ZABBIX 5.0 SECURITY FEATURES AND IMPROVEMENTS

Simon Wetzstein, Convx GmbH
Zabbix Certified Expert
TLS SUPPORT FOR FRONTEND COMMUNICATION WITH DATABASE

ENCRIPT FRONTEND COMMUNICATION WITH THE ZABBIX DATABASE

- Uses already familiar TLS certificate encryption
CONFIGURING THE ENCRYPTION
CONFIGURING THE ENCRYPTION

- TLS certificate based encryption
- Supported for MySQL and PostgreSQL
- With host verification - the database server certificate is checked by comparing the host name specified in the certificate with the name of the host to which it is connected
- If the TLS parameters point to files that are open for writing, a warning will be displayed stating that the files should be read only
TLS SUPPORT FOR SERVER/PROXY COMMUNICATION WITH DATABASE

- Encrypt Zabbix Server communication with the Zabbix database
  - Also uses the already familiar TLS certificate encryption
### Option: DBTLSConnect
DBTLSConnect=

### Option: DBTLSCAFile
DBTLSCAFile=

### Option: DBTLSCertFile
DBTLSCertFile=

### Option: DBTLSKeyFile
DBTLSKeyFile=

### Option: DBTLSCipher
DBTLSCipher=

### Option: DBTLSCipher13
DBTLSCipher13=
EXAMPLE SCENARIO – **JUST ENCRYPT**

- Make encrypted connection to the DB without authenticating or verifying the host identity

```plaintext
### Option: DBTLSConnect
DBTLSConnect=required
```
EXAMPLE SCENARIO – ENCRYPT AND VERIFY

☑ Make encrypted connection to the DB with authenticating and verifying the host identity

### Option: DBTLSConnect
DBTLSConnect= verify_full

### Option: DBTLSCAFile
DBTLSCAFile=/tmp/certs/cafile.crt
CONFIGURABLE CIPHERS

TLS CIPHERSUITS SELECTION
- Ability to configure ciphersuites according to your security policy
- Configurable per component
FUNCTIONALITY AND BENEFITS

- Ability to override the built-in cipher suite selection
- Can override for certificates, PSK and combined
- Ability to override `zabbix_get` and `zabbix_sender` by passing `--tls_cipher13` or `--tls_cipher`
- Allows to select specific ciphers based on your security policy or additional requirements
- Separate configuration parameters for TLS 1.3 and 1.2
- Configurable for incoming/outgoing connections per component
NEW CONFIGURATION PARAMETERS

### Option: TLSCipherCert13
# TLSCipherCert13=

### Option: TLSCipherCert
# TLSCipherCert=

### Option: TLSCipherPSK13
# TLSCipherPSK13=

### Option: TLSCipherPSK
# TLSCipherPSK=

### Option: TLSCipherAll113
# TLSCipherAll113=

### Option: TLSCipherAll
# TLSCipherAll=
04

MASKED MACROS

ABILITY TO MASK YOUR MACROS IN THE FRONTEND

- A simple mask/unmask dropdown
MASK YOUR SENSITIVE MACRO VALUES!

<table>
<thead>
<tr>
<th>Macro</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{$CONNECTION_STRING}</td>
<td>------</td>
<td>ODBC connection string</td>
</tr>
<tr>
<td>{$PASSWORD}</td>
<td>------</td>
<td>User password</td>
</tr>
<tr>
<td>{$SNMP_COMMUNITY}</td>
<td>------</td>
<td>SNMP Community</td>
</tr>
</tbody>
</table>

Add options:
- Update
- Clone
- Full clone
- Delete
- Cancel
When attempting to clone a host/template with secret text macros, the values get reset:

When exporting the host, the value of a secret macro is not exported.

```
<macro>
    <macro>{$PASSWORD}</macro>
    <type>SECRET_TEXT</type>
    <description>Password</description>
</macro>
```

Once defined, you cannot edit it – you have to completely replace the Macro.
FRONTEND PASSWORD HASHING IMPROVEMENTS

