PANDORAFMS

Introduction

29-07-2021

Introduction

Go back to Pandora FMS documentation index

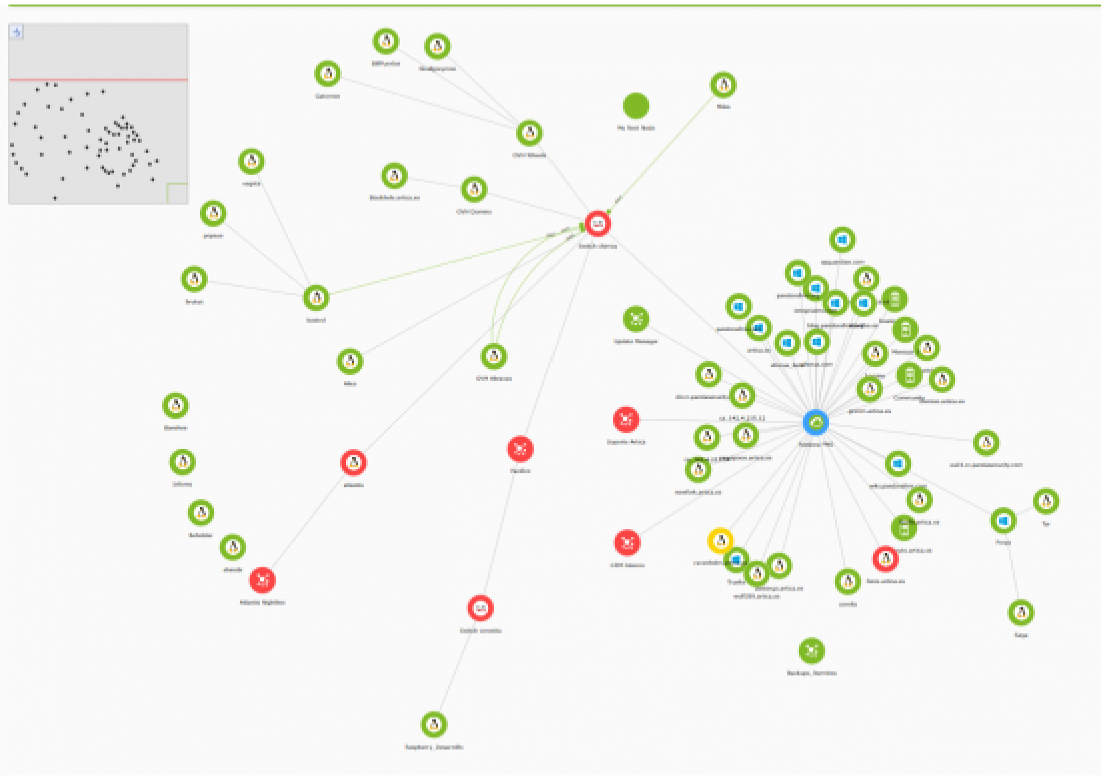
Introduction

Pandora FMS: What is it, exactly?

Pandora FMS is a network monitoring software package, intended for all types of environments. To use the word 'monitoring', in its broad semantic sense, is somehow risky - since there are hundreds of tools available - each one of them adapted to a singular type of environment: monitoring a couple of printers in a small office is not the same as monitoring thousands of interfaces and switches with extremely high network traffic in a data center with thousands of servers.

Pandora FMS is designed to adapt to every role and organization. Its main aim is to be flexible enough to manage and control the complete infrastructure, without the need to invest more time or money in another monitoring tool.

FMS is an acronym for **F**lexible **M**onitoring **S**ystem. Its purpose is to be able to monitor both complex new generation tools and systems with outdated elements that have difficult access and scarce compatibility - all on one platform.



Pandora FMS currently uses agents for every ‘modern’ operating system on the market, describing “software agent ” as the part of that software installed in that system to extract information and report to Pandora FMS’ server.

Pandora FMS can, of course, be used successfully not only as a system monitoring tool, but as a monitoring tool for all sorts of network devices, whether it might use SNMP (versions 1,2,3) or TCP protocol probes (snmp, ftp, dns, http, https, etc), ICMP or UDP.

