Garantir

GaraSign Deployment Planning

Version 1.25.0

Table of Contents

Preface

Document Information

Title	GaraSign Deployment Planning
Product Version	1.25.0
Release Date	July 2024

Trademarks

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 is made up of many components, some required and some optional. This document provides an overview of planning a GaraSign deployment.

Note: while this document is intended to help customers plan a GaraSign deployment, it is still required that Garantir performs or oversees the actual deployment.

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

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

GaraSign Overview

GaraSign is a flexible enterprise cryptographic signing platform that carefully balances security and performance. Unlike most 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 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 made. 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 not only significantly reduces the integration complexities that your clients must deal with but also helps to shield your cryptographic keys from attack and misuse.

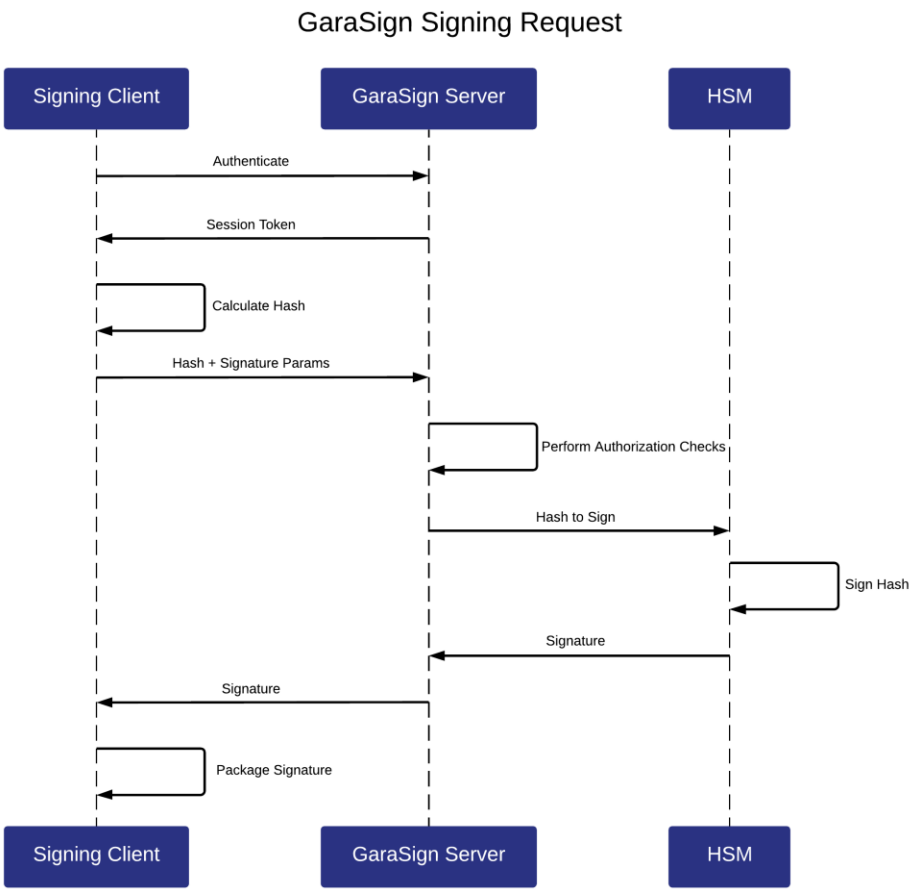

Figure 1 - GaraSign Signing Request

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.

Architecture

This section is a duplicate of the Architecture section from the GaraSign Architecture document. It is provided here as a convenience to the reader but not as a substitute to reading the GaraSign Architecture document.

As shown in Figure 2 below, the entire GaraSign architecture has many components.

