initMAX

ZABBIX PREMIUM PARTNER

ZABBIX CERTIFIED TRAINER

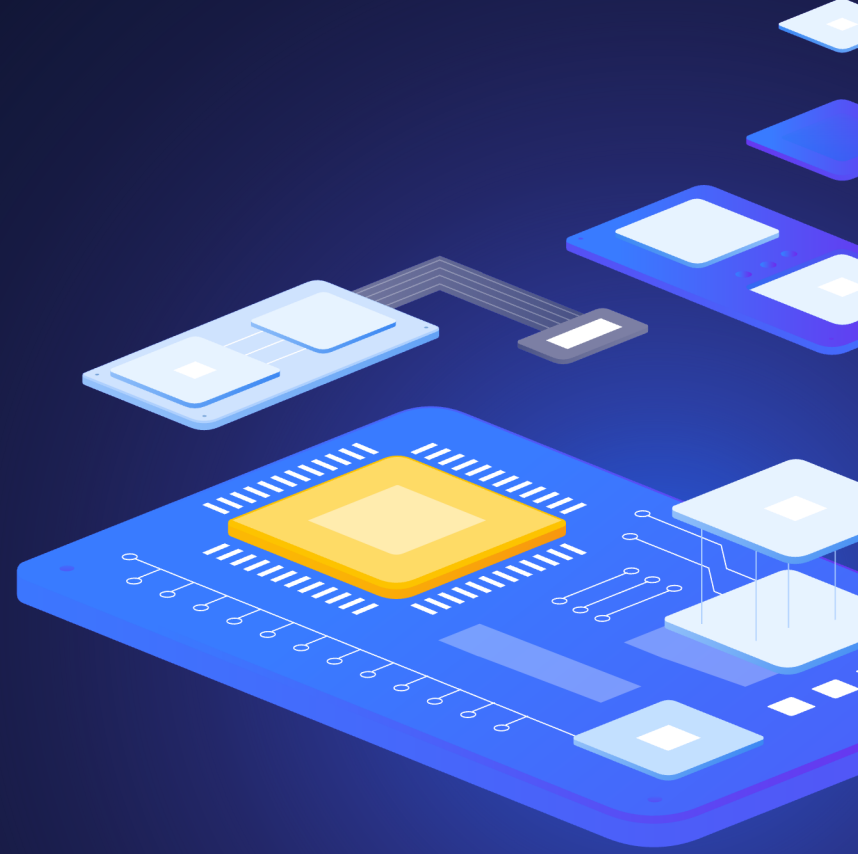
Webinar

Zabbix: Security Best practices + External Vault

all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause



1

Why Security Matters in Infrastructure Monitoring

initMAX

Risks

Historical data sensitivity

- Performance metrics, Log data, Historical trends

Infrastructure documentation

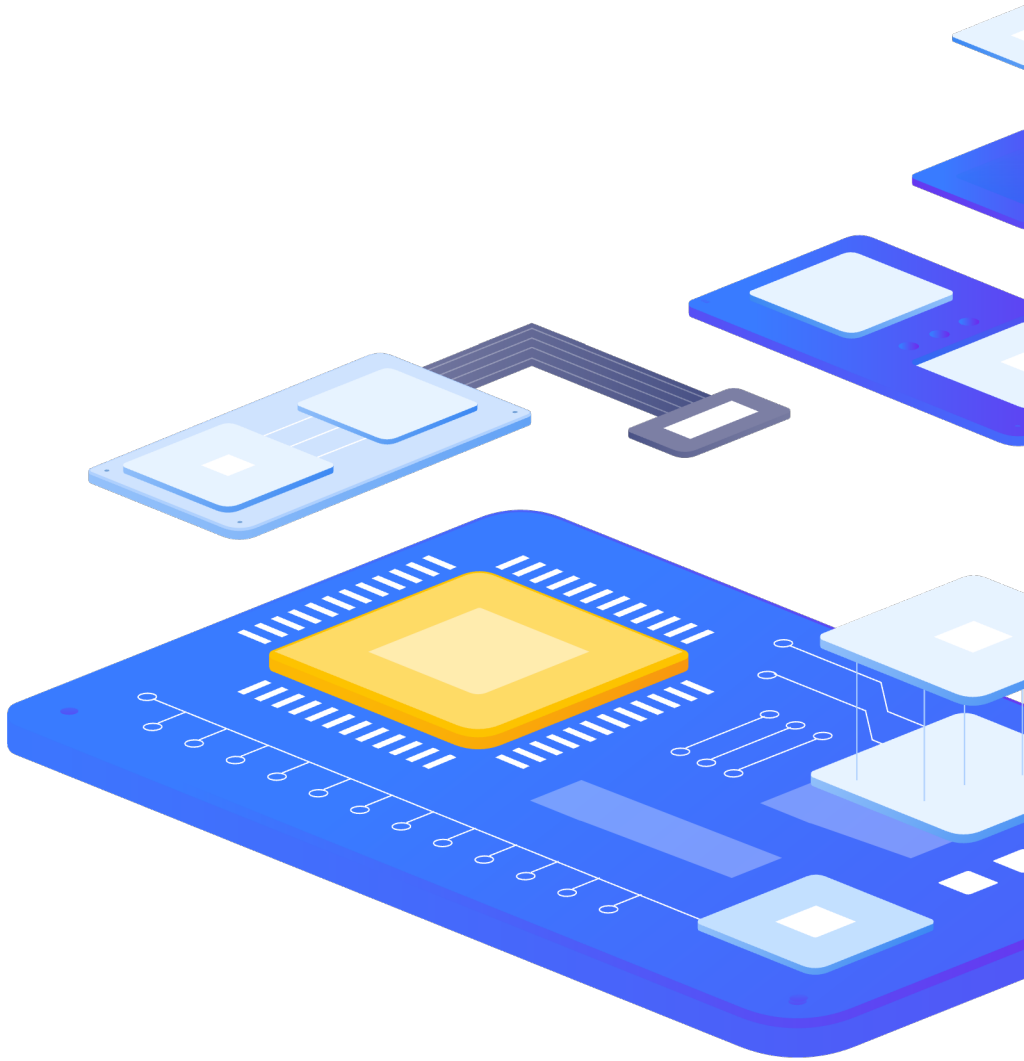
- System inventory, Network topology

Privileged access requirements

- Credentials, Tokens, System accounts

