Alert system

From:
https://pandorafms.com/manual/!776/
Permanent link:
https://pandorafms.com/manual/!776/en/documentation/pandorafms/management_and_operation/01_alerts
2024/06/10 14:34

Alert system

Introduction

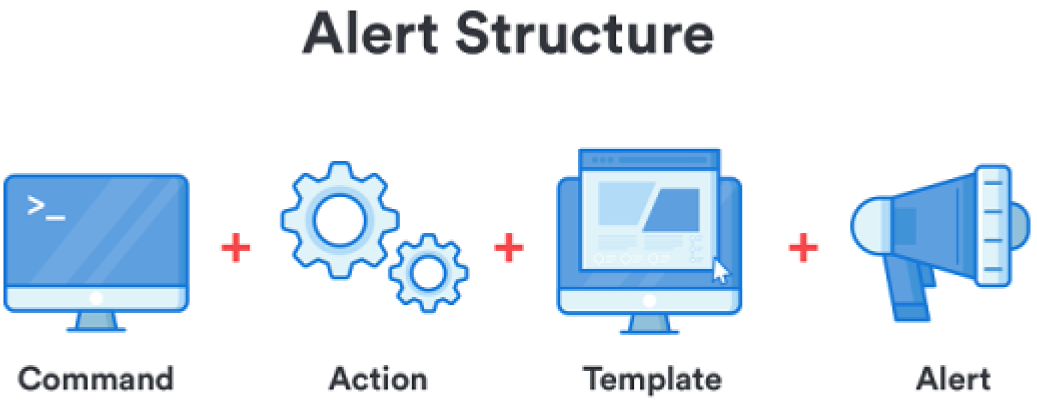
An alert is the reaction of Pandora FMS to an incorrect value of a Module. Said reaction is configurable and can consist of anything that can be triggered by a *script* configured in the Operating System where the Pandora FMS server that processes the Module runs.

In Pandora FMS, the alerts work by defining some trigger conditions, some actions chosen for that alert, and finally the execution of some commands in the Pandora FMS server, which will be in charge of carrying out the configured actions.

There are several types of alerts:

- Alerts simple.
- Alerts about events.
- Alerts about *traps* SNMP.

Structure of an alert



- Commands: Specify *what will be done*; It will be the execution that the Pandora FMS server will carry out when triggering the alert.
- Actions: Specify *how it will be done;* they are the customizations of the command arguments.
- Templates: Specify *when it will be done;* define the conditions for triggering the action(s).

Flow of information in the alert system

Templates and actions have a series of generic fields called `Field1` , `Field2` , `Field3`, (...) , `Field`*n* which they are used to transfer the information from the *template* to the *action* and from the action to the *command*, to finally be used as parameters in the execution of said command.

Said information is transferred whenever the next step does not already bring information defined in its `Fieldn` fields. That is, in case of overlapping fields or parameters, it overwrites the action to the template (for example, if the template has `Field1` defined and the action as well, the `Field1` of the action *overwrites* the template action).

Alert Command

Introduction

Management menu → Alerts → Commands.

The actions that Pandora FMS will carry out in the event of alert situations will ultimately be translated into executions on the server, in the form of commands.

To create alert commands you must access as PFMS superuser.

Creating a command for an alert

Management Menu → Alerts → Commands → Create.

It is recommended to check from the command line if the execution of the command is successful and that it produces the desired result (send an email, generate an entry in a log file, etc).

- Command: Command that will be executed when triggering the alert. It is possible to use macros to replace the parameters configured in the alert declaration.
- Group: This determines which group of alerts you can associate the command with. You can only assign a group to which the user creating the alert command belongs, unless that user explicitly belongs to the group EVERYONE (ALL) .
- Field description and Field values:
    - Available Field Values: A collection of possible values for that field. If this field is set (not empty), the field will be a combo select instead of a text box. The combo needs for each possible value a label (the visible option) and a value (the sent option). The syntax is as follows: value1,label1;value2,label2;value3,label3;valueN,labelN.
    - Hide: *If the field contains any passwords, this option hides the* contents with asterisks.
- It is possible to display an HTML editor in a command field when creating or editing an alert action if that command field has the special token value `_html_editor_` .

It must be taken into account that the commands for alerts executed by the Pandora FMS server are carried out with the same privileges as the user that executes the Pandora FMS server.

Predefined commands

- eMail: Sends an email from the Pandora FMS server. Email messages are sent in HTML format. It must be taken into account that the receiver must be able to access the resources used in the template, such as images.
- Internal audit: Generates an entry in the internal audit system of Pandora FMS. This is stored in the databasee Pandora FMS and can be reviewed with the event viewer from the console.
- Monitoring Event: Create a custom event in the Pandora FMS event console.
- Pandora FMS Alertlog: It is a predefined alert that writes the alerts in plain ASCII format in the file `/var/log/pandora/pandora_alert.log`.
- SNMP Trap: Sends an SNMP *trap* parameterized with the arguments used.
- Syslog: Sends an alert to the syslog using the logger system command.
- Sound Alert: Plays a sound in the sound console of events when an alert occurs.
- Jabber Alert: Send a Jabber alert to a chat room on a predefined server (the `.sendxmpprc` file must be configured first). Put in `field1` the username, `field2` the name of the chat room, and `field3` the text message.
- SMS Text: Sends an SMS to a specific mobile phone. First it is necessary to define an alert and configure a *gateway* for sending SMS that is accessible from the Pandora FMS server.
- Validate Event: Validates all events related to a module. It will be passed the name of the agent and the name of the module.
- Remote agent control: Send commands to agents with UDP server enabled. The UDP server is used to instruct agents (Windows and UNIX) to *refresh* the execution of the agent: that is, to force the agent to execute and send data.
- Generate Notification: Allows you to send an internal notification to any user or group.
- Send report by e-mail and Send report by e-mail (from template): Both options allow you to send a report in different formats (XML, PDF, JSON, CSV) by email, the The second option allows you to use a template for that attached report.

When a public URL is set for a Web Console, emails sent will have that link set.

Editing a command for an alert

Management menu → Alerts → Commands → click on the name of the command to edit. Once the chosen alert has been modified, click the Update button.

The system commands `eMail`, `Internal Audit` and `Monitoring Event` cannot be changed or deleted.

