GÉANT

Monitoring Tools for Network Services and Systems

Best Practice Document

Produced by CSC/Funet led working group on AccessFunet

Author[s]: Kaisa Haapala, Janne Oksanen

13.05.2011

Document No:	GN3-NA3-T4-Monitoring-Tools
Version / date:	13.05.2011
Original language :	Finnish
Original title:	“Verkon palvelujen ja järjestelmien valvontatyökaluja”
Original version / date:	1.0 of 13.05.2011
Contact:	kaisa.haapala (at) csc.fi

CSC/Funet bears responsibility for the content of this document. The work has been carried out by a CSC/Funet led working group on AccessFunet as part of a joint-venture project within the HE sector in Finland.

This translated version is based on the Finnish counterpart approved by the CSC/Funet annual general meeting on 13 May 2011 after an open consultation period of two weeks.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 238875, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3)'.

Table of Contents

1. Introduction

The goals of monitoring:

- Detecting faults with your own monitoring tools, not via customer feedback, and before the backup system fails.
- Gain data for longer-term tracking and raw material for reporting (statistics).
- After a problem situation, the gathered materials allow analysing the causes of the problem and avoid similar situations in the future.
- Detection of misuse in order to alleviate the effects.

What is monitored:

- the functionality, usage statistics and environmental parameters of the network services and systems

Prerequisites:

- Understand what is worthwhile to monitor in the system and how, and especially what not to monitor.
- There is a clear sign whether monitoring is needed: if the service does not interest the management, its maintenance is unorganised and there is no need to care about the end users, monitoring is also a waste of time.

2. Planning Monitoring

When monitoring is being planned, the different parties (monitoring customer/user and the maintainer) must have a clear idea of what should be monitored and how, in particular when there are a lot of services and systems (dozens – hundreds – thousands). Otherwise the implementation of the monitoring is easily left half-baked or will not keep up with changes, and no benefits are gained from the investment. Vagueness in these matters has led at least to the following kinds of problems:

- The system or service is not monitored at all, because it has, for example, worked too well (or poorly) and has therefore been (actively) forgotten. *As an example, the time of vmware virtual servers is not monitored, because keeping them in sync would require changes to the default configuration.*

- The wrong thing is monitored, because either the party that ordered or implemented the monitoring, or both, did not understand what the issue is about and what could or should be monitored. *As an example, only a single system component is monitored, not the entire service.*

- Important alarms drown in a flood of false alarms. *As an example, the alarm limits have* *not been correctly set, or, for example, alerts concerning low disk space are sent to server maintenance even if the service is maintained by another party, to whom the alarms should be sent.*

When the implementation of monitoring is being planned, it is worthwhile to try and take into consideration other perspectives than just maintenance, which has to at least specify the monitoring of its service or system. Different perspectives to be taken into consideration include at least the following:

- Maintenance is mainly interested in noticing and limiting clear faults, reduced redundancy, etc., in all services and systems that are its responsibility.

- The end user is usually only interested in whether the service works correctly or not. In addition to monitoring of single components, it is therefore worthwhile to implement a monitoring method that indicates whether the service works or not from an end user point of view.

- The management should be interested in developing the functionality of the services, monitoring usage (consideration of need), etc. Measurement data collected from the systems for monitoring functionality may be useful in such monitoring, so it should perhaps be at least stored for reporting tools.

Planning the monitoring and ensuring continuity is demanding, in particular when monitoring is implemented in a centralised manner for a large number of services and systems. There must be a maintenance process for all monitored systems and services in order for the monitoring to be useful. A functional method for planning monitoring is to link it tightly as part of maintenance processes.

3. Different Technical Monitoring Methods

Services and systems can be monitored as follows:

- the monitoring system tests whether the service works (e.g. does a WWW-service respond to HTTP queries) or queries the system for its status (whether the file system is full), etc.

- tracking information provided by systems and services on themselves (log entries, SNMP traps, etc. messages).

3.1. The pros and cons of different methods

Checks made by a monitoring system are usually an easier way to get reliable information on the status of the subject of monitoring at the moment of checking. The monitoring system can immediately react to a successful and especially an unsuccessful check, for example by performing N verifying checks before giving an alarm.

The information provided by the monitored system itself (log messages, etc.) can be less reliable, as messages may be lost before they reach the monitoring system. The monitoring system may also be malfunctioning, causing the system to stop sending error messages.

Checks made by the monitoring system have a downside compared to messages sent by the monitored systems themselves – the delay in detecting a fault and the load caused by the checks.

Just handling error messages usually requires less resources than continuous, active checking of functionality. The monitored system itself can also detect a fault and send a notification faster than a check made once in a while from the outside.

Overloading the systems with unnecessary monitoring should be avoided.

4. Maintenance of Monitoring Systems

In order to avoid memory lapses and mistakes, the goal should be that it is enough to add new devices to a single location, from where all monitoring systems retrieve their information. You will usually remember to remove devices from monitoring at the latest when you receive a notification that the device cannot be reached.

