XML API version 4.6.3

Environment

McAfee Database Activity Monitoring (DAM) 4.x
McAfee Vulnerability Manager for Databases 4.x

NOTE: Some parameters were added in the latest product version. To ensure all parameters are supported, upgrade your product to the latest version.

Solution

Use the following procedure for accessing and testing the DAM XML API.

The XML API enables you to query the DAM Server for data via an XML interface.

Enabling the XML API allows you to request information from the DAM Server using a standardized HTTP GET/POST request and receive the response in an XML format. You can find the detailed structure of the XML reply in the XSD file located in the **System**, **Interfaces**, **XML API** tab.

To use the XML API, you must provide the login credentials of a user with the **Use XML API** permission granted. Intel Security recommends that you create a dedicated user for this purpose.

To configure the XML API:

1. Click the **Interfaces** tab on the **System** page and select **XML API**. You will see the **XML API** tab.
2. Select **XML API enabled**.
3. Click **Save**. The XML API is enabled.

To create a dedicated user:

1. On the **Permissions** page, select **Users**, **Create New User** to add a user.
2. Assign the new user All XML API permissions only.

The XML API uses a Representational State Transfer (REST) format for request response processing. The XML REST API receives requests via standardized HTTP GET/POST parameters and the response is an XML response.

To use the XML API:

1. Enable the service through the **System** tab.
2. Access the service by authenticating with a user that has the proper 'XML API' permission (default administrator user has the relevant permission). Authentication is done via HTTP Basic Auth.
3. Use the following base URL for a request: <server URL>**/xmlapi.svc**.

The XML API allows you to perform queries regarding alerts, VA results, DBMSs, sensors, etc.

The type of query is specified with the request parameter named **service**, which can receive one of the following parameters:

- **sensor** - Query sensor status, Start/Stop monitoring
- **alert** - Query alerts
- **varesult** - Query VA results
- **dbms** - Query dbms list including summary of last VA scans
- **scans** - Query VA scans
- **add-db** - Add and configure a VA database
- **update-db** - Update a VA database
- **delete-db** - Delete a database
- **add-db-to-group** - Add a database to a group
- **remove-db-from-group** - Remove a database from a group
- **add-scan** - Add a VA scan
- **update-scan** - Update a VA scan
- **delete-scan** - Delete a VA scan
- **start-scan** - Initialize a VA scan
- **ruleobject** - Set, Delete and List rule objects
- **resolve-results** - Resolving of results
- **resolve-alerts** - Resolving of alerts
- **sensor-restart** – Restart sensors
- **rules** - Perform operations on rules
- **dbgroups** - Perform operation on database groups

To test the XML API, a simple browser request can be used. For example, to get a list of sensors and their current state, issue the following request in a browser (replace localhost with relevant domain name or IP address):

https://localhost:8443/xmlapi.svc?service=sensor

Each service supports a set of parameters that can be used to limit the request. For example, the **alert** service supports the **HH$TimeBackPeriod** parameter, which specifies the time back in milliseconds that an alert was executed at. For example, to get all alerts from the last five minutes, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=alert&HH$TimeBackPeriod=300000

List of supported request parameters

Sensor request parameters:

Description	Usage A - Retrieves Sensors information. The parameters are used to filter which sensors to list. Usage B - start/stop monitoring database .- Once "action" parameter is provided with values: start-db,stop-db
Permission	SENSOR XML API
Version	4.4.9 or higher

- HH$Action=list(default), start-db, stop-db
- HH$Name - String
- HH$Id - long
- HH$Sid - String
- HH$ApprovedBy - User id
- HH$Hostname - String
- HH$Ip - String
- HH$Database - Database id
- HH$Approved - CSV Approved status
- HH$Status - CSV CommStatus status
- HH$Server - Server id
- HH$pageSize - long - the maximum number of results per call, default = -1 (ALL)
- HH$pageNum - long - the number of the page to return, default = 0
- HH$Sensor-db-id - long - (optional) Database-sensor-db - This number reflects the connection between sensor and database
  Notice: you can get this variable value from using the regular "sensor" service.
  find the related xml node <database> of your sensor and inside you may find xml node xml <sensor-db-id>

Examples:

Usage A:

Get all the connected sensors:

https://127.0.0.1:8443/xmlapi.svc?service=sensor&HH$status=ALIVE

Usage B:

Start monitoring database:

https://127.0.0.1:8443/xmlapi.svc?service=sensor&HH$Action=start-db&HH$Sensor-db-id=10000000

Stop monitoring database:

https://127.0.0.1:8443/hedgehog/xmlapi.svc?service=sensor&HH$Action=stop-db&HH$Sensor-db-id=10000000

Alert request parameters:

Description	Retrieves Alerts information. The parameters are used to filter which alerts to list
Permission	ALERT XML API
Version	4.4.9 or higher

- HH$ExecutionTimeFrom - Date (format: dd MMM yyyy HH:mm:ss)
- HH$ResolveReason - String
- HH$Resolves - CSV Resolve ids
- HH$Id - long
- HH$Agents - CSV Agent ids
- HH$TagName - String
- HH$DbGroupName - String
- HH$ExecutionTimeTo - Date
- HH$Databases - CSV Database ids
- HH$Operation - String
- HH$OsUser - String
- HH$ResolvedBy - User id
- HH$Severities - CSV ActionSeverity names (INFO, NOTICE, LOW, MEDIUM, HIGH)
- HH$SourceHost - String
- HH$SourceIP - String
- HH$Rules - CSV Rule ids
- HH$ResolveNames - CSV Resolve name string
- HH$RuleName - String
- HH$QuarantineId - String
- HH$ReleaseTimeAfter - Date
- HH$ReleaseTimeBefore - Date
- HH$ExecUser - String
- HH$DatabaseId - database Long
- HH$ExecProgram - String
- HH$Clientid - String
- HH$Module - String
- HH$ModifyDateFrom - Date
- HH$ModifyDateTo - Date
- HH$Sid - Integer
- HH$TimeBackPeriod - Long
- HH$pageSize - long - the maximum number of results per call, default = 100. The value -1 is for all the results

  NOTE: ResolveNames works only with already existing resolve types.

