



Distributed monitoring with Satellite server



<https://pandorafms.com/manual/!775/>

Permanent link:

https://pandorafms.com/manual/!775/en/documentation/pandorafms/complex_environments_and_optimization/05_satellite

2014/03/18 21:03

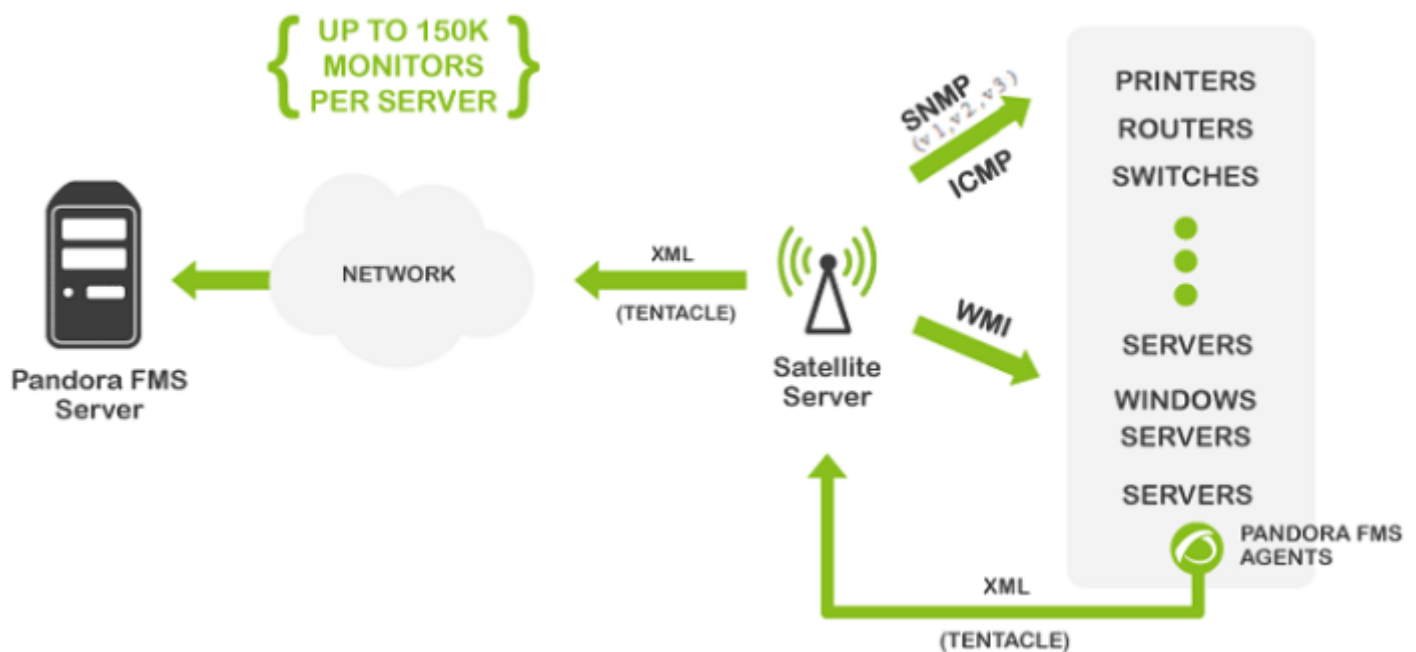


Distributed monitoring with Satellite server

Introduction

E

The Satellite Server is used for **network and remote system both monitoring and discovery**. It can discover network elements (routers, switches, etc) using SNMP or ICMP, or MS Windows® servers (using WMI) and Linux® servers (using SNMP). This is no “ordinary” server, it can be considered to be an agent in **broker mode** with extended features. It is particularly useful to monitor inaccessible remote networks where a software agent is not even an option from Pandora FMS server.



The Satellite server can be used in Windows® and GNU/Linux® (recommended OS) alike, and it has some features that make it more special, highly recommended in certain environments.

- It can execute network tests (ICMP, Latency and SNMP v1 and v2) at an extremely high pace (500 checks per second). For SNMP v3 **configure the access credentials** and due to data encryption the pace of the test will not be that fast.
- It only sends information to the server every N seconds (300 seconds by default), but it can execute latency, ICMP and SNMP tests within a smaller interval (30 seconds for example). That way, it can warn Pandora FMS Server almost instantly when there is a status change. These status changes must have been previously defined if the module type is not *_proc (network interfaces or general network connectivity for instance).
- It does not require connection to the database, rather it is autonomous. It sends all files in XML format the same way as an independent server, similar to a broker agent or an export server.
- It has an auto-discovery system for SNMP and WMI. It creates detected agents (by IP address), it

detects dynamic elements (network interfaces, storage) and monitors them automatically.

- In Windows® systems, it can detect hard drives, CPUs, and memories.
- In systems with SNMP, it can detect interface status, inbound and outbound traffic for each interface and the name of the system.
- Auto-generated modules can be modified, like every other module, managing the agent from the console as if it was an ordinary agent (in *Mass operations menu > Satellite*).
- Agents can be created manually by creating an agent configuration file in the Satellite server directory for agent configuration files (explained later on).

759 NG version or later.

- Since version 759 NG, both the Satellite server and the Enterprise Network Server support IPv6 in all its advanced functionalities. High-performance code that was previously only supported in IPv4 now also applies to IPv6, enhancing existing polling capabilities.

Capacity

The maximum capacity of the Satellite server depends entirely on the server hardware where it runs and the type of checks you want to perform. In the test environment, 500 checks ICMP/SNMP per second have been made, but that depends a lot on the response times of remote devices (it is not the same a device which answers in 0.5ms than one that takes 2 seconds to answer back). Under ideal conditions, an amount of 150,000 checks could be monitored with a single Satellite server. In real conditions, it has been tested in controlled environments (LAN) made of about 50,000 modules with a single Satellite server in a low-end computer hardware (Intel i5, 2GZ, 4GB RAM).

If there are many critical modules, performance will be affected. Take into account the configured timeout, since there is only one check for each critical monitor for timeout. If there are 1000 critical modules and the timeout is configured to 4 seconds, it will take 4000 seconds to execute all the checks with only one thread.

Installation

The Satellite Server is distributed as tarball (GNU/Linux® or .exe (Windows®)), so there is no need to install Perl nor any additional library. It works the same in Windows® or Linux® versions. In Windows® systems, it is installed as a service, and in Linux® systems, it is installed as a daemon. The configuration file and specifications in both cases are the same.



Satellite server Linux® version depends on external packages that are specified in the corresponding version of this documentation.

Online installation tool

E This is a special Pandora FMS feature. An Enterprise license is required for its use. In any case, the mandatory installation parameter is the IP address or FQDN of a Pandora FMS Enterprise server. Please contact the sales team, ask for a quotation or solve your doubts about licenses [in this link](#).

This tool is compatible with Rocky Linux 8.x, AlmaLinux 8.x and RHEL 8.x.

Requirements for the use of the online installation tool:

- Internet access.
- Have curl installed (comes by default in most distributions).
- Meet minimum [hardware requirements](#).
- Be root administrator user.
- To have a supported OS.
- In the case of using RHEL 8, it will be necessary to be previously activated with a license and subscribed to the standard repositories.

To use the online installation tool, just access the command line as provided by your Cloud provider as root administrator user and run:

```
export PANDORA_SERVER_IP='<PandoraServer IP or FQDN>' && curl -Ls  
https://pfms.me/satellite-ent-deploy| bash
```

Custom installation using the online installation tool:

- PANDORA_SERVER_IP: IP address or FQDN of Pandora FMS Enterprise server to which the Satellite server will point. Mandatory parameter.
- TZ: Satellite server time zone. Optional parameter.
- SATELLITE_SERVER_PACKAGE: Custom URL of Satellite server installation tarball package. Optional parameter.

- `SATELLITE_KEY`: Satellite server license to be activated automatically. Optional parameter.
- `REMOTE_CONFIG`: Remote configuration. Optional parameter, enabled by default (value 1).
- `INSTALL_AGENT`: Optional parameter, enabled by default (value 1), allows you to install the Software Agent (you may use all the configuration variables of the [agent's online installer](#)).
- `VMWARE_DEPENDENCIES`: Optional, it allows to install VMware® plugin dependencies, disabled by default (0).
- `ORACLE_DEPENDENCIES`: Optional, it allows to install Oracle® plugin dependencies, disabled by default (0).
- `MSSQL_DEPENDENCIES`: Optional, it allows to install MS SQL Server® plugin dependencies, disabled by default (0).
- `SKIP_KERNEL_OPTIMIZATIONS`: Disable the recommended kernel optimization, disabled by default (0).

Example:

```
env TZ='Europe/Madrid' \  
SATELLITE_KEY='SOPORTEDEV00RS0REB3M2T7ZHIS051IIQH52JISJ47VGHIRM... ' \  
PANDORA_SERVER_IP='192.168.10.10' \  
REMOTE_CONFIG=1 \  
INSTALL_AGENT=1 \  
VMWARE_DEPENDENCIES=1 \  
ORACLE_DEPENDENCIES=1 \  
MSSQL_DEPENDENCIES=1 \  
SKIP_KERNEL_OPTIMIZATIONS=0 \  
sh -c "$(curl -fsSL https://pfms.me/satellite-ent-deploy)"
```

Satellite Server Installation in Linux Systems

The recommended GNU/Linux operating system is RedHat Enterprise (RHEL) 8 / Rocky Linux 8.

You need to install Fping, Nmap and libnsl independently and you must first configure the EPEL repository, visit the following link:

```
https://docs.fedoraproject.org/en-US/epel/#\_quickstart
```

and select the operating system. If Rocky Linux 8 is used:

```
dnf config-manager --set-enabled powertools  
dnf install epel-release
```

Install Perl with the following command:

```
dnf install perl
```

Satellite Server core dependencies: PandoraWMIC (version 762 and later), Fping, Nmap and libnsl. The Braa and PandoraWMIC dependencies are attached to the installer.

```
dnf install fping nmap libnsl
```

Once the binary that contains the Satellite server is downloaded, go to the download directory with root privileges and unzip the binary:

```
tar -xvzf pandorafms_satellite_server_X.XNG.XXX_x86_64.tar.gz
```

Then, a folder called `satellite_server` will be created. Go there typing:

```
cd satellite_server/
```

To install the Satellite Server, execute the installing command:

```
./satellite_server_installer --install
```

```
[root@localhost satellite_server]# ./satellite_server_installer --install
Pandora FMS Satellite Server installer for GENERIC. (c) 2014-2015 Artica ST.

>Installing the Pandora FMS Satellite Server binary to /usr/bin...
>Installing the tentacle_client binary to /usr/bin...
>Installing the braa binary to /usr/bin...
>Installing the pandorafsnmp binary to /usr/bin...
>Installing the wmic binary to /usr/bin...
>Copying configuration file to /etc/pandora...
>Creating agent configuration directory /etc/pandora/conf...
>Copying startup script to /etc/init.d...
>Linking startup script to /etc/rc.d/rc2.d
Creating logrotate.d entry for Pandora FMS log management

Edit the file /etc/pandora/satellite_server.conf and manually configure the Satellite Server.

[root@localhost satellite_server]#
```

Once finished, edit the `satellite_server.conf` file located at:

```
/etc/pandora/satellite_server.conf
```

In versions prior to 761, the license must be entered manually in the `pandora_license` option. With a text editor like **VIM edit** the file, look for this token and proceed to uncomment it. Enter Pandora FMS Enterprise **server license**.

Search for the **server_ip** token and indicate there the IP address or domain of Pandora FMS Enterprise server to which the Satellite server will connect.

