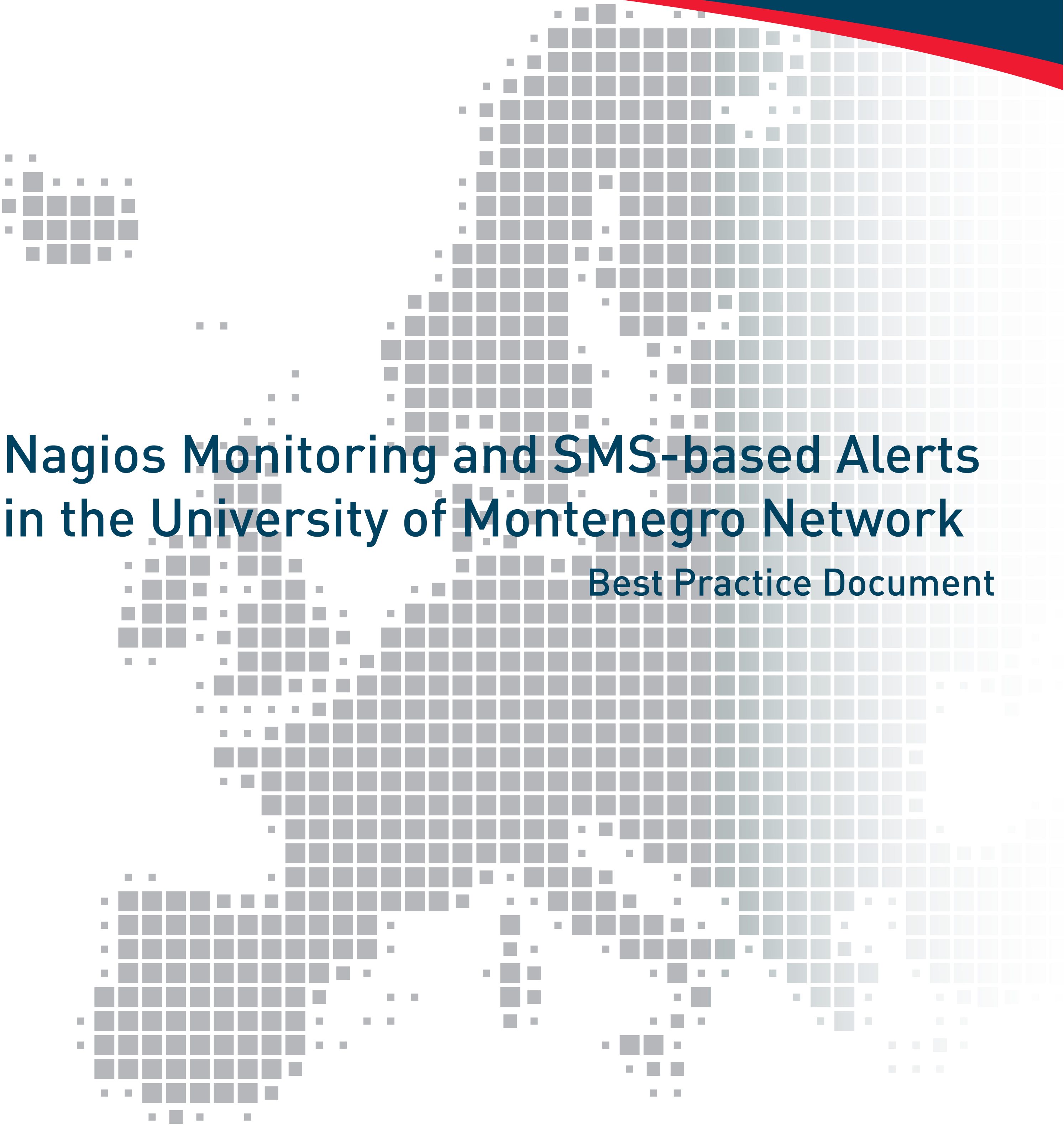
GÉANT
Networks • Services • People



Nagios Monitoring and SMS-based Alerts in the University of Montenegro Network

Best Practice Document

Produced by the MREN-led Campus Networking working group

Authors: Milan Čabak (MREN), Vladimir Gazivoda
(MREN), Božo Krstajić (MREN)

