Mikhail Sharin

AUTOMATIC MONITORING COMPUTER SYSTEM AND NETWORK

Cyberlab Alarm Forwarding

Bachelor's Thesis

Data & Networking

April 2016

KYAMK
University of Applied Sciences

KYAMK
University of Applied Sciences

Author Mikhail Sharin	Degree Bachelor of Data & Networking	Time December 2016
Automatic Monitoring Computer System and Network Cyberlab Alarm Forwarding		31 pages

Commissioned by
Kymenlaakso University of Applied Sciences

Supervisor
Vesa Kankare, Senior Lecturer

Abstract

The main objective of this bachelor’s thesis was to explore a possibility to perform ICTLAB sensor and server monitoring. The main goal was to study the network monitoring topic and configure one of the most popular monitoring systems.

During the thesis, Zabbix monitoring system was configured for ICTLAB alarm forwarding. Also, different forms of network monitoring were studied and other monitoring systems were briefly described.

Also, this thesis aims to introduce network technologies and protocols such as SMTP for emails transferring, SNMP for data gathering, MIB database and OID for network device detecting.

As a result of this thesis, ICTLAB alarm forwarding was established and configured using Zabbix monitoring system. It allows to control the network and the sensor state and notify the system administrator via e-mail.

Keywords
documentation, model, thesis, report writing, network nodes monitoring, Zabbix, SNMP, MIB, alarm forwarding, SMTP, water leak sensor

CONTENTS

ABBREVIATIONS

SaaS	Software as a Service
PaaS	Platform as a Service
SNMP	Simple Network Management Protocol
VM	Virtual Machine
SMS	Short Message Service
DNS	Domain Name System
CPU	Central Processing Unit
CDN	Content Delivery Networks
XMPP	Extensible Messaging and Presence Protocol
HTTP	Hypertext Transfer Protocol
RAM	Random Access Memory
CGI	Common Gateway Interface
TCP	Transmission Control Protocol
IPv4/IPv6	Internet Protocol version 4/6
LAN	Local Area Network
SMTP	Simple Mail Transfer Protocol
OID	Object Identifier
MIB	Management Information Base

1. INTRODUCTION

1.1 Main objectives

Automation is one of the main priorities for direction of progress in the modern world. Modern technology allows us to perform plenty of daily tasks automatically. We no longer need to set an alarm every evening because it works by schedule. We do not have to wash linen by hand because it is performed by a washing machine. Nowadays, technology has made a great leap and introduced smart houses and autopilot for cars.

The development of cloud infrastructure is taking place in the modern world along with automation. SaaS (Software as a Service) and PaaS (Platform as a Service) are now in the trend. Regular offline applications have become useless. This trend makes IT companies take care of permanent internet connection with users and stabilise work of their servers. Competition between internet giants forced them to decrease possibility of failure. This led to a change in measuring of device availability and the use of a monitoring software.

It has led to the development of special software which is responsible for the automatic monitoring of server availability and sensor state. These services work in automatic mode, checking defined parameters and sending information directly to the system administrator.

The objective is to choose and implement one such system and configure it to the alert administrator in case of failure.

Monitoring systems should handle the following tasks:

- Monitoring of current status of defined systems
- Detecting critical states that require network administrator attention
- Gathering information from sensors through SNMP (Simple Network Management Protocol)
- Composing of network security threat messages

- Delivering composed messages to the responsible personnel

There are plenty of systems for monitoring networks. Most of them have a similar function. There are also open source and proprietary solutions among them. Examples of such systems are Nagios, Zabbix, Cacti, OpenNMS, Icinga. They are fit for the execution of required tasks. These programs offer monitoring and alert services for servers, switches, applications and services. They alert users when things go wrong and alert them a second time when the problem has been resolved.

The main objective of the thesis is to implement one monitoring system. This paper describes how to install and configure Zabbix 3.0 for required purposes such as automatic monitoring of network nodes and sensors. Also, theoretical material was devoted to network monitoring in general, and other monitoring systems.

1.2 Environment

This bachelor thesis was designed for KYAMK ICTLAB. ICTLAB is equipped with RITAL CMC III sensors system but it does not forward alarm notifications from it.

For studying purposes, Oracle VM VirtualBox has been used as a network nodes emulator. Also, different operating systems such as Ubuntu Linux distributive and Windows 10 Pro have been used for test monitoring in different environments.

2. NETWORK MONITORING

The network monitoring means a process of gathering information from network nodes which can be used for detecting system malfunction. After data collecting, monitoring system determines severity rating, makes logs, draws graphs and tries to repair occurred problems by built-in tools and scripts. Also, monitoring process provides the notifying the responsible person by e-mail in case of network failure. (Kompyuternye seti i texnologii, 2007)

2.1 Types of network monitoring

Network monitoring can be divided into three parts: Manual check, Scheduled check and Problems monitoring. The difference between the types is in the complexity. The manual check is the simplest method and the problems monitoring is the most complex one.

Manual check

Manual check is one of the most popular methods of network monitoring. It allows to perform a manual ping to server through the console and check its availability. This method is inefficient and useless for large networks because it is impossible to predict the appearance of a problem, connection can disappear at random and availability problems must be decided in a short time. (Habrahabr, 2013)

Schedule check

This way of network monitoring is related to executing problems checking by schedule. The correct time interval of check schedule is the key point of this method. Check interval can be different depending on the importance of the server. Approximately, a 10 minute interval would be enough for the internet site monitoring, but in case of a more important server, interval time may be reduced to seconds. (Habrahabr, 2013)

Problems monitoring

Regular manual or schedule monitorings are insufficient if the availability rate of server is becoming critical. It is necessary to use a schedule check of several parameters with a high inspection frequency. In some special cases, it is required to use different geolocations for check executing. Here are instances of problems for testing:

- DNS-server availability
- Ping time
- Scheduled task execution
- Problems with static content downloading
- Database connection loss
- Various sensors data

This thesis is focused on problem monitoring because it is more difficult and useful monitoring type for data centres and networks. It allows to catch a problem occurrence in time, notify the system administrator and try to resolve accidents in automatic mode. (Habrahabr, 2013)

3. RELIABILITY LEVELS OF PROBLEMS MONITORING

Problem monitoring can also be divided into four parts by reliability levels: Internal monitoring, Infrastructure monitoring, Cloud monitoring, External monitoring. Various types of system monitoring are used in systems of different size. Every case of monitoring is unique and every type of system monitoring must meet the requirements, specified by the system administrator.