After that, save the file and activate the service, executing the following:

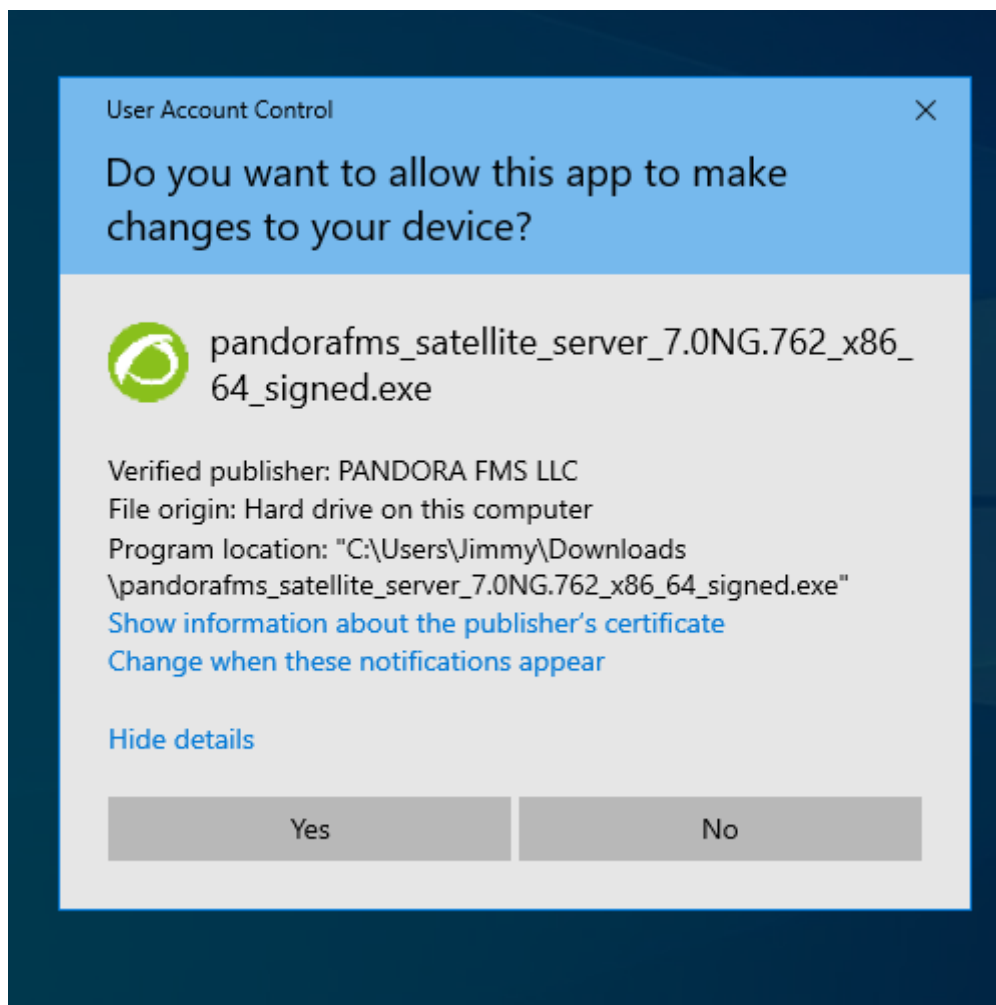
```
sudo /etc/init.d/satellite_serverd start
```

In case of failure, take a look at the log file at:

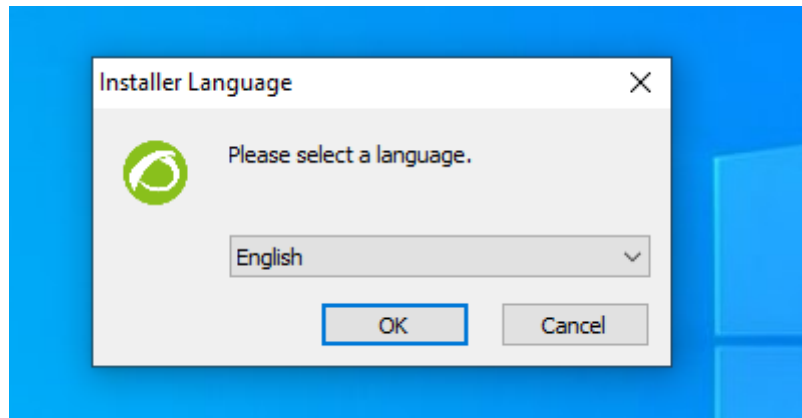
```
/var/log/satellite_server.log
```

Windows Installation

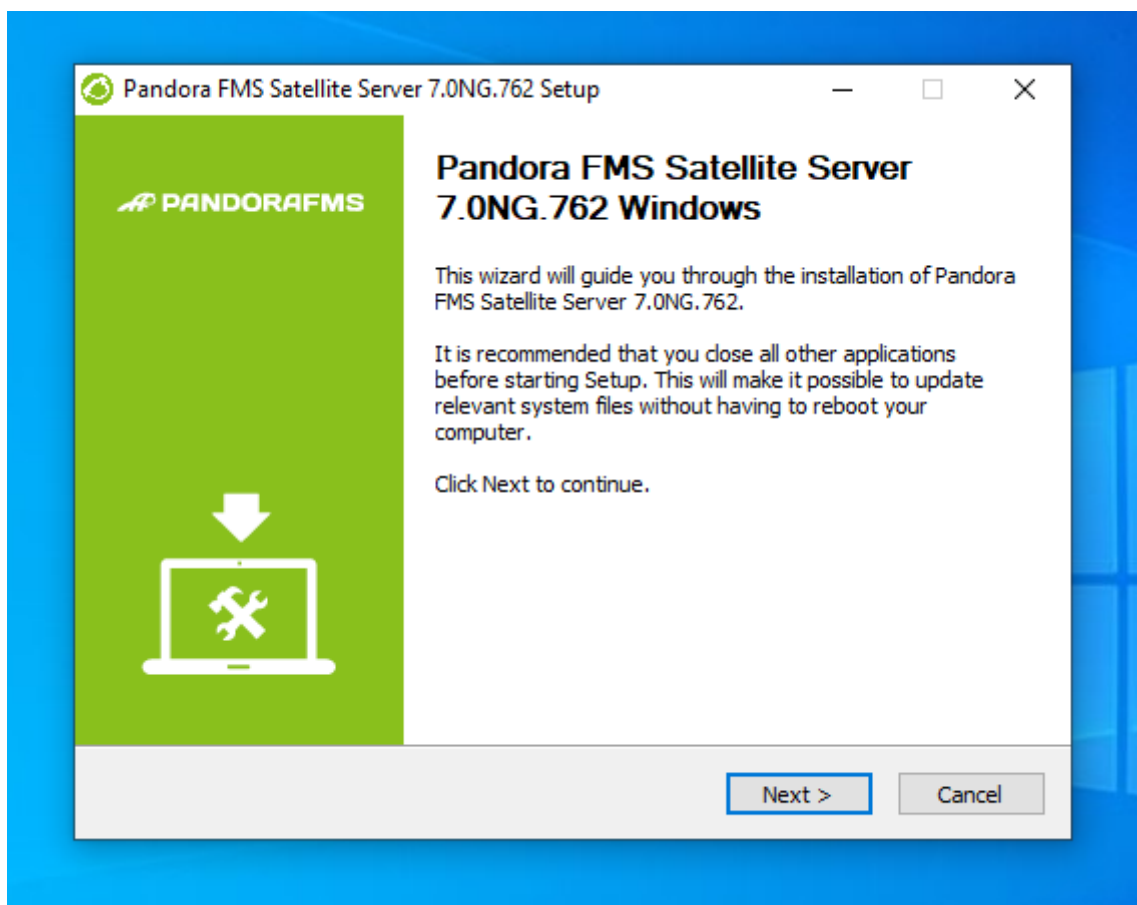
Run the digitally signed installer (version 762 and later), click Yes:



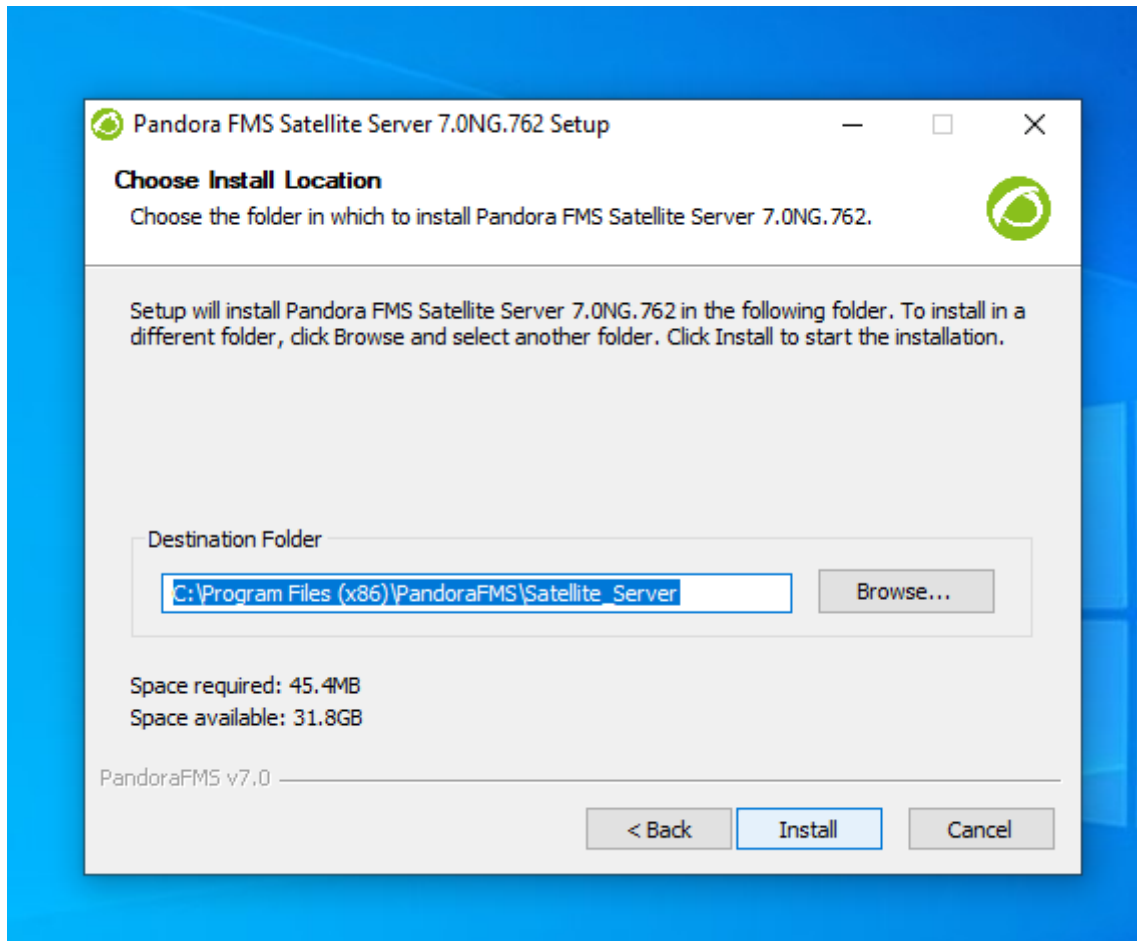
Choose the installation language:



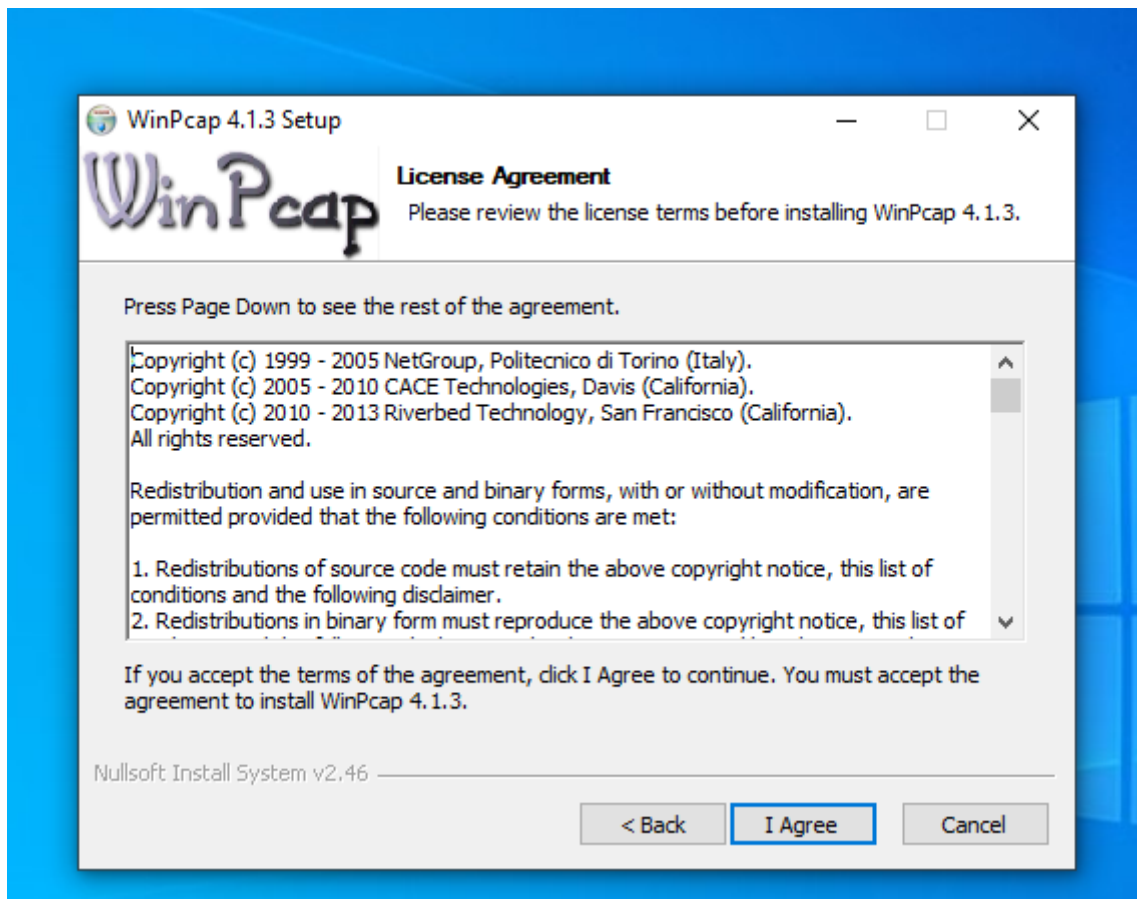
Click Next:



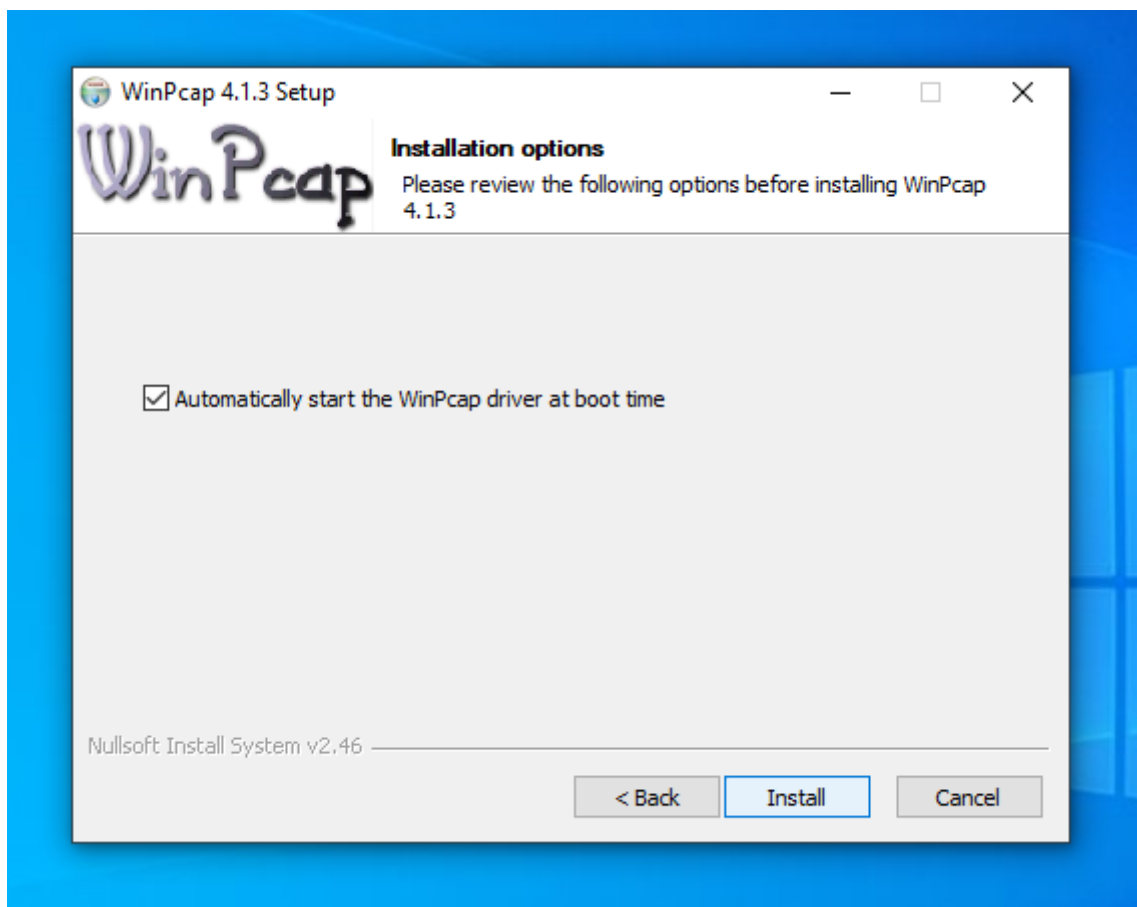
Choose where to install the program:



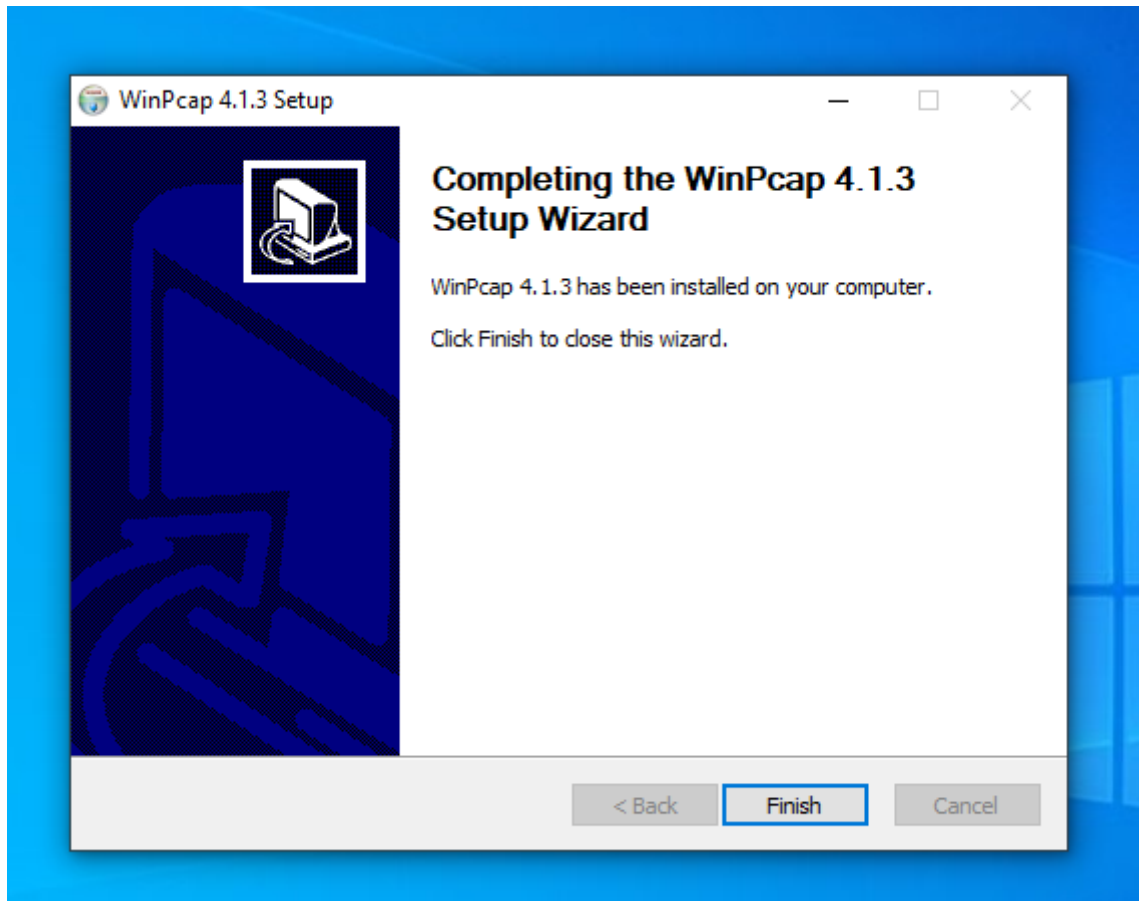
WinPcap installation is required. The installation window will appear at this step of the installation process:



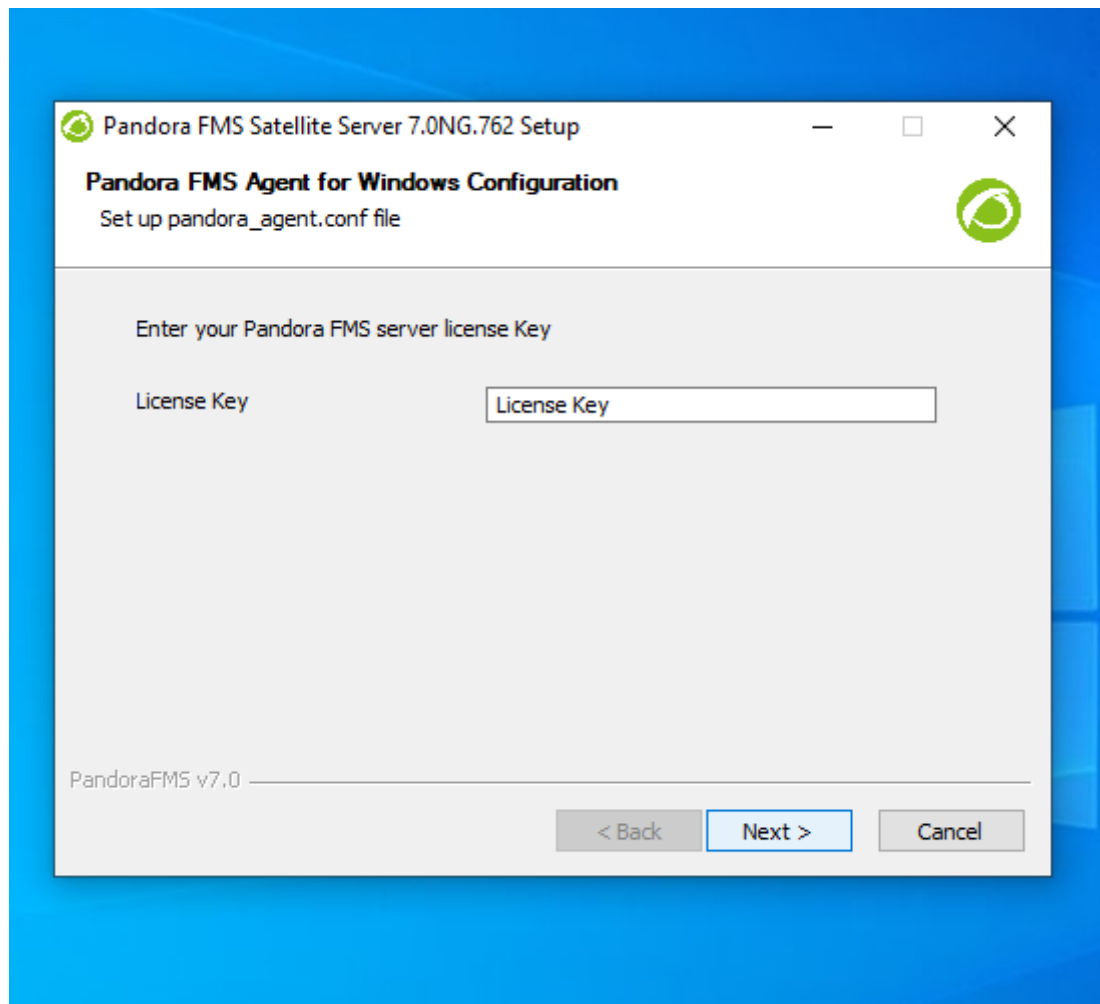
Configure WinPCap to start when the system starts.



