.io
ec_p384	ECDSA	ACTIVE	Garantir Default Key U...	demo.garantir.io
macos_code_sign	RSA	ACTIVE	Garantir Default Key U...	demo.garantir.io
macos_productsign	RSA	ACTIVE	Garantir Default Key U...	demo.garantir.io
rsa_2048	RSA	ACTIVE	Non Prod Code Sign	demo.garantir.io
windows_codesign	RSA	ACTIVE	Garantir Default Key U...	demo.garantir.io
wu_rsa_client_cert	RSA	ACTIVE	Garantir Default Key U...	demo.garantir.io

Figure 3 - Menu Expanded



NAME	TYPE	STATUS	DOMAIN
Virtual_HSM	VIRTUAL_TOKEN	ACTIVE	demo.gar

ID	ALGORITHM	STATUS	POLICY
ec_p256	ECDSA	ACTIVE	Non Prod Code Sign
ec_p384	ECDSA	ACTIVE	Garantir Default Key U...
macos_code_sign	RSA	ACTIVE	Garantir Default Key U...
macos_productsign	RSA	ACTIVE	Garantir Default Key U...
rsa_2048	RSA	ACTIVE	Non Prod Code Sign
windows_codesign	RSA	ACTIVE	Garantir Default Key U...
wu_rsa_client_cert	RSA	ACTIVE	Garantir Default Key U...

Figure 4 - Menu Collapsed

Frequently Asked Questions

Are the keys exportable to the client?

No. The signing keys are generated as non-exportable and GaraSign does not expose any API to retrieve raw key bytes. However, keys marked as exportable in Vault will be exported to the GaraSign Signing and Administration servers. This can result in higher performance but lower security. Note: GaraSign never creates keys as exportable. The only way to use exportable keys in GaraSign is to create them in Vault as exportable and perform a GaraSign Add Key function (instead of Create Key).

How is High Availability (HA) achieved with GaraSign and HashiCorp Vault?

GaraSign makes use of the native Vault HA capabilities. Please see your Vault documentation for more information.

Does using the HashiCorp Vault slow down the process of signing?

No. Since GaraSign generates the hashes client-side, the network usage is minimal which results in fast signatures. No longer do customers have to choose between security and performance.

How fast can GaraSign produce signatures?

Extremely fast. The exact speed will be dependent on your environment (e.g., network latency, computer speeds, etc.) but GaraSign's client-side hashing architecture always results in high performance.

Is it possible to place GaraSign in the cloud but use a Vault instance on-premise?

GaraSign is designed to run on-premise, in the cloud, or in a hybrid environment. For customers who wish to keep their Vault instances on-premise but utilize the cloud for scaling, GaraSign servers in the cloud can make use of the on-premise Vaults provided that network connectivity is available. Customers can also choose to connect their GaraSign instance to a Vault instance in the cloud or hybrid on-premise and cloud Vault instances.

Can I use my own Certificate Authority (CA) with GaraSign?

Yes. GaraSign supports multiple CAs and certificate protocols to allow for easy and flexible issuance of certificates. A single GaraSign deployment can integrate with multiple CAs simultaneously allowing for maximum flexibility.

Does GaraSign support more than just signing?

Yes. GaraSign also supports Elliptic-Curve Diffie-Hellman (ECDH) and RSA decryption. In all cases, the private keys are never exported to the client.

Where can I learn more?

Please get in touch with us via email at info@garantir.io.