3.1. Internal monitoring

Internal monitoring is providing data about physical server's parameters (operations with disk, memory, network and CPU utilization). These are used only for collecting information about devices and services. Theoretically, it is possible to use triggers (scripts) on this monitoring level to perform actions for

repairing network issues but usually it is a work for the next step. (Habrahabr, 2014)
Here are examples of internal monitoring tests: PHP, NGINX or Apache server, MySQL database, Disk usage, Mail daemon, Network state and Operation system.

3.2. Infrastructure monitoring

Infrastructure monitoring is the inspection of network infrastructure in general or its part. There are separate servers for these purposes along with regular servers. They perform hundreds or thousands checks and analyse gathered information. Their objective is to receive monitoring information, determine critical states and notify system administrator by SMS or e-mail in the event of alarm. The most popular software solutions for infrastructure monitoring are Nagios, Zabbix, Icinga. This level of monitoring allows to setup complex triggers to analyse gathered data and to make decision whether to increase or decrease additional capacity. Also, a good practice is to create templates for monitoring network nodes and run automatic observation for every new devices in a network. (Habrahabr, 2014)

3.3. Cloud monitoring

Usually, the first couple of steps are enough for detecting and solving problems with infrastructure, but there is one more step for a cloud based network. Along with the monitoring of server infrastructure, the analysis of queries passing through the cloud also takes place there. Cloud monitoring uses advanced levels of quality control. For example, Amazon provides this type of monitoring. It allows for making decisions about switching power between geo-clusters. (Habrahabr, 2014)

3.4. External monitoring

External monitoring helps to monitor the quality of service from the user’s side. Users may have problems with service using, even if infrastructure is stable, connectivity between servers is not broken, or all servers work in regular mode. It depends on the general condition of the network. There are advanced triggers to switch users to other geo-clusters to increase quality. Also, this level of monitoring may be used for Cloud control and Infrastructure monitoring. It gives additional check of real ping timeout from client side. Usually, this type of monitoring is used in CDN (Content Delivery Networks). For example, it allows to test web page loading time for the client. Also, this method uses for checking servers availability from different parts of the world. If a user tries to get information from non-working servers, he will be switched to an active one. (Habrahabr, 2014)

4. SOLUTIONS FOR MONITORING

4.1. Zabbix

Description

It is an open source system for monitoring and checking the status of networks, services, servers and other equipment. It is composed of several parts and allows to divide them into different servers in case of high load. (Zabbix Documentation, 2016)

Zabbix includes:

- Own server for monitoring, which allows to receive data by schedule, processing, analyzing and notification
- Databases (MySQL, SQLite, Oracle or PostgreSQL)
- PHP web-interface
- Agent - daemon for gathering data from clients

Zabbix features:

- Monitoring up to thousand nodes
- Triggers based on monitoring
- Monitoring of Log files
- Creating reports
- Forwarding an alarm
- Managing templates
- Drawing network map

Principle of operation

Host is the basic unit for monitoring. Every server has a description and an address. Hosts form groups, for example, groups of web-servers or groups of databases. Groups are used for filtering monitoring results.

Every host has items (parameters) for monitoring. For instance, there is item ping. In case of successful query, it equals 1, or 0 in case of failure. There is also another parameter for counting online users. Zabbix allows to change schedule of monitoring or multiplier for each host's items. Besides, it may create charts for different parameters for any time.

Zabbix provides flexible configuration of triggers for accident situations and alarm activation. It can use different means of communication to send a notifying alarm message through e-mail, text messages or XMPP. Likewise, the system creates repairing scripts which would activate in the event of alarm.

Zabbix can be used for the mapping of network logical structure. It displays the host's location, their connection and availability.

4.2. OpenNMS

Description

The main purpose of OpenNMS (Open Network Monitoring System) is the monitoring of different types of servers, services, and network infrastructure in general. For information gathering, the system uses collectors which work through SNMP (Simple Network Management Protocol), HTTP (Hypertext Transfer Protocol) and other protocols. OpenNMS is powered by Java, thereby, this software is platform-independent and can operate on any operating system with PostgreSQL server. (Opennms, 2016)

Principle of Operation

Interface is the basic unit for gathering information. Network device with interfaces forms a node. Online sensor such as temperature detection is also considered an interface. In case of detecting a new interface, "newSuspect" action will be triggered, and included services will be searched by "provisiond" (provisioning daemon responsible for finding services) and formed into a node. OpenNMS provides an automatic scanning of new interfaces by schedule or in manual mode.

In the end of data gathering process, “Policies” define rules for nodes which comply with some requirements, applying policies provided to implement filtering by different parameters.

Here are examples of policies:

- “MatchingSnmpInterfacePolicy” - enables forced data gathering from interfaces whose description matches with a defined keyword.
- “MatchingIpInterfacePolicy” - as in the previous case but with ip address filtering.
- “NodeCategorySettingPolicy” - by nodes filtering.

4.3. Catci

Description

It is a light monitoring system for data gathering and creating charts with RRDtool. Catci collects and shows with graphics all the statistics for defined time intervals. Statistic information is being collected from client devices by default templates such as CPU utilisation, RAM usage, number of running process and input/output network traffic. (Cacti, 2012)

Here are main benefits of the solution:

- Simple realization of web-interface (no flash, no activeX) provides high performance through mobile devices.
- Easy database structure which is ready to implement custom functionality.
- Significant amount of plugins for monitoring network equipment.

Furthermore, Catci provides netflow (Ability to collect IP network traffic as it enters or exits an interface. This technology is provided by Cisco), creation by flowview and flow-tools plugins.