Infrastructure access

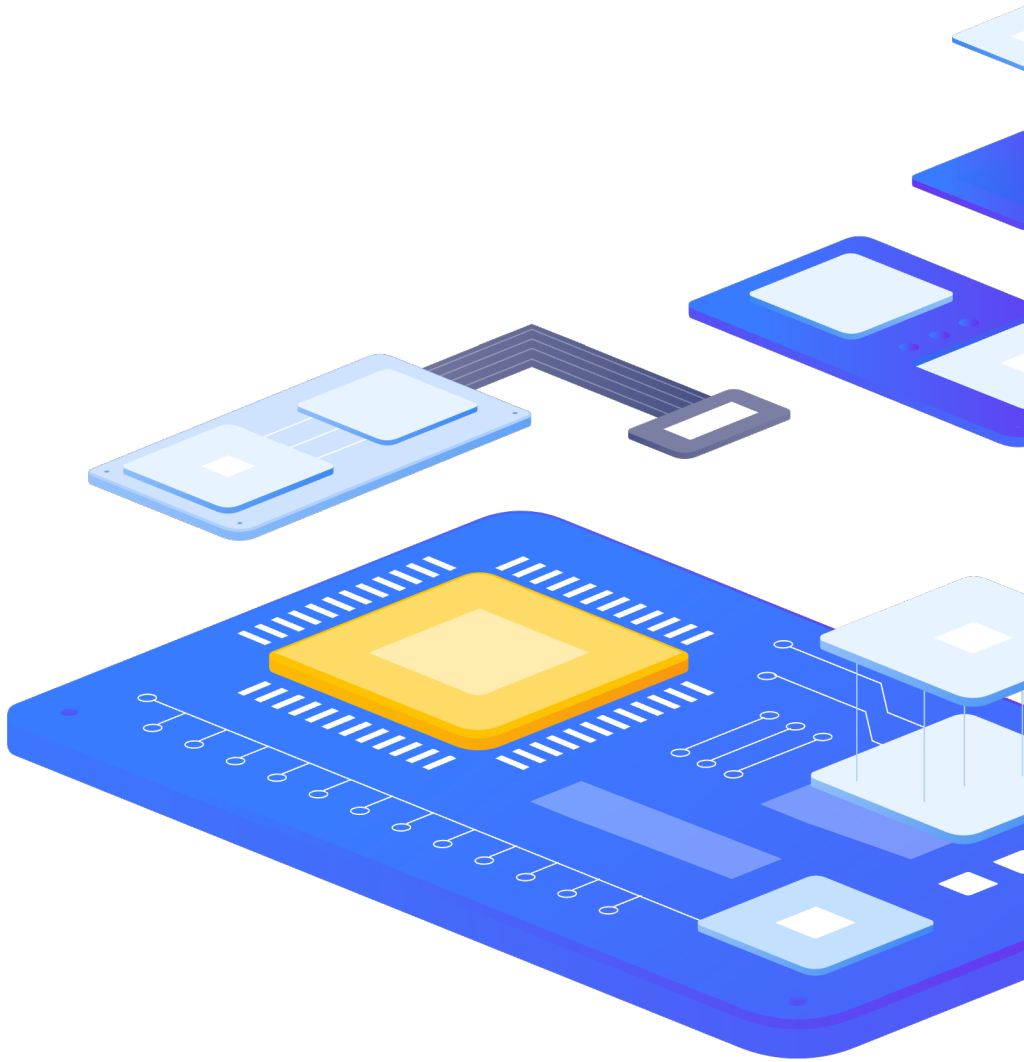
- Network access, Execute remote commands

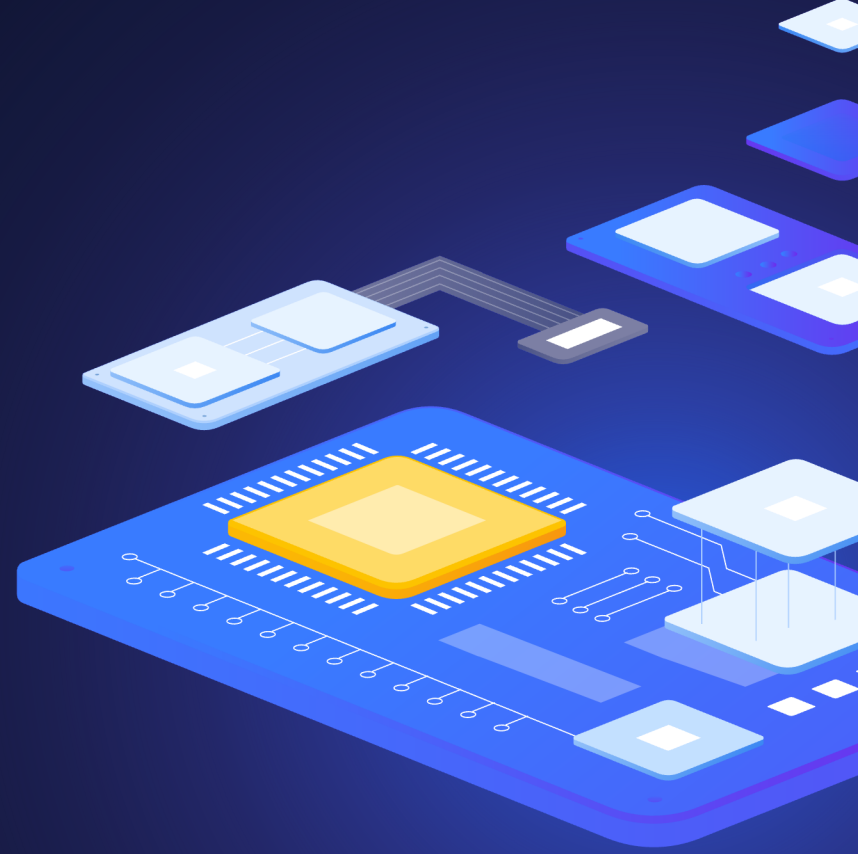


initMAX

Key security principle

- Follow Organization's Security Standards
- Implement Layered Security
- Apply Principle of Least Privilege
- Keep Systems Up-to-date
- Secure Credential Management





2

Frontend Access Security and User Management

initMAX

User access management

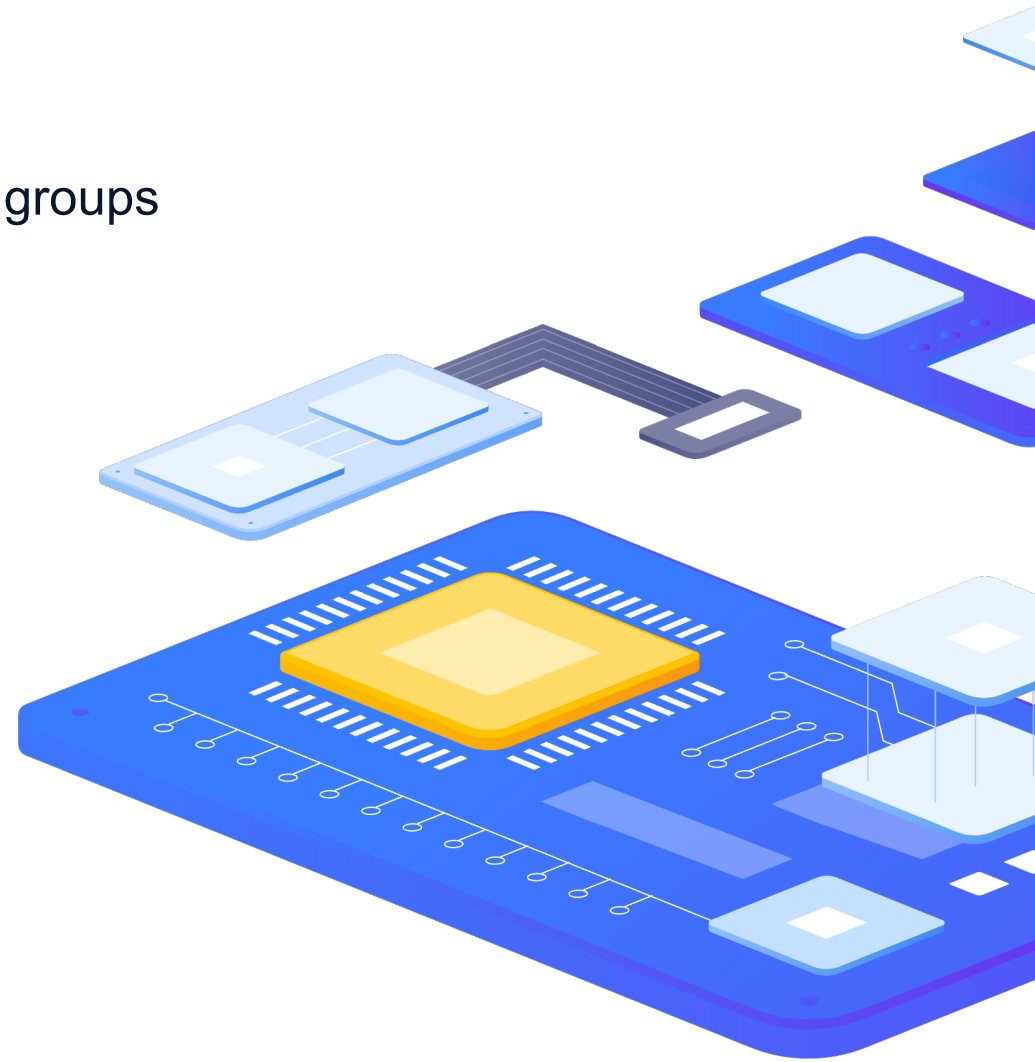
In Zabbix all access rights are based on user groups and host groups

