MessageStorm Detection

White paper

About this paper

This white paper describes how to configure the HPOM for UNIX management server to detect and stop message storms from a managed node or storms caused by a certain policy on a managed node. It is assumed the reader is familiar with HPOM.

Functionality

Event Correlation Services (ECS) circuits are used to prevent message storms (either message-based or policy-based).

All messages that arrive at the management server are directed through the ECS circuit. For the message-based message storm, the rate at which these messages arrive is measured for each node. For the policy-based message storm, on the other hand, the rate at which these messages arrive is measured for each node and each policy. This rate is measured with a moving interval. If the rate of messages received exceeds the allowed message rate, a configurable action is started. While for the message-based message storm the default action calls a script that stops the node from causing a message storm, for the policy-based message storm the default action calls a script that disables the policy on the node. Directly after this action has been executed, a critical message is generated, which informs that node X (for the message-based message storm) or policy X on node Y (for the policy-based message storm) has caused the message storm, and that the configured action has been executed. In parallel, the message received rate for this node (for the message-based message storm) or the policy on the node (for the policy-based message storm) will be reset to zero to allow messages to pass through to the management server again. This reset is performed after a configurable delay that allows the management server processes to process the pending requests. As soon as the circuit receives the first message coming from this node after a storm detection and reset, you will get a message informing you that the message storm is over.

You can configure the circuit so that it does not send the messages that are received by the management server to the message browser until the message storm is stopped.

If you decide to suppress the messages, you are informed of the number of messages that have been suppressed, together with the message informing you that the message storm is over.

Note that for the policy-based message storm it is also possible to create exceptions, so some policies, nodes, or combinations of both are never disabled.

Message flow by example

Circuit configuration:

- Default interval is 5 minutes
- Default message rate is 0.3 messages per second (If over 90 messages are received in 5 minutes, a message storm is indicated).
- Default actions are used

Scenarios

Scenario 1:

- Agent recently installed on a web server
- First template distribution
- Long web server log files
- Logfile template configured to always read from start of file

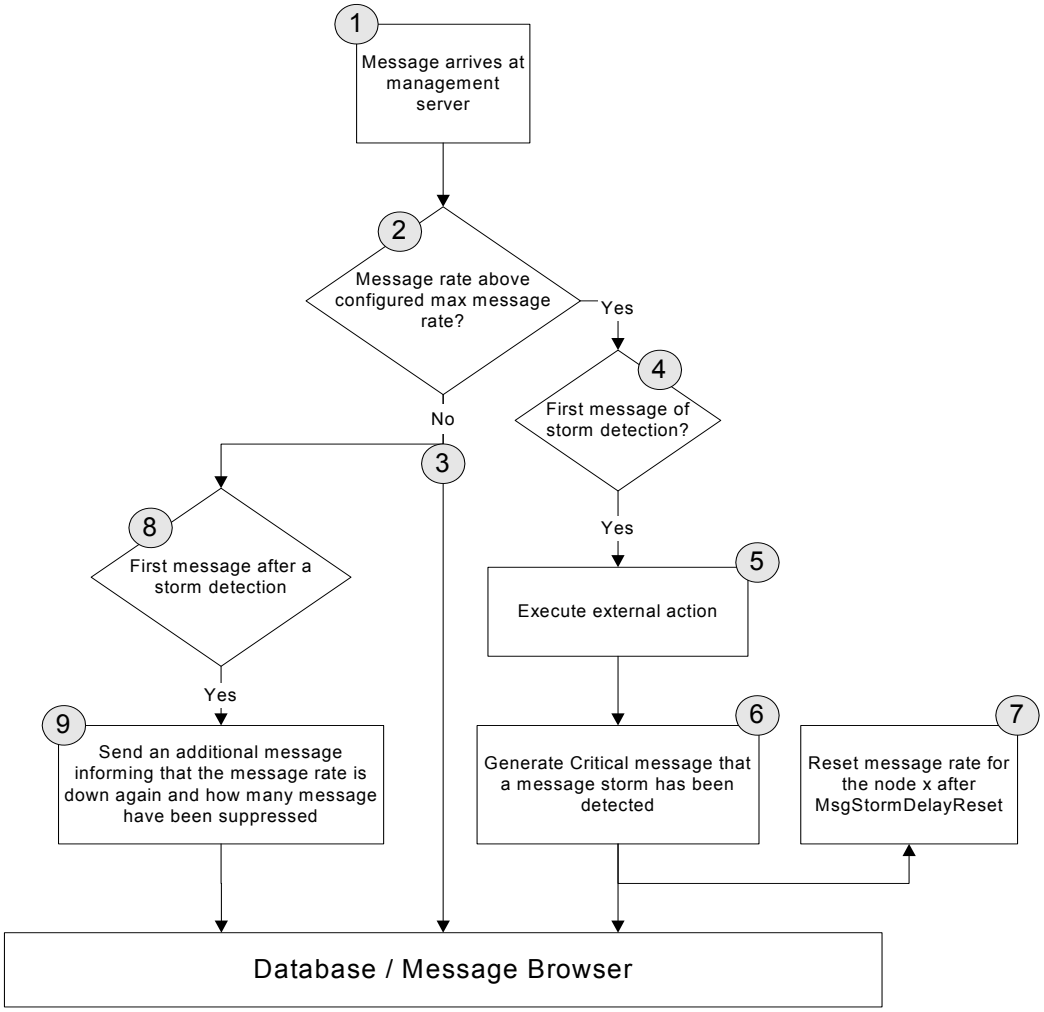
Scenario 2:

- Network segment 10.0.30.x was disconnected from the rest of the network due to a router problem during the weekend
- This caused a lot of problems with various applications in this segment
- Agents in this segment did not have a connection to the management server during this time
- Agents found a lot of problems and generated messages
- Network problem gets fixed on Monday and agents start sending their messages
- Result is a message flood on the management server

Suppression enabled

Default used, which means message suppression is enabled.

Figure A. Message flow when suppression is enabled



In case of the policy-based message storm (see Figure A), the message rate is reset for policy x on node y after PolicyStormDelayReset.

Possible message flows:

- Normal flow 1 -> 2 -> 3
- Flow when detecting a message storm 1 -> 2 -> 4 -> 5 -> 6 -> 7
- Flow after a message storm 1 -> 2 -> 3 & 3 -> 8 -> 9

Steps:

1. Messages arrive at the management server and are intercepted by the circuit.
2. The message rate is calculated for each node (for the message-based message storm) or for each policy on each node (for the policy-based message storm).

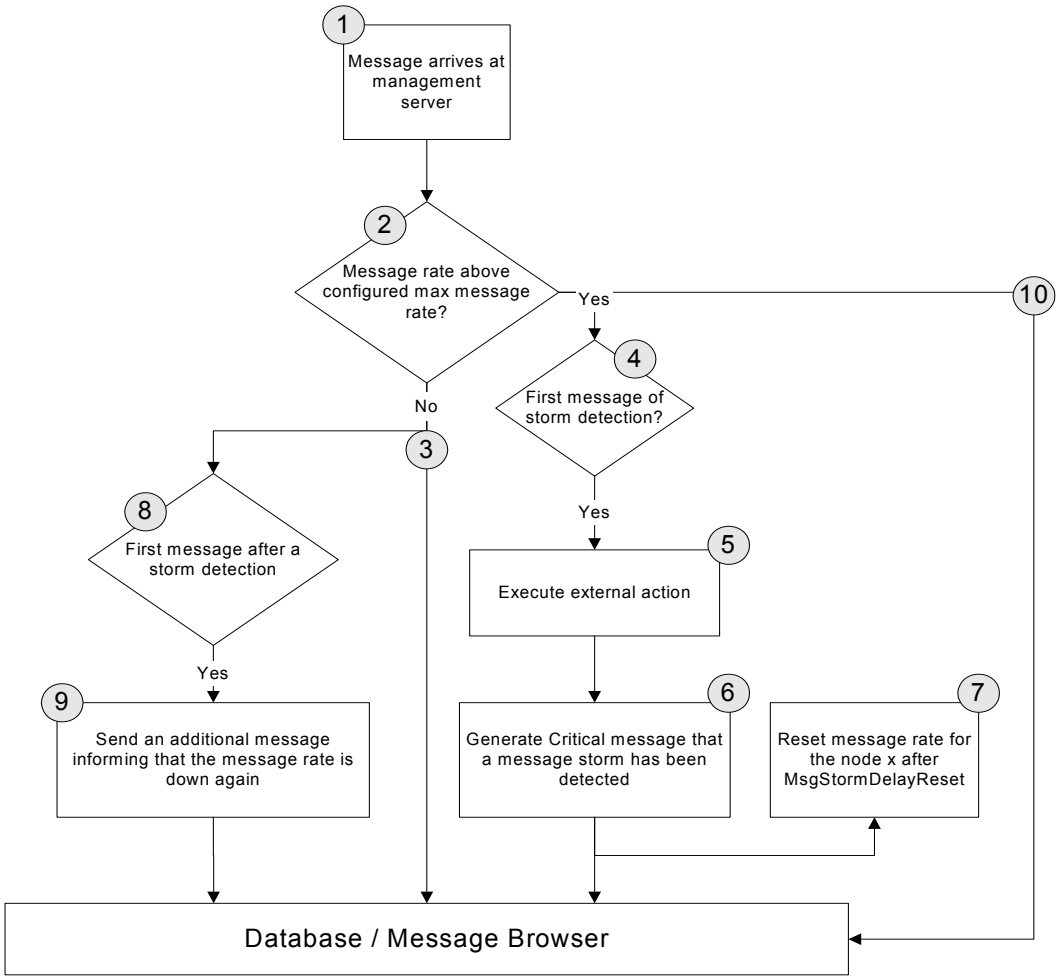
- The message rate is checked to ascertain whether it is below MsgStormRate or PolicyStormRate. In that case, there was no message storm for this node.
- If the message rate is above PolicyStormRate, then exceptions are checked if a policy, a node or a combination of both, causing the storm, should be ignored.
- In case that there was a storm for this node, the message rate must be below MsgStormRecoverRate or PolicyStormRecoverRate.