Action

Introduction

Actions are the alert components in which a command is related to the generic variables `Field 1`, `Field` , ... , `Field 10`.

Actions allow you to define *how* to launch the command.

Creating an Action

Management Menu → Alerts → Actions → Create.

- Group: The group of the action. You can only assign a group to which the user creating the alert command belongs, unless that user explicitly belongs to the group EVERYONE (ALL) . If the associated command has a group other than *All*, only the group associated with the command or the *All* group can be set as the action's group. If for some reason this defers, you will see a warning message for prompt correction by a user with the necessary rights.
- Command: Command that will be used in the event that the alert is executed. You can choose between different predefined commands in Pandora FMS.
- Threshold: An alert action is executed only once within this time interval, regardless of how many times the alert is triggered.
- Command Preview: In this field, *not editable*, the command to be executed in the system will automatically appear.
- Field 1 ~ Field 10: If necessary, these fields define the value of the macros, from `_field1_` to `_field10_`, to be used in the command. These fields can be a text field or a select combo if configured.

When the Fields are assigned a value in the Triggering section, by default these will be the same values for Recovery, unless a different value is assigned.

Edit an Action

Management Menu → Alerts → Actions → click on the name of the action to modify.

Delete an action

Management menu → Alerts → Actions → click on the corresponding trash can icon (Delete column).

Alert Template

Introduction

The templates define the conditions for triggering the alert (*when* to execute the action). They are associated to Modules, in such a way that when the template conditions are met, the associated action(s) will be executed.

Its design allows to generate a small group of generic templates that serve for the majority of possible cases in Pandora FMS.

Creating a Template

Management → Alerts → Templates → Create.

Then follow the three guided steps.

Step 1: Overview

- Group: The group to which the template will be applied. You may only assign a group to which the user who creates the template belongs, unless said user explicitly belongs to the group ALL (ALL).
- Priority: Informative field about the alert. The event generated when the alert is triggered will inherit this priority, useful for filtering alert searches.

Step 2: Conditions

- Use special days list: It sets the special days calendar to be used in the template.
- Time Threshold: Time that must elapse to reset the alert counter. It defines the time interval in which an alert is guaranteed not to fire more times than the number set in Max. number of alerts. After the defined interval, the counter will be restarted. The shot counter reset will not be reset if the alert recovers upon reaching a correct value, unless the value Reset counter for non-sustained alerts is enabled, in which case the counter will be reset immediately after receiving a correct value.
- Min number of alerts: Minimum number of times that the situation defined in the template must take place (always counting from the number defined in the FlipFlop parameter of the Module) to start triggering an alert. The default value is 0, which means that the alert will be triggered when the first value that meets the condition is reached. It works like a filter, useful for ignoring false positives.
- Max number of alerts: Maximum number of alerts that can be sent consecutively in the same time interval (Time Threshold). It is the maximum value of the alert counter. No more alerts will arrive per time interval than those indicated in this field.
- Default Action: This list defines the default action that the template will have. This is the action that will be automatically created when you assign the template to the module. Place one action or none, however *you cannot place place multiple actions by default*.
- Schedule: It establishes the days on which the alert can be triggered. It is possible to see and configure when the alert will be active each day of the week thanks to the built-in editor that is displayed by default in simple mode. In addition, by accessing the detailed mode you may configure the schedules with higher precision.
- Reset counter for non-sustained alerts: Its activation depends on whether the number indicated in

Min. number of alerts is higher than 0. Enabling this token resets the alert counter when the indicated condition does not take place consecutively. For example, if the field Min. number of alerts has a value of 2, it means that the module has to go through the state assigned in Condition type 3 times to trigger the alert. There are two scenarios with the latter token:

- If the reset token is checked, the number of critical states will need to be consecutive, otherwise the counter will be reset.

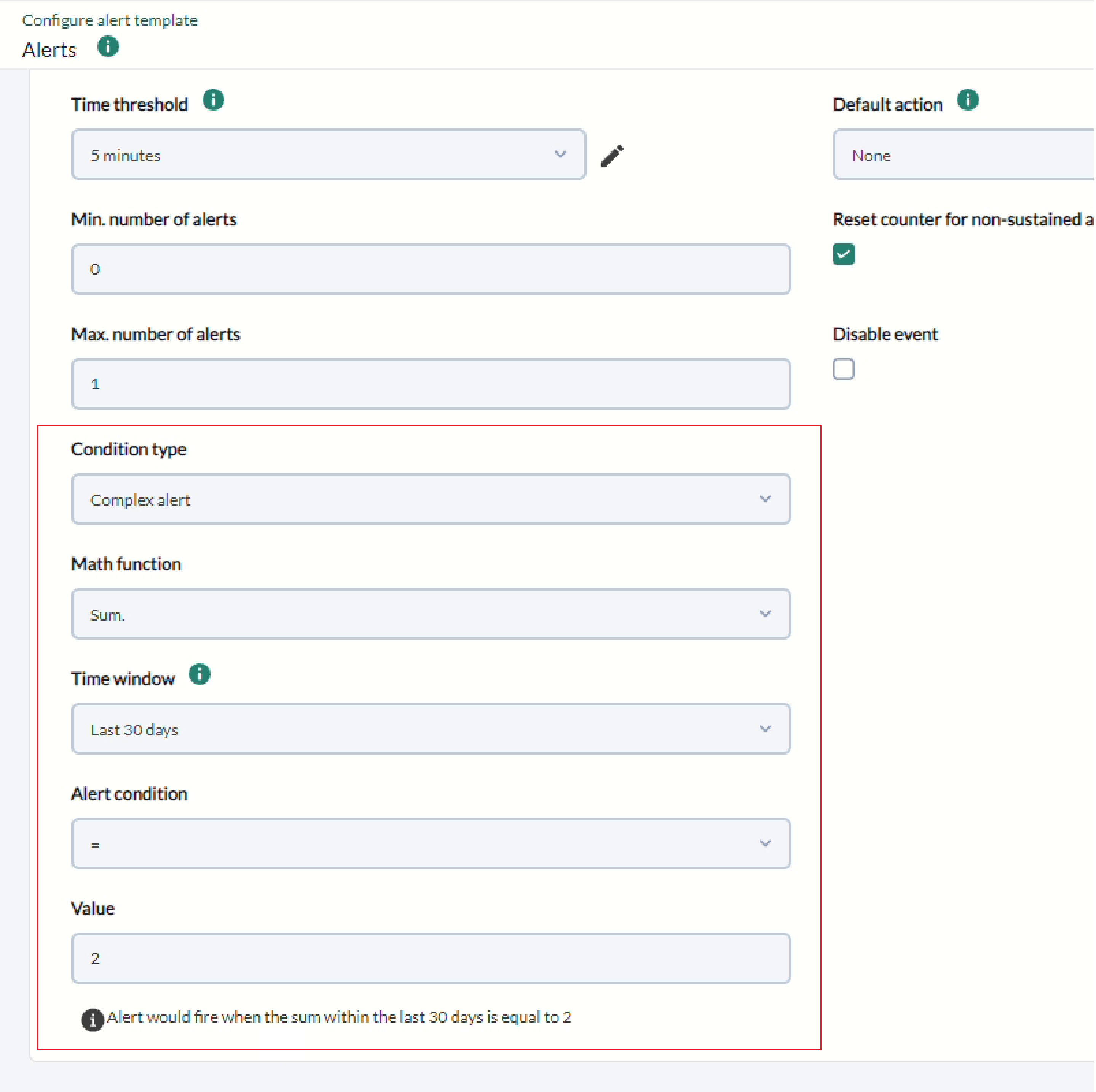
```
normal -> critical -> critical -> critical
```

