“USER ROLES FOR THE ENTERPRISE”

Arturs Lontons
Technical Support Engineer

ZABBIX, LATVIA
PERMISSION GRANULARITY

GRANULAR PERMISSION REQUIREMENTS IN CORPORATE ENVIRONMENTS
PERMISSIONS GRANULARITY

NOC Team role
- Access to Dashboards and maps
- Restrict unnecessary UI elements
- Restrict API access
- Restrict configuration
- Restrict closing problems

Network Administrator role
- Access to Dashboards and Maps
- Access to Configuration
- Restrict API access
- Restrict unnecessary UI elements
ROLES AND MULTI-TENANCY

In multi-tenant environments granular permission play a very important role

✔ The UI should be as intuitive as possible for different roles and tenants

✔ Each tenant can have different monitoring requirements

✔ Restricted access to elements per tenant

✔ Isolation between tenants
USER ROLES IN 5.2

IMPLEMENTING A MORE GRANULAR PERMISSION LOGIC WITH USER ROLES
Starting from 5.2 users will have a User role assigned to them. Depending on the role, a corresponding User type will also be assigned:
USER TYPES

- User types are not being removed
- A role is linked to one of the 3 user types

User roles

<table>
<thead>
<tr>
<th>Name</th>
<th>Admin role</th>
</tr>
</thead>
<tbody>
<tr>
<td>User type</td>
<td>Admin</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Dashboard, Overview, Maps</td>
</tr>
<tr>
<td>View Elements</td>
<td>Problems, Latest data, Discovery</td>
</tr>
<tr>
<td>Admin</td>
<td>Hosts, Screens, Services</td>
</tr>
</tbody>
</table>
USER TYPES

Frontend sections restricted by user type

Superadmin:

Admin:

User:
DEFAULT USER ROLES

Available in Administration – User roles

By default this section contains 4 preconfigured user roles

☑ Super admin role
☑ Admin role
☑ User role
☑ Guest role
SUPER ADMIN ROLE

- Default Super admin role is static – users cannot modify this role

Table of roles:

- **Admin role**
  - Users: 1
  - Users: Tenant Admin

- **Guest role**
  - Users: 1
  - Users: Admin (Zabbix Administrator)

- **Super admin role**
  - Users: 1
  - Users: API, Arturo, guest, Tenant User

- **User role**
  - Users: 4

- This is because at least a single Super admin role MUST exist in your environment

- Newly created roles of type Super admin can be modified
USER ROLES SECTION

Each of the default roles contains the maximum allowed permissions per user type:

<table>
<thead>
<tr>
<th>User roles</th>
<th>Access to UI elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin role</td>
<td>Monitoring</td>
</tr>
<tr>
<td></td>
<td>- Dashboard</td>
</tr>
<tr>
<td></td>
<td>- Overview</td>
</tr>
<tr>
<td></td>
<td>- Maps</td>
</tr>
<tr>
<td></td>
<td>Inventory</td>
</tr>
<tr>
<td></td>
<td>- Overview</td>
</tr>
<tr>
<td></td>
<td>Reports</td>
</tr>
<tr>
<td></td>
<td>- System information</td>
</tr>
<tr>
<td></td>
<td>- Audit</td>
</tr>
<tr>
<td></td>
<td>Configuration</td>
</tr>
<tr>
<td></td>
<td>- Host groups</td>
</tr>
<tr>
<td></td>
<td>- Maintenance</td>
</tr>
<tr>
<td></td>
<td>- Discovery</td>
</tr>
<tr>
<td></td>
<td>Administration</td>
</tr>
<tr>
<td></td>
<td>- General</td>
</tr>
<tr>
<td></td>
<td>- User groups</td>
</tr>
<tr>
<td></td>
<td>- User roles</td>
</tr>
<tr>
<td></td>
<td>- Media types</td>
</tr>
</tbody>
</table>

- Hosts
- Screws
- Services
- Triggers top 100
- Notifications
- Event correlation
- Authentication
- Users
- Queue
UI ELEMENT RESTRICTION

Access to UI elements for each role can be restricted

- **NOC user role** that has access only to Dashobards and maps
UI ELEMENT RESTRICTION

Access to UI elements for each role can be restricted

- NOC user role that has access only to Dashboards and maps
HOST GROUP PERMISSIONS

Note, that User Group access to Host Group still have to be properly assigned!
ACCESS TO API

API Access can also be restricted for each role

✅ Used when creating API specific user roles

Access to API

Enabled

API methods

Allow list  Deny list

Search:
type here to search
Select
API METHOD RESTRICTION

Ability to implement API method Allow or Deny lists

- Useful in environments with many administrative roles
- For example - create a user that can only use get methods

Access to API

Enabled ✓

<table>
<thead>
<tr>
<th>API methods</th>
<th>Allow list</th>
<th>Deny list</th>
</tr>
</thead>
<tbody>
<tr>
<td>event.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>history.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>host.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>item.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>template.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>trend.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>trigger.get</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>type here to search</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Select]
API METHOD RESTRICTION EXAMPLE

If the method execution is not permitted, «no permission to call method» error will be displayed:

```json
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params."
    },
    "data": "No permissions to call \"host.create\"."
},
"id": 1
}
```
ACCESS TO ACTIONS

Each role can have a specific list of actions that it can perform with respect to the role 
User type.