User groups

- Define read/write permissions to hosts and templates
- Users can be members of multiple user groups

User roles

- Super Admin role
- Admin role
- User role



initMAX

User authentication

HTTP

- Web server-based authentication method using standard HTTP protocol mechanisms (Basic, NTLM, Kerberos)

LDAP

- Enterprise directory service protocol for centralized user and authentication management

SAML

- Standard enabling Single Sign-On (SSO) and secure exchange of authentication data between services

SCIM

- Protocol for automating user provisioning and management across multiple systems

Wiki: https://www.initmax.com/wiki/zabbix-automation-of-user-management-jit/

initMAX

Multi-Factor authentication

Purpose

- Enhances login security beyond traditional credentials
- Prevents unauthorized access even if passwords are compromised

Multiple MFA methods are available:

- Time-Based One-Time Password (TOTP)
  - Google authenticator All hashes and code lengths
  - Microsoft authenticator SHA-1 with code length 6 only
- Duo Universal Prompt

Wiki: https://www.initmax.com/wiki/two-factor-authentication-2fa-in-zabbix-7-0/

initMAX

HTTPS

Hypertext Transfer Protocol Secure

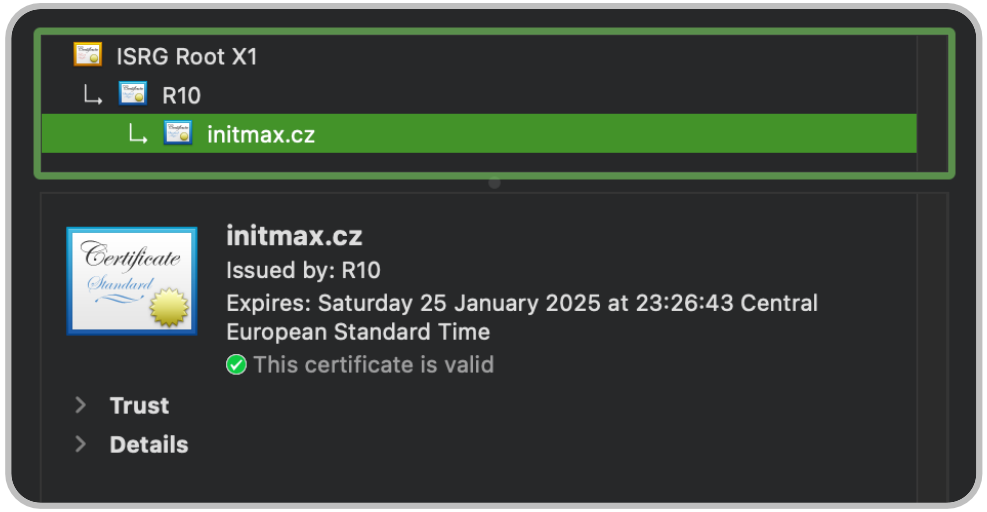
- TLS (Transport Layer Security) - current secure protocol for data encryption
  - Recommended versions 1.2 and 1.3
- SSL (legacy) has been replaced by TLS

Key Benefits

- Encrypted data transfer between client and server
- Protection against man-in-the-middle attacks

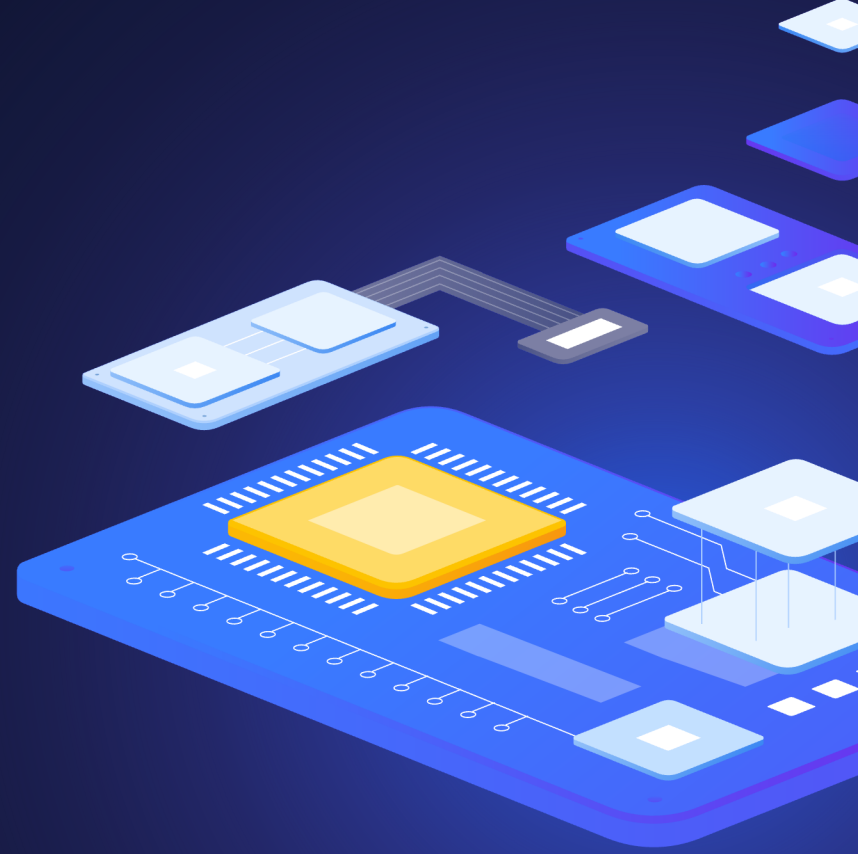
Certificate Validation

- Digital certificates confirm server identity
- Browser validate certificate authenticity
- Visual security indicators in browser interface



3

PSK and Certificates



initMAX

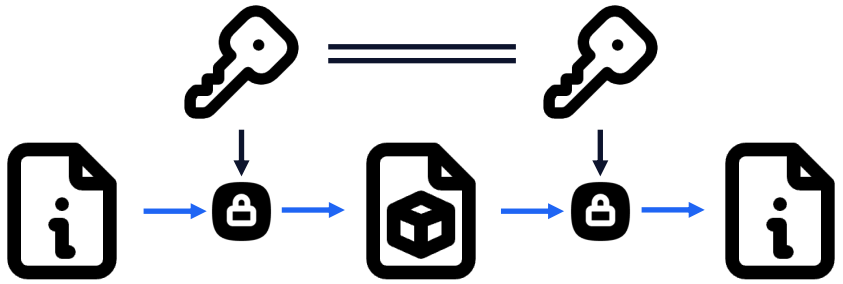
PSK (Pre-Shared Key)