5. Monitoring Tools

See below for a list of results from a network monitoring survey carried out among Funet members in 2010. The tool is first described in brief, followed by comments on usage experience as recounted by the survey respondents. The survey questions and a report of the results is available on AccessFunet’s wiki page [AF]. As the number of respondents was rather small, the results do not show how popular the different tools are.

5.1. Self-written Scripts

Many Funet members use scripts or tools they have written themselves.

As there are many different network environments, tools on the market may not necessarily gather precisely the information needed or desired for monitoring. In such situations, self-written scripts can be the right tool for the job. They can also be used to assist in situations in which two different systems or software cannot directly talk with each other. Scripts allow the automation of certain functions, such as information picking or converting the results into a format readable by some other tool.

When self-written scripts are used, time must be reserved for their planning, and various things must be taken into consideration, such as:

- what information do you wish to generate with the scripts and how?
- assumptions are easily made at the time the scripts are written; they may later become invalid.
- the availability, expandability and information security of the scripts as the environment changes
- which persons are able to and know how to use the scripts?
- who are able to correct any information security holes detected later?
- how is information on all scripts in use managed?
- which of the scripts are necessary, and what are their dependencies with each other and the environment?
- which scripting languages are used, and will there be know-how and support in the future?

5.2. Wireless Networks

Airwave [AIR]

- Monitoring of wireless network access points and controllers, and user statistics.
- Commercial software.
- Works better with standalone access points that with WLAN controllers.
- Occasionally displays erroneous data or no data at all when used with controllers.

Cisco WCS [CISCO]

- Commercial, manufacturer-specific software

7signal Sapphire [SIGNAL]

- For end-to-end monitoring of a wireless network
- Separate active sensors measure the radio way
- Manufacturer-independent
- No knowledge of any implementation in the Funet organisation
- Commercial software

Big Sister [BIG]

- A network monitoring tool, allowing alerts of any error situations.
- Open source code.

5.3. Data Collection and Presentation as Graphs

Cacti [CACTI]

- A tool for collecting packet counters, available memory and other such numerical values and presenting them as graphs, with time as the horizontal axis.
- Open source code.
- Sample application include temperature monitoring of UPS systems and computer server rooms
- Some things may be difficult to configure.
- A useful weather map plug-in http://www.network-weathermap.com/

Cricket [CRICKET]

- Monitoring tool, data collection and presentation of the results as graphs.
- Open source code.
- Easy to take into use; a number of improvements over MRTG [MRTG]
- Downside:
    - No alarm options, monitoring only.
    - In some cases, performance problems have been detected.
    - Development frozen in 2004.

MRTG [MRTG]

- Tool for generating graphs of traffic amounts etc.
- Open source code.
- Many other tools use MRTG [MRTG]

RRDTool [RRD]

- Tool for generating graphs of traffic amounts etc.
- Open source code.
- A popular tool

5.4. Log Collection and Handling

Kiwisyslog [KIWI]

- Log information collection tool.
- Open source code.
- A good tool for collecting log information that can then be analysed, for example, on a UNIX command prompt.
- The server version is commercial.

Splunk [SPLUNK]

- Log information collection from devices.
- Commercial software.
- A good log server that you can modify according to your needs. The software is able to send alarms.

- Supports the simultaneous browsing of many different logs
- Log indexing and searches
- Modifiable; you can create your own searches using regular expressions

5.5. Monitoring of Services/servers

Nagios (also Icinga) [NAGIOS]

- Especially suited to the monitoring of server systems and services running on them.
- Open source code.
- Versatile, scalable and modifiable; not the easiest to use and maintain.
- Performs scheduled checks or only receives results.
- An extensive collection of ready checking plug-ins for the monitoring of various services and systems.
- A simple interface between the Nagios core and the checking plug-ins; easy implementation of your own checking plug-ins.
- Has availability and other such reporting functions, but they are difficult to use; for this reason, Funet, for example, has implemented report generation as regular batch runs utilising Nagios CGI scripts.
- No direct support for creating statistics and graphs from the measurements collected during checking (e.g. /var file system 19 % full); Funet, for example, uses a separate add-on.
- An extensive user base, support agreements can be purchases (even in Finland).
- Icinga, a branch of Nagios, is Nagios-compatible with better reporting and GUI.

Munin [MUNIN]

- For server monitoring
- Open source code.

ManageEngine, OpManager [OP]

- A monitoring tool for the network, servers and services.
- Commercial software.

ManageEngine, DeviceExpert [DEVICE]

- Configuration management and tracking tool for active devices.
- Commercial software

MetaNav [NAV]

- A tool for monitoring extensive campus networks, developed by UNINETT.
- Open source code.
- Laborious to take into use
- Almost all customers of UNINETT use it. Some Funet members have tried it out but have not taken into use.