About the Documentation

All of this power and flexibility comes with an implicit difficulty at setup stages. In spite of Pandora FMS mostly graphical configuration, we are aware that learning how to use it seems complicated at first. That is why we have divided the 800 pages of the User’s Guide into several chapters:

- Chapter I. Understanding Pandora FMS.
- Chapter II. Installation and Configuration.
- Chapter III. Monitoring with Pandora FMS.
- Chapter IV. Operating and Managing Pandora FMS.
- Chapter V. Complex Environments and Best Performance.
- Chapter VI. Technical Appendices
- Chapter VII. Technical References

Besides the official documentation, you can access user's forum where you can post queries in English, Spanish and Japanese to other users. If you require official training, there is an official training program taught by the developers of Pandora FMS.

We have compiled some quick reference guides to help you configure Pandora FMS and implement simple monitoring tasks with Pandora FMS tool. There are also quick reference manuals available, for the installation of software agents, such as Windows and Linux.

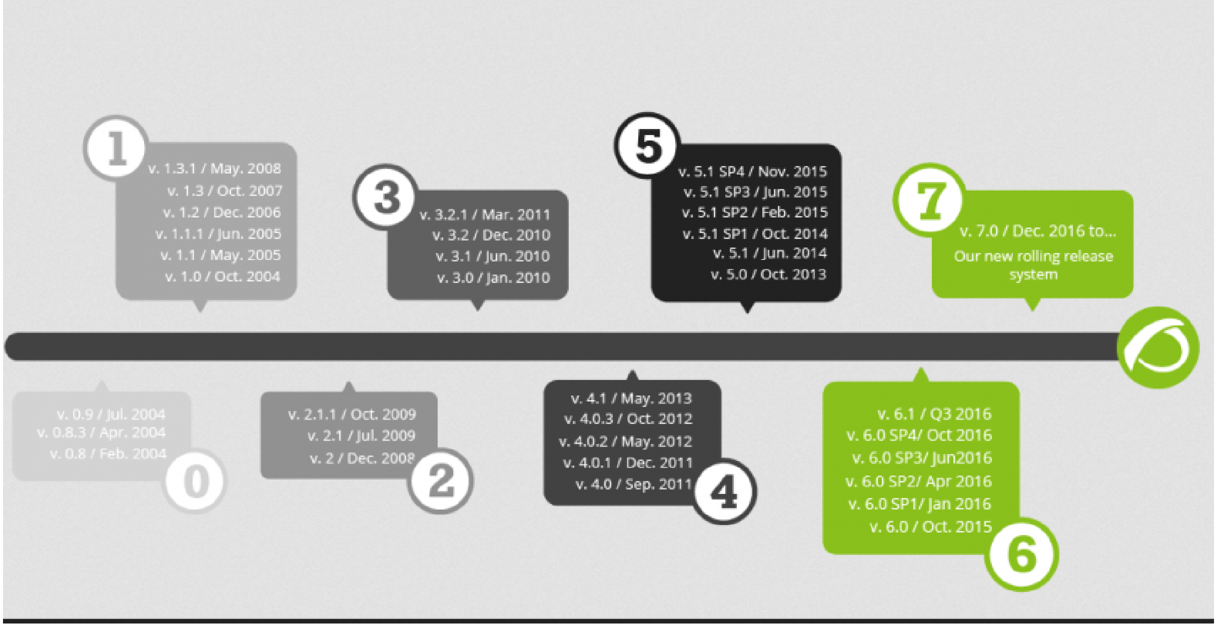
More detailed information about all of the above can be found on our website at http://pandorafms.com

The Evolution of Pandora FMS as a Project

Pandora FMS was born out of a personal development project of its original author, Sancho Lerena, in 2003. Since then, it has gradually evolved to become the resilient, innovative and flexible monitoring tool we offer you today.