3. If no message storm is detected, send the message to the message browser.
4. If the message storm is detected, check whether this is the first time detection.
5. Execute MsgStormAction (when using the default, this will stop the agent) or PolicyStormAction (when using the default, this will disable the policy on the node).
6. Generate the critical message, which informs of a detected message storm.
7. With a delay of MsgStormDelayReset, reset the message rate that is stored for the particular node. With a delay of PolicyStormDelayReset, reset the message rate that is stored for the policy on the node.
8. Messages that are coming from 3 are copied to 7 in order to check whether a recently discovered message storm has actually ended.
9. Generate a message reporting that the message rate of node x (for the message-based message storm) or policy x on node y (for the policy-based message storm) has returned below MsgStormRecoverRate or PolicyStormRecoverRate, and report the number of messages that have been suppressed.

Suppression disabled

MsgStormSuppress is set to "false".

Figure B. Message flow when suppression is disabled



In case of the policy-based message storm (see Figure B), the message rate is reset for policy x on node y after PolicyStormDelayReset.

Possible message flows:

- Normal flow 1 -> 2 -> 3
- Flow when detecting a message storm
  1 -> 2 -> 4 -> 5 -> 6 -> 7 & 2 -> 10
- Flow after a message storm 1 -> 2 -> 3 & 3 -> 7 -> 8

In addition to the steps described for "Suppression enabled", step 10 is performed where messages are sent to the message browser even when a message storm has been detected.

Installation

Package content

The /opt/OV/contrib/OpC/MsgStorm directory consist of the following parts:

- Templates (upload tree for the templates)
- stormstartagt.sh & stormstopagt.sh
  (sample scripts used to start and stop the agent)
- stormenablepolicy.sh & stormdisablepolicy.sh

  (sample scripts used to enable and disable the policy)
- dstore.ds (a sample datastore for the message-based message storm detection, which allows the configuration of the circuit)
- ECpolicy_storm.ds (a sample datastore for the policy-based message storm detection, which allows the configuration of the circuit)

Hints

Users who already make use of the "ECS Management Server" default group should note that required ECS templates (MsgStorm_Dectect and PolicyStorm_Dectect) are placed into this template group after config upload.

In addition, optional message templates (MsgStormMessages and PolicyStormMessages) are placed into the "Management Server" default group.

Required section

1. Upload the policies by using the following command:

   ```
   > opccfgupld -replace -subentity \
   /opt/OV/contrib/OpC/MsgStorm/Templates
   ```

2. Configure the management server.

   Enable the MSI on the management server and divert all messages to MSI by using the following command:

   ```
   /opt/OV/bin/OpC/opcsrvconfig -msi -enable -send divert
   ```

3. Assign and distribute the server policy as follows:

- For message-based policy storm:

  ```
  /opt/OV/bin/OpC/utils/opcnode -assign_pol \
  pol_name='MsgStorm_Detect' pol_type=ec node_name='$MGMTSV' \
  net_type=NETWORK_NO_NODE
  ```

- For policy-based message storm:

  ```
  /opt/OV/bin/OpC/utils/opcnode -assign_pol \
  pol_name='PolicyStorm_Detect' pol_type=ec \
  node_name='$MGMTSV' net_type=NETWORK_NO_NODE
  ```

4. Create a data store file so that it is loaded together with the circuit.
5. To be able to run the circuit on the management serve, run the following command:

   ```
   /opt/OV/bin/OpC/opcragt -distrib -force '$MGMTSV'
   ```

   NOTE: You cannot use both circuits at the same time because duplicate messages could occur and prevent suppression from working.