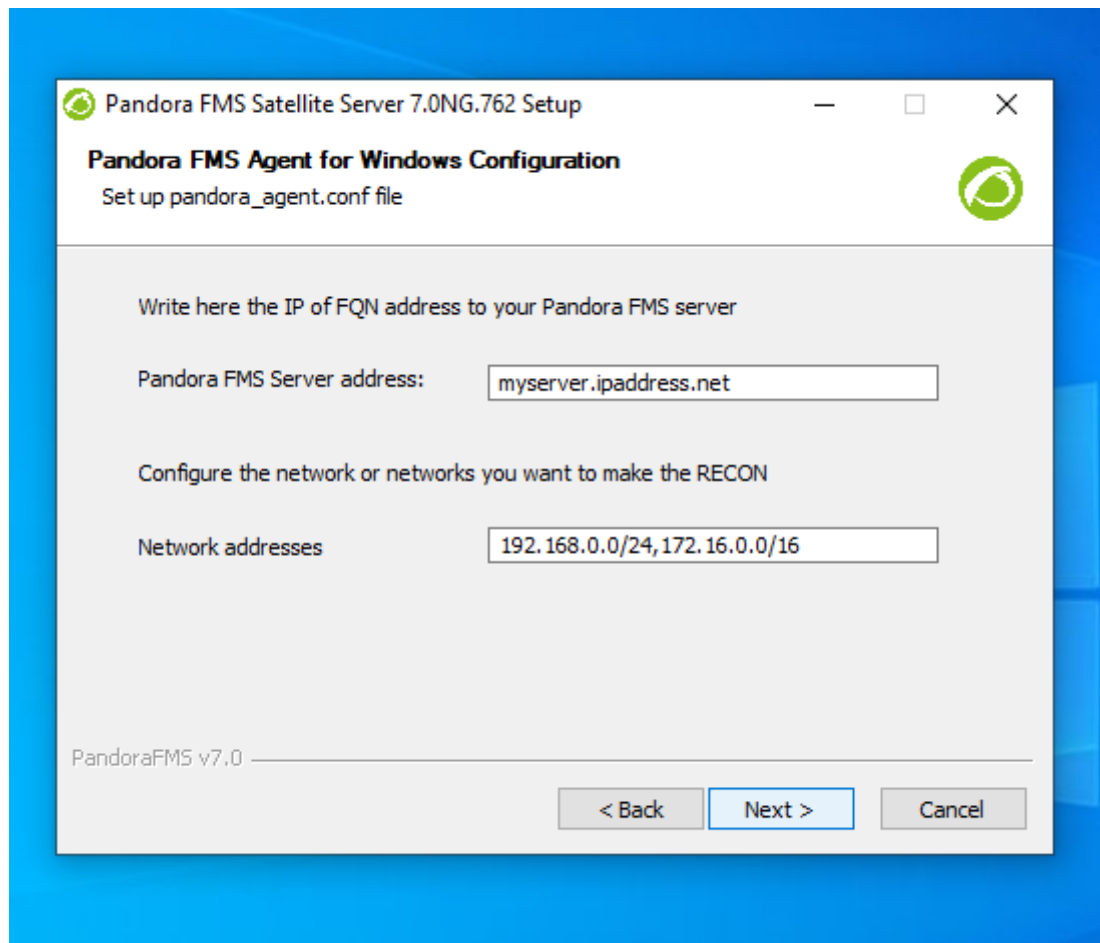
Once WinPCap installation is finished, this window will appear:



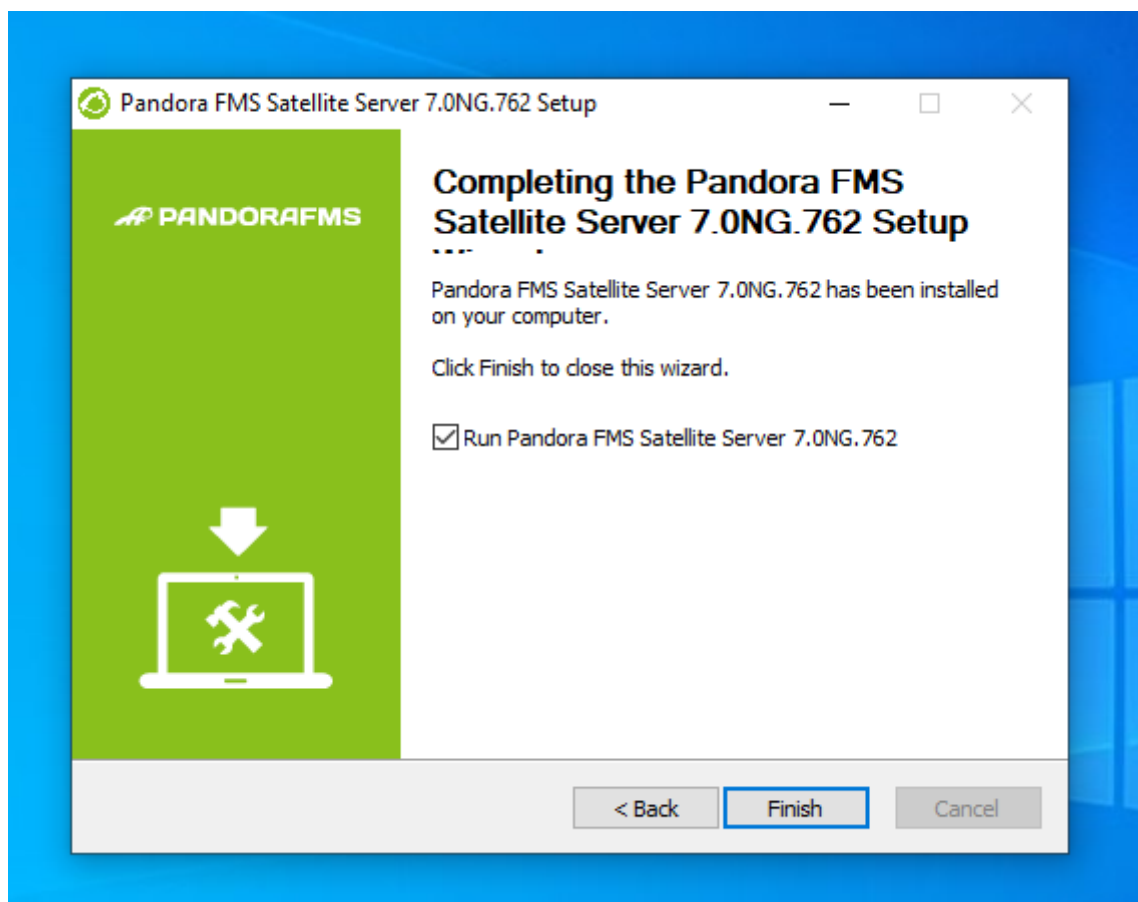
Enter Pandora FMS license number to continue the installation:



Then, set Pandora FMS server address to send data. Define the network recon rules for the Satellite server:



Restart the machine so that all changes are applied.



Once the process is finished, start and stop Pandora FMS Satellite server from the MS Windows® Start menu.

Depending on the year of your MS Windows version you will need to install some of these libraries:

Microsoft Visual C++ Redistributable (latest supported downloads):

- Visual Studio 2015, 2017, 2019, and 2022.
- Visual Studio 2013 (VC++ 12.0).
- Visual Studio 2012 (VC++ 11.0) Update 4.
- Visual Studio 2010 (VC++ 10.0) SP1 (no longer supported).
- Visual Studio 2008 (VC++ 9.0) SP1 (no longer supported).

They are available for both 32-bit (X86) 64-bit (X64) and ARM64 processors at the following link:

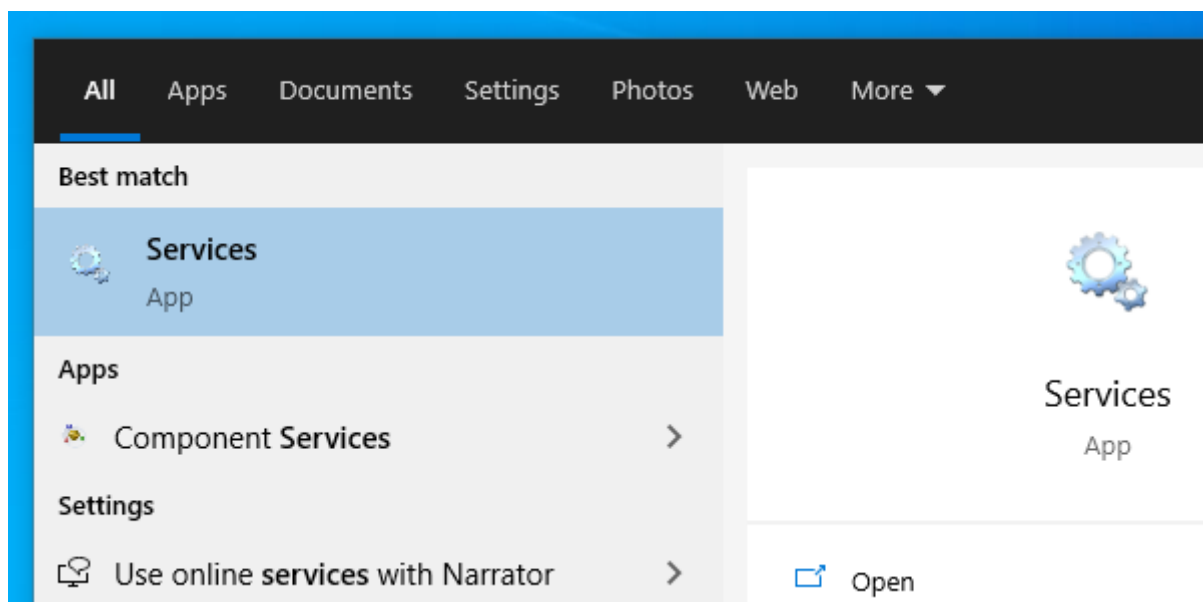
<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170&viewFallbackFrom=msvc-170>