One of available encryption methods

- Uses symmetric encryption
- Both sides know same encryption key
- Simple to implement and manage

PSK in Zabbix consists of two parts:

- PSK Identity
  - Non-secret identifier string
  - Case-sensitive name
- PSK Value
  - Secret string used for encryption
  - Must be unique for each PSK identity



initMAX

Certificates

SSL digital certificates has two main purposes

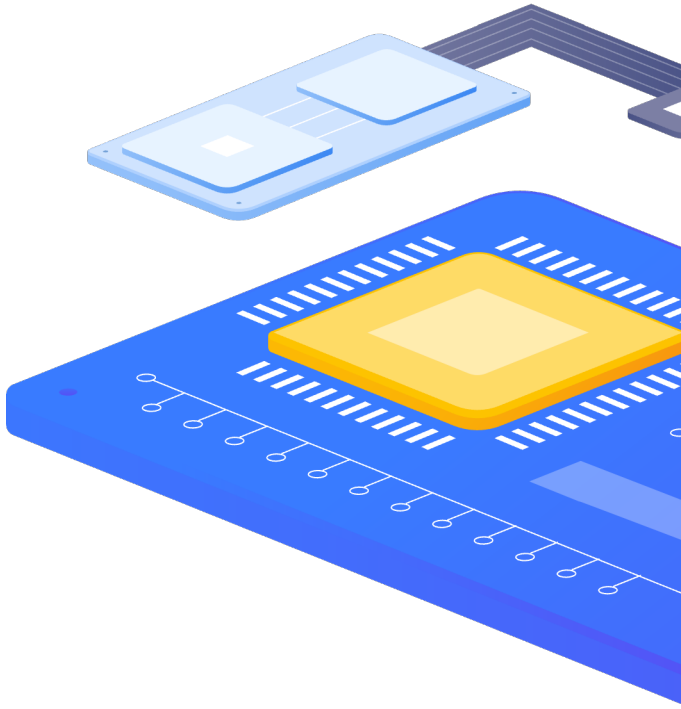
- Authenticates applications identity
- Enables encrypted communication

SSL certificate subject and issuer information

- The issuer represents the trusted authority that issued the certificate
- The subject represents the identified computer system using the certificate
- If issuer and subject match, we are talking about self-signed certificates

SSL certificate private key properties

- Private key and certificate are generated together
- Private key handles decryption operations
- Private key requires strict protection and restricted sharing

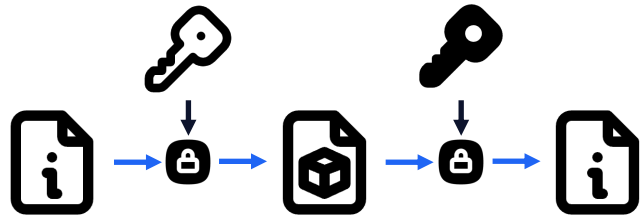


initMAX

Certificates

Authenticates Application Identity

- Certificate is trusted when we know and trust the issuer (CA and intermediate CA)
- Validation chain verifies certificate authenticity
- Makes sure we are connecting to the right system

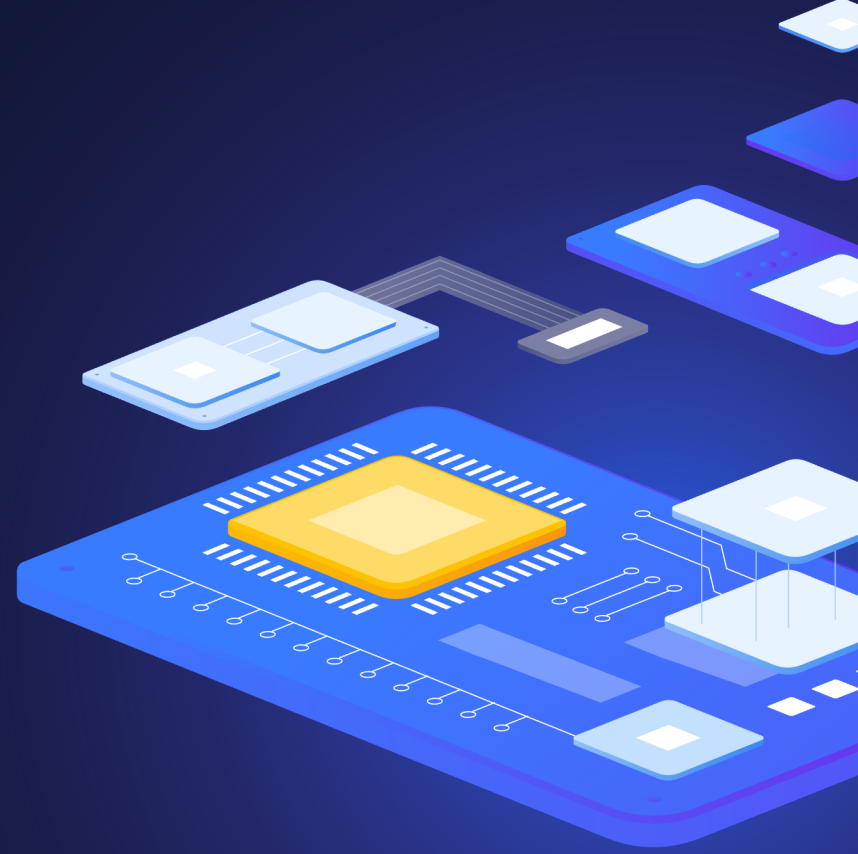


Enables Encrypted Communication

- Uses symmetric encryption for data transfer
- Public key encrypts, private key decrypts
- The server has its own key pair (public/private key), and the client uses the server's public key for encryption.

