Computer Networks Resources Monitoring

M. Matýsek, M. Kubalčík, and M. Mihok

***Abstract*—**In case that a monitoring supervisory system is deployed, then system administrators mostly know about an arisen problem almost immediately. They have it precisely localized and they can it quickly remove. In most cases then it is not necessary to negotiate with a user who may not be an expert in information technology. Of course, deployment of a monitoring system sharply reduces the time of availability and reduces damages which can be consequently caused.

***Keywords*—**Nagios, monitoring system, computer network, SNMP protocol, SAP, Linux, Microsoft Windows.

I. Introduction

AT the present electronic age there is probably no company or organization which can do without information technology. Most people perceive the computer as a tool for sending emails, surfing the Internet and using of the office application software. More technical skilled people use the computer for example for programming applications such as manufacturing tools or use the computers for data collection.

If a surveillance monitoring system is applied, most system administrators know about the problems that arise almost immediately. They have it exactly and immediately located and may solve it quickly. In most of cases then it is not necessary to argue with the user which largely is not an expert in the area of information technologies. Of course, the deployment of the monitoring system sharply reduces time of unavailability of systems and consequently caused damages. With help of these systems it is possible to prevent many problems such as stopping the server due to overfilling the disk array. This problem can be detected even earlier than it ever occurs.

With an appropriate monitoring system it can be supervised even non-critical devices such as network printers that are able to send information about the remaining amount of toner in its cartridge or information about the need of repairs.

If a company has a large number of systems and equipment, the deployment of a monitoring system is necessary. If the company also needs to save money, then the ideal choice is utilization of free software. One of the many options is the systems Nagios, Other possibilities are systems such as Zabbix or Cacti [1], [2].

A typical example of a network administrator working day:

It is ten o'clock on Monday morning. Branch manager is furious because he is waiting for an important email which has not been delivered yet. The administrator finds by fast control that the messages are not stuck in the queue. There is also no reference in the log file and the email from the sender has arrived. So where's the problem? The central mail server is not also responding to the program Ping. This is probably the merit of the problem. But the IT department insists on the fact that the situation is not their fault and that the network is running properly at the headquarters and that the problem must be in the branch network. Searching of the error continues and finally it is found that that the VPN line to the head office was not operational because the back-up line did not set up routing rules. The final result is a lot of minutes spent on finding errors, edgy director (the action for which the email was necessary already expired) and sweaty administrator.

II. Monitoring Tools

Currently there are many tools for monitoring of computer networks available on the market. Particular tools differ mainly in their design and consequently in price. A lot of commercial and professional solutions exist but these induce a large initial investment for most of companies. They rather search for available free products instead of it. But this is to the prejudice of accuracy and limited usability of the products.

A. Basic diagnostic software tools

Software tools are described as tools with large possibilities of creating their configurations. They are mostly fixedly bound to a particular type of an operating system. Many of them are designed for specific purposes e.g. diagnostics of security, exploitation of system resources etc. Probably in each operating system exists basic diagnostic tools which help to the administrator to evaluate quickly, simply and accuratly qualitative parameters of the computer network.

B. Ping

By using this standard command of the TCP/IP (Transmission Control Protocol/Internet Protocol) it is easy to detect and solve problems of availability and connectivity. By means of *ping* it is possible to easily test the availability of a computer, server or other network device not only in a particular local network but, if the Internet is available, anywhere in the world.

F. M. Matýsek is with the Department of Computer and Communication System, Tomas Bata University in Zlín 76005, Zlín, Czech Republic (corresponding author to provide phone: +420 576035205; fax: +420 576035279; e-mail: matysek@ fai.utb.cz).

S. M. Kubalčík, was with the Department of Process Control, Tomas Bata University in Zlín, 76005 Zlín, Czech Republic (e-mail: kubalcik@fai.utb.cz).

T. M. Mihok is with the Department of Computer and Communication System, Tomas Bata University in Zlín 76005, Zlín, Czech Republic (e-mail: mirdam@volny.cz).