WMI module operation in some Windows versions

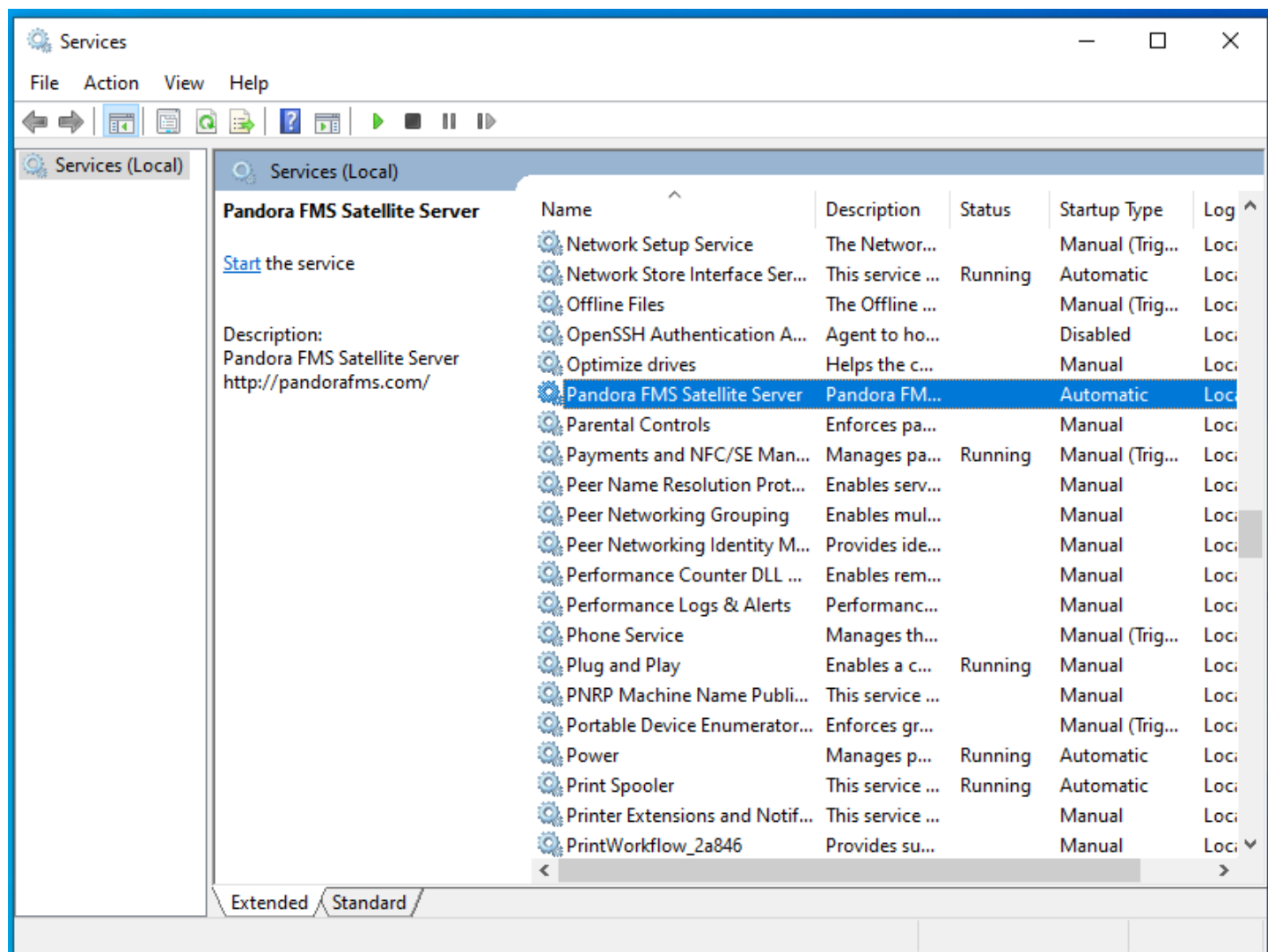
For Windows® security reasons, some versions have limited users who can carry out remote WMI queries. If these queries were not carried out, the solution would be to run the Satellite server service as an Administrator user.

The process to follow is:

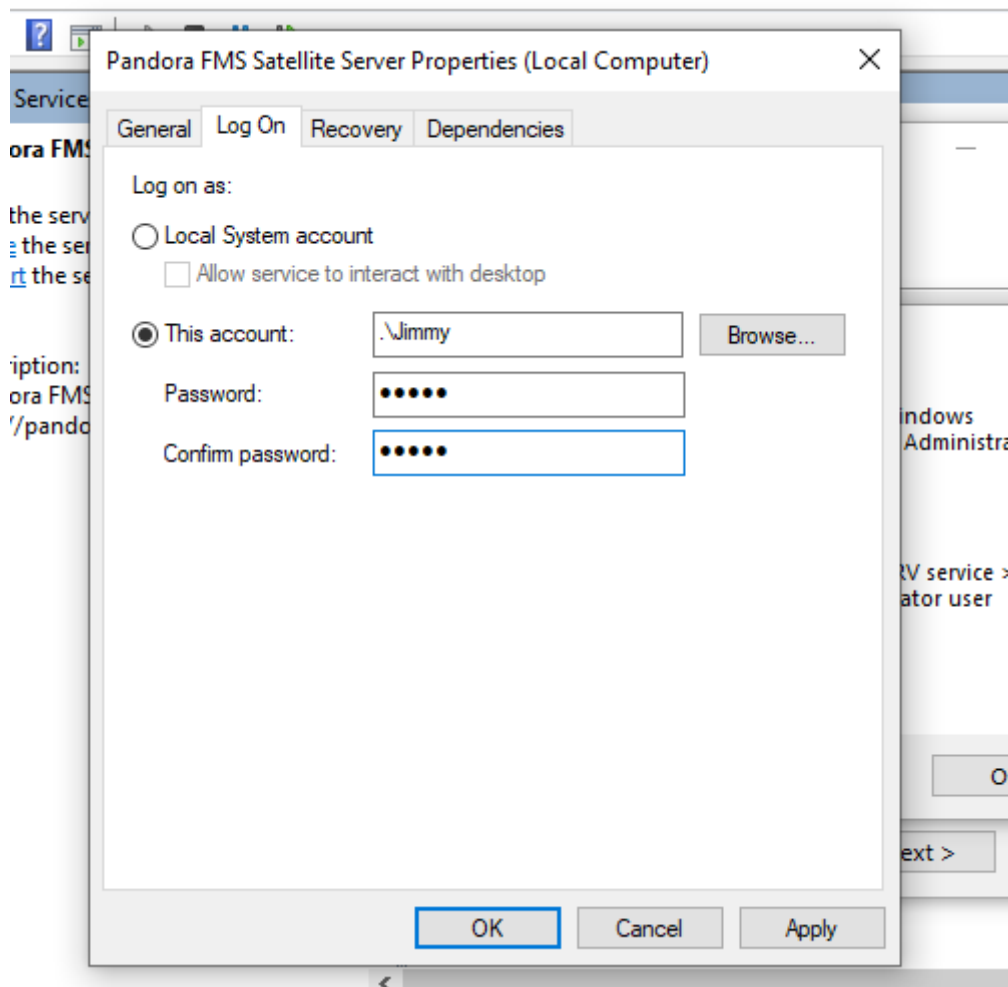
Open the services:



Right click on the service and go to Properties:



On the Log On window, select an account with Administrator permissions and apply changes:



Finally, restart the service to apply the changes.

Configuration

All parameters that require a timeout or some time are specified in seconds, (for example 300 = 5 minutes).

It is important to keep in mind that the latency and snmp intervals are specific for the status change. In case of Boolean checks (port or machine status) the threshold that defines the status change is automatic. For numerical values (latency, network traffic in an interface, disk space, CPU, etc), it is based on a threshold that must be defined in each module.

agent_interval

```
agent_interval xxx
```

300 seconds by default (5 minutes). After that time, information is sent to the server, *regardless of checks done by the network server having a lower interval*. If necessary and by default, it creates agents in the corresponding Pandora FMS server according to the specified time.

If the collected data is different compared to the previous one, it sends it right away. If it is the same, it will send it when the agent interval says so. It is useful to perform intensive checks and notify only in case of status change.

agent_threads

```
agent_threads xxx
```

Number of threads used for sending XML data files.

log_file

```
log_file <path_file>
```

It indicates the file where the Satellite server log is written, by default the path is `/var/log/satellite_server.log`.

recon_task

```
recon_task xxxxx[,yyyy]
```

IP networks and addresses for autodiscovery separated by commas, for example:

```
192.168.50.0/24,10.0.1.0/22,192.168.70.64/26
```

server_ip

```
server_ip <IP>
```

IP address or DNS name of Pandora FMS Server where the information is sent. It is done using the [Tentacle](#) protocol, so communication with the system must be possible through Tentacle port `41121/tcp`.

recon_mode

```
recon_mode <mode_discovery>
```

Autodiscovery mode (`<mode_discovery>`). The system will use the following protocols to discover systems:

- `icmp`: It will just check whether the host is alive (ping) and measure latency time.

- `snmp`: If it is capable of communicating by SNMP (only v1 and v2), it will look for all the interfaces and get its traffic from all of them, as well as its operative status and device name and location. It will try **different communities provided in the configuration file** to connect. To use SNMPv3 whose recognition is required, check [this link](#) on how to configure the known access credentials.
- `wmi`: Similar to the previous case, but in this case showing CPU usage, memory and hard drives (all available ones).

recon_community

```
recon_community <aaa>,<bbb>,<ccc>...
```

It states a list of SNMP `<xxx>` communities to be used in SNMP discovery, separated by commas. It will use this list in SNMP exploration: for each IP found, it will try to see whether it responds to any of these communities.

wmi_auth

```
wmi_auth Administrator%password[,user%pass]
```

It specifies a list of user credential pairs, each of them in `<username>%<password>` format and separated by commas.

For example: `admin%1234,super%qwerty`. It will use this list for WMI exploration. For each IP found, it will try to see whether it responds to any of the combinations.

wmi_ntlmv2

```
wmi_ntlmv2 [0|1]
```

It enables (1) or disables (0) authentication with **protocol NTLMv2** for WMI.

agent_conf_dir

```
agent_conf_dir <path>
```

Path (`<path>`) to the directory automatically created and stored by the configuration files of each agent created by the Sattellite server. By default `/etc/pandora/conf`. Said agents can also be **created manually**.

group

```
group <group_name>
```

It specifies the default group name `<group_name>` for agents created by the Satellite Server. For instance: "Servers".

daemon

```
daemon [1|0]
```

When set to 1, it starts the daemon in the background (value by default).

host_file

```
host_file <path_filename>
```

It is an alternative/complementary method for network scanning.

A file (`< path_filename >`) is provided with an address in each line. It can include the hostname followed by the IP address as well, so that the agent created bears that name and uses that IP for modules (e.g. `193.168.0.2 hostname`). It must be possible to send and fping to those addresses for them to be valid.

pandora_license_key

NG 765 version or later.

```
# Encryption key for the Pandora FMS license.  
# pandora_license_key
```

For safe license transmission to the Satellite server, you will need to configure in the [Web Console](#) or [Metaconsole](#) the same encryption key that you will place in this token.

See also the [server_ip](#) token.

pandora_license

Since version 761 and later the licensing of the Satellite server is done automatically and this token becomes obsolete.

```
pandora_license xxxxxxxx
```

Type in store Pandora FMS Enterprise server license, as shown in the section ►Setup → License from Pandora FMS console.

PANDORAFMS ←

Pandora FMS
the Flexible Monitoring System

Enter keyword 🔍

License
License management

License

Customer key

ARTICAQA0000Z6GN8PJ00WONPW6GCNPW6DZFRW8GZF9TPW7J4F
DUWNKR58D1RGWWNKR51DG5IFRS0KKV5CP2FFRXOLHV5CG4GB
FBQSMGDRW8D0FBQSMGDRW8D0FBQSMGDRW8D0FBQZ56J4KLVV

Support expires

2023/10/03

Current platform count

280
agents

Current platform count (disabled: items)

2
agents

NMS

disabled

You may use the same license in as many Satellite servers as you need, since the total of agents that use the license is verified in Pandora FMS server not the Satellite server.

remote_config

```
remote_config [1|0]
```

It enables remote configuration in detected agents by default. It is mandatory if you wish yo

manage them from the console after detecting them. It also activates Satellite server remote configuration. To find out more, see [Remote configuration](#).

temporal_min_size

```
temporal_min_size xxx
```

If the free space (in MB) in the partition where the temporary directory is located is smaller than this value, data packages are not generated anymore. It prevents the disk from becoming full if the connection with the server is lost during an extended interval for some reason.

xml_buffer

```
xml_buffer [0|1]
```

The default value is 0. If set to 1, the agent will save any XML data files that could not be sent to retry it later on.

