ESTABLISHMENT OF A MONITORING SOLUTION BASED ON EYESOFNETWORK

Mammou Manel, Harabi M.Amine
ur.sciences-tech@time.ens.tn , mammou_manel@outlook.fr
TIME Time University
Tunis, TUNISIA

Abstract— **Network supervision is the fact of collecting the necessary informations on the state of the infrastructure and the entities used therein, analyze and report them. It allows having a global view of the operation and potential problems on a network.**

EON (EyesOfNetwork) is a complete solution to monitor the network while integrating the ITIL approach, it is currently distributed under the GPL and it gathers many tools of supervision: network Supervision with NAGIOS, NAGIOSBP and NAGVIS, Performance Management with CACTI, WEATHERMAP and a configuration interface with backup.

Keywords-component; network supervision, performance management, monitors the network

Introduction:

The concept of network supervising was invented in 1980 during the explosion of computer networks establishment in companies.

Network management systems are nowadays among the most important elements of a successfully functioning computer network. The maintenance and configuration of network devices, servers and services, as well as continual monitoring of the operation of all the devices within the network, are the key elements of a network management system. In order to ensure the reliable and secure management of devices and services, it is necessary to design a network in such a manner that it provides the highest level of security isolation of management traffic from production traffic [1].

The big size of networks and their heterogeneity posed a reel administration problem. Supervision must be capable to adapt with heterogenic environment, to automate the network's control and to generate a group of statistics which give a best vision of the network.

Thus, networks monitoring is both reactive: alerting the administrator with emails for example in case of dysfunction of a part of the information system, and proactive: anticipating the breakdowns and targeting the problem dice their appearance.

In this paper we are focusing on EyesOfNetwork as a solution to administrate the network, we are presenting in this paper firstly EyesOfNetwork and its components, secondly the work done to measure the performance of machines in the network with Raspberry Pi B+ and a simple open source graphical solution, thirdly we presented a solution to remote access the Raspberry Pi from another machine in the network.

I. PRESENTATION OF THE MONITORING TOOL EYESOFNETWORK

EyesOfNetwork is the distribution of Linux based on CentOS; it reassembles many tools of network supervision as present **Figure1**: network Supervision with NAGIOS, NAGIOSBP and NAGVIS, Performance Management with CACTI, WEATHERMAP and a configuration interface with backup. And it's distributed under a GPL license.

EyesOfNetwork is accessible via a unique web interface whose objective is to gather the different actors of the information system: Administrator, Technician…. Each one of them has a view corresponding to his job. All the informations are consolidated with a MYSQL Data base.

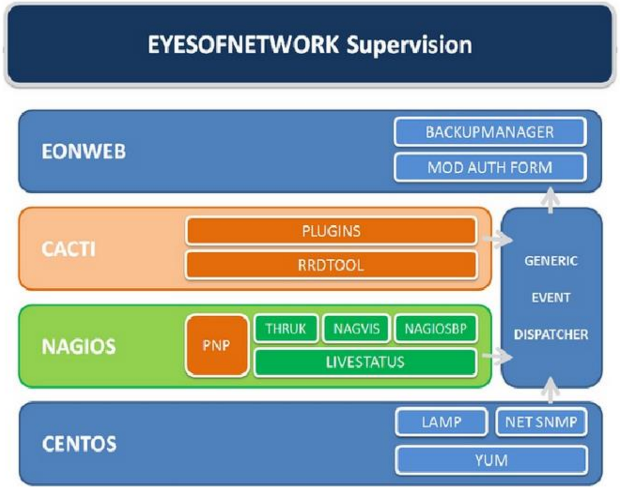


Fig.1 EyesOfNetwork's tools [2]

DOI : 03.IJES.2015.1.8

EyesOfNetwork function thanks to many plugins, without them it's totally incapable to monitor the network. These plugins are external server programs that can be executed online in command lines to test a service. It operates on the basis of sending requests to hosts or services to test their availability.

The relation between the core and the plugins is assured on one hand by the configuration files and on the other by the return code of a plugin.

