EVERTRUST

# Secret Rotator

Version 1.0, 2026-11-02

# Table of Contents

1. Usage . . . . . 4
2. Release notes . . . . . 7

*This section describes how to rotate the secrets encrypted with a Tink keyset in Horizon and Stream*

# Installation

**RHEL**



Secret rotator requires java 17 or 21 installed

**Installation from the EverTrust repository**

Create a `/etc/yum.repos.d/evertrust-secret-rotator.repo` file containing the EverTrust repository info:

```
[evertrust-secret-rotator]
enabled=1
name=Secret-rotator Repository
baseurl=https://repo.evertrust.io/repository/evertrust-secret-rotator-rpm/
gpgcheck=1
gpgkey=https://evertrust.io/.well-known/rpm/gpg.pub
username=<username>
password=<password>
```

Replace `<username>` and `<password>` with the credentials you were provided.

Make sure the Evertrust GPG key is trusted:

```
# rpm --import https://evertrust.io/.well-known/rpm/gpg.pub
```

You can then run the following to install the latest Secret-rotator version:

```
# yum install evertrust-secret-rotator
```

To prevent unattended upgrades when running yum update, you should pin the Secret-rotator version by adding

```
exclude=evertrust-secret-rotator
```

at the end of the `/etc/yum.repos.d/evertrust-secret-rotator.repo` file after installing Secret-rotator.

**Installing from RPM**

Download the latest RPM for Secret-rotator on the Official EVERTRUST repository.

Upload the file `evertrust-secret-rotator-<latest>.noarch.rpm` to the server;

Access the server with an account with administrative privileges;

Install the Secret-rotator package with the following command:

```
# yum localinstall /root/evertrust-secret-rotator-<latest>.noarch.rpm
```

If you wish to verify the signature of the RPM package, the EVERTRUST key can be added to your trusted keys using the following command:

```
# rpm --import https://evertrust.io/.well-known/rpm/gpg.pub
```

The signature can then be verified using the following command:

```
# rpm -K /root/evertrust-secret-rotator-<latest>.noarch.rpm
```

**Debian**



Secret rotator requires java 17 or 21 installed

**Installation from the EverTrust repository**

If you haven't already, to add the EVERTRUST repository to your APT repositories, run the following commands:

1. Install the required tools (`gpg`)

```
# sudo apt install gnupg
```

2. Download and install the EVERTRUST GPG key

```
# curl https://evertrust.io/.well-known/apt/gpg.pub | sudo gpg -o
/usr/share/keyrings/evertrust.gpg --dearmor
```

3. Add the repository

```
# echo "deb [ arch=all,amd64 signed-by=/usr/share/keyrings/evertrust.gpg ]
https://repo.evertrust.io/repository/apt all main" | sudo tee
/etc/apt/sources.list.d/evertrust.list
```

Once the repository has been added, authentication to it must be provided. To do so, edit the `/etc/apt/auth.conf` file and add the following lines:

```
machine repo.evertrust.io
login <your EVERTRUST login>
password <your EVERTRUST password>
```

Once the repository has been added, run the following command to update the APT repository list.

```
# sudo apt update
```

You can then run the following command to install the latest Secret-rotator version:

```
# sudo apt install evertrust-secret-rotator
```

To prevent unattended upgrades when running `apt upgrade`, you should pin the Secret-rotator version by creating a `/etc/apt/preferences.d/evertrust-secret-rotator` file:

```
Package: evertrust-secret-rotator
Pin: version <installed-version>
Pin-Priority: 1001
```

**Installing from DEB**

Download the latest DEB for Secret-rotator on the Official EVERTRUST repository.

Upload the file `evertrust-secret-rotator_<latest>_all.deb` to the server;

Access the server with an account with administrative privileges;

Install the Secret-rotator package with the following command:

```
# apt install /root/evertrust-secret-rotator_<latest>_all.deb
```

**Docker**

The Secret rotator utility is available at:

```
registry.evertrust.io/evertrust-secret-rotator
```

# Chapter 1. Usage

The secret rotator re-encrypts the secrets stored in the Horizon or Stream database with the **current primary key** of the Tink keyset.