The ping command sends an ICMP echo request in order to ask if the computer is accessible. If so, then the interviewed targed host sends a reply by the ICMO echo reply.

It is apparent form Fig. 1 that the size of the sent packet is 32 bytes, the minimum response time 10 ms and the maximum response time 12ms. The value of TTL (time to live) is 248 which means that the packet can pass through even 248 routers before it is discarded.

Every time the packed passes a router the value of TTL is decreased by one which ensures that the packet will not worthlessly wander inside the network in a loop.

ICMP is a service protocol and it is a part of the IP protocol. It is used for signaling of emergencies in computer networks which are based on the IP protocol.

ICMP can indicate a variety of situations but the fact is that a particular implementation supports only a particular portion of the signals. Moreover, for safety reasons, the signals can be discarded on ICMP routers.

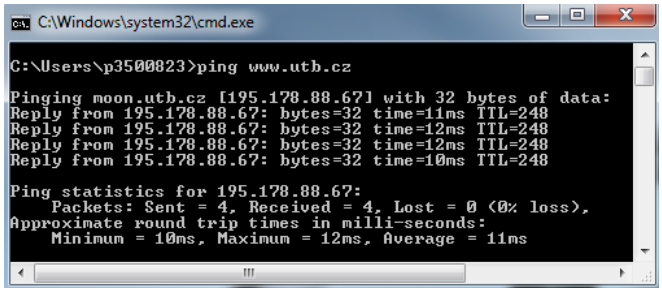

Fig. 1 The output of executing a "ping www.utb.cz"

C. Tracert

The tracert program is one of the other utilities of the TCP/IP protocol. It enables tracking the path which the packets follow to the destination host.

As a test packet for Windows is used the ICMP packet. For Unix the UDP (User Datagram Protocol) packet is used.

The source gradually sends three packets. If these packets reach the host then ICMP sends “time exceeded”, TTL is reduced by one and the packet are further forwarded. In case of the target host ICMP echo reply is sent

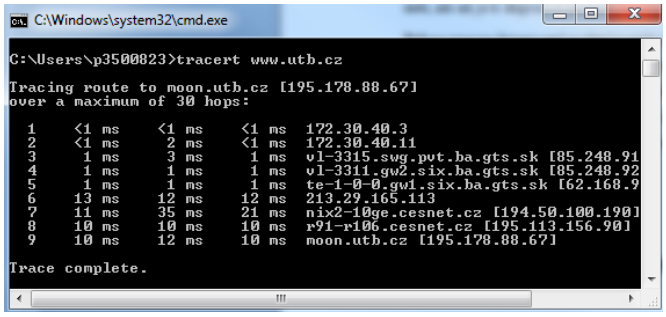

Fig. 2 The output of executing a "tracert www.utb.cz"

Means route list interface routers in the path between the source and destination. As the path we understand a list of routers interface between the source and the target.

When a star appears in the time field of the printout it indicates that the node cannot communicate.

D. Hardware Tools

An advantage of hardware tools is that they can operate separately. They are not dependant on other devices while providing the same capabilities as the software tools. They perform operations on lower levels of communication in computer networks such as tracking of time and volume characteristics of the packet and its classification according to the protocol or other data and consequent processing on higher levels such as long-term statistics claculation or recognizing of possible security attacks according to the content of selected filtered packets.

Necessary hardware monitoring tolls are nowadays available from several manufacturers. Their disadvantage is a high purchase price.

III. Nagios Monitoring System

A. System Description

The development of the program began in 1999. The original name of the project was Netsaint. It was finished in 2002 and it further continued under a new name - Nagios. The author is Mr. Ethan Galstad, who is currently also the president of Nagios Enterprises [3].

It is a very popular monitoring system. This fact also confirms a symposium which was released on the discussion forum for fans of Linux distributions.