- If the reset token is not checked, the alert will be triggered after an alternate or continuous sequence of critical states:

```
normal -> critical -> normal -> critical -> normal -> critical
```

To periodically check modules in unknown status (Unknown status) you can either activate the token `unknown_updates` in the server configuration PFMS.

- Condition type: It allows you to specify the element that will trigger the alert, e.g. a critical state (Critical state option) or simply different from the normal state (Not normal state option). You may also define complex alerts (Complex alert option), for example if the sum is exactly equal to two over the last thirty days:



Configure alert template

Alerts

Time threshold

5 minutes

Default action

None

Min. number of alerts

0

Reset counter for non-sustained a

☑

Max. number of alerts

1

Disable event

□

Condition type

Complex alert

Math function

Sum.

Time window

Last 30 days

Alert condition

=

Value

2

Alert would fire when the sum within the last 30 days is equal to 2

Step 3: Advanced Fields

- Alert recovery: Combo where you may define whether or not to enable alert recovery. In the event that alert recovery is enabled, when the module no longer meets the conditions indicated by the template, the action associated with the arguments specified by the *field* fields defined in this column will be executed.
- In all instances of the fields `field1` ... `field10` (both in the alert template, as well as in the command and in the action) those defined in the macro_list.

Once the configuration is complete, finish by clicking Finish.

Assign Alert Templates to Modules

Alert Management from the submenu of Alerts

Assignment of Alerts from the Alerts submenu

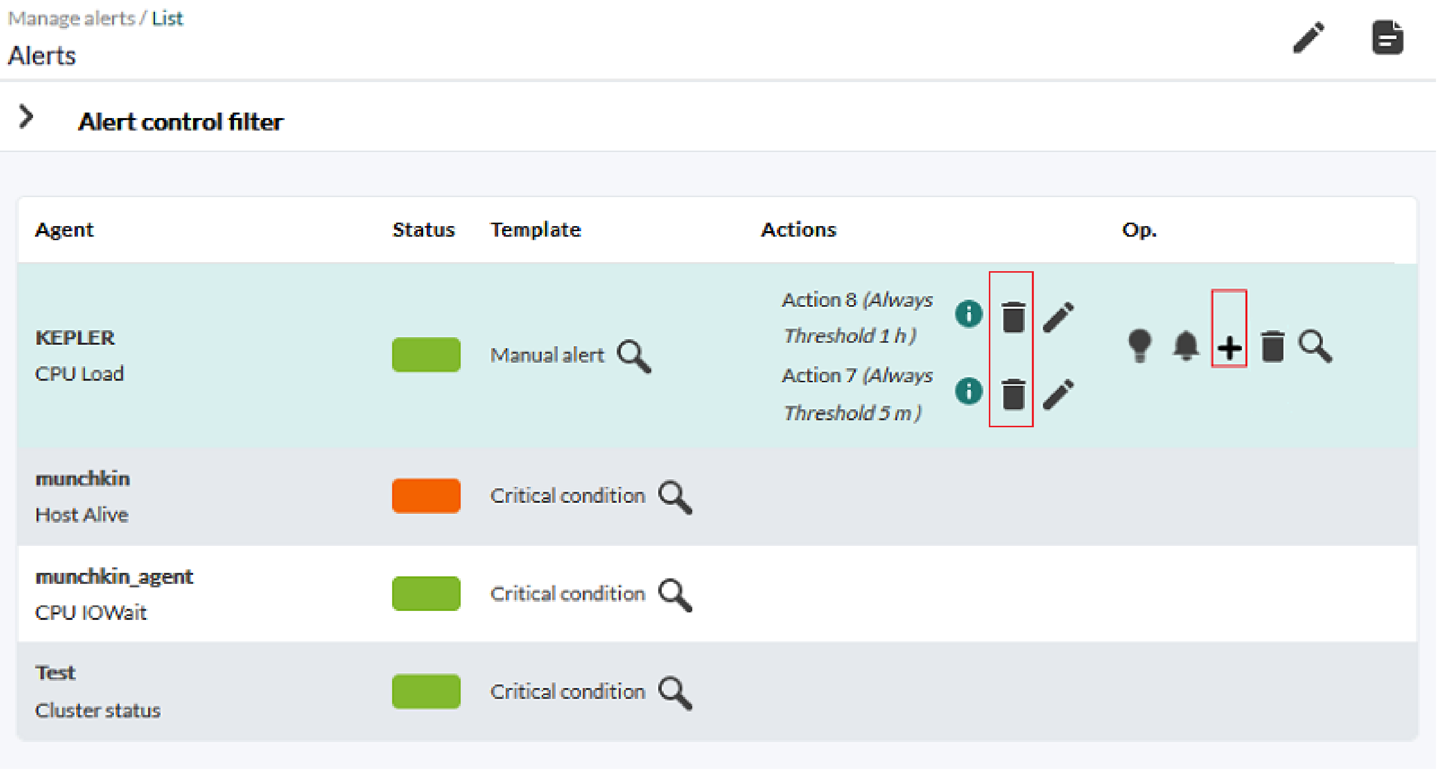
Management menu → Alerts → List of alerts → click on the pencil icon Builder alert.