EyesOfNetwork is based on many configuration files to construct it's infrastructure of monitoring, the most important files are: **Eyesofnetwork.cfg**: Is the principal configuration file of EyesOfNetwork, it contains the list of other configuration files and it includes all the global instructions of performance. **Hosts.cfg**: define all the hosts to supervise in the network; every host is defined by its name, IP address and the test that must be performed to characterize the state of the host.**Contacts.cfg**: define the contacts to prevent in case of incidents and alerts parameters: number of notifications, time slots to send alerts …etc.

A. Components of EyesOfNetwork

EyesOfNetwork gathers many tools that make it a complete simple solution of administration:

1) NAGIOS

NAGIOS is an open source software distributed under the GPL license, it operates on Linux and it allows monitoring hosts, services and active networks. Its role is to alert the administrator in case of a breakdown, the architecture of the information system is composed by five elements like it shows the **Figure2**

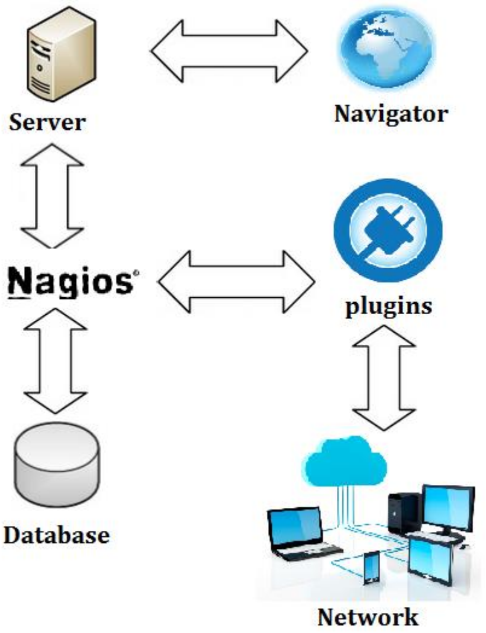


Fig.2. NAGIOS's architecture

Nagios's architecture is composed of:
- Nagios's server that recovers the necessary informations from the plugins.
- A Database that stores all the informations
- The plugins that we choose according to our needs
- The elements to supervise in the network
- A web interfaces that allows the management of all of this

2) CACTI

CACTI create graphics of the bandwidth with SNMP and it's also possible to install other plugins on CACTI to integrate other tools like NTOP or WEATHERMAP.

The **Figure3** shows the functionalities of CACTI

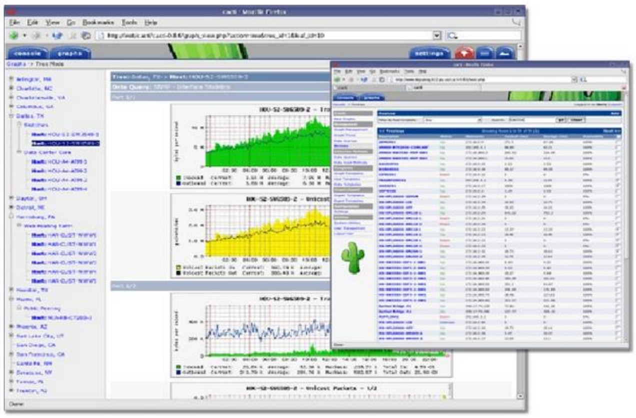


Fig.3: CACTI's functionalities [3]

3) NAGVIS

NAGVIS is written with PHP and JavaScript it allows the aggregation of NAGIOS's supervision informations to obtain cartography of availability, carts are directly created in the web interface.

Administrators can create their own supervision carts with PNG format with indicators like: hosts services…etc.

II. TECHNICAL BACKGROUND

I choosed to work on Cacti because it's a complete graphing solution, it represents graphically the state of the device on which it's installed: CPU usage, state of the processes…

It functions with a web server that contains a MySQL Database, the PHP language, and an RRDTool interface: it's the OpenSource industry standard, high performance data logging and graphing system for time series data. RRDtool can be easily integrated in shell scripts, Perl, python, ruby, Lua or Tcl applications. [4]

A. Cacti's characteristics and features

1) Graphs

Cacti provides an unlimited number of graphics, graph data is manipulated with CDEF math functions built into RRDTool.

2) Data Sources

DOI : 03.IJES.2015.1.8

Each data source can be used to gather local or remote data and place it on a graph. Round robin archive (RRA) settings can be customized giving the user the ability to gather data on non-standard timespans while store varying amounts of data.