4

Database connections



initMAX

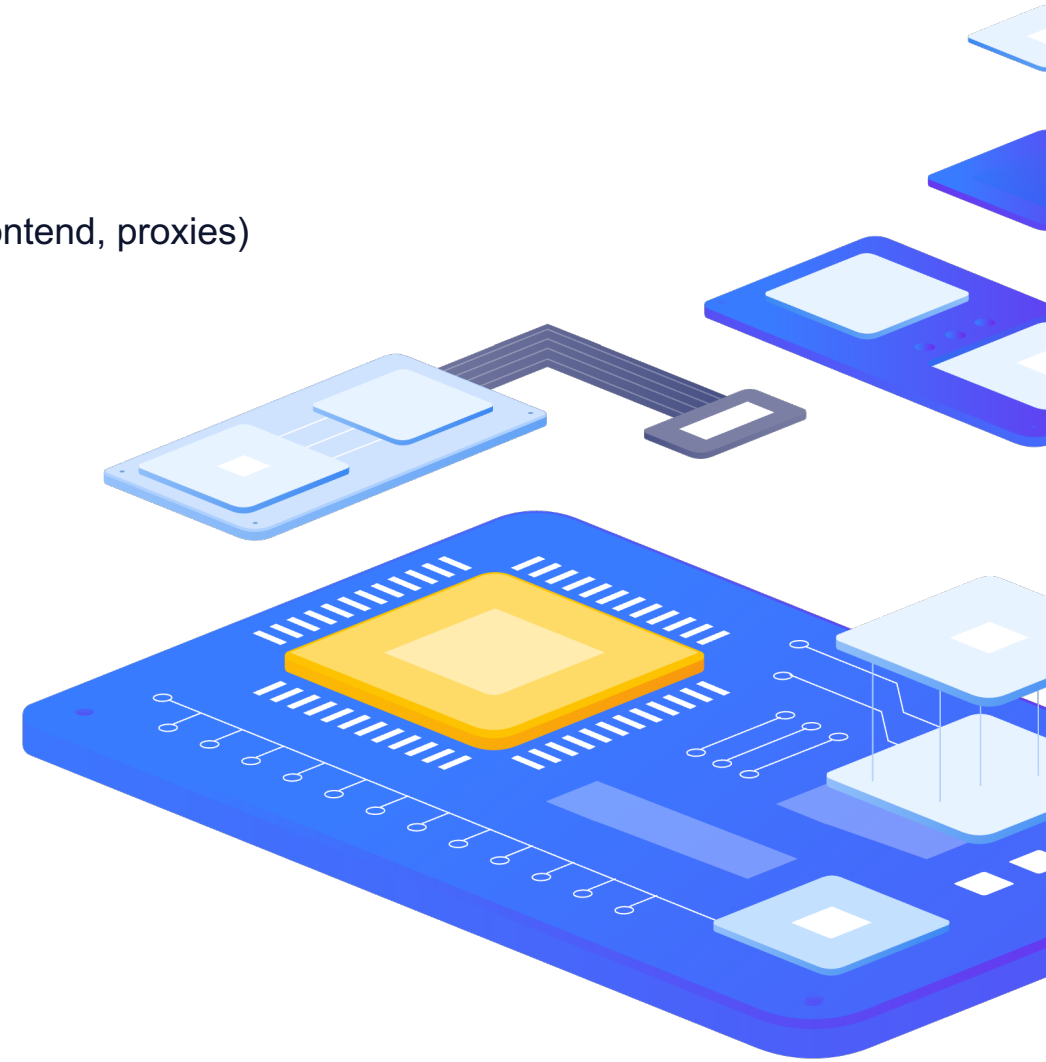
Database connections

Encryption of database communication

- All Zabbix components support encrypted database connections (server, frontend, proxies)
- Uses TLS encryption with certificates issued by database engine
- Protects sensitive data during database communication

Supported Database Encryption

- Native encryption support
  - MySQL, MariaDB
    - Has built-in self-signed SSL certificate
  - PostgreSQL
- Encryption not available for:
  - SQLite3 database engine
  - Socket connections (localhost)



initMAX

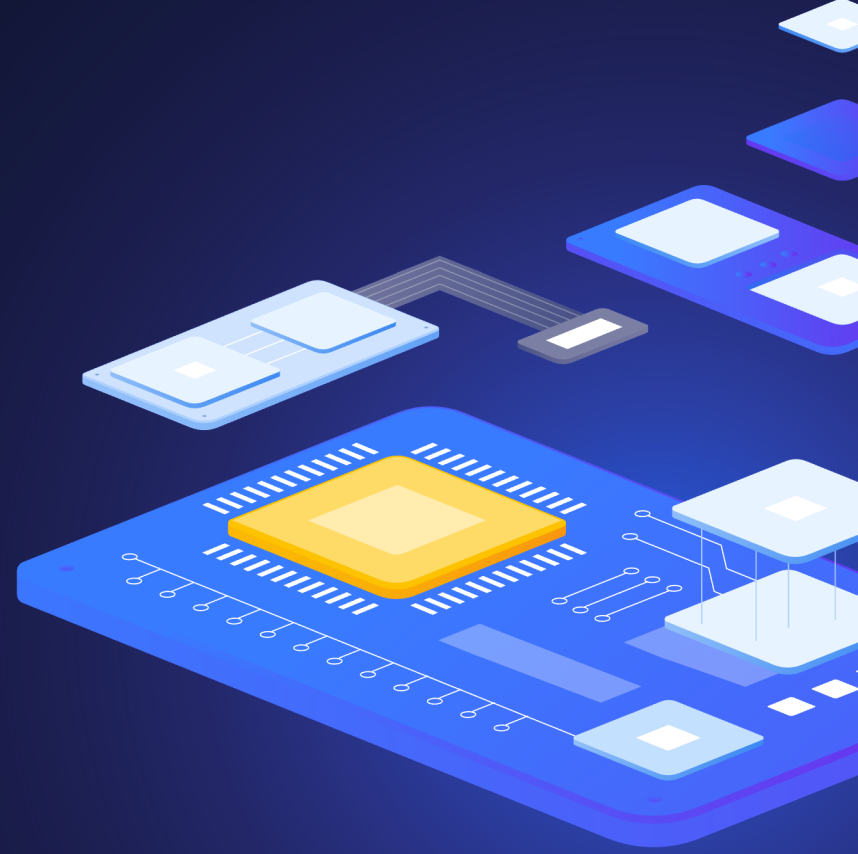
Database connections

Database Connection Encryption Modes

- Server configuration options
  - **required** - use TLS as transport mode without identity checks
  - **verify_ca** - use TLS and verify database certificate
  - **verify_full** - additionally verify that DBHost matches database certificate CN field
- Frontend configuration options with same purpose
  - **ENCRIPTION**
  - **VERIFY_HOST**

5

Internal communication



initMAX

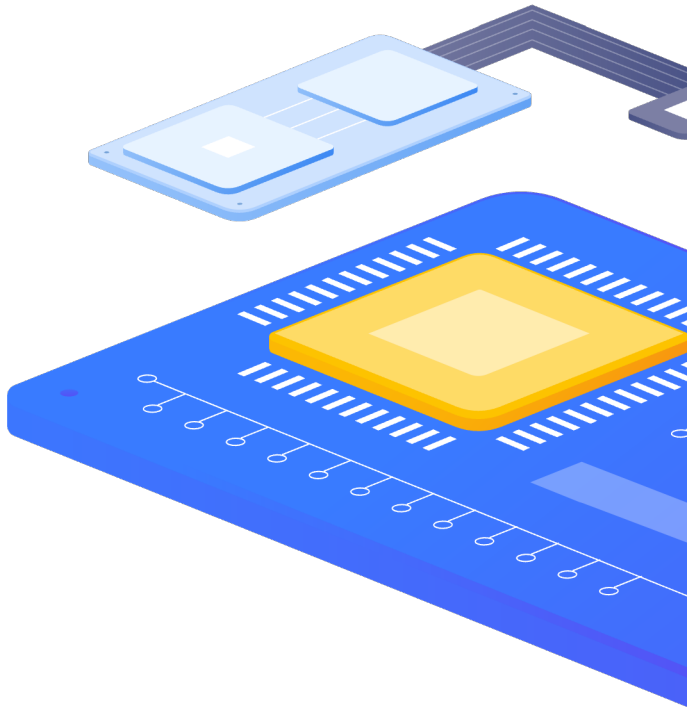
Internal communication

Connection can be encrypted:

- Agent to Server/Proxy (supports both PSK and certificates)
- Proxy to Server (supports both PSK and certificates)
- Web Service to Server
- Java Gateway to monitored Java applications