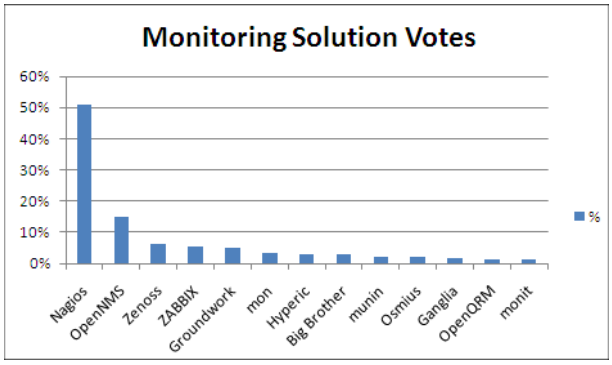

Fig. 3 Poll of the best monitoring tool

B. Requirements on Monitoring System - Requirements on functionality

1) Monitoringof services and states of operating systems.
2) Possibility of testing of network services availability.
3) Measurement of values of important parameters of network services.
4) Possibility of running a network monitoring service on request.

C. Requirements for Monitoring System - Reportingof problems

1) In case of arising problem the administrator will be notified by e-mail.
2) In case of a more serious problem the administrator will be notified by SMS (Short Message Service).
3) The report about a problem will be supplied by sufficient amount of relevant information.
4) Possibility of definition of contact groups.

D. Requirements for Monitoring System - Administration and Management

1) Administration and visualization through a web interface.
2) Ability to define the user management web interface with different privileges.

E. Hardware Requirements

According to discussion forums on the Internet the system with a standard dual core processor and 5 GB of RAM is able to process 2000 services per minute [4].

F. Software Requirements

1) Web server such as Apache and more.
2) PHP 4.3 and more.
3) MySQL 4.1.
4) PEAR Module: HTML_Template_IT 1.1 and more.
5) PHP Extension: gettext.
6) PHP Extension: mysql.
7) PHP Extension: ftp.
8) Javascript enabled in your web browser.

G. Daemon

The Nagios daemon is a major part of the core. After its start are loaded settings from the configuration files and the monitoring of equipment and services begins. The communication of the daemon with the environment is implemented via files in which are stored the outputs as well as are read the input data.

H. Plugins

The core of Nagios is not able to control services as well as to notify their modifications. The control is performed by a plug. Plugs are incorporated between the core of Nagios and monitored hosts and services [5].

Plugs are small independent scripts that are used to control services on remote hosts. They can take a form of the Perl script or the Shell script. They run from the command line. The output of the plugs should always be directed to STDOUT (standard output). The output string should not have more than 80 characters.

Plugs are not distributed with the program core but they can be downloaded from the official website of the program or from pages of volunteer plug developers.

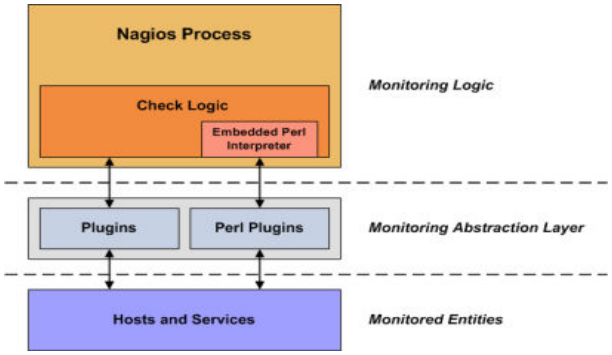

Fig. 4 Block diagram of the integration of plug-in architecture Nagios

I. Return Codes

Nagios evaluates the status of the host or his service via return codes from plug-ins. The following table contains a list of return codes along with the status of the service or host [6].

Table 1 Return codes from the plug

Return code	Status of service	Status of host
0	OK	UP
1	WARNING	UP or DOWN
2	CRITICAL	DOWN/UNREACHABLE
3	UNKNOWN	DOWN/UNREACHABLE

J. Example of using plug-in

As an example was used the plugin check_icmp. By using this plugin we recognize a response of the host or service ICMP packet. Five packets are sent and on the basis of their final values an average value is computed and a resulting status is determined.

mm@ubuntu:/usr/local/nagios/libexec$ ./check_icmp localhost -w 100.0,20% -c 500.0,60%

OK - localhost: rta 0.129ms, lost 0%|rta=0.129ms;100.000;500.000;0; pl=0%;20;60;; rtmax=0.280ms;;;; rtmin=0.077ms;;;;.