3) *Data Gathering*

Contain a data input mechanism that allows user to insert specific data that can be used to gather the necessary information of the network: such as IP addresses.

4) *Templates*

Graph templates permit common graphs to be grouped by templating.

Data source templates enable common data source types to be grouped together by templating. Every field for a normal data source can be templated or specified on a per-data source basis.

5) *Graph Display*

The tree view allows users to create "graph hierarchies" and place graphics on the tree. This is an easy way to manage and organize a large number of graphics.

For the realization of this project we used a Raspberry Pi B+

In the next section we will concentrate on the work we have done using the Raspberry Pi B+

B. *Representing the state of a host with Cacti*

First we installed Cacti on Raspberry Pi B+ and configured it.

We tried to test the performance of Raspberry Pi with Cacti.

The figure above represents a bending of the CPU utilization.

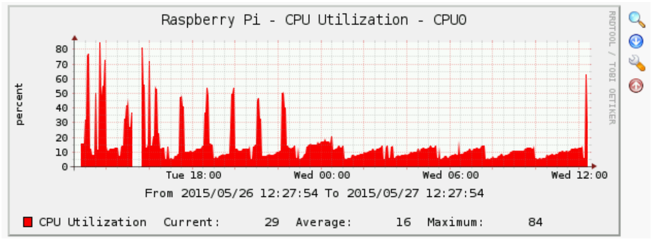


Fig.4: Raspberry Pi’s CPU Utilization

And this figure indicate the load average,

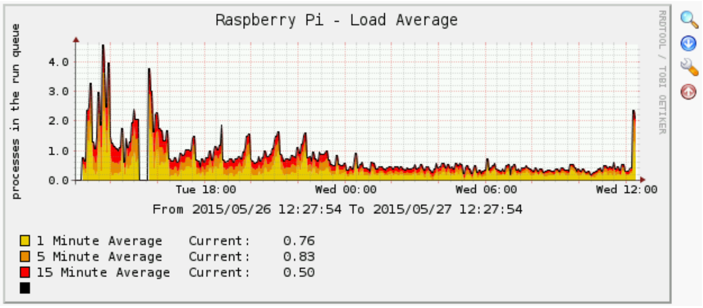


Fig..5: Raspberry Pi’s Load Average

And this figure indicate the interface traffic, it shows the measures of the incoming and out coming traffic in bytes/sec

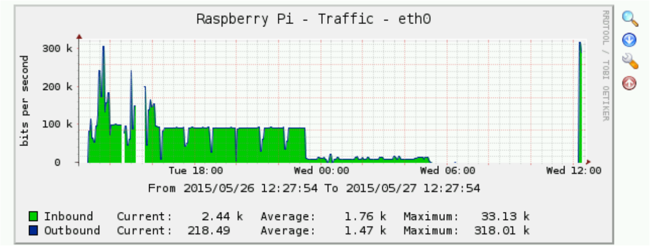


Fig.6: Eth0 traffic

And then, we tried to test the performance of a windows 7 host in the network with Cacti installed on Raspberry Pi.

The Figure7 represents the performance of the processes.