- Agent: Autocomplete to choose the Agent.
- Module: List of modules of the previously selected Agent.
- Actions: Action that will be executed when the alert is triggered. If the template already has a default action it can be left at Default.
- Template: Template that will contain the trigger conditions of the alert.
- Threshold: An alert action will not be executed more than once every `action_threshold` seconds, regardless of the number of times the alert is fired.

Modify alerts from the Alerts submenu

Once an alert has been created, it will only be possible to modify the actions that have been added to the action that has the template.

It is also possible to delete the action that was selected when the alert was created by clicking the gray trash can icon to the right of the action, or add new actions by clicking the + button.



Manage alerts / List

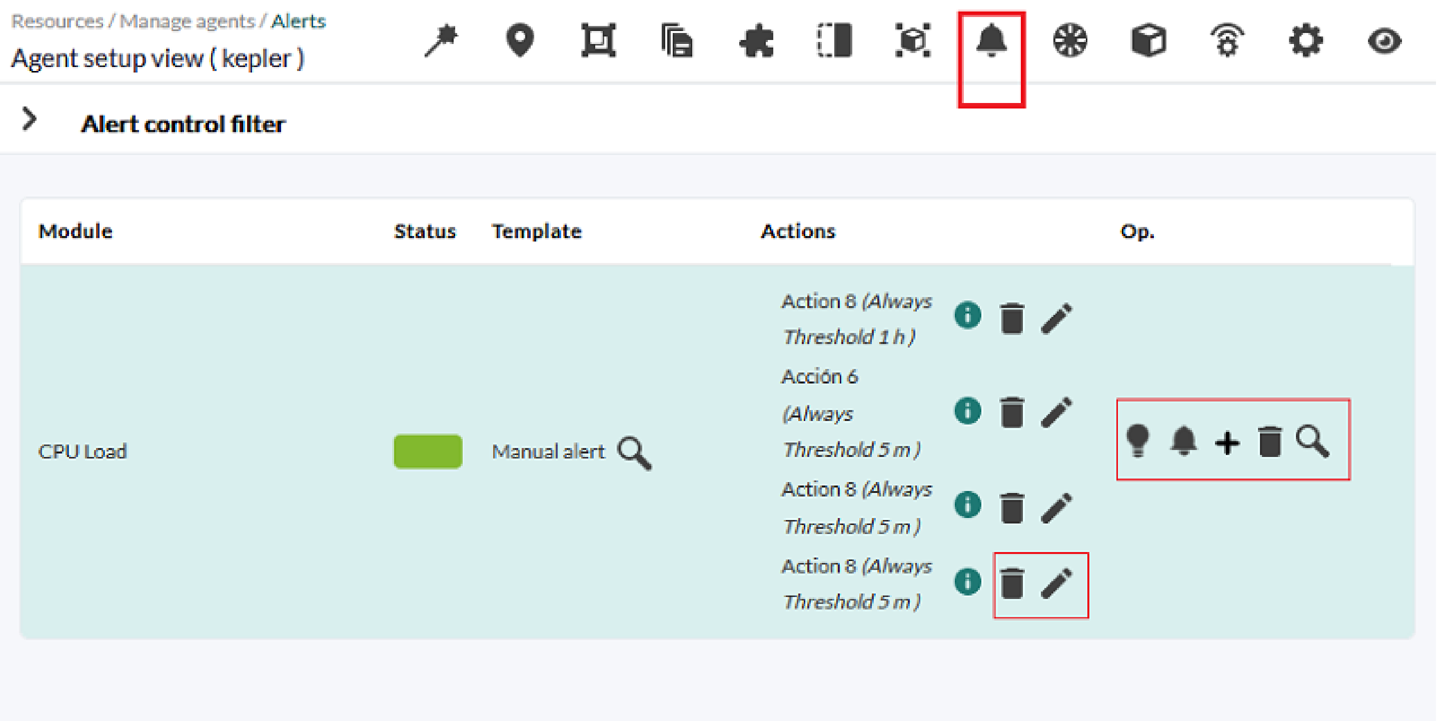
Alerts

Alert control filter

Agent	Status	Template	Actions	Op.
KEPLER CPU Load	🟩	Manual alert	Action 8 *(Always Threshold 1 h)* Action 7 *(Always Threshold 5 m)*	+
munchkin Host Alive	🟧	Critical condition		
munchkin_agent CPU IOWait	🟩	Critical condition		
Test Cluster status	🟩	Critical condition		

Manage alerts from the agent

From the agent administration section you can add new alerts by navigating to the corresponding tab:



Resources / Manage agents / Alerts

Agent setup view (kepler)

Alert control filter

Module	Status	Template	Actions	Op.
CPU Load	🟩	Manual alert	Action 8 *(Always Threshold 1 h)* Acción 6 *(Always Threshold 5 m)* Action 8 *(Always Threshold 5 m)* Action 8 *(Always Threshold 5 m)*	

There you can:

- Edit or delete each and every one of the actions of each alert assigned to the agent (column Actions).
- From the options column (Op.):
  - You can disable or enable.
  - You can put the alert in *standby* mode.
  - You can add an action.
  - You can clear the alert completely.
  - You can view the alert in detail.

Overview of an alert

- Define critical and warning threshold in module.
- Associate the alert to the module, to do so go to the alerts tab within the Agent where the Module is.
  - If necessary, you can create a new action and/or new template, by clicking on those buttons will be redirected to the corresponding sections. Once the new components have been created, you must return to the previous step.
- With the Add alert button the new alert is saved.
- *Alert escalation:* An alert escalation is additional actions that are executed if the alert is repeated a certain number of times consecutively.
  - It is only necessary to add the additional actions and determine between which consecutive repetitions (Number of matching alerts) of the alert this action is going to be executed.
  - When an alert recovers, all actions that have been executed up to that point will be executed again, not just those that correspond to the current Number of alerts match from setting.