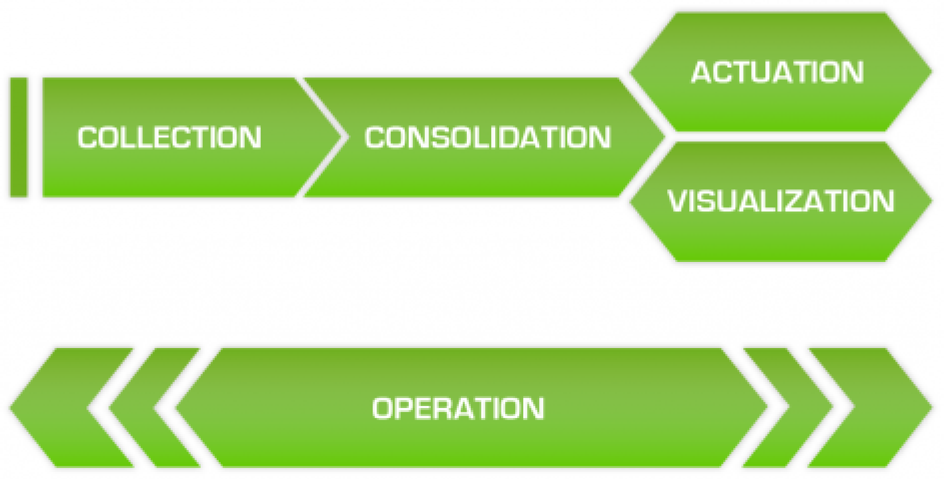
Originally written 100% in open source code, it went through years of experimentation and growth and, after strong demand for the product from large companies and corporations, we felt compelled to launch the Enterprise version. This version offers some specific features designed for conditions which require processing large amounts of information while properly operating with thousands of devices.

The company financing and coordinating all the back up work on Pandora FMS's development is Artica ST, a Spanish company, founded in 2005 by the creator of Pandora FMS.



Pandora FMS can be found to this day among Sourceforge’s top rated, with thousands of downloads and satisfied users all over the world. For more information on Pandora FMS's evolution and to see a road map of the project, please visit http://pandorafms.com

A Quick Glance at Pandora FMS features



- **Auto monitoring**. The default monitoring of Pandora FMS agents allows to detect hard disks, partitions or databases in a database server, among many other things.
- **Auto discovery**. Remotely, using the network, you can detect all network elements, catalog them according to your operating system, and given a profile start monitoring them. It can even detect network topology and build a network scheme based on its routing.
- **Monitoring**. The Agents of Pandora FMS are the most powerful on the market. They are capable of obtaining information - from the execution of a command to the call, at its most basic level- on the Windows API: Events, logs, numerical data, process stages, memory and CPU consumption. Pandora FMS makes use of a default monitor’s library, but one of the greatest advantages of Pandora FMS is how **easy** it is to quickly add, edit and create new monitors.
- **Control**. The agents themselves can activate services, delete temporary files or execute processes. Commands can also be executed remotely from the console, like stopping or starting services. Furthermore, it is possible to program tasks that require periodical execution. In addition, you can use Pandora FMS to access remote systems remotely thanks to *eHorus*, and even use tools like Telnet or SSH, all from a web interface.
- **Alerts and Notifications**. Notifications are just as important as failure detection. Pandora FMS gives you an almost endless variety of notification methods and formats. This includes - but is not limited to - escalation, correlation of alerts and prevention and mitigation of cascading events.
- **Analysis and display**. Monitoring is not just receiving a trap or having a failing

service displayed. Within the Pandora FMS environment, monitoring is also a method to present forecast reports, correlated summary charts of long term gathered data, and to generate user portals, delegate reports to third parties or to define its own charts and tables. Pandora FMS incorporates all of these tools within a Web interface.

- **Inventory**. Unlike other solutions where the idea of CMDB is the base, in Pandora FMS it is an option. The inventory is flexible and dynamic (it can auto-discover, accept remote input, etc.). It can notify changes (e.g. uninstalled software) or simply be used to make listings.

Introduction to Monitoring

Right from the start, every technical manual of a software package will tell you all about configuration, text files, databases, protocols, etc. We very often learn to configure at low levels while remaining ignorant of the full potential of the software under discussion - what can be done with it and in which situations. The purpose of this section is to explain the theory behind monitoring in a brief but systematic way, regardless of the software used for this purpose.

Types of Monitoring

When wondering about the condition of a target item to be monitored, whether it might be a server, a data base, a web element, or a refrigerator, the following questions might arise:

1. How is the information obtainedfrom the target(s)? Is there already something to make this happen, or is it necessary to “ask around”?
2. Is it better having to constantly ask the target's status or waiting for the target warn something has happened?
3. What sort of information does the target provide? Is it something that can be measured in a graphical way and whose progress can be observed?

All of these questions answer the three key points that shape the essence of our monitoring model.