4.4. Nagios

Description

It is one of the commonly known solutions for network monitoring. It provides monitoring and alerting mechanism for operating systems, programs, servers, and services. Nagios has the largest community of developers and significant amount of manuals and instructions. It was developed as expandable module architecture. There are plenty of custom plugins and add-ons which allow to control all types of applications, systems and services. Plugins are used by Nagios for gathering information about server’s ping time and free space in storage. Furthermore, custom plugins may be created by powerful plugins creator with useful user interface. (Nagios, 2016)

Nagios user interface is realized as a web-application. Web-server configuration and CGI-scripts are included in the basic kit. Also, there is a notification system for composing messages about emergency situations.

5. IMPLEMENTING A MONITORING SYSTEM

The practical part of the thesis work is devoted to the instruction for implementing one of such monitoring systems in the real network infrastructure. For experiments Zabbix monitoring system was chosen. Also, Oracle VM VirtualBox system is used for network emulation and Rittal CMC III sensors for gathering information about KYAMK ICTLAB status.

Zabbix use was caused by a number of reasons:

- It is one of the most popular monitoring systems.
- Zabbix is an open source project
- There is a huge multilanguage users and developers community on the internet, which includes English, French, Japanese, Polish, Portuguese and Russian languages.
- New version of Zabbix 3.0 with new features was released in the middle of February 2016.

Oracle VM VirtualBox is a free open source product for virtualization. It fully meets the requirements and has needful features such as creating virtual machines for different operating systems, and emulating network between virtual machines. VirtualBox also has possibility to transfer configured Zabbix server virtual machine to VMware Workstation used by KYAMK for monitoring real ICTLAB sensors. (Oracle VM VirtualBox, 2016).

Rittal CMC III is a control system for monitoring different types of sensors, such as temperature or water leakage. It has following features (Rittal, 2016):

- Support of most useful network protocols (TCP/IPv4, TCP/IPv6, SNMPv1, SNMPv3).
- Built-in temperature and access control sensors.

The Practical part of the thesis work is based on configuring Zabbix server. The final configuration should reach two goals: The first one is regular monitoring of computers in corporation network. This goal will be reached with use of the Zabbix agent on the client side of network. The agent shares gathered information with Zabbix server. The second goal is sensors monitoring in the ICTLAB server room. Monitoring of sensors will be achieved with use of SNMPv2 access to Rittal CMC III ICTLAB sensors.

Zabbix implementation includes 5 steps:

1. Network infrastructure emulation.
   - Creating 3 virtual machines.
   - Making network connection between devices.
2. Zabbix server installation.
   - Installation from packages.
   - Installing additional software, such as PHP, Apache, mySQL.
   - Tweaking of configuration files.
3. Zabbix agent installation.
   - Installation from packages.
   - Tweaking of configuration files.
4. Starting Zabbix monitoring.
   - Tweaking of host, items, triggers and graphs.
5. Monitoring of the ICTLAB sensors.
   - Parsing Rital MIB database.
   - Tweaking of host, items, triggers and graphs in Zabbix.

5.1. Step 1: Network infrastructure emulation

As was said, Oracle VM VirtualBox system will be used for network emulation. Three virtual machines will emulate network infrastructure (as seen in Figure 1). The first virtual machine powered by Ubuntu operating system will be used for hosting Zabbix server and other two powered by Windows and Ubuntu for clients emulation. All virtual machines will be equipped with two network adapters. One of them for the internet connection and the second one for local network emulation. Connection between virtual stations will be established via LAN (Local Area Network). Management of virtual machines will be carried out via SSH (Secure Shell) console.

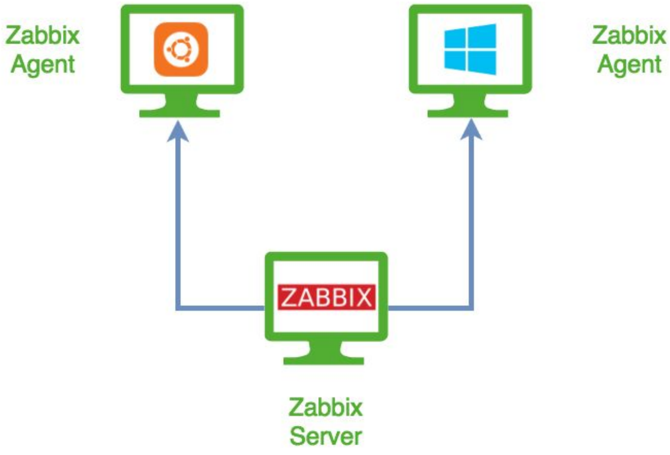


Figure 1: Network infrastructure emulation

5.2. Step 2: Zabbix server installation

Step Description

Zabbix Server is the main part of Zabbix monitoring system. Server executes data gathering from clients' devices, stores information in database, makes data graphs, determines critical cases, performs triggers and sends notifications to the responsible system administrator. Zabbix server contains three main parts: Zabbix server, Web-interface, Database.

Database contains all Zabbix configuration. In case of new host or item creation, it will be saved in the database. Database receives server's requests by schedule and shares data. Further, the server saves active data to the cache and notifies web-interfaces about the changes. This process may produce delay for a few minutes.

Installation

Zabbix provides servers for the most used Linux distributives, such as Ubuntu, Debian, Red Hat Enterprise Linux, CentOS, Oracle Linux. Also, Zabbix offers three ways of the server installation such as: Installation from packages, Installation from sources, Solutions for virtualization.