March 2016

Document No:	GN4P1-NA3-T2-MREN006
Version / date:	January 2016
Original language:	Montenegrin
Original title:	"*Nagios monitoring i SMS notifikacije u Akademskoj mreži Univerziteta Crne Gore*"
Original version / date:	Version 1 / 15 January 2016
Contact:	Milan Čabak, milan@ac.me; Vladimir Gazivoda, vladg@ac.me; Božo Krstajić, bozok@ac.me

MREN is responsible for the contents of this document. The document was developed by the MREN-led working group on network monitoring with the purpose of implementing joint activities on the development and dissemination of documents encompassing technical guidelines and recommendations for network services in higher education and research institutions in Montenegro.

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Table of Contents

Summary

The document describes monitoring and system notifications about the status of the servers and services in the academic network of the University of Montenegro, with the help of the Nagios core server and SMS servers.

The installation and configuration of the Nagios core, NRPE and SMS servers, monitoring Linux servers and sending notifications are contained in the document.

Furthermore, the document will present the procedure for monitoring Ubuntu Linux servers, monitoring server Linux service and sending appropriate notifications to administrators.

Emphasis is placed on monitoring and sending notifications.

1 Introduction

The need for high availability of servers and services stems from their importance in daily business activities and linkages with the work.

Using tools for monitoring servers and services is of great importance for the smooth functioning of the system as a whole and providing high availability. The system and the network administrators for the adequate administration of the system use different monitoring tools. Monitoring tools follow a large number of system parameters and notify the administrators about tracked changes.

The academic network of the University of Montenegro uses Nagios core server for monitoring servers and services. The document describes a complete installation and configuration of the Nagios core server, NRPE server, and SMS server, as well as tracking system parameters and sending notifications.

2 Installation and configuration of the Nagios core server

For the installation and configuration of the Nagios core server in the next steps Ubuntu server 14.04 LTS Linux distribution is selected.

In the academic network, a large percentage of servers and services run on Ubuntu server Linux distribution. A detailed procedure of the installation and configuration of the Nagios core server will be discussed in the subsequent chapters.

2.1 Preparing to install the Nagios server

Before you install the Nagios server, it is necessary to install LAMP server [1]. Basic installation includes running the following command:

```
# sudo apt-get install lamp-server^
```

The installation process requires you to type the password for the MySQL "root" user. After entering the password, the Web server is running and you can run it with the command:

```
# sudo service the Apache2 restart
```

Before installing the Nagios core server, you need to create a new user and group on the server. Commands include creating a new user "Nagios" and group "nagcmd", as well as adding "Nagios" user to "nagcmd" group:

```
# sudo useradd Nagios
# sudo groupadd nagcmd
# sudo usermod -a -G nagcmd Nagios
```

The last step prior to the Nagios server installation process includes the installation of additional packages, required for the operation of the Nagios server. It is necessary to update the existing repository of available packages, and then install the packages required for the operation of the Nagios core server:

```
# sudo apt-get update
# sudo apt-get install build-essential libgd2-xpm-dev openssl libssl-dev xinetd the Apache2-utils unzip
```

To receive notifications from the Nagios server via an email message, it is necessary to install "mailutils” package and then perform the basic settings of "postfix" package:

```
# sudo apt-get install mailutils
```

2.2 Installing Nagios core server

The process of installing the Nagios server starts with downloading the installation package from the official Nagios website [2]:

```
# sudo wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
```

Then you need to unpack the downloaded package:
# tar xvf nagios-4.1.1.tar.gz

Open the unpacked directory:
# cd nagios-4.1.1/

Configuring the Nagios core server requires performing the command which specifies the previously defined user and group, as well as how to send notifications by using the previously installed "postfix" package:
# ./configure --with-Nagios-group=Nagios --with-command-group=nagcmd --with-mail=/usr/bin/mail

In the current directory perform the compiling of the package and the commands listed:
# make all

The installation of the Nagios core servers and HTML files is run with the command:
# sudo make install

Configuring permissions on directories for storing files with external commands:
# sudo make install-commandmode