  - Additionally, a Threshold can be placed as a second parameter, for which an alert cannot be launched more than once during said interval.
- Finally you can configure the sending of alert messages through instant messaging like Telegram, for example.

Standby Alerts

Alerts can be on, off, or in standby mode (*standby*). The difference between alerts that are disabled and alerts on *standby* is that alerts that are disabled simply won't work and therefore won't show up in the alerts view. Instead, alerts on *standby* will show up in the alert view and will work but only at display level. That is, it will show whether or not they are triggered *but they will not carry out the actions that they have programmed nor will they generate events*.

Alerts on *standby* are useful to be able to view them without disturbing other aspects.

Cascading Protection

Cascading protection is a feature of Pandora FMS that allows avoiding a massive bombardment of alerts when a group of Agents is not accessible, due to a failed main connection.

This type of thing happens, for example, when an intermediate network element such as a *router* or a *switch* fail, leaving a large part of the network managed with Pandora FMS inaccessible. Because the network checks would fail in this scenario, alerts would start triggering for downed devices without being true.

For the agent to work with cascading protection enabled, the parent Agent (Advanced options, token Parent) must be correctly configured, on which it depends.

If the parent Agent has at that moment any Module alert in a critical state, it firesda, the lower agent with cascading protection will not execute its alerts. This does not apply for module alerts in `warning` or `unknown` status.

Cascade protection is activated from the Agent configuration, Advanced options section, click on the Cascade protection modules and/or Cascade protection services option.

Service-based cascading protection

Version NG 727 or higher.

It is possible to use the Services to avoid alerts from multiple sources reporting the same incident.

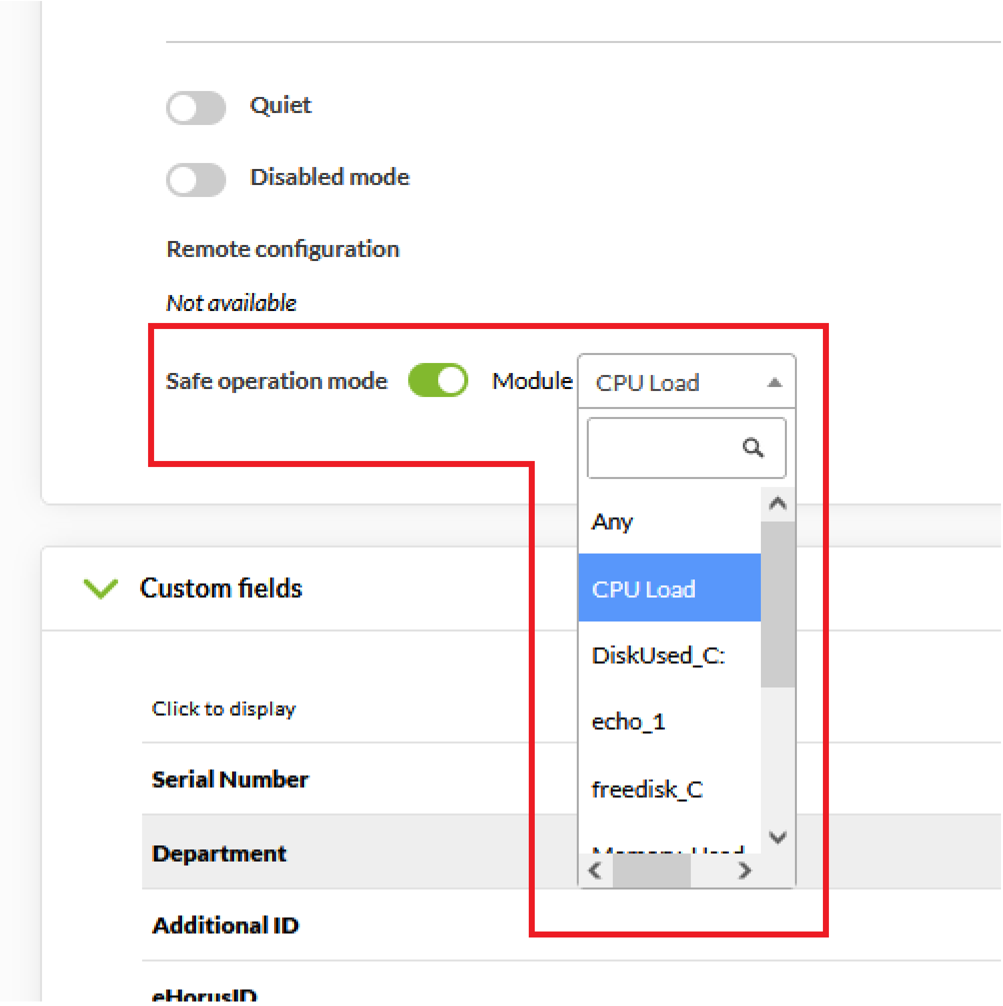
If Service-based cascading protection is activated, Service elements (Agents, Modules or other Services) will not report problems, but the Service itself will alert on behalf of the affected element.

In order to receive this information you must edit or create a new alert template, using the `_rca_` macro for a analisis de causa_raiz (*root cause analysis*).

Module-based cascading protection

You can use the status of a Parent Agent Module to prevent them from sending Agent alerts in case it goes critical.