   HPOM Manager Process (`opcecm`) must be running now. Verify this by using the `opcsv(1m)` command.
6. To allow uncertified remote actions on the management server, run the following command:

   /opt/OV/bin/OpC/opcsrvconfig -msi -allowext noautoactions operactions

   NOTE: This command also disables external automatic actions. Edit the command if you want the external automatic actions be enabled.

Optional section

After finishing the required section, you may proceed with the following optional steps.

Note that this only applies when using the stormstopagt.sh and stormstartagt.sh default scripts for the message-based message storm, or the stormdisablepolicy.sh and stormenablepolicy.sh default scripts for the policy-based message storm. These scripts generate messages that are intercepted by the management server node. The Message Template is used mainly to create the message key pattern in order to acknowledge the message storm detection messages.

1. Assign the "Default" or "Management Server" policy group by running the following command:

   /opt/OV/bin/OpC/utils/opcnode -assign_pol_group pol_group=<pol_group_name> \
   node_name=<node_name> net_type=NETWORK_IP

   where *<pol_group_name> is "Default" or "Management Server"*

You can also assign only the "MsgStormMessages" or "PolicyStormMessages" message policy from the "Management Server" policy group to the management server node. Run the following commands:

- For the "MsgStormMessages" policy:

  /opt/OV/bin/OpC/utils/opcnode -assign_pol pol_name=MsgStormMessages \
  pol_type=msgi node_name=*<node_name>* net_type=NETWORK_IP

- For the "PolicyStormMessages" policy:

  /opt/OV/bin/OpC/utils/opcnode -assign_pol \
  pol_name=PolicyStormMessages pol_type=msgi node_name=*<node_name>* \
  net_type=NETWORK_IP

2. Create a data store file so that it is loaded together with the circuit.
3. To distribute policies to the agent, run the following command:

   /opt/OV/bin/OpC/opcragt -distrib -templates *<node_name>*

Configuration

The circuit can be configured according to your needs by using an ECS datastore. The various values listed below can be changed in the datastore.

In order to make use of a datastore, you must place the datastore in the following directory:

`/var/opt/OV/shared/server/datafiles/policies/ec`

You can use either of the following:

- For the message-based message storm: the global datastore:
  `/opt/OV/contrib/OpC/MsgStorm/dstore.ds`
- For the policy-based message storm: the datastore that is bound to the circuit by using the circuit name:
  `/opt/OV/contrib/OpC/MsgStorm/ECpolicy_storm.ds`

A sample of a datastore file for the message-based message storm is illustrated below:

```
#/var/opt/OV/shared/server/datafiles/policies/ec/ECmsg_storm.ds#03/05/2002#1
#0#
ADD DATA ("MsgStormInterval"       , 5m )
ADD DATA ("MsgStormRate"           , 0.3 )
ADD DATA ("MsgStormRecoverRate"    , 0.15 )
ADD DATA ("MsgStormDelayReset"     , 1m30s )
ADD DATA ("MsgStormSuppress"       , true)
ADD DATA ("MsgStormAction"         ,
             "/opt/OV/contrib/OpC/MsgStorm/stormstopagt.sh")
ADD DATA ("MsgStormActionTimeOut"  , 1m30s )
ADD DATA ("MsgStormOperatorAction" ,
             "/opt/OV/contrib/OpC/MsgStorm/stormstartagt.sh")
ADD DATA ("MsgStormMsgKeyPrefix"   , "EC_OVOMsgStormDetected" )
ADD DATA ("MsgStormMaxTransitDelay", 2m )
ADD DATA ("MsgStormServiceName"    , "")
```