K. Configuration Files

The Nagios tool is a very extensive monitoring system and therefore its configuration is divided into several smaller files

that are defined in the main configuration file nagios.cfg, stored in /usr/local/nagios/etc. In the main configuration file there are also links to files in which is defined information about objects that are used in Nagios.

Table 2 Objects used in Nagios and its configuration file

Configuration file	Definition
cfg_dir=/etc/nagiosql/hosts	monitored devices
cfg_dir=/etc/nagiosql/services	monitored services
cfg_file=/etc/nagiosql/contacts.cfg	contacts used in the notification
cfg_file=/etc/nagiosql/timeperiods.cfg	time interval to send the notification
cfg_file=/etc/nagiosql/commands.cfg	commands that trigger actions
cfg_file=/etc/nagiosql/contactgroups.cfg	define contact groups
cfg_file=/etc/nagiosql/hostgroups.cfg	define groups of hosts
cfg_file=/etc/nagiosql/servicegroups.cfg	define groups for services

L. Defininition of devices (hosts)

The configuration file defines monitored devices such as workstations, servers, printers or devices with assigned IP address or domain name.

By the first three parameters are defined namespaces and contact information of the monitored equipment. The following parameter check_command determines which defined command, whose definition is described in the configuration file commands.cfg, shall be activated.

Another parameter is max_check_attempts which specifies a maximum number of repetitions of the test unless the device is evaluated with an error message. If such an event has occurred then an interval (in minutes) in which further tests will be performed is determined by the parameter retry_interval. Otherwise the tests are performed every five minutes, which is determined by the parameter check_interval.

A notification is sent every 120 minutes 24 hours a day and 7 days a week to a contact group admin in case that the device is in the state DOWN – (d), unreachable in the state UNREACHABLE – (u) or in the state RECOVERY – (r) (in case it started to communicate again).

The last optional parameter assignes images for graphical representation of particular devices.

Example:

define host {
host_name localhost_UBUNTU
alias localhost UBUNTU (Vmware)
address 127.0.0.1
check_command *check-host-alive*
max_check_attempts *10*
check_interval *5*
retry_interval *1*
check_period *24x7*
contact_groups *admins*
notification_interval *120*
notification_period *24x7*
notification_options *d,u,r*
icon_image ubuntu.png
}

M. Definition of test services (services)

Parameter setting of the tested service is very similar as in configuration of the device definition. A difference is that in the explicit example are not defined all parameters (reporting parameters, control parameters etc.). This is because a template (local-service), which has included these parameters, is used.

This option significantly facilitates and makes more transparent the configuration procedure. It eliminates everlasting writing of the most frequently used parameters.

Example:

define service {
host_name localhost_UBUNTU
service_description Root Partition
use local-service
check_command check_local_disk!20%!10%!/
register 1
}

N. Definition of contacts (contacts.cfg)

In this configuration file are defined contact details of persons which shall be informed about arising events.

In the example is defined a user agiosadmin which will be continually informed about failures. The information will be sent to his e-mail address and mobile phone.

Example:

define contact {
contact_name nagiosadmin
alias *Nagios Admin*
email *mm@localhost*
pager *+420608xxxyyy*
host_notification_perio *24x7*
service_notification_period *24x7*
host_notification_options *d,u,r,s*
service_notification_options *w,u,c,r,s*
host_notification_commands *notify-host-by-email,notify-host-by-sms*
service_notification_commands *notify-service-by-email,notify-service-by-sms*
}

O. Definition of the time interval for sending notifications (timeperiods.cfg)

In this step time intervals are defined, in which will the monitoring system generate and send notifications to the persons which were defined in contacts.cfg.

In the example can be seen default settings after installation from which it is easy to read that the notices will be sent continually.