Safe Mode of Operation



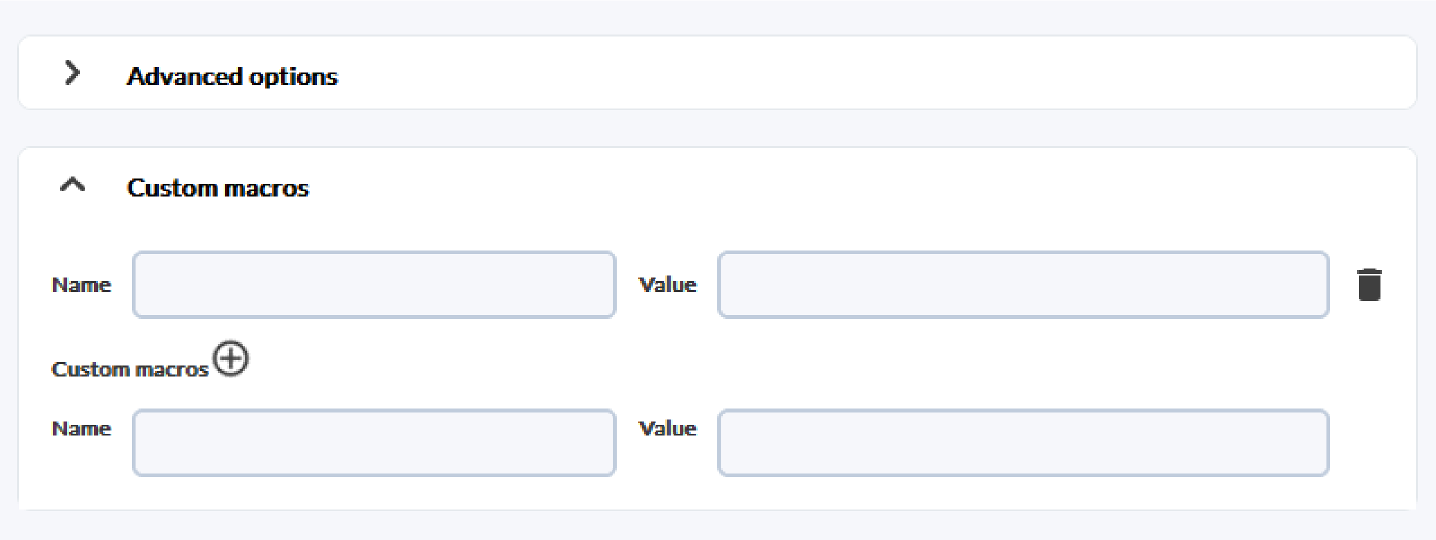
Safe operation mode
Module: CPU Load

Secure operation mode can be enabled in the advanced configuration options of an Agent.

If the status of the selected Module goes to `critical`, the rest of the Agent Modules are disabled until it returns to `normal` or `warning` again. This allows, for example, to disable Remote Modules if connectivity is lost.

Module Alert Custom Macros

These specific macros can be added by expanding the macros section of any module.



- They are defined in the Module.
- Store the data in the database.
- They can have any name, for example `_myMacro`.
- They are not reflected in the local configuration (`.conf`).
- Used exclusively for alerts.
- They cannot be defined at the component level.
- They can be defined in the monitoring policies.
- Set values can be used as part of the fields in the alert definition.

Email configuration for alerts in Pandora FMS

Pandora FMS by itself has the ability to send emails as explained in general configuration of the Console.

However, its flexibility allows sending email with different mail platforms.

Email configuration with Gmail account

In order for the Pandora FMS server to be able to send alerts via Google Mail® (Gmail®) account, proceed to general configuration of the Console or to the configuration of Pandora FMS server and enter your credentials (web domain, usernames, password, etc.).

Action Settings

- An action is added, for example with the name Mail to Admin.
- To configure the mail recipient, use the eMail command, adding the recipients in Destination address Field 1 separated by commas.

Alert Settings

In the Module configuration, Alerts tab, a new alert is created with the action created.

Email configuration with Office365 account

- You must have an account created in Office365.
- Proceed to the general configuration of the Console or to the configuration of the Pandora FMS server and place your credentials (Office365 web domain, usernames, password, etc.).

Correlation of alerts: alerts in events and logs

Alerts can be built based on the events received or on the data collected with the log collection system. Simple or more complex alerts can be built, based on a set of rules with logical relationships.

Log alerts are not executed in Command Center
(Metaconsole).

This type of alerts allows working from a much more flexible perspective, since alerts are not generated based on the status of a specific Module, but on an event that may have been generated by several different Modules, from different Agents.

Event alerts and/or logs are based on filter rules that use the following logical operators:

- and
- `or`
- `xor`
- `nand`
- `Nor`
- `nxor`

These logical operators are used to search for events/expressions in logs that match the configured filter rules, and if matches are found, the alert will be triggered.

When defining alerts about events, it will be essential to
indicate the parameters agent, module and event.

They also use the templates to define some parameters, such as the days on which the alert will work; however, in this case the templates do not determine when the event alert is fired, it is through the filter rules that the matching event alerts will be searched for and fired.

Given the high number of events that the Pandora FMS database can host, the server works on a maximum event window, which is defined in the `pandora_server.conf` configuration file through the `event_window` and `log_window` parameters. Events that have been generated outside this time window will not be processed by the server, so it does not make sense to specify in a rule a time window greater than the one configured on the server.

Creating Correlation Alerts

For the event correlation alerts to work, the event correlation server must be activated with the `eventserver 1` parameter in the Pandora FMS server configuration file.

Correlation Alerts and Templates

Management menu → Alerts → Alert correlation. In this global view, you will have the list of registered correlation alerts and the information about them, as well as options such as operating with the action disabled, in standby mode, adding more actions, editing or deleting the correlated alert.

With the Create button a new correlation alert is added, the process is similar to the creation of Alert Templates. The configuration parameters of the templates for correlation alerts are similar to those of a Module alert, there are only two specific parameters for event alerts:

- Rule evaluation mode: It can be Pass or Drop. The first means that, in case an event matches an alert, the rest of the alerts continue to be evaluated. Drop means that if an event matches an alert, the rest of the alerts are not evaluated.
- Group by: Allows you to group the rules by Agent, Module, alert or group. For example, if a rule is configured to trigger when two critical events are received and is grouped by Agent, two critical events must arrive from the same Agent. It can be disabled.

In case of alerts that contain logs rules, it will only affect the grouping by Agent. If you choose a different grouping, alerts based on log entries will never be honored.

Each rule is configured to jump to a certain type of event or *match* from log; when the logical equation defined by the rules and their operators is satisfied, the alert is triggered.

Rules within a correlation alert

To define the alert rules, it will be necessary to drag the elements on the left side to the drop area on the right side to build your rule.