A sample of a datastore file for the policy-based message storm is illustrated below:

```
#/var/opt/OV/shared/server/datafiles/policies/ec/ECpolicy_storm.ds#03/05/200
8#1#0#
ADD DATA ("PolicyStormInterval"       , 5m )
ADD DATA ("PolicyStormRate"           , 0.3 )
ADD DATA ("PolicyStormRecoverRate"    , 0.15 )
ADD DATA ("PolicyStormDelayReset"     , 1m30s )
ADD DATA ("PolicyStormSuppress"       , true)
ADD DATA ("PolicyStormAction"         ,
          "/opt/OV/contrib/OpC/MsgStorm/stormdisablepolicy.sh" )
```

```
ADD DATA ("PolicyStormActionTimeOut"  , 1m30s )
ADD DATA ("PolicyStormOperatorAction" ,
          "/opt/OV/contrib/OpC/MsgStorm/stormenablepolicy.sh" )
ADD DATA ("PolicyStormMsgKeyPrefix"   , "EC_OVOPolicyStormDetected" )
ADD DATA ("PolicyStormMaxTransitDelay", 2m )
ADD DATA ("PolicyStormServiceName"    , "" )
ADD DATA ("PolicyStormExceptions"     , [ "node1:policy1", ":policy2",
          "node2" ] )
```

Apply configuration changes

NOTE: You cannot apply configuration changes as described in this section to the following message-based and policy-based message storm configuration variables: *MsgStormInterval*, *MsgStormDelayReset*, *MsgStormActionTimeOut*, *MsgStormMaxTransitDelay*, *PolicyStormInterval*, *PolicyStormDelayReset*, *PolicyStormActionTimeOut*, and *PolicyStormMaxTransitDelay*. In these cases, you must first update the ECpolicy_storm.ds file, deassign and distribute the policy, and then assign and distribute the policy again.

- You can make the configuration in the following files:
  - for the message-based message storm:
    `/var/opt/OV/shared/server/datafiles/policies/ec/dstore.ds`
    or
    `/var/opt/OV/shared/server/datafiles/policies/ec/ECmsg_storm.ds`
  - for the policy-based message storm:
    `/var/opt/OV/shared/server/datafiles/policies/ec/ECpolicy_storm.ds`

- To update your changes, call for the message-based message storm:
  `ecsmgr -instance 11 -data_update ECmsg_storm \`
  `/var/opt/OV/shared/server/datafiles/policies/ec/ECmsg_storm.ds`

- To update your changes, call for the policy-based message storm:
  `ecsmgr -instance 11 -data_update ECpolicy_storm \`
  `/var/opt/OV/shared/server/datafiles/policies/ec/ECpolicy_storm.ds`

Message-based message storm configuration variables

MsgStormInterval

Type Duration

Default 5m

Format <hour>h<minute>m<second>s

Description

The time period over which the message flow is analyzed.

MsgStormRate

Type Real

Default 0.3

Description

In conjunction with MsgStormInterval, MsgStormRate specifies how many messages can arrive within the MsgStormInterval variable time period before a message storm is detected.

The rate (in messages per second) is calculated as follows:

$$\text{MsgStormRate} = \frac{\text{Number of messages received in MsgStormInterval}}{\text{MsgStormInterval (in seconds)}}$$

If MsgStormRate measured over MsgStormInterval is higher than the maximum allowable rate configured, it is considered to be a message storm.

For example, if you have selected MsgStormRate of 0.3 and MsgStormInterval of 5 minutes, the circuit will report a message storm as soon as there are more than 90 messages within the 5 minute period or less time.

MsgStormRecoverRate

Type Real

Default 0.15

Description

If you have chosen to suppress messages during a message storm, this value can be used to define the message rate under which a node has to fall for the circuit

to assume that the message storm is over.
Using the default values, the message rate must fall below 45 messages within 5 minutes.
For the calculation of MsgStormRecoverRate, see MsgStormRate.

MsgStormDelayReset

Type	Duration
Default	1m
Format	<hour>h<minute>m<second>s

Description

Delay used between detecting a message storm and resetting the message rate to 0 to allow further messages to pass through.
This delay suppresses further messages from the same node that caused the message storm as they may already be in the server queues but not processed by the message manager.

MsgStormSuppress

Type	Boolean
Default	true
Format	true | false

Description

true – Prevents flooding the database and the message browser with messages.
false – Generates message information of the message storm, but all messages are still sent to the database and to the message browser.

MsgStormAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormstopagt.sh"

Description

Command to call within the circuit as soon as a message storm has been detected. The command will be called with the following parameters:

<Action> <nodename> <MSGID>

MsgStormActionTimeOut

Type	Duration
Default	1m30s
Format	<hour>h<minute>m<second>s

Description

Time period that the circuit waits for the <action> to execute. After this, the circuit will proceed and will no longer wait for the application end. Note that the application will not be stopped, but the return code and output will be ignored.

MsgStormOperatorAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormstartagt.sh"

Description

Operator initiated action which is assigned to the message storm warning to run on <$OPC_MGMTSV>. The default action allows the operator to restart the agent that caused the storm.

The following call is used:

<Action> <nodename> <MSGID>

MsgStormMsgKeyPrefix

Type	String
Default	"EC_OVOMsgStormDetected "

Description

Prefix to add to the message key used in the message generated by the circuit. There are two different formats of keys generated:

1. <MsgKeyPrefix>:start:<nodename> is generated when the message storm is detected.
2. <MsgKeyPrefix>:end:<nodename> is generated when the circuit detects that the message storm is over.

MsgStormMaxTransitDelay

Type	Duration
Default	2m
Format	<hour>h<minute>m<second>s

Description

The ECS engine checks whether incoming messages are older than this MaxTransitDelay. Older messages are not processed with the circuit.
The delay is the difference between the message arrival time on the server and the current time.

MsgStormServiceName

Type	String
Default	""

Description

Service name that should be used within the messages generated by the ECS circuit. As soon as this string is defined, the circuit will generate a service name with the following format: <MsgStormServiceName><node>.

MsgStormAckStartMsg

Type	Boolean
Default	false
Format	true | false

Description

When this variable is set to true, the message key relation is enabled: the Normal messages described in "Message rate below configured threshold level", which indicate the message storm is over, acknowledge the Critical messages described in "Message storm detection", which indicate the message storm was detected.

Policy-based message storm configuration variables

PolicyStormInterval

Type	Duration
Default	5m
Format	<hour>h<minute>m<second>s

Description

The time period over which the message flow is analyzed.

PolicyStormRate

Type	Real
Default	0.3

Description

In conjunction with PolicyStormInterval, PolicyStormRate specifies how many messages can arrive within the PolicyStormInterval variable time period before a message storm is detected.

The rate (in messages per second) is calculated as follows:

$$\text{PolicyStormRate} = \frac{\text{Number of messages received in PolicyStormInterval}}{\text{PolicyStormInterval (in seconds)}}$$

If PolicyStormRate measured over PolicyStormInterval is higher than the maximum allowable rate configured, it is considered to be a message storm.

For example, if you have selected PolicyStormRate of 0.3 and PolicyStormInterval of 5 minutes, the circuit will report a message storm as soon as there are more than 90 messages within the 5 minute period or less time.

PolicyStormRecoverRate

Type	Real
Default	0.15

Description

If you have chosen to suppress messages during a message storm, this value can be used to define the message rate under which a policy on a node has to fall

for the circuit to assume that the message storm is over.
Using the default values, the message rate must fall below 45 messages within 5 minutes.
For the calculation of PolicyStormRecoverRate, see PolicyStormRate.

PolicyStormDelayReset

Type	Duration
Default	1m
Format	<hour>h<minute>m<second>s

Description

Delay used between detecting a message storm and resetting the message rate to 0 to allow further messages to pass through.
This delay suppresses further messages from the same policy on the same node that caused the message storm as they may already be in the server queues but not processed by the message manager.

PolicyStormSuppress

Type	Boolean
Default	true
Format	true | false

Description

true – Prevents flooding the database and the message browser with messages.
false – Generates message information of the message storm, but all messages are still sent to the database and to the message browser.

PolicyStormAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormdisablepolicy.sh"

Description

Command to call within the circuit as soon as a message storm has been detected. The command will be called with the following parameters:

<Action> <nodename> <policy> <MSGID>

PolicyStormActionTimeOut

Type	Duration
Default	1m30s
Format	<hour>h<minute>m<second>s

Description

Time period that the circuit waits for the <action> to execute. After this, the circuit will proceed and will no longer wait for the application end. Note that the application will not be stopped, but the return code and output will be ignored.

PolicyStormOperatorAction

Type	String
Default	"/opt/OV/contrib/OpC/MsgStorm/stormenablepolicy.sh"

Description

Operator-initiated action which is assigned to the message storm warning to run on <$OPC_MGMTSV>. The default action allows the operator to reenable the policy on the node that caused the storm.