Setting the init script in "/etc/init.d" directory to run Nagios service:
# sudo make install-init

Installing the display configuration files on location "*/usr/local/Nagios/etc*":
# sudo make install-config

Setting apache configuration file for access to the Nagios Web interface:
# sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf

The final step when installing the Nagios core server involves adding a new user to manipulate and configure the Nagios server by using the Web interface:
# sudo usermod -G nagcmd www-data

2.3 Installing the supporting core Nagios plugins

The process of installing Nagios core plugins starts with downloading the installation package from the official Nagios website [3]:
# sudo wget http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz

Then you need to unpack the downloaded package:
# tar xvf nagios-plugins-2.1.1.tar.gz

Open the unpacked directory:
# cd nagios-plugins-2.1.1/

Configuring the Nagios plugin package requires performing the command that defines the user, group, and using security openssl protocol:

```
# ./configure --with-Nagios-user=Nagios --with-Nagios-group=Nagios --with-openssl
```

In the current directory compile the package with the command:

```
# make
```

Run the installation of the Nagios plugin package with the command:

```
# sudo make install
```

2.4 Installing NRPE service on Nagios core server

NRPE allows monitoring Linux server resources, such as CPU, memory, storage space, and other parameters.

The process of installing Nagios NRPE services starts with downloading the installation package from the official Nagios website:

```
# sudo wget http://kent.dl.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
```

Then you need to unpack the downloaded package:

```
# tar xvf nrpe-2.15.tar.gz
```

Open the unpacked directory:

```
# cd nrpe-2.15/
```

Configuring the Nagios NRPE service requires performing the command:

```
# ./configure --enable-command-args --with-Nagios-user=Nagios --with-Nagios-group=Nagios --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86_64-linux-gnu
```

In the current directory compile the package with command:

```
# make all
```

To install the Nagios NRPE service and supporting scripts run the following commands:

```
# sudo make install
# sudo make install-xinetd
# sudo make install-daemon-config
```

Modify xinetd script and add the address of the Nagios core server:

```
# nano /etc/xinetd.d/nrpe
only_from = 127.0.0.1 192.168.1.10
```

Save the changes and restart xinetd service:

```
# sudo service xinetd restart
```

2.5 Configuring Nagios server

After the installation of the Nagios core, server side packages, and services, it is necessary to perform basic configuration of the server.

In the basic Nagios core server configuration file, you need to enable access to the directory that contains configuration files from the monitored remote servers, by removing the comments from the lines below:

```
# sudo nano /usr/local/Nagios/etc/nagios.cfg
cfg_dir=/usr/local/Nagios/etc/servers
```

Save changes and create the corresponding directory in the specified location to store the configuration files for future servers whose parameters will be monitored using Nagios core server:

```
# sudo mkdir /usr/local/Nagios/etc/servers
```

The next configuration step involves modifying contact address, i.e. defining the person(s) who will be alerted with a message about the condition of the system that is being monitored. The following chapters will describe a more detailed configuration.

You need to modify the following configuration file and to save changes:

```
# sudo nano /usr/local/Nagios/etc/objects/contacts.cfg
email     mail@domen
```

Having installed and configured the NRPE service in the previous step, you need to modify the Nagios configuration file with commands and to add the following lines at the end of the cfg file:

```
# sudo nano /usr/local/Nagios/etc/objects/commands.cfg
define command{
        command_name        check_nrpe
        command_line  $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

The additional lines define “check_nrpe” command which will be used to check the service status on the servers that are monitored using Nagios core server.

Save the changes to "commands.cfg" file.

2.6 Configuring the Apache server

The next configuration step involves configuring the Apache server installed in the previous installation steps.

Configuration involves enabling rewrite and cgi modules with commands:

```
# sudo a2enmod rewrite
# sudo a2enmod cgi
```

You need to create a new "nagiosadmin” user to access the Web interface:

```
# sudo htpasswd -c /usr/local/Nagios/etc/htpasswd.users nagiosadmin
```

Creating a symbolic link and activating the Nagios Web interface:

```
# sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/
```

Start Nagios service with command:

```
# sudo service Nagios start
```

Restart the Apache service with command:

```
# sudo service the Apache2 restart
```

Access to the Nagios server can be restricted only on the local network or a specific IP address. It is possible to limit access by modifying the Apache cfg file:

```
# sudo nano /etc/apache2/sites-available/nagios.conf
```

Comment the following two configuration lines:

```
# Order allow,deny
# Allow from all
```

Remove the comment from the following configuration lines and add network or IP address which will be allowed access to the Nagios Web interface:

```
Order deny,allow
Deny from all
Allow from 127.0.0.1 192.168.1.0/24
```

Save changes and restart Nagios and the Apache services:

```
# sudo service Nagios restart
# sudo service the Apache2 restart
```

Access to the Nagios server Web interface is possible via an Internet browser at the address:
http://192.168.1.10/nagios

To automatically start the Nagios service when you restart the server you need to perform the following command:

```
# sudo ln -s /etc/init.d/Nagios /etc/rcS.d/S99nagios
```

3 Installation and configuration of the client server

After the installation and configuration of the Nagios core server, you need to install and configure the NRPE server on the client. The client is a server whose services are monitored.

3.1 Installation and configuration of NRPE server

NRPE server installation and configuration steps are shown on the example of Ubuntu Linux server distribution. NRPE server checks the parameters specified by the client server and the information on the status of the service is passed to the Nagios core server on request.

3.1.1 Installing NRPE server on the client

Update software repository:

```
# sudo apt-get update
```

The first step requires the installation of the Nagios NRPE server and the supporting plugins:

```
# apt-get install -y Nagios-plugins Nagios-nrpe-server
```

3.1.2 Configuring NRPE server on the client

After you have installed the NRPE server on the client it is necessary to perform its configuration. Configuration involves defining the parameters that will monitor the services on the client server.

Modify the NRPE configuration file:

```
# nano /etc/Nagios/nrpe.cfg
```

For "server_address" parameter in the configuration file set the address of the client server:

```
server_address=192.168.1.30
```

Modify the parameter "allowed_hosts" and add the address of the Nagios core server:

```
allowed_hosts=127.0.0.1,192.168.1.10
```

To check the availability of storage space on the server it is necessary to modify the existing command "check_hda1" and add the name of the "root" file system at the end of the command. The name is checked with command *"df -h"*:

```
command[check_hda1]=/usr/lib/Nagios/plugins/check_disk -w 20% -c 10% -p /dev/mapper/server-root
```

Checking RAM and swap memory requires an entry for the next commands for checking the memory on the server:

```
command[check_swap]=/usr/lib/Nagios/plugins/check_swap -w 10% -c 5%
command[check_memory]=perl /usr/lib/nagios/plugins/check_memory.pl -w 50% -c 25%
```

3.1.3 Plugin for checking RAM memory

The default plugins installed with the previous command on the client server do not include a plugin for checking RAM memory [4]. Previously added command for checking the RAM requires an additional plugin to check "check_memory.pl".

The procedure for the installation of the additional plugin involves downloading from the Internet:

```
# wget https://exchange.icinga.org/exchange/check_memory/files/784/check_memory.pl
```

Then you need to move the downloaded plugin into the appropriate directory:

```
# mv check_memory.pl /usr/lib/nagios/plugins/check_memory.pl
```

Add the execution right:

```
# chmod +x /usr/lib/nagios/plugins/check_memory.pl
```

Install the pearl module required to execute the downloaded plugin:

```
# apt-get install libnagios-plugin-perl
```

Test the download plugin command:

```
# perl /usr/lib/nagios/plugins/check_memory.pl -w 50% -c 25%
```

The result of the test command:
“CHECK_MEMORY OK – [...] free [...]“.

On the server client you should specify the [check_memory] command, creating "memory.cfg" cfg file:

```
# sudo nano /etc/Nagios-plugins/config/memory.cfg
```

Add the following content in the configuration file:

```
# 'check_memory' command definition
define command{
        command_name        check_memory
        command_line        perl    /usr/lib/nagios/plugins/check_memory.pl    -w    $ARG1$    -c
$ARG2$
        }
```

After the installation and configuration of NRPE server it is necessary to restart the process on the client:

```
# sudo service Nagios-nrpe-server restart
```

3.1.4 Configuring UFW *firewall*

In case the server uses UFW firewall you need to pass the appropriate port from the IP address of the Nagios core server:

```
# ufw allow from 192.168.1.10 to any port 5666 proto tcp
```

3.2 Application for monitoring windows server

For monitoring windows server you need to download the client application from the following location [5]:
http://nsclient.org/nsclient/

4 Monitoring Linux server

After you install the NRPE server on the client server, you need to create a configuration file for the client server on the Nagios core server. The configuration file contains information about the client server and the services whose parameters are monitored.

4.1 Creating configuration file on the Nagios core server for client servers

On the Nagios core server you need to create configuration files for each client server that is monitored:

# sudo nano /usr/local/Nagios/etc/servers/client.cfg

Each configuration file should contain a description of the client that is monitored in the following form:

```
define host {
        use                     Linux-server            ; Used template
        host_name               client                  ; Client name
        alias                   Client server           ; Description
        address                 192.168.1.30            ; IP address
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
```

The same configuration file should contained the configurations of the service that is monitored. An example of most commonly used services is given. The first example shows the client server response to the "ping" command. The service checks the availability of the server on the network:

```
define service {
        use                     generic-service
        host_name               client
        service_description     PING
        check_command           check_ping!100.0,20%!500.0,60%
}
```

Services for checking the availability of the client server via SSH connection and HTTP Web access:

```
define service {
        use                     generic-service
        host_name               client
        service_description     SSH
        check_command           check_ssh
}
```

```
define service{
        use                     generic-service
        host_name               client
        service_description     HTTP
        check_command           check_http
}
```

Earlier defined services are performed directly on Nagios core server. The following services need to be defined on the client server. Detailed configuration is described in the previous chapter 2.1. For the services below you need to use the command 'check_nrpe' followed by the name of the service on the client server:

```
define service{
        use                     generic-service
        host_name               client
        service_description     RAM
        check_command           check_nrpe!check_memory
        }
define service {
        use                     generic-service
        host_name               client
        service_description             Current Users
        check_command           check_nrpe!check_users
}
define service {
        use                     generic-service
        host_name               client
        service_description     Current Load
         check_command          check_nrpe!check_load
}
define service {
        use                     generic-service
        host_name               client
        service_description     Root Partition
        check_command           check_nrpe!check_hda1
}
define service {
        use                     generic-service
        host_name               client
        service_description     Total Processes
        check_command           check_nrpe!check_total_procs
}
define service{
        use                     generic-service
        host_name               client
        service_description     Swap
        check_command           check_nrpe!check_swap
}
```

Save the changes and restart the Nagios server:

```
# sudo service Nagios reload
```

The names of the commands correspond to the services that are checked on the client server. According to the parameters defined in the configuration file "max_check_attempts", "check_period" mentioned services will be monitored by the Nagios core server. The parameters "notification_interval", "notification_period" define the frequency and time of sending notifications of changes to service status. The next chapter describes the process of sending notifications via email and SMS messages.

5 Notifications about the status of the client server and service

Notifications about the status of client server and services are supplied to the server administrators. The notifications contain basic information about the service name and its current status. Notify the administrator via email and SMS messages.

5.1 Mail notifications

For successful and timely notification to the administrator via e-mail messages, you need to follow the steps already described in chapters 1.1 and 1.2, which describe the installation of the mail client and integration with Nagios core server.

The next step is to define the contact and the group which will receive notifications via mail messages for clients. It is necessary to modify the file:

```
# sudo nano /usr/local/Nagios/etc/objects/contacts.cfg
```

Define new contacts:

```
define contact{
        contact_name        nagiosadmin         ; dministrator name
        use                 generic-contact     ; using template
        alias               Nagios Admin        ; Full user name
        email               mail@domain         ; Mail address
}
```

The template is located on the site "/usr/local/Nagios/etc/objects/templates.cfg" and contains the parameters of time frames and how to notify the administrator. Data defined in the template can be specifically defined for each administrator by specifying the parameters when creating a new contact.

Besides defining individual contacts, you can create groups in which contacts will be listed. Define new groups:

```
define contactgroup{
        contactgroup_name       admins                  ; Group name
        alias                   Nagios Administrators   ; Description
        members                 nagiosadmin, admin      ; Contacts
        }
```

In the previous chapter 3.1, when defining the client you need to modify the file "client.cfg" and add the following two lines:

```
define host {

contacts                nagiosadmin             ; Contact
contact_groups          admins                  ; Group
}
```

By specifying the name of the contacts and/or group all members will be notified via mail message about the status of the server and/or services that are monitored.
To implement SMS notifications first you need to perform additional configuration steps.

5.2 SMS notifications

SMS notifications are system of notifications through mobile networks and SMS messages. An administrator can receive notification of change in the status of the server and/or service in their phone.

Timely notification and troubleshooting problems is of great importance for certain services. By modifying the "templates.cfg" cfg file it is possible to add SMS templates for contacts and services. A separate template for SMS notifications is created when it is necessary to modify the method of notification system in relation to mail notifications.

Edit:

```
# sudo nano /usr/local/Nagios/etc/objects/templates.cfg
```

Add a separate SMS contact template:

```
define contact{
name                          SMS-contact
service_notification_period   24x7
host_notification_period      24x7
service_notification_options  c,r
host_notification_options     d,r
service_notification_commands notify-service-by-email, notify-service-by-SMS
host_notification_commands    notify-host-by-email, notify-host-by-SMS
register                      0
}
```

The added template for SMS contacts in the example above is changed in the notification method for services and hosts so that notifications are sent only in the specified cases (c-critical, d-down, r-recovery). You should define commands for notifications "host-notify-by-SMS" and "notify-service-by-SMS" which will be described below in more details.

As in the previous example, it is possible to create a separate template for the services whose changes will be tracked via SMS notifications. All the parameters from the template can be adjusted with SMS notifications:

```
define service{
name                          SMS-service
active_checks_enabled         1
passive_checks_enabled        1
parallelize_check             1
obsess_over_service           1
check_freshness               0
notifications_enabled          1
event_handler_enabled          1
flap_detection_enabled         1
failure_prediction_enabled     1
process_perf_data              1
retain_status_information      1
retain_nonstatus_information   1
is_volatile                    0
check_period                   24x7
max_check_attempts             3
normal_check_interval          10
retry_check_interval           2
contact_groups                 SMS
notification_options           w,u,c,r
notification_interval          60
notification_period            24x7
register                       0
}
```

After entering SMS template and customization of SMS notifications, you should define the contacts that will receive this type of information.

Edit configuration file:
# sudo nano /usr/local/Nagios/etc/objects/contacts.cfg

Add a new contact:

```
define contact{
contact_name                   Nagios         ; Name
use                            SMS-contact    ; Downloading defined values from SMS template
alias                          Nagios Admin   ; Full user name
service_notification_period    workhours      ; Notification period for services
host_notification_period       workhours      ; Notification period for servers
email                          nagios@domain; email
pager                          38269123456    ; Phone number
}
```

In the example of adding a new contact a created SMS template was used. The parameters "service_notification_period" and "host_notification_period" use values that will be defined later in this document. The values of these parameters change the values previously defined in template.

Contacts can be grouped in a separate SMS group. All members of the SMS group, if they use "SMS-contact" template, will receive SMS notifications.

Add a new group:

```
define contactgroup{
contactgroup_name              SMS
alias                          SMS Group
members                        nagios,nagiosadmin
}
```

In the example of adding a new contact a "workhours" notification period is defined. Time frames where notification sending period is adjusted are located in "timeperiods.cfg" configuration file.

Edit:

```
# sudo nano /usr/local/Nagios/etc/objects/timeperiods.cfg
```

In the configuration file change the working hour time frame:

```
# 'workhours' timeperiod definition
define timeperiod{
timeperiod_name          workhours
alias                    Normal Work Hours
Monday                   08:00-16:00
Tuesday                  08:00-16:00
Wednesday                08:00-16:00
Thursday                 08:00-16:00
Friday                   08:00-16:00
}
```

Contact that uses "workhours" timeline of changes to the system will be informed only in the specified period.

On the Nagios core server, install the console Web browser "*lynx*":