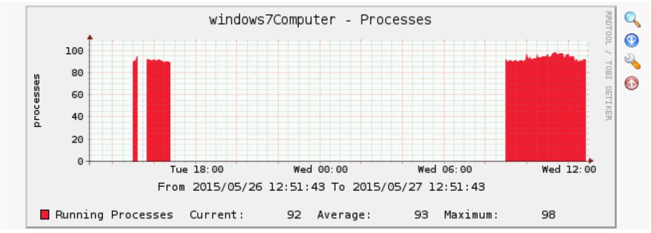


Fig.7: Windows 7 processes’s performance

This graph illustrate the Physical memory exploitation

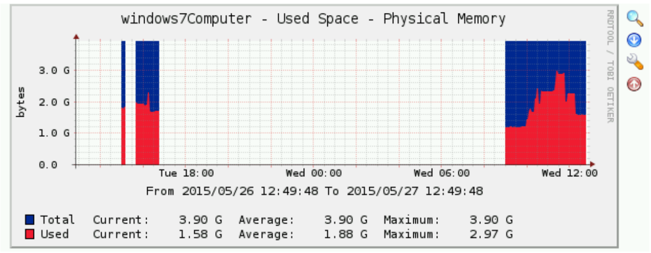


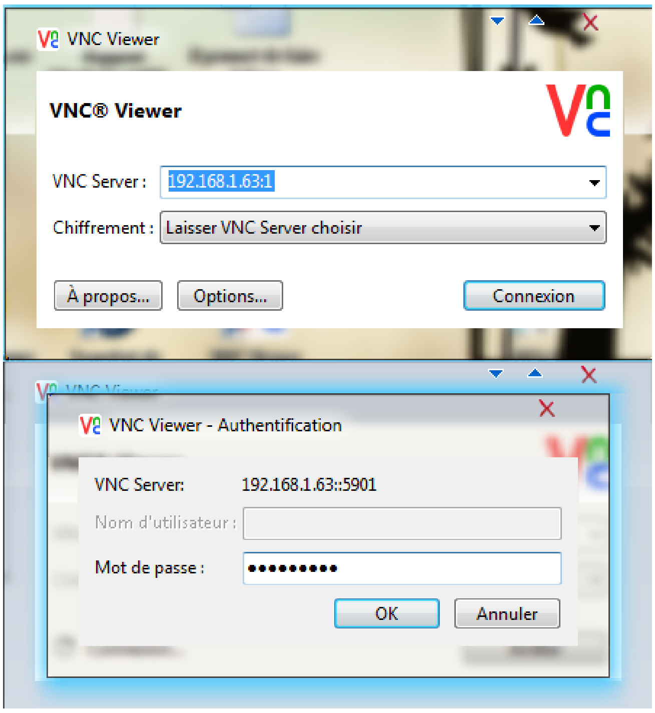
Fig.8: Windows 7 physical memory’s exploitation

C. *Remote access of Raspberry Pi from another host*

Our goal was to remote access the Raspberry Pi’s graphical interface.

DOI : 03.IJES.2015.1.8

We decided to use VNC Viewer (Virtual Network Computing), on a machine in the network.
We specified the Raspberry Pi's ip address and password on the VNC client to connect to the VNC Server and take control of it as illustrated in the figure above


VNC Server : 192.168.1.63:1
Chiffrement : Laisser VNC Server choisir

VNC Viewer - Authentification
VNC Server: 192.168.1.63::5901
Nom d'utilisateur :
Mot de passe : •••••••••

Fig.9: VNC Viewer

And as represent the Figure10, now we are able to see the Raspberry Pi's desktop on a windows machine and we are able to control it as though we are working on the raspberry Pi itself.

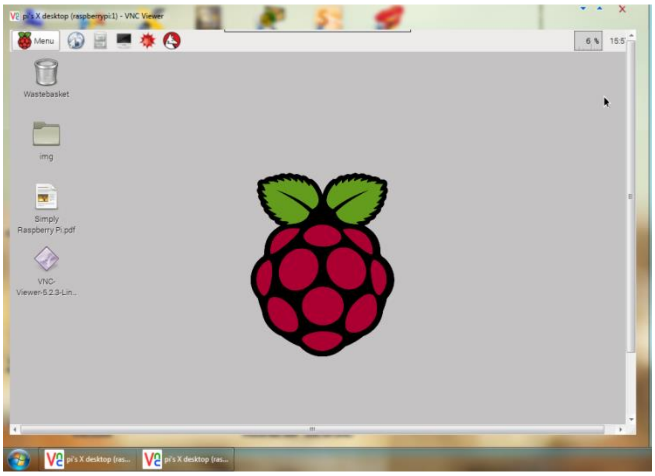


Fig.10: Raspberry Pi desktop on windows 7 machine

III. Conclusion

This paper explores the notion of network supervision, performance management. During this research, we have aimed to present a simple and accurate way for monitoring the performance and the load of machines in the network.

And also w presented a way to remote access and supports the Raspberry Pi from any machine in the network.

References

[1]: http://www.gwcnetworks.com/network-monitoring-management.html

[2]: https://www.eyesofnetwork.com/?lang=fr

[3]: http://www.cacti.net/

[4]: http://oss.oetiker.ch/rrdtool/

DOI : 03.IJES.2015.1.8