In a safe UNIX environment, consider changing the temporal directory, since `/tmp` gives writing permissions to all users.

snmp_version

```
snmp_version xx
```

SNMP version to use by default (only 1 and 2c are supported). 1 by default. To use SNMP v3 check [this link](#) on how to configure the known access credentials.

Some modules could stop working if you change this value.

braa

```
braa <path>
```

Path (<path>) to the braa binary. Value `/usr/bin/braa` by default.

fping

```
fping <path>
```

Path (<path>) to the Fping binary. Value by default /usr/sbin/fping.

fsnmp

```
fsnmp <path>
```

Path (<path>) to the Fsnmp binary (SNMPv3). Value by default /usr/bin/pandorafsnmp.

latency_packets

```
latency_packets xxx
```

Number of xxx ICMP packages sent by latency request.

nmap

```
nmap <path>
```

Path (<path>) to the Nmap binary. Value by default /usr/bin/nmap.

nmap_timing_template

```
nmap_timing_template x
```

A value x that specifies the level of aggressiveness of Nmap, from 1 to 5. 1 means slower but more reliable, 5 means faster but less reliable. Default value: 2.

ping_packets

```
ping_packets xxx
```

Number of ICMP packages sent for each ping.

recon_enabled

```
recon_enabled [0|1]
```

It enables (1) or disables (0) equipment autodiscovery.

recon_timing_template

```
recon_timing_template xxx
```

Like [nmap_timing_template](#) but applied to network scanning.

server_port

```
server_port xxxxx
```

Tentacle server port number.

server_name

```
server_name xxxxx
```

Name of the Satellite server (by default it takes the machine's *hostname*).

server_path

```
server_path <path>
```

Path <path> where XML files are copied if [transfer_mode](#) is in local (by default `/var/spool/pandora/data_in`).

server_opts

Server parameters sent to Tentacle.

transfer_mode

```
transfer_mode [tentacle|local]
```

File transfer mode. It can be `tentacle` or `local` (by default `tentacle`).

snmp_verify

```
snmp_verify [0|1]
```

It enables (1) or disables (0) the verification of SNMPv1 modules that make braa fail in real time.

These modules will be discarded and stop being executed. See both [snmp2_verify](#) and [snmp3_verify](#).

snmp2_verify

```
snmp2_verify [0|1]
```

It enables (1) or disables (0) the verification of SNMPv2 module that make braa fail in real time. These modules will be discarded and stop being executed. See both [snmp2_verify](#) and [snmp3_verify](#).

Verifying SNMP version 2 modules can take lots of time!

snmp3_verify

```
snmp3_verify [0|1]
```

It enables (1) or disables (0) the verification of SNMPv3 modules that make braa fail in real time. These modules will be discarded and stop being executed. See both [snmp2_verify](#) and [snmp3_verify](#).

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_seclevel

Security level used for SNMPv3 messages (noauth, authnopriv or authpriv).

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_secname

Security name used for SNMPv3 messages.

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_authproto

Authentication protocol (md5 or sha) for authenticated SNMPv3 requests.

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_authpass

Authentication password for authenticated SNMPv3 request.

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_privproto

Privacy protocol (des or aes) for encrypted SNMPv3 requests.

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

snmp3_privpass

Privacy password for encrypted SNMPv3 messages.

To use SNMP v3 check [this link](#) on how to configure the known access credentials.

startup_delay

```
startup_delay xxx
```

It waits xxx seconds before sending XML data files for the first time.

temporal

```
temporal <directory>
```

Temporal directory where XML files are created, /tmp by default.

tentacle_client

```
tentacle_client <path>
```

Full <path> to the Tentacle client (/usr/bin/tentacle_client by default).

wmi_client

```
wmi_client <path>
```

Full <path> to the wmic. By default /usr/bin/wmi.

```

â euclides root ~ /usr/bin/wmic --usage
Usage: [-?|--help] [--usage] [-d|--debuglevel DEBUGLEVEL] [--debug-stderr]
[-s|--configfile CONFIGFILE] [--option=name=value]
[-l|--log-basename LOGFILEBASE] [--leak-report] [--leak-report-full]
[-R|--name-resolve NAME-RESOLVE-ORDER]
[-O|--socket-options SOCKETOPTIONS] [-n|--netbiosname NETBIOSNAME]
[-W|--workgroup WORKGROUP] [--realm=REALM] [-i|--scope SCOPE]
[-m|--maxprotocol MAXPROTOCOL] [-U|--user [DOMAIN\]USERNAME[%PASSWORD]]
[-N|--no-pass] [--password=STRING] [-A|--authentication-file FILE]
[-S|--signing on|off|required] [-P|--machine-pass]
[--simple-bind-dn=STRING] [-k|--kerberos STRING]
[--use-security-mechanisms=STRING] [-V|--version] [--namespace=STRING]
[--delimiter=STRING]
//host query

Example: wmic -U [domain/]adminuser%password //host "select * from Win32_ComputerSystem"
â euclides root ~

```

snmp_blacklist

```
snmp_blacklist <path>
```

Path (<path>) to the SNMP module blacklist file
(/etc/pandora/satellite_server.blacklist by default).

add_host

```
add_host <IP_addr> [agent_name]
```

It adds the given host to the list of monitored agents. The name for the agent can be specified after the IP address. Multiple hosts may be added, one per line. For example:

```
add_host 192.168.0.1
add_host 192.168.0.2 localhost.localdomain
```

ignore_host

```
ignore_host <agent_name>
```

It removes the given host from the list of monitored agents, even if it is found in a network scan by a recon task. The host must be identified by agent name. Multiple hosts may be ignored, one per line. For example:

```
ignore_host 192.168.0.1
ignore_host localhost.localdomain
```

delete_host

```
delete_host <agent_name>
```

It permanently removes the given host from the list of monitored Agents by deleting its configuration file. The host must be identified by the Agent name. Multiple hosts can be deleted, one per line. *For example:*

```
delete_host 192.168.0.1  
delete_host localhost.localdomain
```

keepalive

```
keepalive xxx
```

Satellite Server reports its status and checks remote configuration changes (from agents and its own) every xxx seconds. It is 30 seconds by default.

credential_pass

```
credential_pass xxx
```

Password used to encrypt credential box passwords. It must match the one defined in Pandora FMS console. The hostname is used by default.

timeout_bin

```
timeout_bin <path>
```

If defined, the timeout program (usually `/usr/bin/timeout`) will be used to call the Tentacle client.

timeout_seconds

```
timeout_seconds xxx
```

Timeout in seconds for the timeout command. The `timeout_bin` parameter must be configured.

proxy_traps_to

```
proxy_traps_to <dir_IP[:port]>
```

It redirects SNMP traps received by the Satellite server to the given address (and port). Port 162 is used by default.

proxy_tentacle_from

```
proxy_tentacle_from <dir_IP[:port]>
```

It redirects data received by Tentacle server from the specified address and port. Port 41121 is used by default.

proxy_tentacle_to

```
proxy_tentacle_to <dir_IP[:port]>
```

It redirects Tentacle client requests received by the Satellite Server to the given address (and port). Port 41121 is used by default.

This option may be in conflict with remote agent configuration.

This happens if the Satellite server is intended to be used as proxy for some software agents and monitor them remotely from the Satellite server itself (ICMP, SNMP, etc.) and remote configuration is enabled in both cases.

In this situation, it is necessary to either use different agents for the performed checks (i.e. with different `agent_name`), or leave the remote configuration enabled only on one of them (Satellite Server or software agents).

dynamic_inc

```
dynamic_inc [0|1]
```

When set to 1, it moves dynamic auto-discovered modules (SNMP, WMI...) to separate files so that they do not interfere with remote agent configuration.

vlan_cache_enabled

```
vlan_cache_enabled [0|1]
```

It enables (1) or disables (0) the VLAN cache in the auto-discovered hosts.

verbosity

```
verbosity <0-10>
```

Detail log level, where 10 is the highest information detail level.

agents_blacklist_icmp

Version NG 713 or later.

```
agents_blacklist_icmp 10.0.0.0/24[,8.8.8.8/30]
```

ICMP check blacklist. This field can be configured with a list of IPs, using CIDR notation to prevent ICMP-type modules from running. To specify multiple subnets, separating them with commas.

agents_blacklist_snmp

Version NG 713 or later.

```
agents_blacklist_snmp 10.0.0.0/24[,8.8.8.8/30]
```

SNMP check blacklist. This field can be configured with a list of IPs, using CIDR notation to prevent SNMP-type modules from running. You may specify multiple subnets separating them with commas.

agents_blacklist_wmi

Version NG 713 or later.

```
agents_blacklist_wmi 10.0.0.0/24[,8.8.8.8/30]
```

WMI Check blacklist. This field can be configured with a list of IPs, using the CIDR notation to prevent WMI-type modules from running. You may specify multiple subnets by separating them with commas.

general_gis_exec

Versión NG 734 o later.

```
general_gis_exec xxx
```

By enabling this option, a GIS positioning script will be used for all agents detected by the Satellite server. The script must have running permissions and must print on screen the coordinates with the format <longitude>,<latitude>,[<altitude>]. The third parameter, altitude, is optional.

forced_add

```
force_add [0|1]
```

If set to 1, hosts added manually (via [host_file](#) or [add_host](#)) will always be created, even if they do not respond to ping, with a configuration file without modules.

agent_block

```
agent_block XX
```

Number of XML data files sent in a single call to the Tentacle client, by default 50.

conf_interval

```
conf_interval XXX
```

Remote configuration check interval, by default 300 seconds.

exec_interval

```
exec_interval XXX
```

Time between exec checks, by default 300 seconds.

exec_threads

```
exec_threads X
```

Number of threads used for exec modules, 5 by default .

It depends on the capacity (CPU and RAM) of the machine. The higher the threads, the higher the load on the machine but the processing capacity will be higher. The performance may become poor when exceeding 20 threads, depending on each system.

latency_block

```
latency_block XXX
```

Number of hosts processed in a single call to nmap (latency), 400 by default.

It forces the server to execute checks into blocks of XXX checks. The higher the number (500 tops) the more capacity it will have, but at the expense of an increased latency. Sometimes, it might be recommended to lower that number.

latency_interval

```
latency_interval XXX
```

Time between latency checks, 180 seconds by default.

latency_retries

```
latency_retries X
```

Number of retries for latency modules, 2 by default.

latency_threads

```
latency_threads X
```

Number of threads used for latency checks, 4 by default.

latency_timeout

```
latency_timeout X
```

Timeout for latency checks in seconds, 1 by default.

ping_block


```
ping_block XXX
```

Number of hosts processed in a single call to nmap (ping), 400 by default.