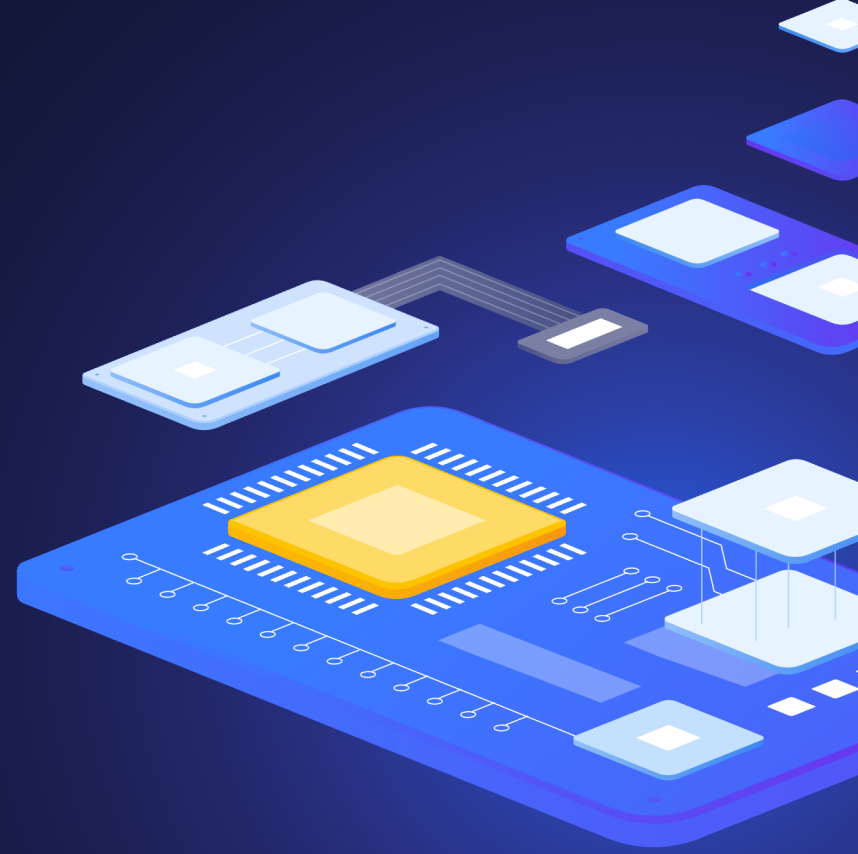
Connection cannot be encrypted:

- Frontend to Server (Test button, queue info, system data)
- Server/Proxy to Java Gateway



6

SNMPv3



initMAX

SNMPv3

SNMPv3 Security Levels

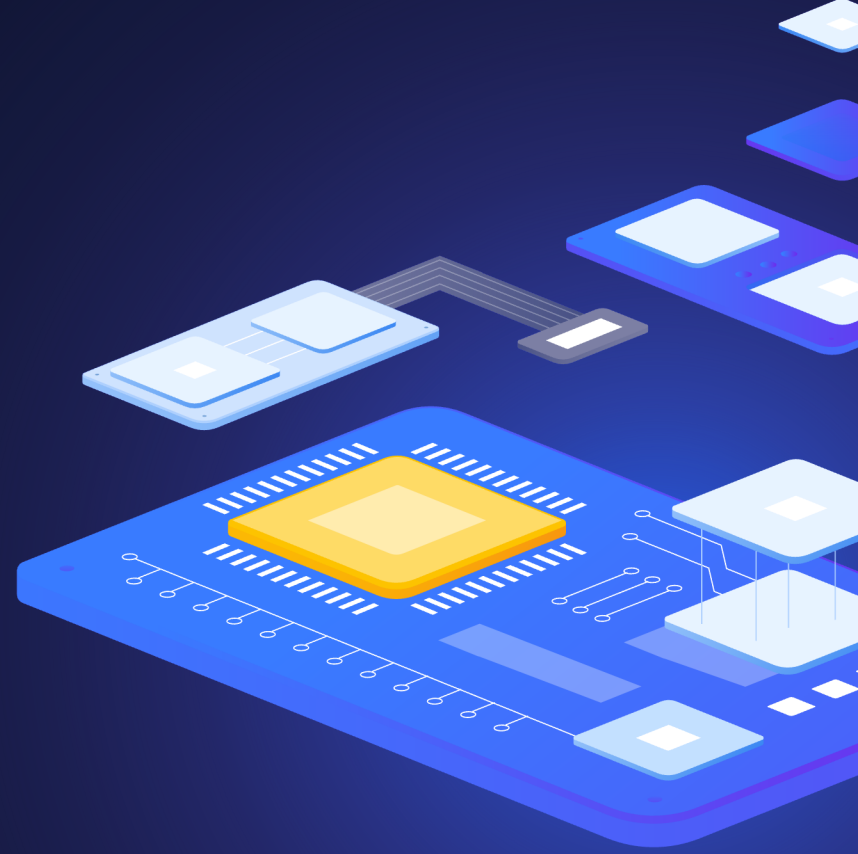
- noAuthNoPriv - Plain text authentication, no data encryption
- authNoPriv - Secure authentication (SHA), no data encryption
- authPriv - Secure authentication (SHA) with data encryption (DES/AES)

We invite you to join our webinar dedicated to SNMP

- A video recording in Czech is available for you
  https://www.initmax.cz/webinar/jak-pouzivat-snmp-a-snmp-traps-v-zabbixu-7-0/
- You can also download the presentation in English
  https://www.initmax.cz/wp-content/uploads/2024/08/snmp-a-snmp-traps-in-zabbixu-7.0.pdf

7

External Vault



initMAX

External Vault

Available Vault Solutions

- HashiCorp Vault - industry standard for secrets management, widely adopted
- OpenBao - open-source alternative based on HashiCorp Vault 1.14
- CyberArk - enterprise privileged access management solution

Usage in Zabbix

- Store database credentials (server, proxy, frontend)
- Secure macro values as “Vault secret” value
  - “Secret text” values are still stored in the database as **plaintext**
- Store API tokens
- Centralized secrets management
- Zabbix needs only read access to vault secrets

initMAX

HashiCorp Vault - overview

Licensing

- Free for non-commercial use (development/testing).
- Commercial use requires a license.

Policy Engine & Audit Logging

- Granular access control and detailed auditing for compliance.
- Captures system operations for compliance (e.g., PCI DSS, HIPAA).

Encryption & Key Management

- Advanced encryption capabilities to protect data at rest and in transit.
- Supports dynamic secrets with lease and revocation for added security.

initMAX

HashiCorp Vault - overview

PKI Infrastructure for Certificate Management

- Automated certificate generation and distribution.

Deployment Flexibility

- Single-node or high availability (HA) clusters.
- Install via native, containers, Kubernetes, or cloud.