In case of using Ubuntu operating system, it is more convenient for the server hosting to install Zabbix from packages via repository. Zabbix provides its own repository for downloading. It should be added to Ubuntu before downloading. After that, Zabbix server is available for installation. Also, PHP and MySQL have to be installed with Zabbix. In case of Ubuntu operating system, it looks like:

```
shell> apt-get install zabbix-server-mysql
zabbix-frontend-php
```

Then, MySQL Database should be created and configured.

```
shell> mysql -uroot -p<password>
mysql> create database zabbix character set utf8
collate utf8_bin;
mysql> grant all privileges on zabbix.* to
zabbix@localhost identified by '<password>';
mysql> quit;
```

Also, Zabbix provides the possibility to work with different types of databases, such as PostgreSQL, IBM DB2, SQLite.
Further, PHP required options should be configured as in the example:

```
    php.conf> php_value max_execution_time 300
    php.conf> php_value memory_limit 128M
    php.conf> php_value post_max_size 16M
    php.conf> php_value upload_max_filesize 2M
    php.conf> php_value max_input_time 300
    php.conf> php_value always_populate_raw_post_data -1
    php.conf> php_value date.timezone Europe/Helsinki

    shell> service apache2 restart
```

Next step will be Zabbix server configuring. Required fields in configuration file should be filled. For example:

```
    zabbix_server.conf> DBHost=localhost
    zabbix_server.conf> DBName=zabbix
    zabbix_server.conf> DBUser=zabbix
    zabbix_server.conf> DBPassword=zabbix
```

Further, Zabbix server should be started as a Linux service:

```
    shell> service zabbix-server start
```

5.3. Step 3: Zabbix agent installation

Step Description

Zabbix agent is a client side daemon for monitoring client device state, such as CPU utilization, memory usage and other characteristics. Information gathered by the agent is sent to the server for further processing. Zabbix agent uses the native system api for data collection because it is very effective. Agent daemon provides passive and active checks. In case of the passive check, the agent responds to server’s queries only. The active check sends data to the server without any queries. It’s more complicated process based on the gathering information according to the checklist.

Zabbix agent can be used on different kinds of platforms, such as Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows. It allows to use Zabbix monitoring almost everywhere.

Installation

Zabbix agent will be installed in the client virtual machine as a package from Zabbix repository. After downloading, it should be started using terminal commands.

```
shell> apt-get install zabbix-agent
shell> service zabbix-agent start
```

The agent will also be configured via changing configuration file and restarted.

```
zabbix_agentd.conf> Server=192.168.56.101
zabbix_agentd.conf> Hostname=Example

shell> service zabbix-agent restart
```

5.4. Step 4: Starting Zabbix monitoring

Step Description

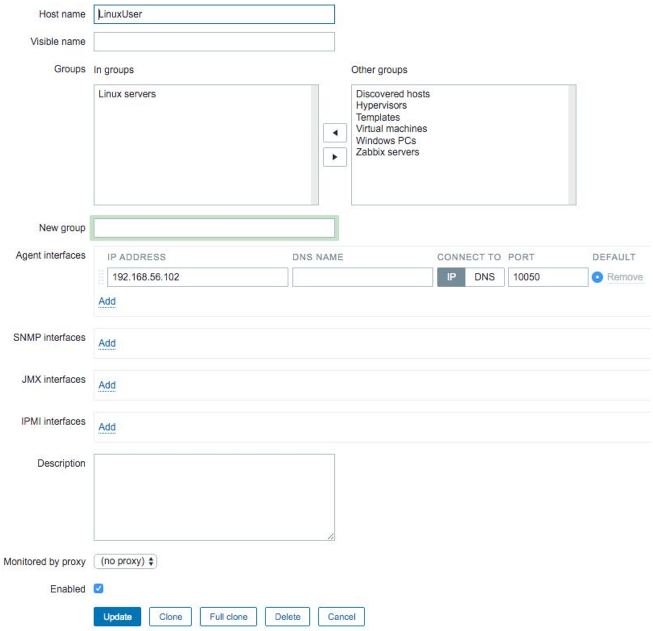
First of all, it is required to define some terms commonly used in Zabbix monitoring.

- Host - a network device for monitoring via IP or DNS addressing.
- Item - a particular data element from the host device.
- Trigger - a logical expression which means critical level of item value.
- Notification is a message about an occurred alarm case sent via chosen channel.

Monitoring

Zabbix is managed via the web-interface. Frontend is available at http://server-address/zabbix in the browser. The first time Zabbix server is started, user have to authorize with default user and password. New accounts can be created via control panel in section Administration/Users. Zabbix users are divided in groups and has different permissions for management.

The first step to launching monitoring is **creating a host** (Figure 2). Virtual machine powered by Ubuntu with Zabbix agent will be used as the host for monitoring. Information about configured Zabbix hosts is available in Configuration/Hosts panel. Host creating has some required fields. Host name must be the same as the client device’s host name. Also, hosts are divided into groups and it is required to choose one. Further, it is necessary to determine a client’s interface for connection to Zabbix agent. Zabbix provides connection via IP or DNS.



Host name	LinuxUser
Visible name	
Groups – In groups	Linux servers
Groups – Other groups	Discovered hosts Hypervisors Templates Virtual machines Windows PCs Zabbix servers
New group	

Agent interfaces

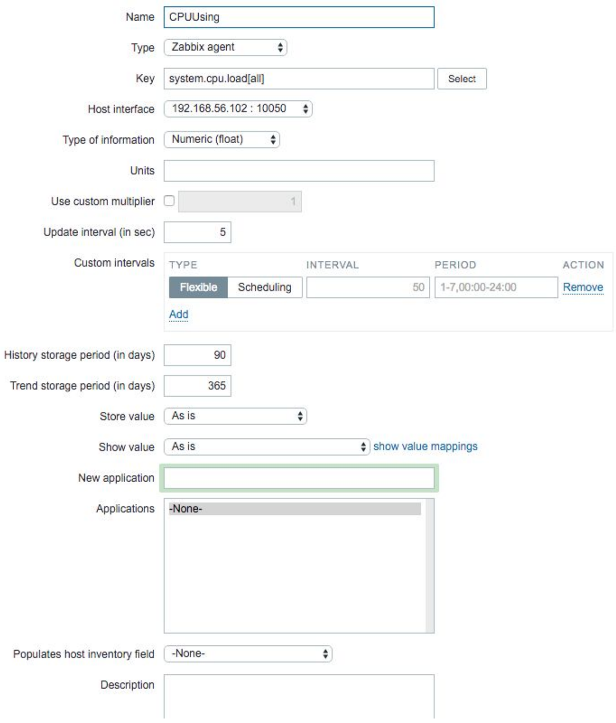
IP ADDRESS	DNS NAME	CONNECT TO	PORT	DEFAULT
192.168.56.102		IP	10050	●

Description	
Monitored by proxy	(no proxy)
Enabled	☑

Figure 2: Creating a host

The next step will be **new item creating** inside host (Figure 3). New items can be created inside a host. Items also have some required fields. Item Name will be displayed in Zabbix dashboard. Zabbix provides different types of data gathering from item. The Most useful is collecting by agent. One of them must be defined with use of special agent key. For instance, there are some most useful keys:

- system.cpu.load - CPU utilization
- proc.mem - memory usage by a process
- system.users.num - number of logged in users
- system.boottime - system boot time



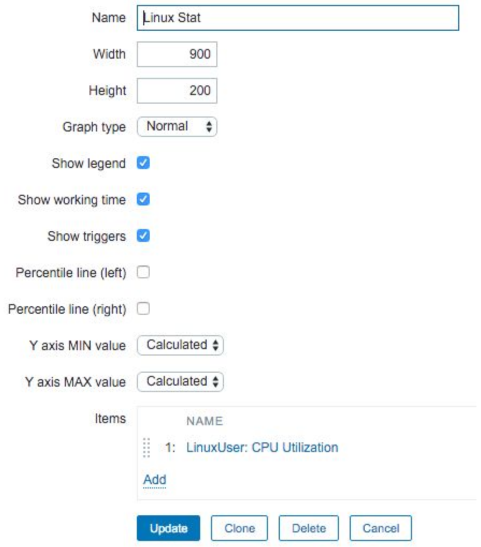
Name	CPUUsing
Type	Zabbix agent
Key	system.cpu.load[all]
Host interface	192.168.56.102 : 10050
Type of information	Numeric (float)
Units	
Use custom multiplier	□ 1
Update interval (in sec)	5
Custom intervals	
History storage period (in days)	90
Trend storage period (in days)	365
Store value	As is
Show value	As is
New application	
Applications	-None-
Populates host inventory field	-None-
Description	

Custom intervals:

TYPE	INTERVAL	PERIOD	ACTION
Flexible Scheduling	50	1-7,00:00-24:00	Remove

Figure 3: New item creating

Zabbix provides **graphs** for visual demonstration items status (Figure 4). New graph can be created inside a host. Graph creating process is similar to item creating. Required fields, such as name, size and other, have to be filled and the target item should be selected.



Name: Linux Stat

Width: 900

Height: 200

Graph type: Normal

Show legend ☑

Show working time ☑

Show triggers ☑

Percentile line (left) □

Percentile line (right) □

Y axis MIN value: Calculated

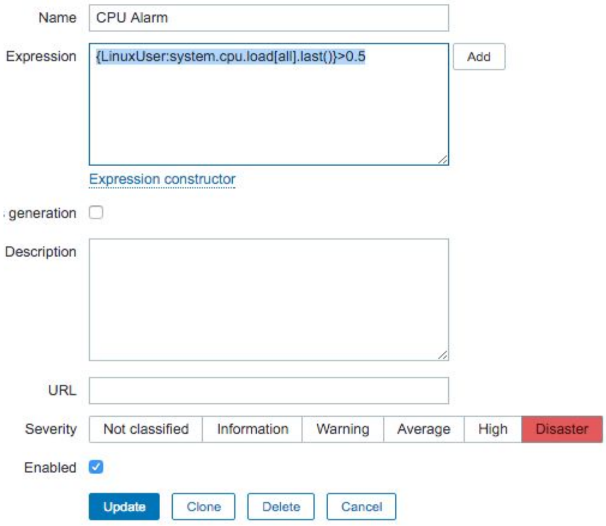
Y axis MAX value: Calculated

Items:

	NAME
1:	LinuxUser: CPU Utilization

Figure 4: Graph creating

Critical cases have to be handled by **triggers** (Figure 5). Triggers allow to determine critical changing of item value. They also should be configured via configuration menu inside a host. Trigger is based on regular expression. It contains a host name, an item for monitoring and a critical value. Zabbix provides a built-in expressions constructor. Also, triggers are divided in groups by severity level such as information, warning and disaster. Triggers do not execute anything but they provide mechanisms for actions calling.



Name: CPU Alarm

Expression: {LinuxUser:system.cpu.load[all].last()}>0.5

generation □

Description:

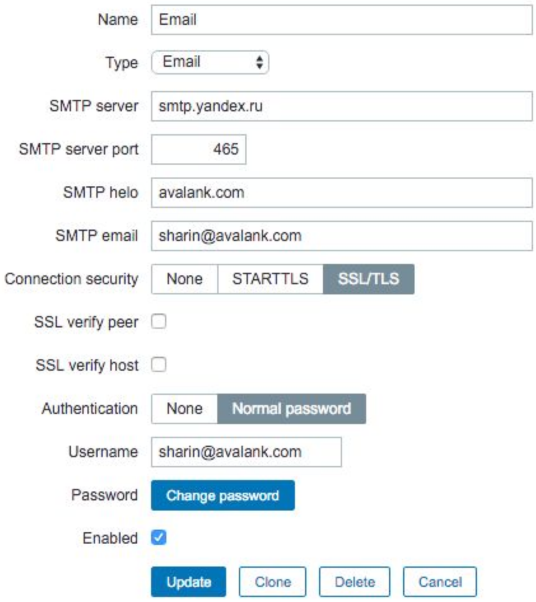
URL:

Severity: Not classified | Information | Warning | Average | High | [✓ Disaster]