REPLACED MD5 ALGORITHM

- Passwords hashes by using bcrypt
- Much more secure approach
BENEFITS OF SWITCHING TO **BCRYPT**

- Based on the Blowfish algorithm
- A lot slower than MD5
- Not feasible for hardware acceleration – less vulnerable to brute-force attacks
- Old MD5 hashes replaced with bcrypt hashes after initial login
- Uses Unique salt value
- Not feasible for Rainbow table attacks
OUT OF THE BOX SAML SUPPORT

INTEGRATE ZABBIX WITH SAML SINGLE SIGN-ON NATIVELY
CONFIGURING INTEGRATION WITH SAML
CONFIGURING INTEGRATION WITH SAML

- A corresponding user must exist in Zabbix, however, its Zabbix password will not be used.
- Need to preconfigure the identity provider
- Default location for private key and certificate is ui/conf/certs/
- Some settings - SP key, SP cert, IDP cert and additional settings can be configured in zabbix.conf.php file
BLACKLISTING AND WHITELISTING OF ITEM KEYS

RESTRICT EXECUTION OF ITEM KEYS
- Key whitelist/blacklist per agent
- Specify individual keys or use wildcards
CONFIGURING KEY RESTRICTIONS

- EnableRemoteCommands still required!
- Rule check stops after first match
- AllowKey can be used only if DenyKey is specified
- If a specific item key is disallowed in the agent configuration, the item will turn unsupported
- Zabbix agent with –print (-p) command line option will not show keys that are not allowed by configuration;
- Zabbix agent with –test (-t) command line option will return "Unsupported item key." status for keys that are not allowed by configuration.
### Option: DenyKey
DenyKey=system.run[*]

### Option: AllowKey
AllowKey=system.run[ls -la /tmp]

THE CONFIGURATION ORDER MATTERS!
### Option: AllowKey

AllowKey=system.run[ls -la /tmp]

### Option: DenyKey

DenyKey=system.run[*]
ODBC CONNECTION STRING SUPPORT

ANOTHER WAY TO SPECIFY A CONNECTION STRING

- dsn parameter now optional
- New connection string parameter
DSN OR CONNECTION STRING?

☑️ In some cases users may not have access to the odc.ini file
☑️ Connection string works around that by defining the connection parameters on the item level
☑️ Either dsn OR connection string should be present. If both are present – the dsn will be ignored
☑️ The connection string may contain drivers specific arguments
Old key:
db.odbc.select[<unique short description>,dsn]

New key:
db.odbc.select[<unique short description>,<dsn>,<connection string>]

Example connection string:
“Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=3306”
LEGACY ENCRYPTION LIBRARY SUPPORT DROPPED

MBED TLS SUPPORT DISCONTINUED

Currently supported mbed TLS versions have reached end of life
WHY?

- Previous versions supported mbed TLS 1.3.9 and later 1.3.x versions – by now these versions have reached end of life.
- Lack of interest/use cases from the community
- Saves the development overhead
- Better focus on support of OpenSSL and GnuTLS
USE API TO RETRIEVE AUDIT LOG

- Audit log object added
- auditlog.get method added
EXAMPLE AUDITLOG.GET CALL

```json
1 {  2   "jsonp": "2.0",  3   "method": "auditlog.get",  4   "params": {  5     "output": "extend",  6     "sortfields": "clock",  7     "sortorder": "DESC",  8     "limit": 2  9   }, 10   "id": 1, 11   "auth": "5c4eb8ee5e42b55035f1a6e51187767f" }
```

```
1 {  2   "jsonp": "2.0",  3   "result": {  4     {  5         "auditid": "1998",  6         "userid": "4",  7         "clock": "2019-08-16T03:19:15",  8         "action": "I",  9         "resourcetype": "0", 10         "note": "", 11         "ip": "192.168.1.144", 12         "resourcet": "", 13         "resourcename": "" 14     }, 15     { 16         "auditid": "197", 17         "userid": "1", 18         "clock": "2019-08-16T03:18:15", 19         "action": "I", 20         "resourcetype": "0", 21         "note": "", 22         "ip": "192.168.1.144", 23         "resourcet": "5", 24         "resourcename": "user2" 25     } 26   }, 27   "id": 1 28 }
```
AUDIT LOG OBJECT AND METHOD