```
# sudo apt-get install lynx-cur
```

Using the lynx browser the parameters that are sent by SMS messages are forwarded to SMS server. It is necessary to change the "commands.cfg" configuration file and add commands for sending SMS messages
"*notify-host-by-SMS*" and "*notify-service-by-SMS*":

```
# sudo nano /usr/local/Nagios/etc/objects/commands.cfg
```

Add commands to forward notification message parameters to SMS server:

```
# 'notify-host-by-SMS' command definition
define command{
command_name             notify-host-by-SMS
command_line             lynx
"http://SMS.ac.me/send.php?code=12345&number=$CONTACTPAGER$&message=$HOSTNAME$"%20"$SERVICESTATE$"%20"tip:"%20"$NOTIFICATIONTYPE$"%20"servis:"%20"$SERVICEDESC$"%20"$SERVICEOUTPUT$"
}

# 'notify-service-by-SMS' command definition
define command{
command_name             notify-service-by-SMS
command_line             lynx
"http://SMS.ac.me/send.php?code=12345&number=$CONTACTPAGER$&message=$HOSTNAME$"%20"$SERVICESTATE$"%20"tip:"%20"$NOTIFICATIONTYPE$"%20"servis:"%20"$SERVICEDESC$"%20"$SERVICEOUTPUT$"
}
```

6 SMS server

For the implementation of SMS server "Kannel" WAP and SMS gateway open code software are used, as well as HSDPA modem with SIM card. SMS server is implemented as a separate server with Ubuntu server operating system.

6.1 Installing kannel gateway software

The process of installing "Kannel" WAP and SMS gateway software starts by checking whether the required library is installed:

```
# dpkg -l libxml2
```

If there are no required packages you need to install them:

```
# sudo apt-get install libxml2
```

The installation of kannel service is performed with the command:

```
# sudo apt-get install kannel
```

Kannel service can be managed with the commands:

```
# sudo service kannel start
# sudo service kannel stop
# sudo service kannel restart
```

6.2 Configuring kannel gateway software

Configuring gateway kannel software involves modifying cfg file:

```
# sudo nano /etc/kannel/kannel.conf
```

All configuration lines in 'kannel. conf' file are described in detail in Kannel user manual [6].

For SMS Kannel configuration setup it is necessary to define multiple groups, as follows: *core, SMSc, SMSbox, SMS-service and sendSMS-user* [7].

The configuration file can have only one core group that contains basic settings:

```
group = core
admin-port = 13000
SMSbox-port = 13001
admin-password = 12345
box-deny-ip = "*.*.*.*"
box-allow-ip = "127.0.0.1"
```

The next set of options defines SMS centre that will send SMS messages through a USB modem:

```
group = SMSc
SMSc = at
```

```
modemtype = ZTE_CDMA_Technologies_MSM
device=/dev/ttyUSB2
my-number = +38267123456
SMS-center = +38267100100
connect-allow-ip = 127.0.0.1
log-file = "/var/log/kannel/modem.log"
log-level = 0
```

Kannel requires defining an SMS box that represents the module in charge of SMS traffic and management of SMS messages that have been received by the SMS center:

```
group = SMSbox
bearerbox-host = 127.0.0.1
sendSMS-port = 13013
global-sender = +38267123456
log-level = 0
```

After defining SMS centre and SMS box parameters, the next step is to enter parameters for SMS service:

```
group = SMS-service
keyword = default
get-url = "http://localhost/service?msg=%q"
max-messages = 0
```

The next set of settings defines the user who can send SMS messages:

```
group = sendSMS-user
username = cis
password = 12345
concatenation= true
max-messages = 10
dlr-url= "http://localhost/service?status=%a"
```

The last set of commands in the configuration file defines SMS modem parameters:

```
group = modems
id = ZTE_CDMA_Technologies_MSM
name = "ZTE CDMA Technologies MSM"
detect-string = "ZTE"
init-string = "AT+CNMI=2,1,2,2,0"
```

When the server restarts in case the modem is not automatically initialized, you need to create a script that will automate this process:

The script is created in "/etc/init.d/" directory with the following commands:

```
# sudo nano /etc/init.d/kannelstart.sh
```

```
#!/bin/bash
sleep 15
chmod a+rw /dev/ttyUSB2
service kannel restart
sleep 10
```