User type: User

Access to actions

- Create and edit dashboards and screens
- Create and edit maps
- Create and edit maintenance
- Acknowledge problems
- Close problems
- Change severity
- Add problem comments
- Execute scripts

User type: Admin

Access to actions

- Create and edit dashboards and screens
- Create and edit maps
- Create and edit maintenance
- Acknowledge problems
- Close problems
- Change severity
- Add problem comments
- Execute scripts
RESTRICTING ACTIONS EXAMPLE (1)

Restricting the role

Access to actions

- Create and edit dashboards and screens
- Create and edit maps
- Create and edit maintenance
- Acknowledge problems
- Close problems
- Change severity
- Add problem comments
- Execute scripts

Result in the frontend

Update problem

Problem: Zabbix agent is not available (for 3m)

Message

History

Scope

Only selected problem
Selected and all other problems of related triggers 1 event

Change severity

Not classified Information Warning Average High Disaster

Acknowledge
Close problem
RESTRICTING ACTIONS EXAMPLE (2)

Unrestricted role

- Create and edit dashboards and screens
- Create and edit maps
- Create and edit maintenance
- Acknowledge problems
- Close problems
- Change severity
- Add problem comments
- Execute scripts

Result in the frontend

Update problem

Problem: Zabbix agent is not available (for 3m)

Message:

History

Scope

- Only selected problem
- Selected and all other problems of related triggers

Change severity

- Acknowledge
- Close problem
DEFAULT ACCESS

Default access for new elements of different types can be enabled or disabled for user roles:

Default access to new actions  ✔
Default access to new modules  ✔
Default access to new UI elements  ✔

If enabled, whenever a new element is added, the user belonging to this role will automatically have access to it.
ROLE ASSIGNMENT POST-UPGRADE

After migration to 5.2, the users will have the pre-created Admin/User/Super admin roles assigned to them:
EXAMPLE USE CASES

HOW ROLES CAN BE USED IN DIFFERENT ENVIRONMENTS
READ ONLY ROLE

A role that has no ability to create or modify any elements

- Read only access to dashboards
- Read only access to problems
- No access to API
- No permissions to execute frontend scripts
READ ONLY ROLE

First off, let’s decide on User type and sections, which this role should have access to
READ ONLY ROLE

We also need to restrict access to actions, API and decide on the new UI element and module permission logic.
READ ONLY ROLE

- No option to create or edit a dashboard
- Access to dashboards is granted and evaluated based on Dashboard sharing options and User group – Host group relationship.
READ ONLY ROLE

- Restricted UI elements hidden
- Acknowledge button is not clickable for this Role
RESTRICT ACCESS TO ADMINISTRATION SECTIONS

A Superadmin type role that has no access to User configuration and General Zabbix settings

- Ability to Create and manage proxies
- Ability to define media types and frontend scripts
- Access to queue section to see the Zabbix server and proxy health status
RESTRICT ACCESS TO ADMINISTRATION SECTIONS

User type – Super admin. General and User sections are restricted for this role

<table>
<thead>
<tr>
<th>User type</th>
<th>Super admin</th>
</tr>
</thead>
</table>

**Access to UI elements**

- **Monitoring**
  - Dashboard
  - Overview
  - Maps

- **Inventory**
  - Overview

- **Reports**
  - System information
  - Audit
  - Availability report
  - Action log

- **Configuration**
  - Host groups
  - Maintenance
  - Discovery
  - Templates
  - Actions
  - Services

- **Administration**
  - General
  - User groups
  - Media types
  - Proxies
  - Scripts
  - Authentication
  - Users
  - Queue
RESTRICT ACCESS TO ADMINISTRATION SECTIONS

- Restricted Administration elements are hidden
- The Monitoring Super admin user still has the ability to create new Proxies, Media Types, Scripts and has access to the Queue section
Zabbix Dashboards and maps are used by multiple tenants to provide monitoring data

- Read only access to dashboards
- Read only access to maps
- No access to API
- No access to configuration
- Isolation per tenant
ROLES FOR MULTI-TENANT ENVIRONMENTS

We will be creating a user type role with very limited access to UI

<table>
<thead>
<tr>
<th>Access to UI elements</th>
<th>Monitoring</th>
<th>Inventory</th>
<th>Reports</th>
<th>Configuration</th>
<th>Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Dashboard</td>
<td>Overview</td>
<td>System information</td>
<td>Host groups</td>
<td>General</td>
</tr>
<tr>
<td>User type</td>
<td></td>
<td>Overview</td>
<td>Audit</td>
<td>Maintenance</td>
<td>User groups</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Discovery</td>
<td>Media types</td>
</tr>
<tr>
<td>* Name</td>
<td>Tenant NOC</td>
<td>Maps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User type</td>
<td>User</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Monitoring: Dashboard, Maps
- Inventory: Overview
- Reports: System information, Audit
- Configuration: Host groups, Maintenance, Discovery
- Administration: General, User groups, Media types

- Problems, Latest data, Discovery, Availability report, Action log, Triggers top 100, Notifications
- Hosts, Screens, Services, Templates, Actions, Events correlation
- Triggers top 100, Notifications
- Hosts, Users, Queue
ROLES FOR MULTI-TENANT ENVIRONMENTS

Let’s use tag-based permissions to isolate our Hosts per tenant
ROLES FOR MULTI-TENANT ENVIRONMENTS

Don’t forget to tag your problems! This time I’m doing the tagging on the host level:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>Zabbix SIA</td>
<td>Remove</td>
</tr>
<tr>
<td>DC</td>
<td>Riga</td>
<td>Remove</td>
</tr>
</tbody>
</table>

Add

Update  Clone  Full clone  Delete  Cancel
ROLES FOR MULTI-TENANT ENVIRONMENTS

- The UI is restricted only to the required monitoring sections
- Tag-based permission ensure that we are seeing problems related to our tenant
ROLES – WHAT’S NEXT?

Implementing user roles can help you manage your Zabbix environment!

- Improve auditing
- Restrict configuration
- Restrict API access
- Remove unwanted UI elements