It forces the server to execute checks into blocks of XXX checks. The higher the number (500 tops) the more capacity it will have, but at the expense of an increased latency. Sometimes, it might be recommended to lower that number.

ping_interval

```
ping_interval XXX
```

Time between ping checks, 120 seconds by default.

ping_retries

```
ping_retries X
```

Number of retries for latency modules, 2 by default.

ping_threads

```
ping_threads X
```

Number of threads used for ping checks, 4 by default.

ping_timeout

```
ping_timeout 1
```

Timeout for ping checks in seconds, 1 by default.

plugin_interval

```
plugin_interval XXX
```

Time between plugin checks, 300 by default.

plugin_threads

```
plugin_threads X
```

Number of threads used for plugin checks, 2 by default.

plugin_timeout

```
plugin_timeout XX
```

Timeout for plugin checks in seconds, 10 by default.

recon_interval

```
recon_interval XXXXXX
```

Time between network scans in seconds, 604800 by default.

snmp2_block

```
snmp2_block XX
```

Number of hosts processed in a single call to braa (SNMPv2c), 50 by default.

snmp2_interval

```
snmp2_interval XXX
```

Time between SNMP checks (SNMPv2c), 180 seconds by default.

snmp2_retries

```
snmp2_retries X
```

Number of retries for SNMP modules (SNMPv2c), 2 by default.

snmp2_threads

```
snmp2_threads X
```

Number of threads used for SNMP checks (SNMPv2c), 8 by default.

snmp2_timeout

```
snmp2_timeout X
```

Timeout for SNMP (SNMPv2c) checks in seconds, 5 by default.

snmp3_block

```
snmp3_block XX
```

Number of hosts processed in a single call to braa (SNMPv3), 50 by default.

snmp3_interval

```
snmp3_interval XXX
```

Time between SNMP checks (SNMPv3), 180 by default.

snmp3_retries

```
snmp3_retries X
```

Number of retries for SNMP modules (SNMPv3), 2 by default.

snmp3_threads

```
snmp3_threads X
```

Number of threads used for SNMP checks (SNMPv3), 4 by default.

snmp3_timeout

```
snmp3_timeout X
```

Timeout for SNMP (SNMPv3) checks in seconds, 5 by default.

snmp_block

```
snmp_block XX
```

Number of hosts processed in a single call to braa (SNMPv1), 50 by default.

snmp_interval

```
snmp_interval XXX
```

Time between SNMP checks (SNMPv1), 180 seconds by default.

snmp_retries

```
snmp_retries X
```

Number of retries for SNMP modules (SNMPv1), 2 by default.

ssh_interval

```
ssh_interval XXX
```

Time between SSH checks, 300 seconds by default.

ssh_threads

```
ssh_threads X
```

Number of threads used for SSH modules, 5 by default.

ssh_timeout

```
ssh_timeout X
```

Timeout for SSH checks in seconds, 2 by default.

tcp_interval

```
tcp_interval XXX
```

Time between TCP checks, 300 seconds by default.

tcp_threads

```
tcp_threads X
```

Threads dedicated to TCP checks, 5 by default.

tcp_timeout

```
tcp_timeout X
```

Timeout for TCP checks in seconds, 1 by default.

snmp_threads

```
snmp_threads X
```

Number of threads used for SNMP checks (SNMPv1), 8 by default.

snmp_timeout

```
snmp_timeout X
```

Timeout for snmp checks in seconds (SNMPv1), 5 by default.

wmi_interval

```
wmi_interval XXX
```

Time between WMI checks, 300 seconds by default.

wmi_threads

```
wmi_threads X
```

Threads dedicated to WMI polling, 5 by default.

ipam_task

```
ipam_task <id IPAM TASK> , <CIDR>
```

Comma-separated list of networks (in SLASH notation) to scan for IPAM. It must be preceded by the ID of the IPAM task in Pandora FMS (the Discovery server field should be left unassigned and then assigned to a Satellite server). E.g. 1, 192.168.0.0/24 .

ipam_interval

```
ipam_interval XXXXXX
```

Time between IPAM scans in seconds.

Secondary Server

```
secondary_mode [on_error|always]
```

An special kind of general configuration parameter is the definition of a secondary server. This allows defining a server to send data to, in a complementary way to the server defined the standard way. The secondary server mode works in two different ways:

- on_error: It sends data to the secondary server only when it cannot send them to the primary one.
- always: It always sends data to the secondary server, both if it can contact the main server or not.

Configuration example:

```
secondary_server_ip      192.168.1.123
secondary_server_path    /var/spool/pandora/data_in
secondary_mode           on_error
secondary_transfer_mode  tentacle
secondary_server_port    41121
```

Remote configuration

Remote file configuration

You may access the advanced editor for remote configuration of the Satellite server on the PFMS Enterprise server the Satellite server belongs to through the menu ►Servers → Manage servers. Once you have loaded the page in your web browser, click on the Remote configuration icon.

Pandora FMS
the Flexible Monitoring System

Enter keywords 🔍

Servers / Manage Servers
Pandora FMS servers

pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	2:0	3 seconds	
pandorafms	🟢	🔧	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📊	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📊	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	📱	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	🔧	📘	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	🟢	🔧	📘	7.0NG.770 (P) 230505	N/A	N/A	1:1	3 seconds	
pandorafms	🟢	🔧	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	🔧	📘	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	🟢	📶	📘	7.0NG.770 (P) 230505	0 of 0	-/0	0:0	14 seconds	🔧 ⚙️ 🗑️

Remote configuration

Operation Management

- Discovery
- Resources
- Profiles
- Configuration
- Alerts
- Servers
 - Manage servers
 - Manage consoles
 - Manage database HA
 - Plugins
 - Register Plugin
 - Export targets
 - Manage Satellite Server

Then click on the Advanced editor icon:

Pandora FMS
the Flexible Monitoring System

Enter keywords 🔍

Servers / Pandora FMS servers
Remote configuration ⓘ

Advanced editor

Delete remote conf server files 🗑️ ⓘ

Configuration

```
# -----
# Pandora FMS Satellite Server configuration file
# (c) 2014-2023 Pandora FMS
# -----

# Directory where agent configuration files are stored.
agent_conf_dir /etc/pandora/conf

# Number of XML data files sent in a single call to the Tentacle client.
agent_block 50
```

In the text box corresponding to Configuration you may edit and/or add each of the tokens described in previous sections. Once you are finished editing, save the changes by clicking

◆ Update at the end of the page.

New token synchronization and loading will take some time. Please wait a few seconds for the changes to propagate.

Remote configuration graphical interface

NG 764 version or later.

You will be able to access the graphic interface on PFMS Enterprise server to which the Satellite server belongs, remotely, by means of the menu Servers → Manage servers and then click on the Remote configuration icon.

The screenshot shows the Pandora FMS web interface. The sidebar menu on the left has 'Servers' expanded to 'Manage servers'. The main content area is titled 'Pandora FMS servers' and contains a table with the following data:

Server Name	Status	IP	Version	Token Count	Sync Status	Sync Time	Last Sync
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	2:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	1:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	N/A	N/A	1:1	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	4:0	3 seconds	
pandorafms	Online	7.0NG.770 (P) 230505	0 of 0	-/0	0:0	14 seconds	Remote configuration

Use the Dynamic search to enter key text (one letter or more) and search for a specific token.

Pandora FMS
the Flexible Monitoring System

Enter keywords

Servers / Pandora FMS servers
Remote configuration ?

Operation Management

- Discovery
- Resources
- Profiles
- Configuration

Dynamic search

Some tokens only accept two values (ON /OFF): use the first button to change this value. If you enable or disable the corresponding token with the second button, the Satellite server will take the default value assigned to it (1 or 0) *regardless of the value set by the first button*.

Pandora FMS
the Flexible Monitoring System

Enter keywords

Servers / Pandora FMS servers
Remote configuration ?

Dynamic search

General server configuration

VLAN cache enabled ?

Default value: 1

Enable XML buffer

Default value: 0

Once you have made all the changes click Update to save your preferences.

Agent creation in Satellite Server

There are three ways of creating an agent in the Satellite server: Recon Task, Satellite_hosts.txt file, or manually creating the .conf of the agents to monitor.

Agent creation through Recon Task

The creation of agents through Recon Task is the most used by Pandora FMS users. To be able to

do it, go to the Satellite server configuration file and set the following parameters:

- `recon_community`: Specify a list of SNMP communities to use in SNMP discovery separated by commas (in case of performing a recon of the SNMP type).
- `recon_enabled`: It must be set to 1 to enable the recon task of the Satellite server.
- `recon_interval`: Time interval where a certain network is scanned, in seconds (604800 seconds by default, 7 days).
- `recon_mode`: Recon task mode (snmp,icmp,wmi) separated by commas.
- `recon_task`: List of networks to be recognized, separated by commas.
- `recon_timing_template`: A value that specifies how aggressive nmap must be, from 1 to 5. 1 means slower but more reliable, 5 means faster but less reliable (3 by default).

An example of Recon Task would be:

```
recon_community public
recon_enabled 1
recon_interval 604800
recon_mode icmp,snmp,wmi
recon_task 192.168.0.0/24,192.168.1.0/24
recon_timing_template 3
```

Once the data has been configured, run the satellite server using the command:

```
/etc/init.d/satellite_serverd start
```

Agents without modules in their configuration files will be ignored by the Satellite Server.

Agent creation through a file

First, in order to create an agent through the `satellite_hosts.txt` file, go to the configuration file of the Satellite server and uncomment the line:

```
host_file /etc/pandora/satellite_hosts.txt
```

Secondly, create the file `satellite_hosts.txt` with the IP of the host that you wish to create by entering IP and name of the agent to create:

```
192.168.10.5 Server5
192.168.10.6 Server6
192.168.10.7 Server7
```

In order for these IPs to be created, it is necessary to be able to make the `fping` call to each one of the IPs in the list, otherwise it will not be created.

Once the data has been configured, run the satellite server using the command:

```
/etc/init.d/satellite_serverd start
```

The reading of the indicated file is done every recon_interval seconds.

Manual Agent Creation

Firstly, look at the configuration file of the Satellite server in the /etc/pandora/conf directory, which is where the new agent configuration files are created. Open a terminal and go to that folder:

```
cd /etc/pandora/conf
```

Once this path is located, create a .conf for example "archivo.conf". Fill in the following fields:

- agent_name: Agent name.
- agent_alias: Agent alias.
- address: IP of the element to monitor.
- group: Group to assign the agent to.
- gis_exec: Positioning script (optional). It overwrites the [general_gis_exec](#) location of the Satellite server.
- Modules to be monitored by the agent.

An example would be:

```
agent_name Example
agent_alias This is an example
address 127.0.0.1
group Servers

module_begin
module_name Ping
module_ping
module_end

module_begin
module_name Latency
module_latency
module_end
```

Once the data has been configured, run the Satellite server using the command:

```
/etc/init.d/satellite_serverd start
```

Agent removal in Satellite Server

You may fully remove the agents or delete them partially.

First back up the folders and their files before proceeding.

For agent total removal, take into account the method used in agent creation.

- **Manual:** First remove the `.conf` files from the agents created in the `/etc/pandora/conf` folder and then remove the agents from the console.
- **Satellite_hosts.txt file:** Delete the file as well as the `.conf` that have been created in the folder `/etc/pandora/conf` and later delete the agents from the console.
- **Recon_task:** Deconfigure the `recon_task` from the `.conf` file of the Satellite server, then remove the `.conf` created in the folder `/etc/pandora/conf` and then remove the agents from the console.

For agent partial removal also take into account the method used in the agent creation.

- **Manual:** First of all, remove the `.conf` files from the agents you wish to delete in the `/etc/pandora/conf` folder and then remove the agents from the console.
- **Satellite_hosts.txt file:** Delete the lines of the IPs from the file as well as the `.conf` that have been created in the folder `/etc/pandora/conf` with those IPs and then delete the agents from the console.
- **Recon_task:** Configure the blacklist of the `recon_task` in the `conf` file of the Satellite server, then remove the `.conf` created in the folder `/etc/pandora/conf` with those IPs and then remove the agents from the console.

Custom settings by agent

In addition to “automatic” modules, all kinds of available TCP, SNMP or WMI tests can be added, using a similar syntax to the local modules in [software agents](#). Here are some module examples valid for Satellite server, just as they are autogenerated after being detected by the system.