Example:

```
define timeperiod {
   timeperiod_name      24x7
   alias                24 Hours A Day, 7 Days A Week
   friday               00:00-24:00
   thursday             00:00-24:00
   wednesday            00:00-24:00
   tuesday              00:00-24:00
   monday               00:00-24:00
   sunday               00:00-24:00
   saturday             00:00-24:00
   }
```

P. Scheduling of Tests

Nagios core contains a very sophisticated scheduler with many user defined options.

Q. Check Interval

All internal processes of Nagios, including host and service controls, are located in the global event queue. Schedule of control actions can be defined by a user, but not using the specified absolute date or time in cron (Unix/Linux) or Task Scheduler (Windows). This is caused by inability of Nagios to check how long it will be performed a monitoring program (plugin). Instead of it, we can tell you to Nagios how long to wait before it can run again after its finish.

It should be noted that the inspection interval is sufficient to define only for control of services. It is also possible to specify it during specification of the control interval of the target guest. But this is not necessary because host checks are usually carried out after failure of control service. If services are not working then it is assumed that the host is not accessible. The default interval length of control is 60 seconds [7].

R. Interval control after evaluating the status of

If the check of the service returns a different code from 0 (OK) Nagios reschedules the check interval to other value than original 60 seconds.

This setting is called retry_check_interval. In case that unaccessibility of the service (WARNING) is detected, Nagios will perform further three repeated checkings in order to make sure that the service is DOWN. Only after that the service is marked as unaccessible and a notification is sent by e-mail or SMS.

Of course also a maximum number of repetitions can be configured.

S. Distribution of load

When starting, Nagios usually reads a long list of hosts and services. The primary task is to find out a status of each element as soon as possible. Theoretically the list should be browsed from the beginning to the end and then again from the beginning. But this possibility is not an optimal path because it generates too much load on the remote host. For example in case of a server which is on the top of the list and has configured 15 services Nagios would try to check all of them one by one.

Instead of this procedure is used so called Interleave factor which is depicted in Fig. 5.

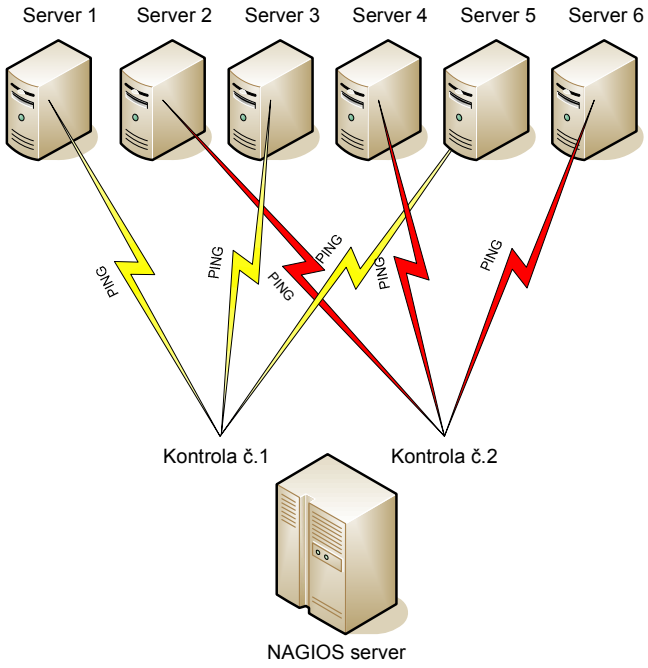

Fig. 5 Load distribution with the interleave factor

T. SNMP Monitoring

The SNMP protocol was originally created for a remote control of devices of computer networks. It is an application protocol that provides services for management over the UDP protocol [8]. The SNMP is based on the client/server model. The client program called network manager creates a virtual connection to the server. A SNMP agent is running on the monitored network device. The agent monitors the devices status and provides information about it by the manager. Information provided by the agent is arranged according to the MIB database (Management Information Base) that is the structure corresponding to the device.