The first question dictates whether an agent-based monitor will be used inside the surveilled device or, on the contrary, monitoring will be carried out externally, by means of an internet connection. There are monitoring systems that operate one way or the other, and devices that can only be monitored via either model. Pandora FMS supports every model.

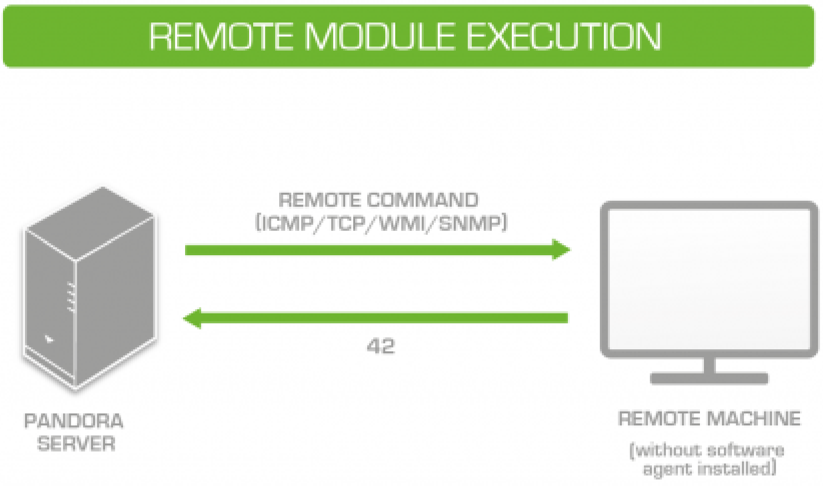
The second question concerns whether the monitoring is synchronous (every X number of seconds it asks itself, regardless of any information changes taking place or not) or asynchronous (it only receives information when something relevant has taken place). When using synchronous monitoring with 10 million elements, collecting data at 5 minute intervals will create a considerable load, but if it is done every 50 minutes instead, it will be much more manageable, the down side of the second option being that if something takes place in between, it will take 50 minutes before finding out about it. Using asynchronous monitoring (e.g. with SNMP traps or logs) can save many processing resources, but it will not be possible to draw graphics or create historical graphs, except those directly related to the incidents that occurred. Many tools are based solely on one of the models, sometimes known as 'performance' or 'capacity' tools, and there are other tools based on event management. They are not often exchangeable in their functions. However, Pandora FMS supports both approaches.

The third question refers to what is relevant in a given moment in time. The result can be a text chain (a descriptive event), a floating point number (to be able to draw graphics) or simply a status (down, up). Being able to work with different kinds of data allows more flexibility. Pandora FMS supports all types of data.

These three “paradigms” condition the monitoring environment greatly, and dictate the appropriate tool chosen to monitor it. Acknowledge the type of information needed and the best approach to obtain it. Plan around the available information elements and how to monitor them.

Remote Monitoring

Remote monitoring means that Pandora FMS’s server probes, ('polling') in a **synchronous** way, the devices it intends to monitor. Remote Monitoring does not refe to the 'local' monitoring, based on agents installed on the devices to be observed.


REMOTE MODULE EXECUTION
REMOTE COMMAND (ICMP/TCP/WMI/SNMP)
42
PANDORA SERVER
REMOTE MACHINE
(without software agent installed)

Generally speaking, remote monitoring is done with two different purposes:

- To make sure something is 'alive' (e.g. interface, or active system)
- To obtain a numerical value (e.g. to measure the web traffic or the number of active connections)

Synchronous monitoring is always conducted in the same direction: From the monitoring server to the monitored element (target).

The opposite process may also be interesting: receiving a notification when an incident takes place. This is called **asynchronous** monitoring, and in case of remote monitoring, we usually refer to it as SNMP traps.

Synchronous monitoring is usually done by using the SNMP protocol, which is the most widely used in methodology for observing and collecting status-related information. WMI, a similar protocol owned by Microsoft, is an alternative method of observing and collecting status-related information.

Basically, both protocols work in a similar fashion, which is as follows: A server sends a request for a particular configuration element of the ‘SNMP agent’ or ‘WMI service’ available in the target device. This particular element is called OID, in SNMP and in WMI it can be identified by a WQL query. The request could be for the free available memory, the router’s number of connections or the traffic in a given interface - or a wide variety of other reportable information.