Netdisco [DISCO]

- This tool can be used to, for example, find out the location of devices on a network with the help of IP/MAC.

- Open source code.
- Able to create a map of the network topology, for example.
- Displays diverse inventory data of the hardware.

Smokeping [SMOKE]

- Measuring of service and device availability and response times.
- Open source code.
- A handy tool with good visualisation of network lag and the ability to send e-mail alarms.

What's UP [UP]

- Monitoring of network connections.
- Commercial software.
- Affordable and easy to use
- able to create a map of the network topology and collect various types of log information
- Add-ons are available for the tool, and you can also create them yourself

Zabbix [ZABBIX]

- A monitoring tool for servers and applications.
- Open source code.
- Quickly detects problems.
- A commercial support agreement also available.

Zino [ZINO]

- Tool for collecting information such as network device connection counter data, and presenting it as graphs with time as the horizontal axis.
- Open source code.
- Topology maps from different layers and the counter graphs of individual connections can be linked with each other.
- Topology maps feature link colouring according to load.
- E-mail reporting of network connection errors as compilations.
- Partially dates to last millennium; the (open) source code uses, for example, Scotty and Just (a TCL library), due to which laborious to make it work.
- Can be modified with Perl, TCL and Bourne Shell scripting.
- Funet has RPM-packaged all that is needed, including their own hacks.
- Probably only in use at Funet, NORDUnet and Uninett.
- Scales well to monitoring of at least up to 200 routers.

5.6. Information Security

Nfsen/Nfdump [NFSEN][NFDUMP]

- Tools designed for handling Netflow.
- Open source code.
- The commant line tool Nfdump collects and prosesses Netflow data.
- Nfsen is a web-based user interface for the Nfdump tool
- Application includes tracking of traffic amounts, investigation of misuse, identification of

worms, port scanning and protocol distributions.

Snort [SNORT]

- An IDS/IPS tool analysing traffic.
- Open source code.
- Easy to take into use, but maintenance is laborious

Rancid [RANCID]

- Backup copying of network device configurations and comparison of changes.
- Open source code.

5.7. Tools Offered by Funet

Funet Scanner [CERT]

- Software such as Nessus available for finding network vulnerabilities.
- CSC/Funet offers Funet user organisations the possibility of investigating information security holes in their own networks from outside using the Nessus software.
- You can find a service description at Funet extranet (only accessible to Funet members)

Service Status

- im.funet.fi [IM]
    - Connection quality measurement using ping.
    - A public view showing what is Funet network monitoring’s idea of the status of a customer’s Funet connection.
- Monitoring views implemented using Nagios (only accessible to Funet members).
- Monitoring views for the Funet router network:
    - Weathermap (Zino) [ZINO] [WM]
    - Looking glass [GLASS]
- Multicast monitoring view:
    - Multicast Beacons [BEACON]

6. References

[AIR]	http://www.arubanetworks.com/products/airwave_management.php
[AF]	https://info.funet.fi/wiki/AccessFunet/ (only for Funet members)
[BEACON]	http://noctv.funet.fi/funet/
[BIG]	http://www.bigsister.ch/
[CACTI]	http://www.cacti.net/
[CERT]	https://info.funet.fi/palvelut/cert/ (only for Funet members)
[CISCO]	http://www.cisco.com/en/US/products/ps6305/index.html
[CRICKET]	http://cricket.sourceforge.net/
[DEVICE]	http://www.manageengine.com/products/device-expert/index.html
[DISCO]	http://www.netdisco.org/
[GLASS]	http://www.csc.fi/funet/status/tools/lg.pl
[IM]	http://im.funet.fi
[KIWI]	http://www.kiwisyslog.com/
[MRTG]	http://oss.oetiker.ch/mrtg/
[MUNIN]	http:// munin-monitoring.org/
[NAGIOS]	http://www.nagios.org/
[NAV]	http://metanav.uninett.no/
[NFDUMP]	http://sourceforge.net/projects/nfdump/
[NFSEN]	http://sourceforge.net/projects/nfsen/
[OP]	http://www.manageengine.com/network-monitoring/
[RANCID]	http://www.shrubbery.net/rancid/
[RRD]	http://oss.oetiker.ch/rrdtool/
[SIGNAL]	http://www.7signal.com/
[SMOKE]	http://oss.oetiker.ch/smokeping/
[SNORT]	http://www.snort.org/
[SPLUNK]	http://www.splunk.com/
[UP]	http://www.whatsupgold.com/
[WM]	http://www.csc.fi/funet/status/tools/wm
[ZABBIX]	http://www.zabbix.com/
[ZINO]	http://www.network-weathermap.com/

7. Glossary

GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
MAC	Media Access Control
SNMP	Simple Network Management Protocol
UPS	Uninterruptible Power Supply
WLAN	Wireless Local Area Network
WWW	World Wide Web