High Availability & Disaster Recovery

- HA mode ensures continuous availability.
- Disaster recovery features for cross-region replication.

initMAX

HashiCorp Vault - overview

Integration Capabilities

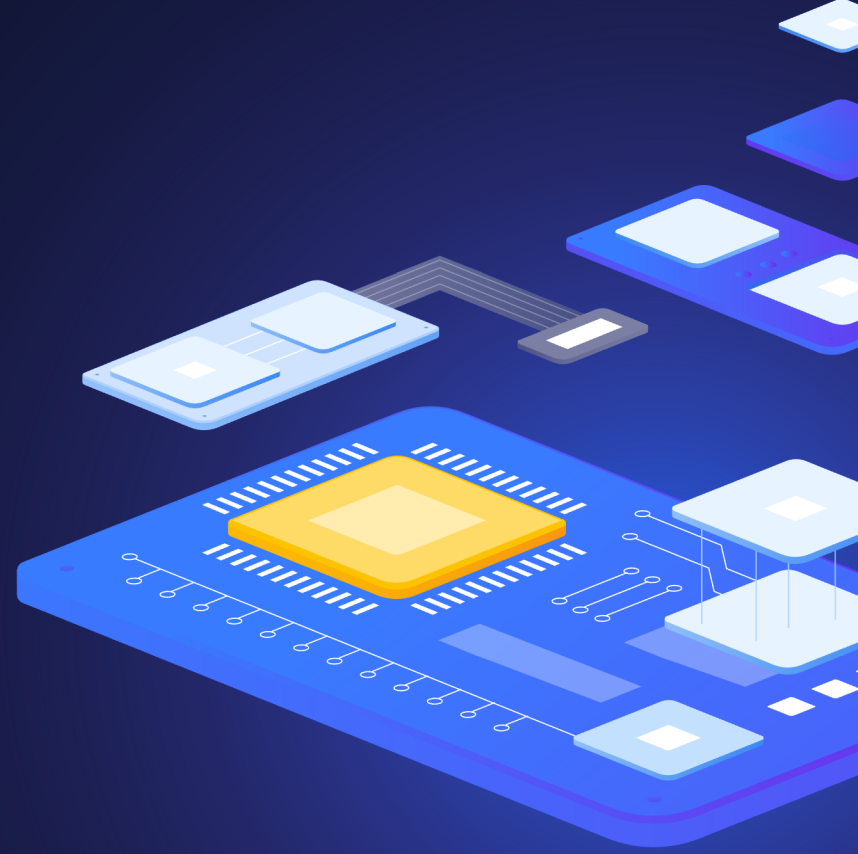
- Authentication: Username/Password, Tokens, LDAP/AD, SAML/OIDC.
- Cloud & third-party service integrations (AWS, Azure, Okta).
- Extensive API support for automation and integration.

Storage Backends

- Options: File, Raft, PostgreSQL, cloud storage.

8

Demo



initMAX

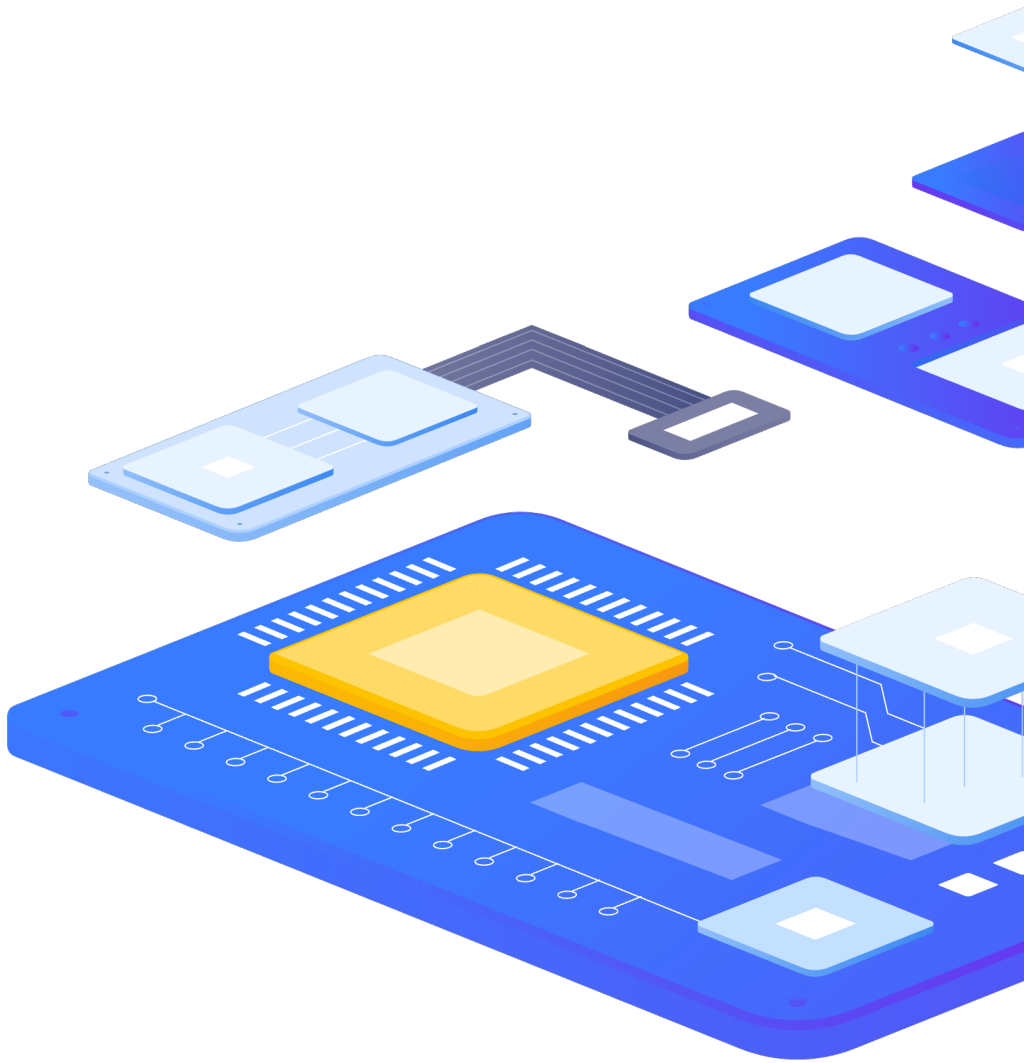
Demo: Lab Environment Setup

zabbix.lab.initmax.cz (10.210.10.80)

- RockyLinux 9.4
- Zabbix 7.0.5 (server, frontend, agent2)
- PostgreSQL 17
- Certificates from „lab_initmax_ca“

vault.lab.initmax.cz (10.210.10.81)

- RockyLinux 9.4
- Certificates from „lab_initmax_ca“



initMAX

Demo: Vault installation

Install vault package

```
sudo yum install -y yum-utils

sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo

sudo yum -y install vault
```

Certificates

```
ls -l /etc/pki/tls/certs/vault.lab.initmax.crt \
      /etc/pki/tls/private/vault.lab.initmax.key \
      /etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt

sudo update-ca-trust extract
```

initMAX

Demo: Vault installation

Setup privileges for certificates

```
sudo groupadd app_certs

sudo usermod -aG app_certs vault

sudo chown root:app_certs /etc/pki/tls/certs/vault.lab.initmax.crt
sudo chown root:app_certs /etc/pki/tls/private/vault.lab.initmax.key
sudo chown root:app_certs /etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt

sudo chmod 644 /etc/pki/tls/certs/vault.lab.initmax.crt
sudo chmod 640 /etc/pki/tls/private/vault.lab.initmax.key
sudo chmod 644 /etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt
```

initMAX

Demo: Vault installation

Setup privileges for certificates

```
groups vault

ls -l /etc/pki/tls/certs/vault.lab.initmax.crt \
      /etc/pki/tls/private/vault.lab.initmax.key \
      /etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt

sudo -u vault cat /etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt
```