If the monitoring is mainly based on internet environments, it is important to know SNMP **in detail**, since it will be the monitoring tool's most widely used function. The asynchronous monitoring through SNMP is also vital. Together with a monitoring tool, you will need an external explorer of SNMP devices, access to the MIBS collections from the makers of your target devices (which are like OID’S libraries) and, of course, a lot of patience to investigate, given that each device usually has its own collection of OID’s, but among the thousands that each device has, you will only be interested in some of those elements.

If you are monitoring Windows servers and you are not interested in installing agents on the machines, WMI remote monitoring can be very powerful and well suited. The WMI interface is even more powerful (and better organized than SNMPs). With WMI, you will be able to obtain almost any data, status or event on your Windows servers.

Unix and Windows systems can also use SNMP, but the information returned is limited. Furthermore, you will need to activate and configure the SNMP agents of the operating system, which can be much more complex than simply installing a Pandora FMS monitoring agent.

Finally, you can always monitor networked elements through the use of TCP or ICMP tests. ICMP is mainly used for two purposes:

- To verify whether a system responds (ping)
- To find out the latency time of that device (in milliseconds)

Through TCP tests, it is possible to test whether a web server responds properly, or a mail server (SMTP) sends the mail properly and in a timely fashion. These types of tests are not intended to just get the server to 'open the port' but also to get it to 'communicate': that is, the sending mail command receives an OK to confirm its functionality or the answer from the web server is ‘200 OK’ (a valid reply in the HTTP protocol).

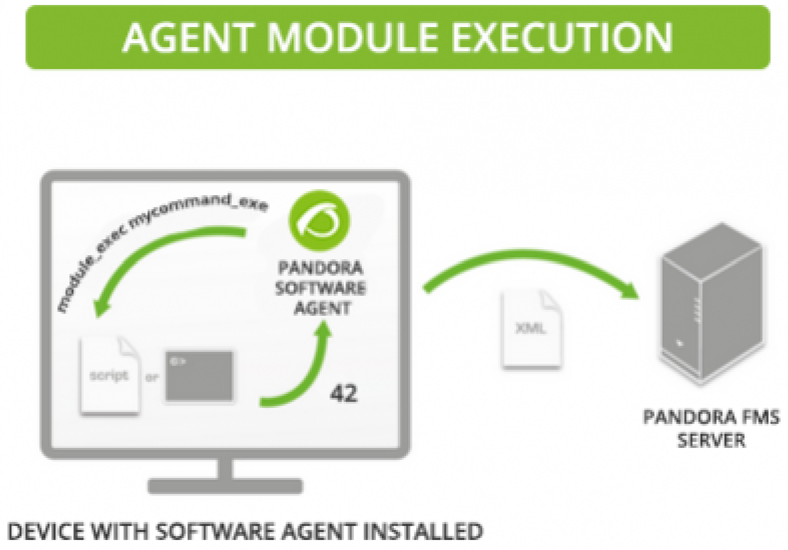
By default, Pandora FMS supports a series of plugins for TCP testing, but it can easily implement its own tests by adapting its own scripts or developing new ones. Integration with Pandora FMS does not require an API, complex structures or proprietary libraries.

Given the importance of the topic, Web Transaction Monitoring and remote monitoring receive a separate chapter.

Local Monitoring (by Agents)

Regarding systems and applications, the best way to obtain information is definitely from the target system. This is done by executing commands, or querying the system data sources from the same machine to be monitored. This means executing a command or script, or to investigate the system or the application. To that end, use Pandora FMS monitoring agent, a specific software module to take care of those small monitoring tasks.

According to the nomenclature used by Pandora FMS, ‘agent’ is used to refer to the entity containing the information and ‘software agent’ as the part of that software installed in that system to retrieve information and report back to Pandora FMS server. The software agent is executed constantly in the system (as a service) and reports information periodically.