ICMP/TCP queries

Conectivity to a machine (using PING):

```
module_begin
module_name ping
module_type generic_data
module_ping 192.168.70.225
module_end
```

Port check (using TCP):

```
module_begin
module_name Port 80
module_type generic_proc
module_tcp
module_port 80
module_end
```

WMI queries

CPU WMI usage check (percentage):

```
module_begin
module_name CPU
module_type generic_data
module_wmicpu 192.168.30.3
module_wmiauth admin%none
module_end
```

Memory free WMI check (percentage):

```
module_begin
module_name FreeMemory
module_type generic_data
module_wmimem 192.168.30.3
module_wmiauth admin%none
module_end
```

General WMI query:

```
module_begin
module_name GenericWMI
module_type generic_data_string
module_wmi 192.168.30.3
module_wmiquery SELECT Name FROM Win32_ComputerSystem
module_wmiauth admin%none
module_end
```

SNMPv1 and SNMPv2 queries

Make sure OIDs start with a dot, otherwise SNMP modules will not work!

Interface status through SNMP. The Satellite server detects automatically each interface:

```
module_begin
module_name if eth1 OperStatus
```

```
module_description IP address N/A. Description: The current operational state of
the interface. The testing(3) state indicates that no operational packets can be
passed.
module_type generic_data_string
module_snmp 192.168.70.225
module_oid .1.3.6.1.2.1.2.2.1.8.3
module_community artica06
module_end
```

To force the module to use SNMP version 2c, add the line:

```
module_version 2c
```

To force the module to use SNMP version 1, add the line:

```
module_version 1
```

For example:

```
module_begin
module_name if eth1 OperStatus
module_description IP address N/A. Description: The current operational state of
the interface. The testing(3) state indicates that no operational packets can be
passed.
module_type generic_data_string
module_snmp 192.168.70.225
module_version 2c
module_oid .1.3.6.1.2.1.2.2.1.8.3
module_community artica06
module_end
```

General SNMP check. In this case, the server retrieves automatically the traffic from each interface with its "real" descriptive name:

```
module_begin
module_name if eth0 OutOctets
module_description The total number of octets transmitted out of the interface,
including framing characters.
module_type generic_data_inc
module_snmp 192.168.70.225
module_oid .1.3.6.1.2.1.2.2.1.16.2
module_community public
module_end
```

SNMPv3

To configure an SNMPv3 module, set `module_version` to 3 and specify

- `module_seclevel`: Security level (`noauth`, `authnopriv` or `authpriv`).
- `module_secname`: Security name.
- `module_authproto`: Authentication protocol (`md5` or `sha`).
- `module_authpass`: Authentication password.
- `module_privproto`: Security protocol (`aes` or `des`).
- `module_privpass`: Privacy password, if needed. For example:

```
module_begin
module_name snmp_noauth
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.1.0
module_seclevel noauth
module_secname snmpuser
module_end
```

```
module_begin
module_name snmp_authnopriv
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_seclevel authnopriv
module_secname snmpuser
module_authproto md5
module_authpass 12345678
module_end
```

```
module_begin
module_name snmp_authpriv
module_type generic_data_string
module_snmp 127.0.0.1
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_seclevel authpriv
module_secname snmpuser
module_authproto sha
module_authpass 12345678
module_privproto aes
module_privpass 12345678
module_end
```

SNMPv3 specific configuration can be shared between modules by placing it outside the module declaration, in case it is the same for all of them (it can also be shared between agents by moving it to the Satellite's configuration file):

```
agent_name snmp
address 127.0.0.1
```

```
seclevel authpriv
secname snmpuser
authproto md5
authpass 12345678
privproto des
privpass 12345678

module_begin
module_name snmp_authpriv_1
module_type generic_data_string
module_snmp
module_version 3
module_oid .1.3.6.1.2.1.1.1.0
module_end

module_begin
module_name snmp_authpriv_2
module_type generic_data_string
module_snmp
module_version 3
module_oid .1.3.6.1.2.1.1.2.0
module_end
```

For the creation of SNMP modules (including SNMPv3) via PFMS Web Console [visit this video](#). For the creation of component groups (including SNMPv3) see “[SNMP wizard](#)”.

Default Satellite Server configuration file for SNMPv3:

You will need to put in your own values and/or credentials, as well as change any necessary protocols or encryption methods. You will need to restart the PFMS server for the new configuration values to be read and put into memory.

```
# Security level used for SNMPv3 messages (noauth, authnopriv or authpriv).
#snmp3_seclevel authpriv

# Security name used for SNMPv3 messages.
#snmp3_secname

# Authentication protocol (md5 or sha) for authenticated SNMPv3 requests.
#snmp3_authproto sha

# Authentication password for authenticated SNMPv3 request.
#snmp3_authpass

# Privacy protocol (des or aes) for encrypted SNMPv3 requests.
#snmp3_privproto des

# Privacy password for encrypted SNMPv3 messages.
#snmp3_privpass
```


SSH queries

SSH queries on Satellite servers installed on MS Windows® is still under implementation. The PFMS development team is working on it.

Generic SSH command:

```
module_begin
module_name GenericSSH
module_type generic_data
module_ssh 192.168.30.3
module_command ls /tmp | wc -l
module_end
```

To add a threshold, do it both in the module's text definition and threshold definition in the web interface (`module_min_warning`, `module_min_critical`). For example:

```
module_begin
module_name Latency
module_type generic_data
module_latency 192.168.70.225
module_min_warning 80
module_min_critical 120
module_end
```

Execution modules can be created manually. The scripts or commands executed by the Satellite server must be previously established and available for the server to use. It works the same as an agent `module_exec`. Bear in mind that the use of `module_exec` can make the performance of the Satellite server to become poor:

```
module_begin
module_name Sample_Remote_Exec
module_type generic_data
module_exec /usr/share/test/test.sh 192.168.50.20
module_min_warning 90
module_min_critical 95
module_end
```

Plugins queries

From Pandora FMS version 7 on, plugins can be added. Like those, note that the plugins will run on the machine where the Satellite server is running. Therefore, it will be necessary to implement in these plugins some method to connect to the remote computer you wish to monitor. The

advantage over the previous ones is their great flexibility. That way, preconditions and other mechanisms for which a `module_exec` falls short can be implemented. The syntax is the same as that of the agents. An example of using a plugin might be as follows:

```
module_plugin /usr/share/pandora/remote_advanced_checks.sh 192.168.0.1
```

Credential boxes

Unless key-based authentication is configured with private and public keys, SSH modules require a username (`<user>`) and a password (`<pass>`) in order to work. These are configured in the main configuration file, `satellite_server.conf`, using credential boxes (`credential_box`) with the following format:

```
network/mask,username,password
```

```
network/mask,username,[[encrypted password]]
```

SSH queries on Satellite servers installed on MS Windows® is still under implementation. The PFMS development team is working on it.

For example:

```
credential_box 192.168.1.1/32,<user>,<pass1>  
credential_box 192.168.1.0/24,<user>,<pass2>
```

Credential boxes are searched from more restrictive to less restrictive masks.

Passwords can be encrypted using Blowfish in ECB mode. Make sure `credential_pass` is defined, otherwise the hostname will be used as the default encryption password. The hexadecimal representation of the ciphertext should be enclosed in double brackets:

```
credential_box 192.168.1.0/24,<user>,[ [80b51b60786b3de2 | ] ]
```

General view of all agents in the console

If the configuration of the satellite server is correct, you should an agent view similar to this one:

Agent	Description	Remote	OS	Interval	Group	Type	Modules	Status	Alerts	Last contact
192.168.70.157	Created by SatServer			5 minutes			2 : 1 : 1			4 minutes 27 seconds
192.168.70.159	Created by SatServer			5 minutes			2 : 2			4 minutes 27 seconds
192.168.70.165	Created by SatServer			5 minutes			2 : 2			4 minutes 27 seconds
192.168.70.168	Created by SatServer			5 minutes			2 : 2			4 minutes 27 seconds

Generally, in all machines ICMP (Ping and Latency) modules will be created, but in some machines SNMP and WMI modules can be created. In machines where WMI is enabled, the following modules will be generated if available:

F.	P.	Type	Module name	Description	Status	Thresholds	Data	Graph	Last contact
			CPU Load	CPU Load (%)		N/A - N/A	21 %		39 seconds
			Free memory	Total free memory in kilobytes		N/A - N/A	7,635,884 KB		39 seconds
			FreeDisk C:	Available disk space in kilobytes		N/A - N/A	214,845,685,284 KB		39 seconds
			FreeDisk D:	Available disk space in kilobytes		N/A - N/A	78,945,619 KB		39 seconds

In machines with SNMP enabled, the following modules will be generated if available:

F.	P.	Type	Module name	Description	Status	Thresholds	Data	Graph	Last contact
			ipInReceives	The total number of input datagrams received from interfaces...		N/A - N/A	2		3 minutes 34 seconds
			ipOutRequests	The total number of IP datagrams which local IP user-protocol...		N/A - N/A	1.6		3 minutes 34 seconds
			sysName	An administratively-assigned name for this managed node. By...		N/A - N/A	pacifico		3 minutes 34 seconds
			sysUpTime	The time (in hundredths of a second) since the network manag...		N/A - N/A	1378258510		3 minutes 34 seconds
			X0_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	43,870.2		3 minutes 34 seconds
			X0_ifOperStatus	MAC C0:EA:E4:6E:9B:20 IP 192.168.80.1. Description: The curr...		N/A - N/A	1		3 minutes 34 seconds
			X0_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	60,051.9		3 minutes 34 seconds
			X1_ifInOctets	The total number of octets received on the interface, includ...		N/A - N/A	213,040.1		3 minutes 34 seconds
			X1_ifOperStatus	MAC C0:EA:E4:6E:9B:21 IP 192.168.90.254. Description: The cu...		N/A - N/A	1		3 minutes 34 seconds
			X1_ifOutOctets	The total number of octets transmitted out of the interface,...		N/A - N/A	1,609,405		3 minutes 34 seconds

In the massive operations menu of Pandora FMS console, there is a specific section for the Satellite server where different edition and deletion actions can be performed on agents and modules massively.

Bulk operations » Edit Satellite modules in bulk

Action: Edit Satellite modules in bulk

Agent group: All

Filter agent:

Filter module:

Agents: 192.168.70.1, 192.168.70.100, 192.168.70.102, 192.168.70.107, 192.168.70.109, 192.168.70.114, 192.168.70.116, 192.168.70.12, 192.168.70.123, 192.168.70.125

When selecting agents: Show common modules

Filter module: Any, Latency, Ping

Warning status: Min. , Max. , Str.

Critical status: Min. , Max. , Str.

Update

NG 763 version or later.

Resources / Manage agents / Setup

NODO1MANU ?

Agent name: Nodo1Manu ID 1859

Interval: 5 minutes

Alias: Nodo1Manu

OS: Satellite

IP Address: 192.168.80.222

Unique IP: Fix IP address: Delete selected IPs:

Primary group: Servers

Server: koldo

Satellite: koldo

Description: Created by koldo

View agent QR code

Custom ID:

SNMP blacklist

When monitoring big networks, SNMP modules that return invalid data may affect the performance of the Satellite server and many modules may become Unknown. To avoid that, the Satellite Server can read a blacklist of SNMP modules that will be discarded at startup before execution.

To create a new blacklist, edit the `/etc/pandora/satellite_server.conf` configuration file and make sure `snmp_blacklist` is uncommented and configured with the path of the file where blacklist modules are saved. Then run:

```
satellite_server -v /etc/pandora/satellite_server.conf
```

Restart the Satellite server. The blacklist can be regenerated as many times as needed.

The format of the blacklist file is:

```
agent:0ID
agent:0ID
...
```

For example:

```
192.168.0.1:.1.3.6.1.4.1.9.9.27
192.168.0.2:.1.3.6.1.4.1.9.9.27
```



[Go back to Pandora FMS documentation index](#)