```
service kannel restart
```

Assign the execution right to the created script:

```
# chmod a+x /etc/init.d/kannelstart.sh
```

Then you need to execute a command to automatically run the script when the server restarts:

```
# update-rc.d /etc/init.d/kannelstart.sh defaults
```

6.3 Script for sending SMS messages

Script for sending SMS messages takes over the parameters 'code', 'number' and 'message', which have been forwarded by Nagios core server, when defining 'notify-host-by-SMS', 'notify-service-by-SMS' commands, and sends SMS notification on the basis of the received parameters.

Sending SMS messages is performed by Kannel gateway which has been previously installed and configured on the SMS server.

The script is called from the '/var/www' directory, and the send.php script code is given in the following example:

```
<?php
function sendNotification($pwd,$number,$message)
{
$codedMessage=urlencode($message);
$url="http://localhost:13013/cgi-
bin/sendSMS?username=cis&password=".$pwd."&to=00".$number."&text=".$codedMessage."&dlr-
mask=31";
sleep(5);
$ch= curl_init();
curl_setopt($ch,CURLOPT_URL,$url);
curl_exec($ch);
curl_close($ch);
}

$pwd=$_GET["code"];
$number=$_GET["number"];
$message=$_GET["message"];
sendNotification($pwd,$number,$message);
?>
```

Calling the script sends the SMS notification about the change in status of the server or the service monitored on Nagios core server, to the administrators with the aim of timely notification and troubleshooting problems.

7 Conclusion

The implementation of the described monitoring tools has improved the availability of the academic network of the University of Montenegro (AMUCG) server. Servers in AMUCG are predominantly based on Ubuntu Linux server distribution. Ubuntu server distribution was selected because of proven stability, long-term support (LTS), ease of administration and the availability of numerous free tools.

Timely notifications of changes to the system are crucial for increasing its availability. In order that the availability of the servers and services should be at a satisfactory level it is necessary to implement monitoring and information systems described herein. The recommendations include the installation and configuration of the Nagios core, NRPE, and SMS servers.

The document describes an example of monitoring a system with the installation and configuration of the Nagios core and NRPE server. It also describes and provides recommendations for the realization of the notification system by email and SMS messages.

References

[1] Mixeduperic.com, "*How to Install LAMP, phpMyAdmin and Collabtive on Ubuntu 10.04 Server*", http://mixeduperic.com/ubuntu/how-to-install-lamp-phpmyadmin-and-collabtive-on-ubuntu-1004-server.html#collabtive

[2] Nagios core, Nagios, "*Ubuntu Quickstart*", https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart-ubuntu.html

[3] Mitchell Anicas, Digital Ocean, "*How to Install Nagios 4 and Monitor Your Servers on Ubuntu 14.04*", https://www.digitalocean.com/community/tutorials/how-to-install-nagios-4-and-monitor-your-servers-on-ubuntu-14-04

[4] Christosoft, en.christosoft.de, "*Nagios / Icinga: Monitor (local) memory usage*", http://blog.christosoft.de/2013/01/nagios-icinga-memory-usage/

[5] NSClient++, http://nsclient.org/nsclient/

[6] Kannel.org, "*Kannel 1.4.4 User's Guide*", http://www.kannel.org/download/1.4.4/userguide-1.4.4/userguide.html#AEN189

[7] Stevan Šandi, Specijalistički rad, "*Primjer SMS notifikacije i upravljanja računarskim sistemom*", Elektrotehnički fakultet, Podgorica, jul 2013. godine

List of abbreviations

AMUCG	Akademska mreža Univerziteta Crne Gore
HSDPA	High-Speed Downlink Packet Access
HTML	Hyper Text Markup Language
LAMP	Linux, the Apache, MySQL, PHP
LTS	Long Term Support
MySQL	Structured Query Language
NRPE	Nagios Remote Plugin Executor
SIM	Subscriber Identity Module
SMS	Short Message Service
WAP	Wireless Application Protocol

Complete BPDs are available at http://services.geant.net/cbp/Pages/Home.aspx

campus-bp-announcements@geant.org