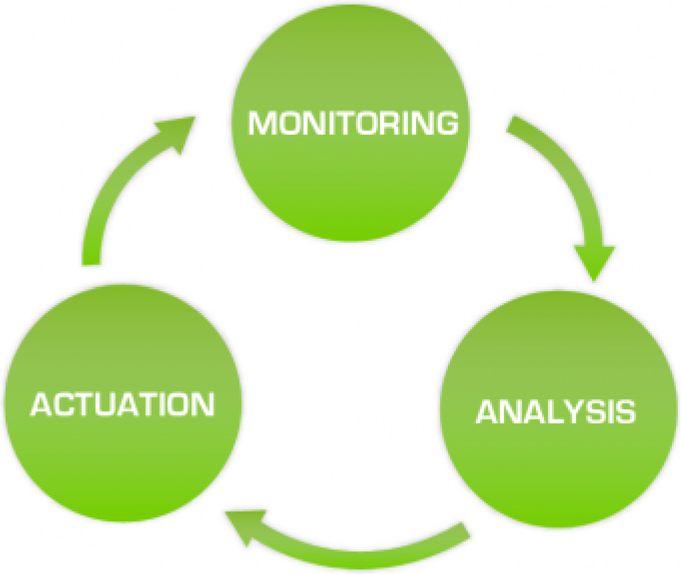
The agents allow to do more than obtain information through commands, for example to obtain inventory information. Agents can also be configured to react in case of a problem or a failure, interacting automatically with the system, deleting a temporary file or executing a given command.

To obtain *precise and specific* information that may be relevant, refer to the manuals of the application to monitor, because even when having 'generic' monitors, internal application monitoring involves some additional complexity and specific elements.

In Windows, there is a wide range of accesses to the information: WMI, Perfcounters, Eventlog, system logs, registry, commands, powershell scripts, API (by Windows NT) etc. In fact, Microsoft's architecture is one of the easiest, most powerful and best documented, when it comes to obtaining the information from the system. In Unix / Linux systems, the capability of the software agent to execute any command allows you to benefit from the full power of the shell.

The Monitoring Procedure

Before starting deploying, is is important to set the key points of the technological platform to be monitored. That way, before having information about specific data on the systems, it is clear what it is for and how to make full use of it without wasting time on researches or trivial things.



In your case, what do you think describes your monitoring needs better?

- To avoid losses → Availability.
- To analyze degradations → Performance.
- To evaluate growth → Capacity planning.

For each of those answers, the focus of your monitoring solution will be different in certain aspects.

Availability You are mostly interested in event-based monitoring and remote monitoring will probably be enough for your needs. It is faster to deploy and will give you fairly quick results. SLA reports will be the most useful in this case.

Performance Its strength is graphics and numbers, collecting information through agents or remotely, even though you will probably require agents to get in-depth information on their systems. Group reports and combined graphics are your primary interest.

Capacity Planning Much more specific. It is necessary to obtain data, as in the second instance, but to parse and manipulate the data, with predictive monitors and very specialized projective reports. Establishing early alerts will be of great help and you are required to have good knowledge of the WARNING and CRITICAL status meanings, besides elaborating serial event management policies to prevent the problem from happening, which is without a doubt the most complex and interesting case.

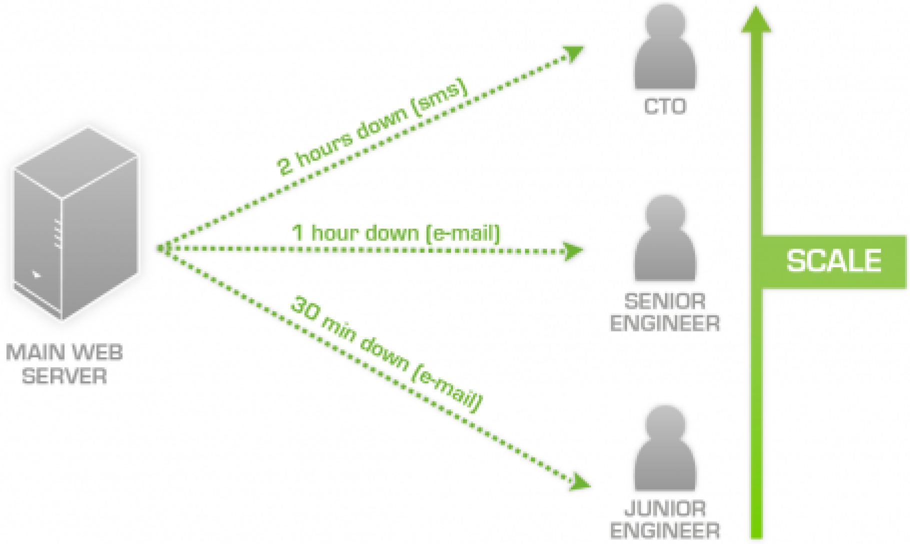
Once you know which model you will follow, you are left to wonder what to do when the system tells you the service is down, or worse, what will happen if the server's capacity reaches its limit next Friday?