The tool never modifies the keyset. The primary key must be rotated beforehand, as described in the Tinkey key rotation section of the Horizon or Stream documentation, and the previous keys must stay `ENABLED` so that the existing secrets can still be decrypted. Once the rotation succeeds, the previous keys can be disabled.

## Listing the supported versions

To list the supported product versions:

**RHEL**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator list
```

**Debian**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator list
```

**Docker**

```
docker run registry.evertrust.io/evertrust-secret-rotator list
```

## Checking the secrets

The examples below target Horizon. For Stream, use `--product stream` and the Stream keyset (`/opt/stream/etc/stream.keyset` by default).

The `check` command verifies that every secret can be decrypted with the keyset, and counts the secrets that still need to be rotated. Nothing is written to the database.

**RHEL**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator check
--product horizon --mongo-uri "<mongo URI>" --keyset-path
/opt/horizon/etc/horizon.keyset
```

**Debian**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator check
```

```
--product horizon --mongo-uri "<mongo URI>" --keyset-path
/opt/horizon/etc/horizon.keyset
```

**Docker**

```
docker run registry.evertrust.io/evertrust-secret-rotator check --product
horizon --mongo-uri "<mongo URI>" --keyset-path /opt/horizon/etc/horizon.keyset
```

# Rotating the secrets



Stop the product and back up the database before running the `rotate` command. Documents modified while the rotation runs are not overwritten and are reported as conflicts: run the rotation again to handle them.

To re-encrypt the configuration secrets with the primary key of the keyset:

**RHEL**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator rotate
--product horizon --mongo-uri "<mongo URI>" --keyset-path
/opt/horizon/etc/horizon.keyset
```

**Debian**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator rotate
--product horizon --mongo-uri "<mongo URI>" --keyset-path
/opt/horizon/etc/horizon.keyset
```

**Docker**

```
docker run registry.evertrust.io/evertrust-secret-rotator rotate --product
horizon --mongo-uri "<mongo URI>" --keyset-path /opt/horizon/etc/horizon.keyset
```

For Horizon only, to re-encrypt the escrowed private keys with the primary key of the escrow vault keyset, use the `escrow` mode:

**RHEL**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator rotate
--mode escrow --product horizon --mongo-uri "<mongo URI>" --keyset-path
/path/to/escrow.keyset
```

**Debian**

```
/opt/evertrust/evertrust-secret-rotator/bin/evertrust-secret-rotator rotate
--mode escrow --product horizon --mongo-uri "<mongo URI>" --keyset-path
/path/to/escrow.keyset
```

**Docker**

```
docker run registry.evertrust.io/evertrust-secret-rotator rotate --mode escrow
--product horizon --mongo-uri "<mongo URI>" --keyset-path
/path/to/escrow.keyset
```

The rotation is idempotent: the secrets already encrypted with the primary key are skipped. If the primary key uses a `RAW` output prefix, every secret is re-encrypted.

The command exits with a non-zero code if any secret could not be decrypted with the keyset. These secrets are reported and left untouched.

# Parameters

Parameters are only read from the command line. Environment variables and `/etc/default` files are never read.

| Parameter | Description |
|---|---|
| `--product` | **Required**. The product owning the database: `horizon` or `stream` |
| `--mongo-uri` | **Required**. The database URI, including the database name |
| `--keyset-path` | **Required**. The path of the Tink keyset, whose primary key has already been rotated |
| `--master-key-uri` | The master key URI, only for a wrapped keyset (PKCS#11, AWS KMS or GCP KMS) |
| `--credentials-path` | The KMS credentials, requires `--master-key-uri`. The KMS default credentials are used if not set |
| `--mode` | `config` (default) to rotate the configuration vault secrets, `escrow` to rotate the escrow vault private keys (Horizon only) |
| `--product-version` | Overrides the product version detected from the `misc` collection |

# Chapter 2. Release notes

## v1.0

Initial release. The following product versions are supported:

- Horizon 2.10.0 to 2.10.10
- Horizon 2.11.0 and 2.11.1
- Stream 2.1.7 to 2.1.12



Stream 2.1.0 to 2.1.6 are not supported. Upgrade Stream to 2.1.7 or later before rotating its secrets.