The following call is used:

<Action> <nodename> <policy> <MSGID>

PolicyStormMsgKeyPrefix

Type	String
Default	"EC_OVOPolicyStormDetected "

Description

Prefix to add to the message key used in the message generated by the circuit. There are two different formats of keys generated:

3. <MsgKeyPrefix>:start:<nodename>:<policy> is generated when the message storm is detected.
4. <MsgKeyPrefix>:end:<nodename>:<policy> is generated when the circuit detects that the message storm is over.

PolicyStormMaxTransitDelay

Type	Duration
Default	2m
Format	<hour>h<minute>m<second>s

Description

The ECS engine checks whether incoming messages are older than this MaxTransitDelay. Older messages are not processed with the circuit.
The delay is the difference between the message arrival time on the server and the current time.

PolicyStormServiceName

Type	String
Default	""

Description

Service name that should be used within the messages generated by the ECS circuit. As soon as this string is defined, the circuit will generate a service name with the following format: <PolicyStormServiceName><node>:<policy>.

PolicyStormExceptions

Type	List of strings
Default	[ ]

Description

List of <node>:<policy> combinations, which should be excepted from the message storm detection. If "<node>" is specified, then all policies from this node will be excepted. If ":<policy>" is specified, then the policy will be excepted on all nodes.

PolicyStormAckStartMsg

Type	Boolean
Default	false
Format	true | false

Description

When this variable is set to true, the message key relation is enabled: the

Normal messages described in “Message rate below configured threshold level”, which indicate the message storm is over, acknowledge the Critical messages described in “Message storm detection”, which indicate the message storm was detected.

Limitations

There are some limitations when you apply these circuits.

Using several ECS circuits on the management server

The ECS engine always processes all circuits in parallel. This means that when you have more than the message-based message storm ECS circuit or the policy-based message storm ECS circuit, the other circuits may also process messages and pass them to the message browser even when a message storm has been detected. All actions described above will be taken and you will also get a message about a detected message storm, but you may still receive messages from that node even when you selected to suppress all messages after a message storm has been detected.

Message storm caused by the agent on the management server (for the message-based message storm only)

In case the agent on your management server generates a message storm and you use the default action or your own script to stop the agent, all operator-initiated actions, which are added to the message storm detection warnings, will fail since they would be executed on that node.

The consequence is that you have to restart the agent on the management server manually.

To avoid this, you can modify the stormstopagt.sh script to behave differently when stopping the node on the management server.

Proxy nodes

In case of a node acting as a proxy for another device that does not have an agent installed, the proxy node is stopped (or the policy is disabled on it for the policy-based message storm) when using the default scripts. This may happen when using the OS390/SPI that sends all messages by using the node on the management server.

This can be handled by modifying the stormstopagt.sh or stormdisablepolicy.sh script.

Default Values

The default values, used to define whether a message storm occurs or not, might not be the best for your environment and can be changed according to your needs.

Message storm caused by the policy assigned to the virtual node in the cluster environment (for the policy-based message storm only)

It is currently not possible to enable or disable templates assigned to virtual nodes in the cluster environment. The templates are automatically set to disabled or enabled depending on which physical node the virtual package is running. It is also not possible to disable the templates though the package is running.

The consequence is that the policy-based message storm does not work on the templates, which are assigned to the virtual nodes in the cluster environment.

Appendix: Messages generated by the ECS circuit

For the message-based message storm

1. Message storm detection

1.1 Action successfully executed

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> Rate is : <configured message rate> AutoAction successfull executed, Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent a reoccuring of the message storm after agent start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

1.2 Action returned an exit code different from 0

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> action <MsgStormAction> failed, ExitCode: <Action exit code> Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent a reoccuring of the message storm after agent

	start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

1.3 Action timed out

Severity	critical
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message Storm detected on <node> automatic action <MsgStormAction> timed out (time out is set to <MsgStormActionTimeOut>) Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent a reoccuring of the message storm after agent start
Message key	<MsgStormMsgKeyPrefix>:start:<node>
Service name	<MsgStormService><node>

2. Message rate below configured threshold level

2.1 With suppression

Severity	normal
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message rate from node <node> went down under the configured rate of <MsgStormRecoverRate> after a Messagestorm has been detected. <suppress count> messages have been suppressed during the storm. The message storm which has been detected for this node seems to be over since the message rate of this node is down under the configured recover rate. It can also happen that the storm didn't end but the rate has been reset since the storm remains longer than the configured reset delay after a storm detection.
Message key	<MsgStormMsgKeyPrefix>:start:<node>[1]
Service name	<MsgStormService><node>

[1] This key relation is set only if the MsgStormAckStartMsg variable is set to true. Otherwise, it is empty.