You need to think about action procedures.

Action Procedures

In order to be able to draw up action procedures, it will be necessary to take into account several factors:

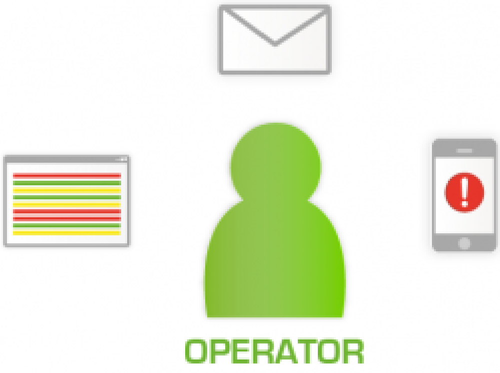
***Urgency of the event**: being able to distinguish something normal from something rare or critical. ***Form of notification**: email, sms, Telegram, sound alert… ***Scaling**: different forms of warning in face of a recurrent problem. A common case is notification to a manager after a certain amount of time without solving a problem. Before entering any configuration, it is advisable to have these concepts clear, draw up schemes with the critical elements, how to monitor them, what to do with all the information gathered and how to report problems that arise.



By focusing on the most critical issues first, you reach a logical starting point that defines **what** the most important issues for your organization are. Once you know what the most critical elements are, you can define **how** to monitor the target(s), while considering **who** will be responsible for the resolution of the reported problems in those systems as well as how to notify the appropriate people of the existence of a problem.

Supervision Models

By supervision models, we are stating that a monitoring system is designed to report information and work automatically, but this is supervised by a human being in a direct or indirect way. This person often receives the title of *operator*, which is the person who looks at the screen or otherwise receives the events, whether it might be by means of a smartphone device or similar, by e-mail or logs registered with another tool. The “how” does not matter, the important thing is the fact that someone is minding the system.



On the other hand, there are certain people called *system administrators* in general or *infrastructure staff*, those who, when something happens, receive a call from the operator saying: “Hey, we have got a problem here,” or a direct notification sent automatically by the system, warning them of an event, which is frequently sent by SMS or email.

Here we can already see the differences:

- The **direct supervision model** implies a person or several people, constantly watching over the system, so that if something critical takes place, it will be detected immediately. The monitoring package can usually notice small, non-critical changes, and has much greater flexibility in how it reports this information. It is not necessary to define 'notifications' (alerts under Pandora FMS) for each possible case, but rather it is enough to go through the events (some sort of visual indicator to detect status changes) to have an idea of what is going on in the system at any given time. It is possible to define many screens and also to define alerts to support that supervision. This model is used in large environments, given that it does not matter how much we define an alert policy.
- The **indirect supervision model** implies that there is no one permanently looking at the screen, so it is necessary to define, beforehand, the automatic notifications (alerts) that the system is going to have; given that the events, graphics and maps are not going to be looked at by anyone. This system is suitable when having few devices, or when what is critical has been already identified as well as how to face the problem (solution and notification).

For teamwork that involves operators, administrators and third level personnel, Pandora FMS provides meaningful tools like: event ticketing, incident creation, notification scaling, internal mail, notice board and chat among the users of Pandora FMS.

And what Now?

The following chapters are exclusively devoted to Pandora FMS. Up to this point, we have been discussing general issues which were probably important for you to know before we continue to explore Pandora FMS. You probably know many of these things already. You may have used other monitoring programs. Perhaps, you may have heard that this or that application is always monitored in a certain way because it is the best way possible.

Maybe, but from our experience, each client works in a certain way and regardless of how much we know about monitoring, we may not know more about how your infrastructure was configured than you do. Monitoring easy tasks presents no problems, the hard job is to adapt monitoring to your business without having to adapt your business to monitoring. Not an easy task. More than 800 pages await, if you wish to discover the best way to monitor your organization with Pandora FMS. It is a challenge, but one we believe is well worth the effort.

From:
https://pandorafms.com/manual/ - **Pandora FMS Documentation**

Permanent link:
https://pandorafms.com/manual/en/documentation/01_understanding/01_introduction

Last update: **2021/07/27 08:49**