The advantage of this solution is that it is necessary to have a password to the privileged mode. The community string can be used for access to the device, which can be defined as read-only.

IV. Installation

Nagios can be installed in two ways. Either we use a package which is created for a particular distribution or we use installation from binary files. This paper describes the second method. For the compilation are necessary packages and libraries: gcc, make, autoconf and automake.

The entire installation process will be performed on the operating system Linux Ubuntu 9.10 (Karmic Koala) installed on VMware Workstation on 7.0.1 build-227600th.

A. Download of the installation package

From the home website http://www.nagios.org/download/ it is necessary to download the kernel package, which consists of a daemon and a web interface. Further it is necessary to download a package which includes plug-ins. http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.1.tar.gz.

http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz

Nagios needs for its installation only a few dependent packages. Most of them are not mandatory.

In case of using the Web interface it is necessary to have installed a Web server which supports CGI (Common Gateway Interface). Further it is necessary to have installed three graphic libraries (libpng, libjpeg, gd library) in order to display pretty generated pictures and graphs.

During installation of the plugins it is necessary to have installed more parts: PING program, some tools BIND (Berkeley Internet Name Domain) as HOST, DIG, NSLOOKUP, further library OpenSSL (Secure Sockets Layer) and PERL (Practical Extraction and Report Language).

For interrogating of network objects by SNMP it is necessary to install net-SNMP, perl-SNMP.

B. Creating of required users groups

Before the installations start it is necessary to create manually two groups (Nagios, nagcmd) and a user Nagios which will be assigned to these groups:

```
mm@ubuntu:~$sudo –s
root@ubuntu:~#/usr/sbin/useradd -m -s /bin/bash nagios
root@ubuntu:~#passwd nagios
root@ubuntu:~#/usr/sbin/groupadd nagios
root@ubuntu:~#/usr/sbin/usermod -G nagios nagios
root@ubuntu:~#/usr/sbin/groupadd nagcmd
root@ubuntu:~#/usr/sbin/usermod -a -G nagcmd nagios
root@ubuntu:~#/usr/sbin/usermod -a -G nagcmd www-data
```

C. Compiling and Installing Nagios (core program)

From the directory, where the installation package is extracted, run the configuration script:

```
root@ubuntu:~#./configure --with-command-group=nagcmd
```

Compiling the source code:

```
root@ubuntu:~#make all
```

Installation of binary files, initialization scripts, sample configuration files and setting competencies for a directory with external commands:

```
root@ubuntu:~#make install
root@ubuntu:~#make install-init
root@ubuntu ~#make install-config
root@ubuntu:~#make install-commandmode
```

D. Compiling and Installing Nagios (plugins)

From the directory, where the installation package is extracted, run the configuration script:

```
root@ubuntu:~#./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

Compiling the source code:

```
root@ubuntu:~#make
root@ubuntu:~#make install.
```

V. Monitoring Systems

A. SAP System Monitoring

There are several ways of monitoring the SAP (System Analyse und Programmentwicklung) systems. The simplest way is to control the ports on which the SAP system is running. It is usually running on ports 3200/3300 for numbers 00 and ports 3201/3301 for numbers 01, etc. This simple check can be made by check_tcp plug. In case that the internal services in SAP fail, the terminal user will not be able to sign to the system even if the ports will be available. If it is necessary to test complex interactions between components of the SAP then communication is needed at the application layer [9].

B. Control Using sapinfo

The program sapinfo is a part of an optional RFC-SDK package (Remote Function Call Software Development Kit) used in RFC interfaces. The package can be downloaded from SAP portal at http://service.sap.com, but for login it is necessary to have a customer number.

C. Control via Plug Check_sap.sh

Plug check_sap.sh is a script that is based on the program sapinfo. It is included in a package with plugins for Nagios in the contrib directory. Since it is not installed automatically it must be manually copied to the plugins directory:

```
/usr/local/nagios/libexec
```

It is also necessary to modify the variable in the script sapinfocmd check_sap.sh entering the path to the sapinfo:

sapinfocmd=' /usr/local/sap/rfcsdk/bin/sapinfo '

D. Control via CCMS

The SAP system has its own monitoring system called CCMS (Computing Center Management System) in which local agents, determined for data collection, collect data from different hosts. CCMS is not just for SAP systems but it can monitor also external third-party applications.

Fortunately, the developers thought to the possibility of monitoring the CCMS and programmed plug-ins for the data collection system Nagios.

E. Monitoring of Linux/Unix

There are several different ways how to monitor attributes on remote Linux/Unix server. One of the possible approaches is a method with help of the SSH keys (Secure Shell), created SSL (Secure Sockets Layer) connection and plug modul Nagios check_by_ssh. The plugins are activated on a remote server. The disadvantage of this method is an extreme load on monitoring server in case we want to monitor hundreds of services or possible destruction of an encrypted SSH connection.

The second method uses NRPE (Nagios Remote Plugin Executor) which enables activating of plugins on a monitoring remote server [10].

F. Direct Control Using the NRPE

Fig. 6 shows the most common use of the NRPE daemon. In this case, only local sources on a remote server are monitored, for example load of the CPU, availability of memory or disk array, swap, number of registered users etc.

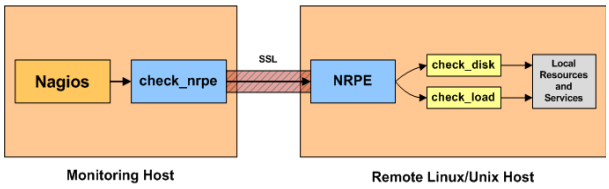

Fig. 6 Block diagram of the direct control of NRPE

G. Indirect Control Using the NRPE

Fig. 7 shows how to use the NRPE for control of services and resources on remote servers that are not accessible from the monitoring machine which is running Nagios.

In this case, the NRPE daemon acts as a proxy server.

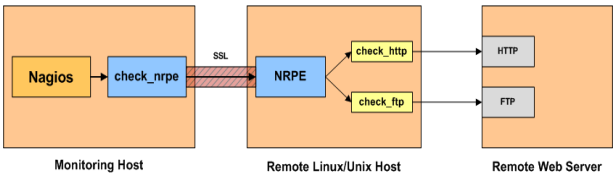

Fig. 7 Block diagram of indirect control NRPE

H. Monitoring of Microsoft Windows

Monitoring of running services on systems with the Microsoft Windows using Nagios is relatively easy to implement. One of the possibilities is utilization of the standard functionality of Windows and installs the SNMP protocol support from the installation CD.

Another option is using of an agent NSCClient++.

I. Agent NSCClient++

It is a simple and secure monitoring agent written for Microsoft Windows operating systems. This agent serves as a proxy between the Nagios plug-in and a monitoring service or attribute on the Microsoft Windows server. Private services such as availability of the memory, disk space or CPU load cannot be monitored without this client.

In case of monitoring of public services such as HTTP, FTP, POP3, it is possible to use Nagios plugins check_http, check_ftp, check_pop.

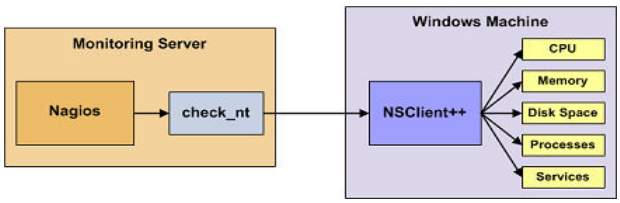

Fig. 8 Block diagram of the control agent using NSClient++

Fig. 8 shows the principle of collecting of the monitoring data using the Nagios plug check_nt and agent NSClient++. During questioning about the data, NSClient++ will request for them. Further it will store the data in an internal stack and subsequently it will provide them to a plug-in check_nt for further processing.

J. Sending Notification

Only few computer network administrators are able constantly monitor changes in the monitoring tool. Therefore, also Nagios has the option of sending the notice by mail, SMS, pager or VoIP (connection to Asterisk). In order Nagios not to become a "spam server", it should be considered when the notifications will be sent, in what quantities and to how many recipients. It is not always needed to send a notification to the system administrator.

K. Sending Notification via E-mail Messages

In order to deliver alarm notifications from Nagios to the terminal recipient, it is necessary to use either a functional mail server or to install it on the server with Nagios. A possible solution is, for example, Postfix mail server that is simple to install and initially configure.