Enabled ☑

Figure 5: Trigger creating

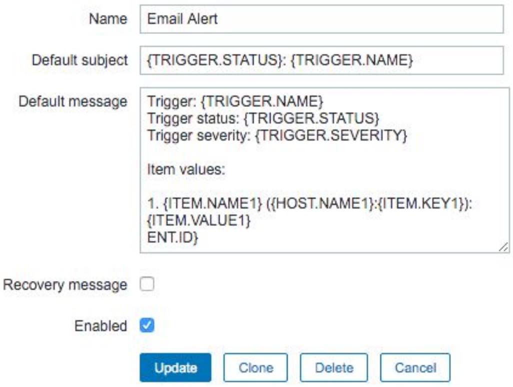
One of the most useful **actions** is sending a notification to the system administrator (Figure 6). Zabbix provides different ways for notifying. It is necessary to create an action for implementation of e-mail sending. First of all, media types for notifications delivery should be configured. Configuration of media is produced in the administration panel. E-mail notifications require an established SMTP (Simple Mail Transfer Protocol) server. Yandex SMTP server will be used for demonstration of e-mail delivering. SMTP configuration fields must be filled and saved. (Yandex support, 2016)



Name	Email
Type	Email
SMTP server	smtp.yandex.ru
SMTP server port	465
SMTP helo	avalank.com
SMTP email	sharin@avalank.com
Connection security	None STARTTLS [✓ SSL/TLS]
SSL verify peer	☐
SSL verify host	☐
Authentication	None [✓ Normal password]
Username	sharin@avalank.com
Password	Change password
Enabled	☑

Figure 6: Action creating

The next step will be setting up an action for **messages delivering** (Figure 7). Subject and body of action’s message consist of regular expressions such as {TRIGGER.STATUS}, {TRIGGER.NAME} and a type of media which was created earlier.



Name	Email Alert
Default subject	{TRIGGER.STATUS}: {TRIGGER.NAME}

Default message	Trigger: {TRIGGER.NAME}
	Trigger status: {TRIGGER.STATUS}
	Trigger severity: {TRIGGER.SEVERITY}
	
	Item values:
	
	1. {ITEM.NAME1} ({HOST.NAME1}:{ITEM.KEY1}):
	{ITEM.VALUE1}
	ENT.ID}

Recovery message	☐
Enabled	☑

Figure 7: Trigger creating

It was the last step of basic implementation of alarm messaging. Now, in case of emergency situation, e-mail will be delivered to the system administrator. Delivering time depends on checking schedule and SMTP server delay. In this case, it took approximately one minute. It is a very good result for monitoring system.

5.5. Step 5: ICTLAB sensors monitoring.

Description

KYAMK ICTLAB uses Rittal CMC III monitoring system. CMC III sensors monitor the physical ambient conditions. Every sensor has a specific measuring or control task. CMC III system provides a number of different sensors, such as water leak sensor, voltage monitor, temperature sensor, smoke alarm, access sensor and vandalism sensor.

Water leak sensor monitoring will be used as an example in this thesis. It was chosen because it is possible to test it in the real environment. Water leak sensor contains a metal line on the floor which detects water leak (Figure 8). It means, alarm system testing requires only water on the floor.

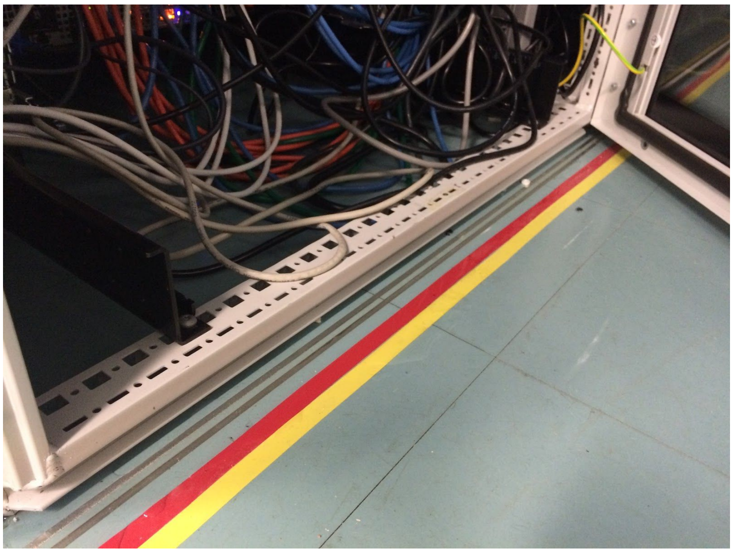


Figure 8: Water leak sensor

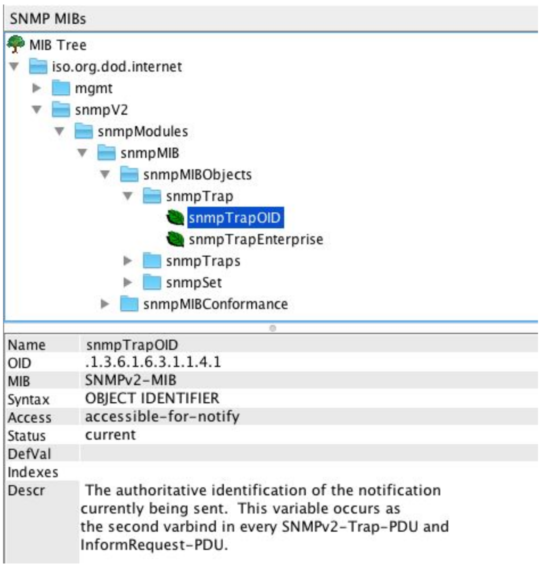
First of all, it is necessary to explain some terms:

SNMP (Simple Network Management Protocol) is an Internet-standard protocol for collecting and organising information about managed devices from

IP networks and modifying that information to change device behaviour. (OID Net-snmp, 2013)

OID (object identifier) is an commonly known instrument for network nodes identification. It is used for network object naming. Structurally, an OID contains numbers of nodes from a tree separated by dots (Figure 9). This algorithm allows to create unique names for every node in a tree. OIDs are contained in MIB database. (OID Repository, 2015)