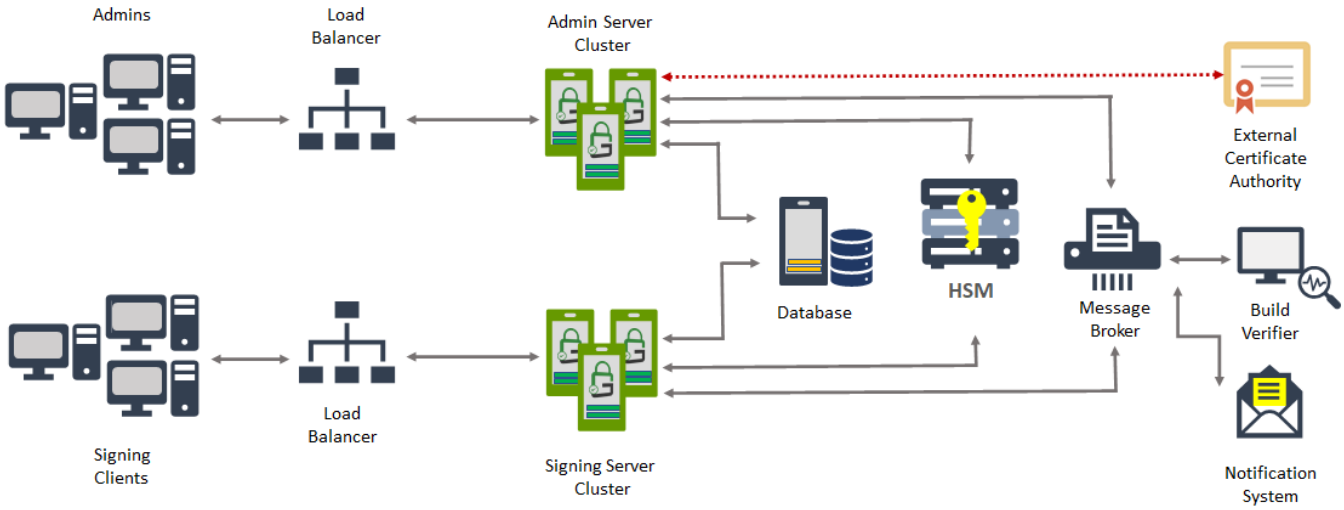

Figure 2 - GaraSign Architecture Diagram

GaraSign is made up of the following components:

1. Cryptographic Token – The cryptographic device(s) that store the signing keys (usually one or more HSMs)
2. GaraSign Signing Server – The REST server that sits in front of the cryptographic tokens that store the signing keys
3. GaraSign Signing Clients – Clients that allow the signing tools they integrate with to locally hash data and offload signature generation to the GaraSign Signing Server
4. GaraSign Administration Server – The stateless REST server that allows administrators to perform their duties (e.g., user management, key management, etc.)
5. GaraSign Administration Clients – The client software that allows administrators to interface with the Administration Servers to perform administrative duties
6. GaraSign Hash Validator – The servers configured to validate hash values either before or after signing occurs
7. GaraSign Notification Server – The server that sends out notifications, usually in the form of email
8. GaraSign Message Broker – The messaging system that allows the Signing, Administration, Hash Validator, and Notification servers to communicate
9. Database – The database that stores persisted information such as users, key metadata, and signature history
10. Load Balancer(s) – Used to balance load in front of the various service endpoints

For more details on the GaraSign's architecture, please see the GaraSign Architecture document.

Planning Your Deployment

Cryptographic Token

The core of any GaraSign deployment is its cryptographic tokens. Since GaraSign integrates with multiple different cryptographic tokens, these tokens and their software libraries drive some of the decisions to be made about the Signing and Admin servers. For example, some cryptographic tokens require the Signing and Admin servers to be run on particular operating systems. Your Garantir representative can inform you of any such requirements that your cryptographic token(s) have.

Notifications

GaraSign makes use of notifications for security purposes. Notifications are currently implemented as email or Slack messages. You should decide which notification method your deployment will use. While not often done, it is possible to change the notification method of a GaraSign deployment.

Scalability

All GaraSign nodes are designed to be horizontally scalable. The number of each node type deployed is up to the customer although Garantir strongly recommends that more than one of each node type is deployed and that customers choose appropriate sizing based on their expected worst-case usage. If more than one signing or admin server is to be used (as is recommended), load balancers are recommended, although load balancing can also be achieved via DNS techniques. Your Garantir representative can help you select the right sizing based on your expected use.

Authentication

GaraSign supports five first factor authentication methods – username/password, client-certificate, Kerberos, OpenID Connect, and SAML (admin interface only). Customers are free to make use of different authentication methods for different users, but customers should be aware that the choice of authentication can have an impact on how load balancers are configured. Most notably, if traditional client-certificate authentication is used, TLS should **not** be terminated at the load balancer, although GaraSign supports an application-level approach of client-certificate authentication that does allow for TLS termination at the load balancer.