Available setting items:

These elements will be enabled to guide the user in complying with the grammar of the rule. The following is a simplified explanation of the grammar to be used:

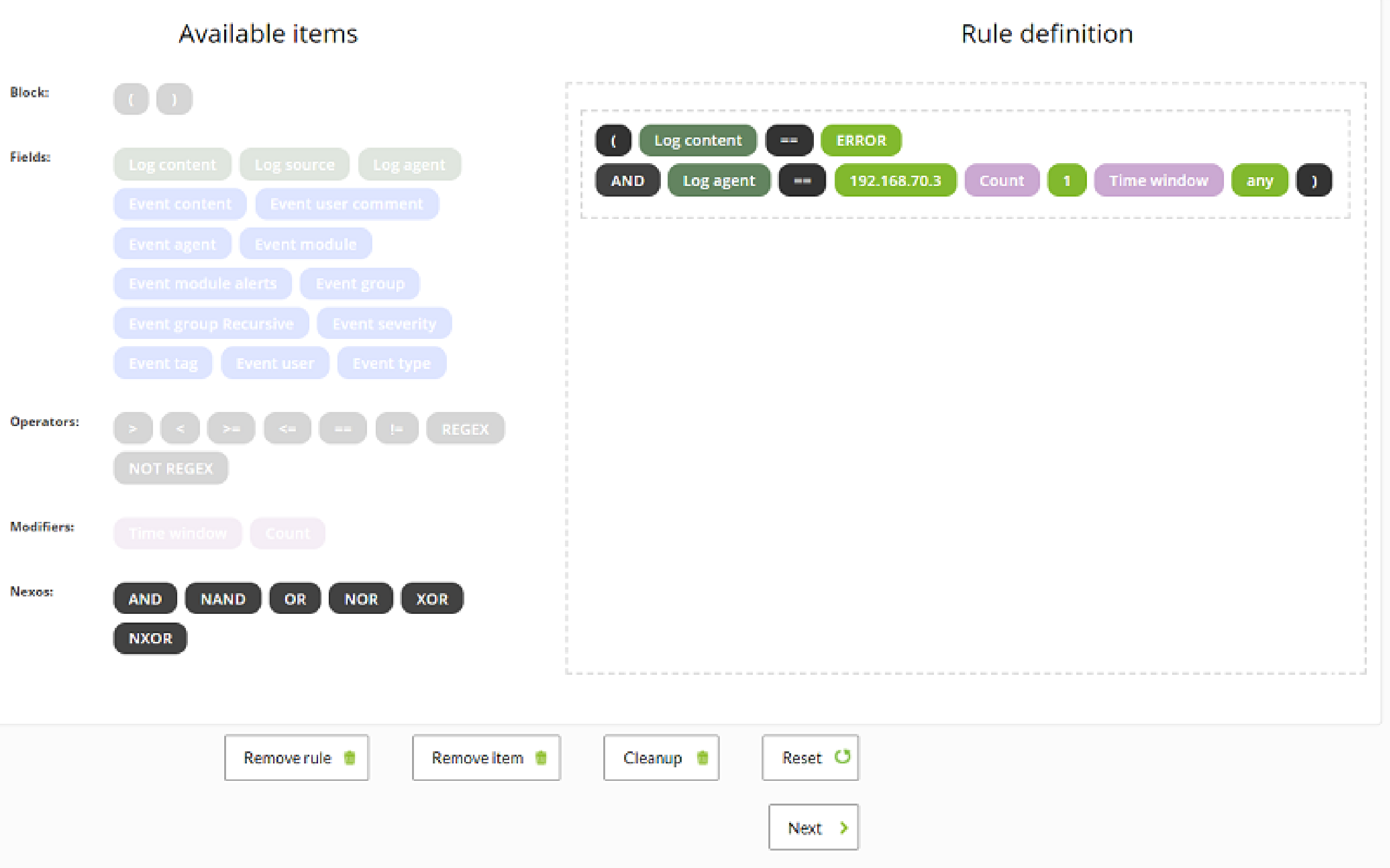
```
S → R | R + NEXUS +R

R → FIELD + OPERATOR + C | FIELD + OPERATOR + C + MODIFIER

C → VARIABLE
```

Where S is the set of rules defined for the correlated alert.

It will be necessary to drag the element over the area of definition of rules, in such a way that the image is similar to this one for example:



(Log content == ERROR
AND Log agent == 192.168.70.3 Count 1 Time window any)

The comparison operators == and != compare text strings literally. For more flexibility consider using the REGEX operator which uses Regular Expressions.

To clean and undo all changes there are two buttons: Cleanup and Reset.

It will only save the changes when you click the Next button.

Remember: The blocks have simultaneity when fulfilling the condition. Look at the following theoretical examples.

(A and B)

It forces the analyzed element (whether event or log) to fulfill A and B simultaneously.

A and B

It forces both rules (A) and (B) to be fulfilled in the evaluation window. This means that there must exist in the last few seconds (defined by the `log_window` and `event_window` parameters) entries that satisfy both rules.

Fields inside a correlation alert

Version 764 or later:

The macros related to modules and agents are not available in the Fields of the recovery section since the recovery of these alerts is executed when the threshold ends and lacks an event recovery to obtain such information.

In the previous section "Alerts System" the operation of the fields in alerts is explained in more detail.

Triggering within a Correlation Alert

In this section you must configure the actions that will be carried out when the alert is triggered and indicate at what intervals and how often said action will be executed.

- Actions: Action that needs to be executed.
- Number of alerts match: Number of intervals that have to pass since the alert was triggered for the action to be executed. If you need it to be always, you must leave these fields blank.
- Threshold: Interval that has to pass for the action to be executed again once the alarm is triggered.

Then display the list of configured actions. In this listing the triggering field shows at which alert intervals the action will be executed, as configured in number of alerts match. Also, in the Options column you can delete or modify the configured actions.

Multiple Correlated Alerts

When you have multiple alerts, they have a specific evaluation order. They will always be evaluated in order, starting first with the first in the list.

If the PASS rule evaluation mode is configured, if a correlated alert is executed, the following ones will be evaluated as well. It is *normal* mode.

If the DROP rule evaluation mode is configured, if a correlated alert configured with this mode is executed, it will stop the evaluation of the rules below it. This feature gives us the possibility of cascading alert protection.

The rest of the correlation rules (action fields and application of actions) work similar to the rest of Pandora FMS alerts.

Event Alert Macros

The macros that can be used within the configuration of an event alert are in macrolist.

Macro List

The Command Macros, Action Macros and Event Alert Macros are common to each other but with the following exceptions: `_modulelaststatuschange_` , `_rca_` and `_secondarygroups_`.