MIB (management information base) is a formal description of a set of network objects (Figure 9) that can be managed using the Simple Network Management Protocol (SNMP). The format of the MIB is defined as part of the SNMP. (IBM Knowledge Center, 2012)



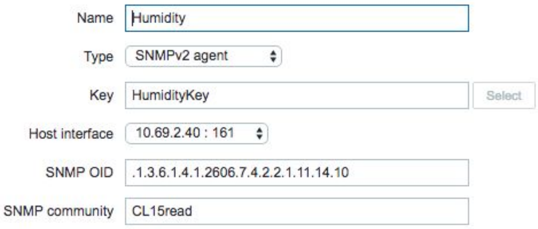
```
SNMP MIBs
MIB Tree
▼ iso.org.dod.internet
    ▶ mgmt
    ▼ snmpV2
        ▼ snmpModules
            ▼ snmpMIB
                ▼ snmpMIBObjects
                    ▼ snmpTrap
                          snmpTrapOID
                          snmpTrapEnterprise
                    ▶ snmpTraps
                    ▶ snmpSet
                ▶ snmpMIBConformance
```

Name	snmpTrapOID
OID	.1.3.6.1.6.3.1.1.4.1
MIB	SNMPv2-MIB
Syntax	OBJECT IDENTIFIER
Access	accessible-for-notify
Status	current
DefVal	
Indexes	
Descr	The authoritative identification of the notification currently being sent. This variable occurs as the second varbind in every SNMPv2-Trap-PDU and InformRequest-PDU.

Figure 9: MIB tree & OID

For monitoring water leak sensor, new Zabbix host should be created in Zabbix monitoring server. It is the same procedure as in case of client computer monitoring. The difference appears in Zabbix item configuration.

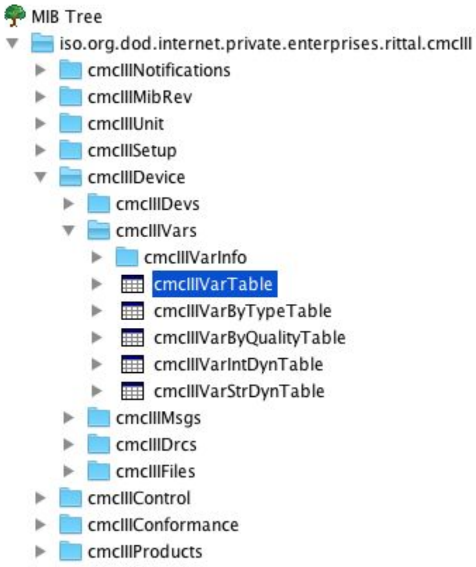
In item configuration, SNMPv2 agent will be used as a type of gathering data from CMC III device (Figure 10). The most difficult part of creating a new item with SNMPv2 agent is finding the right SNMP OID in MIB tree.



Name	Humidity
Type	SNMPv2 agent
Key	HumidityKey
Host interface	10.69.2.40 : 161
SNMP OID	.1.3.6.1.4.1.2606.7.4.2.2.1.11.14.10
SNMP community	CL15read

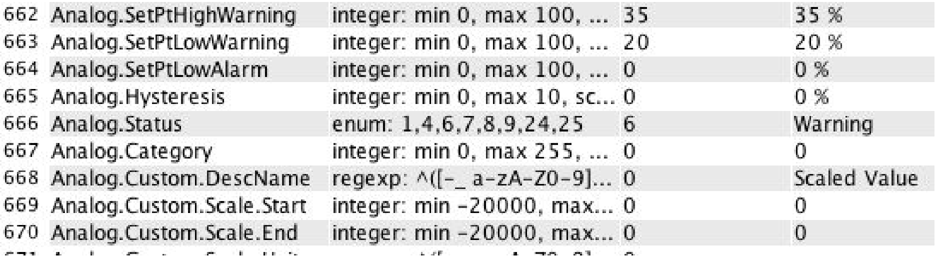
Figure 10: Gathering data from CMC III device

Rital company provides MIB tree database for management of their devices (Figure 11). One of the difficulties was related to finding the appropriate MIB file in Rital website. MIB tree contains tables for monitoring different devices. The second problem was related to choosing the correct value for monitoring. Required variable is called Analog.Status (Figure 12). It is enum variable and it can take a number of values. If value is bigger than 4, alarm notification has to be sent to the system administrator.



```
MIB Tree
▼ iso.org.dod.internet.private.enterprises.rittal.cmcIII
    ▶ cmcIIINotifications
    ▶ cmcIIIMibRev
    ▶ cmcIIIUnit
    ▶ cmcIIISetup
    ▼ cmcIIIDevice
        ▶ cmcIIIDevs
        ▼ cmcIIIVars
            ▶ cmcIIIVarInfo
            ▶ cmcIIIVarTable
            ▶ cmcIIIVarByTypeTable
            ▶ cmcIIIVarByQualityTable
            ▶ cmcIIIVarIntDynTable
            ▶ cmcIIIVarStrDynTable
        ▶ cmcIIIMsgs
        ▶ cmcIIIDrcs
        ▶ cmcIIIFiles
    ▶ cmcIIIControl
    ▶ cmcIIIConformance
    ▶ cmcIIIProducts
```