High-Level Configuration

Signing Servers

Windows or Linux servers (exact type dictated by cryptographic tokens but should be the same as the GaraSign Administration Servers) with Java 8 and Tomcat 9 installed. GaraSign software is deployed as a .war file on Tomcat. Cryptographic-token-specific client software must also be installed on these servers.

Minimum System Requirements:

- CPU/vCPU: 1
- RAM: 4 GB
- Hard Disk: 30 GB

Administration Servers

Windows or Linux servers (exact type dictated by cryptographic tokens but should be the same as the GaraSign Signing Servers) with Java 8 and Tomcat 9 installed. GaraSign software is deployed as two .war files on Tomcat. Cryptographic-token-specific client software must also be installed on these servers.

Minimum System Requirements:

- CPU/vCPU: 1
- RAM: 4 GB
- Hard Disk: 30 GB

Database

MySQL (version 8+) or SQL Server (version 2019+) database. The exact configuration is deployment-specific but common configurations are a Galera cluster, primary-secondary replication, or cloud-provider-managed database (e.g., AWS RDS). Appropriate sizing should be made based on the expected volume of signatures and certificates. Additional considerations should be made for replication to disaster recovery sites.

Message Broker

Server running ActiveMQ 5.15.X or later. Cloud-provided managed services such as AmazonMQ may be used. Multiple instances are recommended to be used with the *failover* protocol.

Notification Server

Windows or Linux server later running Java 8. GaraSign software deployed as a runnable jar file.

Build Verifier

The Build Verifier is an optional component that is used for Automated Hash Validation. In large enterprises there may be multiple Build Verifiers on different operating systems to support different build configurations. At a minimum, all Build Verifiers must have Java 8 installed. The GaraSign software is deployed as a runnable jar file.

Firewall Rules

Signing Servers

Depending on the deployment model, the clients either communicate directly with the signing servers or communicate with one or more load balancers which then communicate to the signing servers. When a load balancer is placed between the clients and the signing servers, the signing clients should not have direct network access to the signing servers.

Recommended Inbound Port: 443 or 8443

Outbound Network Access: Cryptographic Token(s), Database, Message Broker, and CA CRL and/or OCSP responder endpoints.

Administration Servers

Depending on the deployment model, the administrative clients either communicate directly with the administration servers or communicate with one or more load balancers which then communicate to the administration servers. When a load balancer is placed between the administrative clients and the

administrative servers, the administrative clients should not have direct network access to the administrative servers.

Recommended Inbound Port: 443 or 8443

Outbound Network Access: Cryptographic Token(s), Database, Message Broker, and CA endpoints (CRL and/or OCSP, API endpoints for issuance, revocation, renewal, etc.).

Database

The database is accessed by the signing and administrative servers. Depending on the deployment model, there may be other database nodes communicating for replication and/or HA purposes.

Recommended Inbound Port: 3306

Message Broker

The message broker is accessed by the signing, administrative, notification, and build verifier servers. Depending on the deployment model, there may be other message broker nodes communicating for replication and/or HA purposes.

Recommended Inbound Port: 61443.

Notification Server

Messages are sent to the notification server by the signing and administrative servers via the message broker. Depending on the deployment model, there may be other message broker nodes communicating for replication and/or HA purposes.

Outbound Network Access: Message Broker and external notification system (e.g., SMTP, Slack, etc.). The latter depends on the notification method(s) chosen.

Build Verifier

Messages are sent to the build verifier by the signing server via the message broker. The build verifier must be able to retrieve source and/or binaries from the preconfigured trusted repositories.

Outbound Network Access: Message Broker and external repositories (e.g., source code repository). The latter depends on the repositories chosen.

Advanced Configuration

In some cases (e.g., when evaluating GaraSign in a proof-of-concept environment), it may be desirable to deploy GaraSign with the minimal number of (virtual) machines possible, at the expense of reduced high-availability and performance. In this case, it is possible to merge any of the components listed above, except the Administration and Signing Servers. Both of those servers must be hosted on separate virtual machines.