- Filter by audit ID’s and/or user ID’s
- Ability to search by old value and new value
- The object contains information about Action type, Resource type, IP address, Resource ID’s, Names and other details
- Potentially very useful for parsing audit data and notifying you when critical changes have been made
HTTP PROXY IN WEBHOOKS

ADDED ABILITY TO USE WEBHOOKS WITH HTTP PROXY
HTTP PROXY WEBHOOK CONFIGURATION

- Ability to specify HTTP proxy when configuring a webhook

- Same logic as in HTTP agent

- Need to specify the HTTPProxy parameter:

  ```
  [protocol://][username[:password]@]proxy.mycompany.com[:port]
  ```
DB CHARACTER SET CHECK

WARN USERS OF CHARACTER SET MISCONFIGURATION
- Warning is displayed during the initial frontend setup
- Warning is displayed on already running instances
**EXAMPLE OF AN ISSUE**

<table>
<thead>
<tr>
<th>Hostid</th>
<th>Hostname</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10325</td>
<td>ABC</td>
<td></td>
</tr>
<tr>
<td>10326</td>
<td>Abc</td>
<td></td>
</tr>
<tr>
<td>10327</td>
<td>AbC</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing host statuses](image-url)

**Details:** Cannot update host

Host "Abc" already exists. (hosts.php:856 → CApiWrapper->__call() → CFrontendApiWrapper->callMethod() → CApiWrapper->callMethod() → CFrontendApiWrapper->callClientMethod() → CLocalApiClient->callMethod() → CHost->update() → CHost->massUpdate() → CApiService::exception() in include/classes/api/services/CHost.php:1063)
Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Details ▲ Cannot connect to the database.

Unsupported charset or collation for tables: acknowledges, actions, alerts, application_discovery, application_prototype, applications, auditlog, auditlog_details, autoreg_host, conditions, config, config_autoreg_tis, cor_condition_tag, cor_condition_tagpair, cor_condition_tagvalue, correlation, dashboard, dchecks, drules, dservices, event_tag, events, expressions, functions, globalmacro, graph_theme, graphs, graphs_items, group_discovery, group_prototype, history_log, history_str, history_text, host_discovery, host_inventory, host_tag, hostmacro, hosts, housekeeper, hstgrp, httpstep, httpstep_field, httptest, httptest_field, icon_map, icon_mapping, ids, images, interface, interface_smp, item_condition, item_discovery, item_preproc, item_rdata, items, iid_MACRO_PATH, iid_override, iid_override_condition, iid_override_operation, iid_override_ophistory, iid_override_opprior, iid_override_opptrig, iid_override_opp trends, maintenance_tag, maintenances, mappings, media, media_type, media_type_message, media_type_param, module, opcommand, opconditions, operations, opmessage, problem, problem_tag, profiles, proxy_autoreg_host.
Zabbix supports only "utf8_bin" collation. Database "zabbix" has default collation "utf8_general_ci"

character set name or collation name that is not supported by Zabbix found in 421 column(s) of database "zabbix"

only character set "utf8" and collation "utf8_bin" should be used in database
WHY DOES COLLATION MATTER?

- Ensures that the DB backend is aware of object case sensitivity
- No way to detect before 5.0 but to take a look at the DB table structure from the DB side
- If utf8_bin collation is not used, the user may eventually encounter duplicate SQL errors or other unexpected behavior in the frontend
MAIN BENEFITS OF DETECTING MISCONFIGURATION

- Letting users know that there’s DB misconfiguration
- It’s better to catch any issues and fix them during the initial setup of the DB
- Users should avoid running any extra queries on DB’s that have over time substantially grown in size
THANK YOU!