Figure 11: Rital CMC III MIB tree



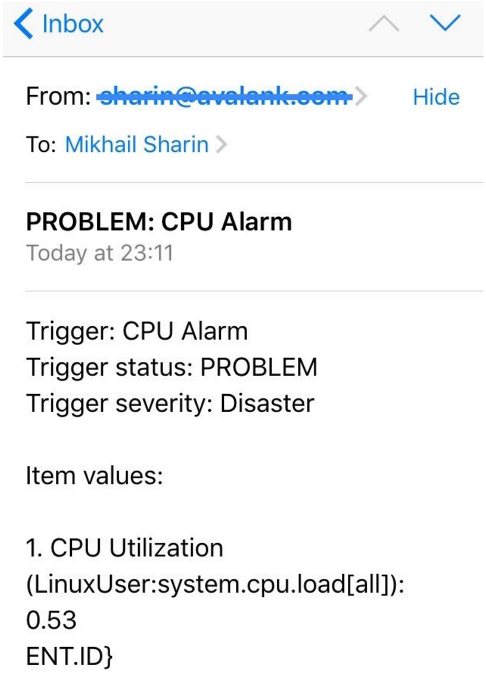
662	Analog.SetPtHighWarning	integer: min 0, max 100, ...	35	35 %
663	Analog.SetPtLowWarning	integer: min 0, max 100, ...	20	20 %
664	Analog.SetPtLowAlarm	integer: min 0, max 100, ...	0	0 %
665	Analog.Hysteresis	integer: min 0, max 10, sc...	0	0 %
666	Analog.Status	enum: 1,4,6,7,8,9,24,25	6	Warning
667	Analog.Category	integer: min 0, max 255, ...	0	0
668	Analog.Custom.DescName	regexp: ^([-_ a-zA-Z0-9]...	0	Scaled Value
669	Analog.Custom.Scale.Start	integer: min -20000, max...	0	0
670	Analog.Custom.Scale.End	integer: min -20000, max...	0	0

Figure 12: MIB tree monitoring values

The next steps of the host configuration are the same as in regular client configuration. It is required to configure a trigger, an action and a graph if it is needed. Finally, in case of emergency situation, notification about water leak will be sent to the system administrator's e-mail address.

5.6. Results of alarm forwarding and Zabbix configuration.

As a result: Zabbix server, two Zabbix clients with agents and water leak sensor are ready for monitoring. Also, Zabbix server is gathering information about client computers CPU utilization and checking ICTLAB sensor. Visual graphs are displaying the current devices' status. Triggers are monitoring dangerous level of indicators. In case of devices' emergency state, an alarm message with a full description will be delivered to the system administrator via e-mail (Figure 13).



From: ~~sharin@avalank.com~~

To: Mikhail Sharin

PROBLEM: CPU Alarm

Today at 23:11

Trigger: CPU Alarm
Trigger status: PROBLEM
Trigger severity: Disaster

Item values:

1. CPU Utilization
(LinuxUser:system.cpu.load[all]):
0.53
ENT.ID}

Figure 13: E-mail from Zabbix

6. CONCLUSION

As a result of the thesis work, the author became acquainted with the monitoring technologies and types such as Manual check, Scheduled check and Problem monitoring; researched reliability levels of problems monitoring:

Internal monitoring, Infrastructure monitoring, Cloud monitoring and External monitoring. Also, a number of the most popular monitoring systems were researched and briefly described. Zabbix monitoring system has been used for practical purposes.

As a practical part, network infrastructure was emulated by Oracle VM VirtualBox. Network nodes were emulated by three virtual machines. Zabbix configuration was explored and Zabbix server was started on Linux Ubuntu VM for monitoring. Two virtual machines with Zabbix agents on Linux Ubuntu VM & Windows 10 Pro and one water leak sensor were monitored by server. Water leak sensor has been provided by KYAMK ICTLAB and located in the server room. It allowed to use Zabbix in real environment and perform water leak test.

As a part of Zabbix configuration, new technologies has been learned by the author: transferring e-mails via SMTP, using MIB database with OIDs, gathering data from network devices via SNMP and other technologies among them.

Configured Zabbix server allowed to gather information from agents and a sensor, store it in a database and create reports and graphs. Also, real time triggers were responsible for determining critical state of monitoring values and notifying the system administrator via e-mail with a full problem description.

7. REFERENCES

- Cacti (2012) Cacti features. Available at: http://www.cacti.net/features.php (Accessed: 25 March 2016)
- Habrahabr (2009) Universalnaya sistema monitoringa zabbix — Vvedenie. Available at: https://habrahabr.ru/post/73338/ (Accessed: 01 April 2016)
- Habrahabr (2014) Monitoring sobytij informacionnoj bezopasnosti s pomoshhyu ZABBIX. Available at: https://habrahabr.ru/post/215509/ (Accessed: 01 April 2016)
- Habrahabr (2016) Vyshel Zabbix 3.0. Available at: https://habrahabr.ru/company/zabbix/blog/277265/ (Accessed: 01 April 2016)
- IBM Knowledge Center (2012) MIB types and objects. Available at: https://www.ibm.com/support/knowledgecenter/SSGU8G_11.70.0/com.ibm.snmp.doc/ids_snmp_050.htm (Accessed: 04 April 2016)
- Kompyuternye seti i texnologii (2007) - Monitoring sistemy i poisk neispravnostej. Available at: http://www.xnets.ru/plugins/content/content.php?content.156.7 (Accessed: 20 March 2016)
- Nagios (2016) Nagios Overview. Available at: https://www.nagios.org/about/overview/ (Accessed: 25 March 2016)
- Net-snmp (2013) Net-snmp. Available at: http://net-snmp.sourceforge.net/ (Accessed: 02 April 2016)
- OID Repository (2015) Object Identifier (OID) repository. Available at: http://www.oid-info.com/ (Accessed: 04 April 2016).
- Opennms (2016) OpenNMS Wiki. Available at: https://wiki.opennms.org/wiki/Main_Page (Accessed: 25 March 2016)
- Oracle VM VirtualBox (2016) User Manual. Available at: https://www.virtualbox.org/manual/ (Accessed: 02 April 2016)
- Rittal (2016) CMC III – Monitoring system Available at: https://www.rittal.com/com-en/product/list.action?c=/System%20accessories/Monitoring/CMC%20III%20%E2%80%93%20Monitoring%20system&categoryPath=/PG0001/PG0900ZUBEHOER1/PG1538ZUBEHOER1/PGR9560ZUBEHOER1 (Accessed: 10 April 2016)

- Yandex support (2016) Setting up email clients. Available at: https://yandex.com/support/mail/mail-clients.xml (Accessed: 09 April 2016)
- Zabbix Documentation 3.2 (2016) Zabbix Manual. Available at: https://www.zabbix.com/documentation/3.2/manual (Accessed: 01 April 2016)