2.2 Without suppression

Severity	normal
Group	OpC
Application	OpC
Object	MsgStorm
Message text	Message rate from node <node> went down under the configured rate of <MsgStormRecoverRate> after a Messagestorm has been detected. The message storm which has been detected for this node seems to be over since the message rate of this node is down under the configured recover rate. It can also happen that the storm didn't end but the rate has been reset since the storm remains longer than the configured reset delay after a storm detection.
Message key	<MsgStormMsgKeyPrefix>:start:<node>[2]
Service name	<MsgStormService><node>

Note that the service name is only created when MsgStormService is set.

[2] This key relation is set only if the MsgStormAckStartMsg variable is set to true. Otherwise, it is empty.

For the policy-based message storm

3. Message storm detection

3.1 Action successfully executed

Severity	critical
Group	OpC
Application	OpC
Object	PolicyStorm
Message text	Message Storm detected on <node> policy: <policy> Rate is : <configured message rate> AutoAction successfull executed, Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent the message storm from reoccurring after enabling policy
Message key	<PolicyStormMsgKeyPrefix>:start:<node>:<policy>
Service name	<PolicyStormService><node>:<policy>

3.2 Action returned an exit code different from 0

Severity	critical
Group	OpC
Application	OpC
Object	PolicyStorm
Message text	Message Storm detected on <node> policy: <policy> action <PolicyStormAction> failed, ExitCode: <Action exit code> Output: <action output> Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent the message storm from reoccurring after enabling policy
Message key	<PolicyStormMsgKeyPrefix>:start:<node>:<policy>
Service name	<PolicyStormService><node>:<policy>

3.3 Action timed out

Severity	critical
Group	OpC
Application	OpC
Object	PolicyStorm
Message text	Message Storm detected on <node> policy: <policy> automatic action <PolicyStormAction> timed out (time out is set to <PolicyStormActionTimeOut>) Hints for further processing: - find the cause of the message storm - check queues (msgagtq) and buffers (msgagtdf) on the node which caused the message storm (UX: /var/opt/OV/tmp/OpC, WIN: <drive>:\usr\OV\tmp\OpC) Tools: opcqchk and opcdfchk (opcqchk msgagtq, opcdfchk msgagtdf -p) - in case the check of the temp file shows a high number of items left on the node it might be better to remove the file to prevent the message storm from reoccurring after agent start
Message key	<PolicyStormMsgKeyPrefix>:start:<node>:<policy>
Service name	<PolicyStormService><node>:<policy>

4. Message rate below configured threshold level

4.1 With suppression

Severity	normal
Group	OpC
Application	OpC
Object	PolicyStorm
Message text	Message rate from node <node> policy: <policy> went down under the configured rate of <PolicyStormRecoverRate> after a message storm has been detected <suppress count> messages have been suppressed during the storm. As the message rate of this node and policy is again below the configured recover rate, the message storm detected for this node and policy seems may be over. Or, the message storm is still not over, but the rate has been reset since the storm remains longer than the configured reset delay after storm detection.
Message key	<PolicyStormMsgKeyPrefix>:start:<node>:<policy>[3]
Service name	<PolicyStormService><node>:<policy>

[3] This key relation is set only if the PolicyStormAckStartMsg variable is set to true. Otherwise, it is empty.

4.2 Without suppression

Severity	normal
Group	OpC
Application	OpC
Object	PolicyStorm
Message text	Message rate from node <node> policy: <policy> went down under the configured rate of <PolicyStormRecoverRate> after a message storm has been detected. As the message rate of this node and policy is again below the configured recover rate, the message storm detected for this node and policy seems may be over. Or, the message storm is still not over, but the rate has been reset since the storm remains longer than the configured reset delay after storm detection.
Message key	<PolicyStormMsgKeyPrefix>:start:<node>:<policy>[4]
Service name	<PolicyStormService><node>:<policy>

Note that the service name is only created when MsgStormService is set.

[4] This key relation is set only if the PolicyStormAckStartMsg variable is set to true. Otherwise, it is empty.

© 2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.
HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products. Intel®, Itanium®, and Pentium® are trademarks of Intel Corporation in the U.S. and other countries.
Java is a registered trademark of Oracle and/or its affiliates.
Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.
Oracle is a registered trademark of Oracle and/or its affiliates.
UNIX® is a registered trademark of The Open Group.

May 2014