`_address_`

Address of the Agent that triggered the alert.

`_addressn_n_`

The address of the Agent that corresponds to the position indicated in n. Example: `addressn_1_` , `addressn_2_`

`_agent_`

Alias of the Agent that triggered the alert. If no alias is assigned, the Agent name is used.

`_agentalias_`

Alias of the Agent that triggered the alert.

`_agentcustomfield_n_`

Custom field number n of the Agent (eg `_agentcustomfield_9_`).

`_agentcustomid_`

Agent custom identifier.

`_agentdescription_`

Description of the Agent that triggered the alert.

`_agentgroup_`

Agent group name.

`_agentname_`

Name of the Agent that triggered the alert.

`_agentos_`

Agent operating system.

`_agentstatus_`

Current agent state.

`_alert_critical_instructions_`

Instructions contained in the Module for a `critical` state.

`_alert_description_`

Alert description.

`_alert_name_`

Alert name.

`_alert_priority_`

Numeric priority of the alert.

`_alert_text_severity_`

Alert text priority (Maintenance, Informational, Normal, Minor, Warning, Major, Critical).

`_alert_threshold_`

Alert threshold.

`_alert_times_fired_`

Number of times the alert was fired.

`_alert_unknown_instructions_`

Instructions contained in the Module for an unknown state.

`_alert_warning_instructions_`

Instructions contained in the Module for a `warning` state.

`_all_address_`

All the addresses of the Agent that triggered the alert.

`_critical_threshold_min_`

Minimum critical threshold.

`_critical_threshold_max_`

Maximum critical threshold.

`_data_`

Data that caused the alert to be triggered.

`_dataunit_`

It displays the unit type specified in the Unit field (located in the Advanced options section of an agent's module).

`_email_tag_`

Email mailboxes associated to the *tags* of Modules.

`_event_cf_text_`

(Only event alerts). It get all the information from *custom data* in text mode (with line breaks).

`_event_cf_json_`

(Only event alerts). It gets the information from *custom data* in JSON format.

`_event_cfX_`

(Only event alerts). Key of the custom field of the event that triggered the alert. For example, if there is a custom field whose key is IPAM, its value can be obtained using the `_event_cfIPAM_` macro.

`_event_description_`

(Only event alerts) Textual description of the Pandora FMS event.

`_event_extra_id_`

(Event alerts only) Extra identifier.

`_event_id_`

(Event alerts only) Identifier of the event that triggered the alert.

`_event_text_severity_`

(Event alerts only) Priority in text of the event that triggers the alert (Maintenance, Informational, Normal Minor, Warning, Major, Critical).

`_eventTimestamp_`

`Timestamp` in which the event was created.

`_fieldX_`

User-defined X field.

`_group_contact_`

Group contact information. It is configured when creating the group.

`_groupcustomid_`

Custom group identifier.

`_groupother_`

Other information about the group. It is configured when creating the group.

`_homeurl_`

It is a public URL link that must be configured in the general configuration options.

`_id_agent_`

Agent identifier, useful to build access URL to the Pandora FMS console.

`_id_alert_`

Alert identifier, useful for correlating the alert in third-party tools.

`_id_group_`

Agent group identifier.

`_id_module_`

Module identifier.

`_interval_`

Module execution interval.

`_module_`

Module name.

`_modulecustomid_`

Module custom identifier.

`_moduledata_X_`

Using this macro (“X” is the name of the Module in question) you collect the last data from this Module and if it is numeric, it returns it formatted with the decimals specified in the console configuration and with its unit (if it has one). It would be useful, for example, when sending an email when a Module alert is skipped, to also send additional (and perhaps very relevant) information from other modules of the same Agent.

If “X” (name of the Module in question) contains spaces,
these must be placed as an HTML entity:

`&#x20;`

You can view a list of HTML entities on Wikipedia.

`_moduledescription_`

Module description.

`_modulegraph_nh_`

(Only for alerts using the eMail command) It returns a base64-encoded image of a module graph with a period of n hours (eg `_modulegraph_24h_`). It requires the correct configuration of the connection from the server to the console through API, which is done in the server configuration file.

`_modulegraphth_nh_`

(Only for alerts that use the `_email_tag_` command) The same operation as the previous macro but only with the critical and warning thresholds of the Module, if they are defined.

`_modulegroup_`

Module group name.

`_modulestatus_`

Module status.

`_modulelaststatuschange_`

(For Command Macros only) *Timestamp* at which the module's last state change occurred.

`_moduletags_`

URLs associated with the *tags* of modules.

`_name_tag_`

Name of the *tags* associated to the Module.

`_phone_tag_`

Telephones associated to the *tags* of modules.

`_plugin_parameters_`

Module *plugin* parameters.

`_policy_`

Name of the policy to which the module belongs (if applicable).

`_prevdata_`

Previous data before the alert was triggered. It is necessary to uncomment the following section in the Pandora FMS server configuration file:

```
# Default texts for some events. The macros _module_ and _data_ are supported.
text_going_down_normal Module '_module_' is going to NORMAL (_data_) with
previous data (_prevdata_)
#text_going_up_critical Module '_module_' is going to CRITICAL (_data_)
#text_going_up_warning Module '_module_' is going to WARNING (_data_)
#text_going_down_warning Module '_module_' is going to WARNING (_data_)
#text_going_unknown Module '_module_' is going to UNKNOWN
```

The server process must be restarted for the new changes to be applied.

`_rca_`

Root cause analysis chain (for Services only).

`_secondarygroups_`

It shows the child groups of the Agent (only for command macros and action macros).

`_server_ip_`

IP address of the server to which the Agent is assigned.

`_server_name_`

Name of the server to which the Agent is assigned.

`_target_ip_`

IP address of the target of the Module.

`_target_port_`

Module target port.

`_timestamp_`

Time and date the alert was triggered.

`_time_down_human_`

Time in long format, for example: “1day 10h 35m 40s” (this macro only works for recovery alerts).

`_time_down_seconds_`

Time in seconds (this macro only works for recovery alerts).

`_timezone_`

The time zone represented by `_timestamp_`.

`_warning_threshold_max_`

Maximum warning threshold.

`_warning_threshold_min_`

Minimum warning threshold.

Back to Pandora FMS Documentation Index