L. Sending Notification via SMS

Nowadays almost everyone owns a mobile phone and it would be a shame if the mobile phones are not used also for sending notifications from Nagios via SMS. Even if its length is limited to 160 characters, basic information about the host,

failure, time and condition of devices or services fits to it just fine. This option provides a huge benefit to administrators who may be informed of any problems encountered anywhere, even if they are not directly at their computer.

There are many solutions how to send SMS messages from your computer to your mobile phone. On Linux distributions, we can implement SMS as an additional installation by utilities (Gnokii and Yaps) that communicate with a mobile phone which is connected to the computer.

Another solution is using of SMS gateways available on the Internet. Some Service Company service which provides SMS messages over HTTPS/API (Hypertext Transfer Protocol Secure/Application Programming Interface) was used in this work. It is necessary to create a login account on this page and enable communication HTTPS/API. Further it is necessary to make settings in the configuration file command.cfg, which defines a command which will be called when an alarm in the monitoring tool for sending notices occurs. This service is chargeable.

Some mobile operators provide a service through which you can send an email to their email account and it is automatically forwarded to your mobile phone.

VI. Conclusion

The aim of this work was to test and prepare for common use a free supervisory system for monitoring of computer networks. The paper is focused on different methods and tools for monitoring. The Nagios monitoring tool is described comprehensively including its configuration. The paper also focuses on installation and application of its functionalities on the current most widely used operating systems. The system is able to send a notice by e-mail, SMS messages and generate statistical graphs of the measured values.

The practical result of this work is the implementation of the Nagios monitoring system for an unnamed company, which will certainly be a great benefit for the company. The company management particularly appreciated network monitoring of the SAP system, which the previous pre-paid network monitoring system did not enable.

The only initial investment of the open-source Nagios product is its installation and configuration. The time spent on its launch will however quickly return in the form of a solid system for monitoring of computer networks from one central point.

References

[1] M. Matysek, Monitoring of computer networks and applications using Nagios, in Proceedings of the 11th WSEAS International Conference on Data Networks, Communications, Computers. Sliema, Malta: WSEAS Press, 2012, pp. 63-67.

[2] T. Shimomura, Q. L. Chen, N. S. Lang, and K. Ikeda, in Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications, Athens, Greece: WSEAS Press, 2007, pp. 188-196.

[3] C. Burgess. (2010, Feb 02). The Nagios Book [Online]. Available: http://www.nagiosbook.org/html/index.html

[4] M. Sysel, and S. VITÁSEK, Client-Server Hardware Detection Tool, in Proceedings of the 15th WSEAS International Conference on Computers, Corfu Island, Greece : WSEAS Press, 2011, pp. 340-343.

[5] M. Schubert, Nagios 3 Enterprise Network Monitoring Including Plug-Ins and Hardware Devices, Burlington: Syngress Publishing, Inc., 2008.

[6] E. Galstad. (2010, Jan 28). Official Nagios Documentation [Online]. Available: http://support.nagios.com/knowledgebase/officialdocs

[7] J. David, Building A Monitoring Infrastructure With Nagios, Boston: Pearson Education, Inc., 2007.

[8] M. Sysel, MATLAB/Simulink TCP/IP Communication, in Proceedings of the 15th WSEAS International Conference on Computers. Corfu Island, Greece: WSEAS Press, 2011, pp. 71-75.

[9] W. Barth, Nagios, System and Network Monitoring, Munich: Open Source Press GmbH, 2006.

[10] J. Kretchmar, Open Source Network Administratio, Upper Saddle River: Prentice Hall Professional, 2003.