initMAX

Demo: Vault installation

Create PostgreSQL database for vault on zabbix.lab.initmax.cz

```
sudo -u postgres psql -c "CREATE USER vault WITH PASSWORD 'SuperSecretPass123';"
sudo -u postgres psql -c "CREATE DATABASE hcvault OWNER vault;"

sudo -u postgres psql -d hcvault

  SET ROLE vault;

  CREATE TABLE vault_kv_store (
    parent_path TEXT COLLATE "C" NOT NULL,
    path TEXT COLLATE "C",
    key TEXT COLLATE "C",
    value BYTEA, CONSTRAINT pkey PRIMARY KEY (path, key)
  );

  CREATE INDEX parent_path_idx ON vault_kv_store (parent_path);
```

initMAX

Demo: Vault installation

Set vault configuration

```
sudo nano /etc/vault.d/vault.hcl

  storage "postgresql" {
    connection_url = "postgres://vault:SuperSecretPass123@10.210.10.80:5432/hcvault?sslmode=verify-full"
    max_parallel = "100"
    table = "vault_kv_store"
    max_idle_connections = 100
  }

  listener "tcp" {
    address = "10.210.10.81:8200"
    tls_cert_file = "/etc/pki/tls/certs/vault.lab.initmax.crt"
    tls_key_file = "/etc/pki/tls/private/vault.lab.initmax.key"
    tls_client_ca_file = "/etc/pki/ca-trust/source/anchors/lab_initmax_ca.crt "
  }

  api_addr = "https://10.210.10.81:8200"

  log_level = "warn"
```

initMAX

Demo: Vault installation

Start and enable vault

```
sudo systemctl enable vault

sudo systemctl start vault
```

```
sudo systemctl status vault

journalctl -u vault
```

initMAX

Demo: Vault installation

Open firewall port

```
sudo firewall-cmd --list-all

sudo firewall-cmd --zone=public --permanent --add-port=8200/tcp

sudo firewall-cmd --reload

sudo firewall-cmd --list-all
```

```
export VAULT_ADDR='https://10.210.10.81:8200'

vault status
```

initMAX

Demo: Vault initialization

Open in web browser

```
https://10.210.10.81:8200/ui
```

- Initialize your vault with 5 keys
- Required for unseal are 3 of them
- Save root token with generated keys
- Unseal vault

initMAX

Demo: Vault initialization

Add your secrets (Initialize new KV(v2) secrets engine)

- zbx_db/server

```
{"username": "zabbix_server","password": "pass_zabbix_server"}
```

- zbx_db/web

```
{"username": "zabbix_web","password": "pass_zabbix_web"}
```

- secret_macros/zabbix_db

```
{"username": "monitoring","password": "monitoring"}
```

initMAX

Demo: Vault initialization

Set ACL Policies

- zabbix-server-policy

```
path "kvzabbix/*" {
  capabilities = [ "read", "list" ]
}
```

- zabbix-web-policy

```
path "kvzabbix/data/zbx_db/web" {
  capabilities = [ "read", "list" ]
}
```

initMAX

Demo: Vault initialization

Create and save new tokens for Zabbix server and Frontend

```
vault login
#with root token

#Server
vault token create \
  -policy="zabbix-server-policy" \
  -orphan=true \
  -period=768h

#Frontend
vault token create \
  -policy="zabbix-web-policy" \
  -orphan=true \
  -period=768h
```

initMAX

Demo: Zabbix server setup

```
sudo nano /etc/zabbix/zabbix_server.conf
```

```
#DBUser=
#DBPassword=

Vault=HashiCorp
VaultToken=
VaultURL=https://10.210.10.81:8200
VaultPrefix=/v1/kvzabbix/data/
VaultDBPath=zbx_db/server
```

```
sudo systemctl set-environment VAULT_TOKEN=<<TOKEN>>

sudo systemctl restart zabbix-server && tail -fn 100 /var/log/zabbix/zabbix_server.log
```

initMAX

Demo: Zabbix frontend setup

```
sudo nano /etc/zabbix/web/zabbix.conf.php
```

```
$DB['USER’]             = '';
$DB['PASSWORD’]         = '';

$DB['VAULT’]            = 'HashiCorp';
$DB['VAULT_URL’]        = 'https://10.210.10.81:8200';
$DB['VAULT_PREFIX’]     = '/v1/kvzabbix/data/';
$DB['VAULT_DB_PATH’]    = 'zbx_db/web';
$DB['VAULT_TOKEN’]      = '<<TOKEN>>';
$DB['VAULT_CERT_FILE’]  = '';
$DB['VAULT_KEY_FILE’]   = '';
```

```
ls -l /etc/zabbix/web/zabbix.conf.php

sudo systemctl restart nginx php-fpm
```

initMAX

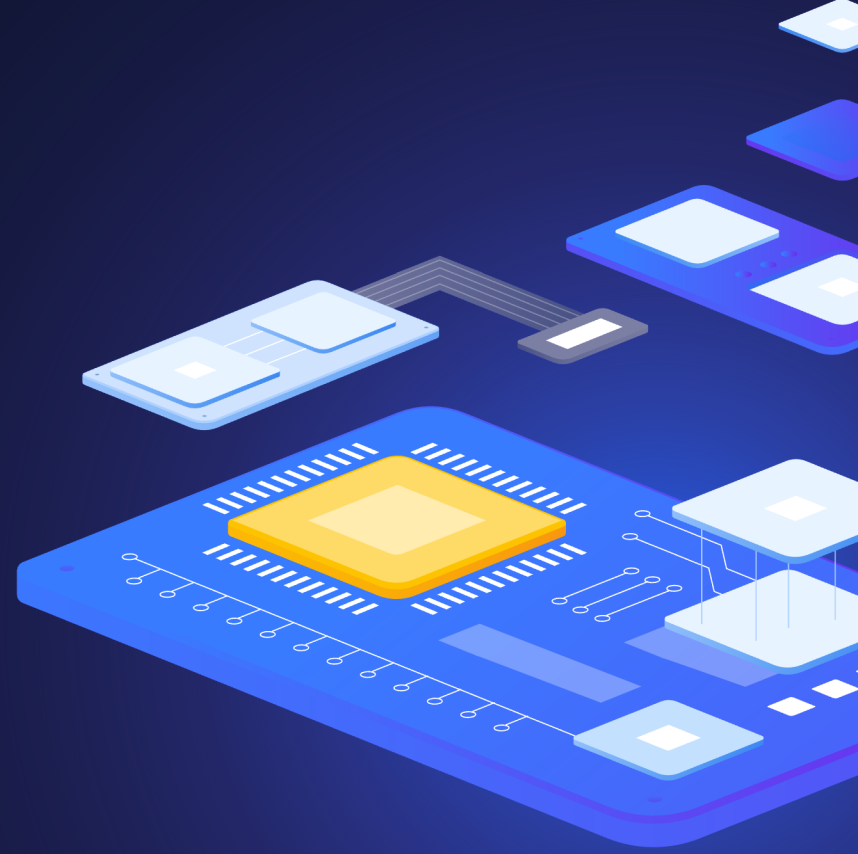
Demo: Use Vault secret macro

- Link template “PostgreSQL by Zabbix” agent 2 with host “zabbix.lab.initmax.cz”
- Set inherited macros {$PG.USER} and {$PG.PASSWORD}
  - Change value type on “**Vault secret**”

- Reload secrets in Zabbix server configuration cache

```
sudo zabbix_server -R secrets_reload
```

Questions?



initMAX

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.cz
Email:	tomas.hermanek@initmax.cz
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184