For example, to get maximum of 5000 unresolved alerts from host “myhost”, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=alert&HH$SourceHost=myhost&HH$ResolveNames=Unresolved&HH$pageSize=5000

VA Result request parameters:

Description	Retrieves VA Results information. The parameters are used to filter which VA results to list
Permission	VA Result XML API
Version	4.4.9 or higher

- HH$ExecutionTimeFrom - Date (format: dd MMM yyyy HH:mm:ss)
- HH$ResolveReason - String
- HH$Resolves - CSV Resolve ids
- HH$Id - long
- HH$TagName - String
- HH$DbGroupName - String

- HH$ExecutionTimeTo - Date
- HH$Databases - CSV Database ids
- HH$ResolvedBy - User id
- HH$Severities - CSV ActionSeverity names (INFO, NOTICE, LOW, MEDIUM, HIGH)
- HH$RuleName - Tests that generated the result
- HH$ScanNames - Scan names that generated the result
- HH$Categories - CSV VA result categories to include in the response
- HH$SqlFix - VA results containing the specified SQL fix.
- HH$pageSize - long - the maximum number of results per call, default = 100. The value -1 is for all the results

For example, to get VA results with severity HIGH from scan name “myscan”, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=varesult&HH$Severities=HIGH&HH$ScanNames=myscan

---

DBMS request parameters:

Description	Retrieves Databases information. The parameters are used to filter which databases to list
Permission	DBMS XML API
Version	4.6.0 or higher

- HH$Id - long
- HH$Name - String
- HH$VaConfigured - Boolean - search for DBMSs configured for VA
- HH$MonitorStatus - DAM monitoring status: (FULL, PARTIAL, NONE)
- HH$Agents - Comma-separated list of Agent ids
- HH$Description - String
- HH$MajorVersion - String - version of the DBMS
- HH$DbType - String - one of these database types
- HH$$RuleStats - Boolean - if set to false then the total number of rules (custom and predefined) are not calculated. The default value is true
- HH$$ScanStats- Boolean - if set to false then the scan summary are not included. The default value is true
- HH$pageSize - long - the maximum number of results per call, default = -1 (ALL)
- HH$pageNum - long - the number of the page to return, default = 0

For example, to get a list of DBMSs with VA enabled, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=dbms&HH$VaConfigured=true

To get the list of database without calculating the total number of rules:

https://127.0.0.1:8443/xmlapi.svc?service=dbms&HH$$RuleStats=false

Note:

The parameters **HH$$RuleStats** and **HH$$ScanStats** are helpful when this service call takes a fairly long time to complete. It eliminates rules and scans statistics from the database output, since these operation can take long time when the network connection to the backend database is slow.

Scans request parameters:

Description	Retrieves Scans information. The parameters are used to filter which scans to list
Permission	Scans XML API
Version	4.4.9 or higher

- HH$name - String
- HH$tags - String
- HH$Deleted - Boolean
- HH$Valid - Boolean

- HH$Enabled - Boolean
- HH$Modified - Boolean
- HH$Databases - Comma-separated list of database group names
- HH$SysIdentifiers - Comma-separated list of test SYS identifiers
- HH$pageSize - long - the maximum number of results per call, default = -1 (ALL)
- HH$pageNum - long - the number of the page to return, default = 0

For example, to get the list of active scans, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=scans&HH$Valid=true

---

Add VA Database request parameters:

Description	Adds a VA database
Permission	Add DB XML API
Version	4.4.9 or higher

NOTE: Depending on the database type, you must provide either an instance name or a port number. For advanced scenarios, it is possible to provide a full connection URL

NOTE: On MSSQL you are not allowed to provide both paramtesr: instance & port number.

- HH$Name - Custom display name for the DB
- HH$Description - String
- HH$DbType - One of these database types
- HH$Host - String
- HH$Port - Integer
- HH$Instance - String
- HH$EnableVA - Boolean - enable/disable VA section (Default: true) .
- HH$Username - String (mandatory)
- HH$Password - String
- HH$ConnUrl - Full connection URL (for advanced scenarios)
- HH$Properties - Additional properties (name=value pairs on each line)
- HH$OUs - Comma-separated list of OUs
- HH$ExcludeUsers - a list of users to exclude
- HH$validateCredentials - - Boolean - enable/disable verify connection (Default: true for add db) .

If support for OS scanning is required, use the following:

- HH$OSType - One of: DCOM, SSH
- HH$OSUsername - String
- HH$OSPassword - String
- HH$OSCert - String
- HH$OSCertPassword - String
- HH$OSPort - Integer (SSH port)

If SSH tunneling is required, use the following:

- HH$OSTunnelUsername - String
- HH$OSTunnelPassword - String
- HH$OSTunnelCertificate - String
- HH$OSTunnelCertificatePassword - String
- HH$OSTunnelHost - String
- HH$OSTunnelPort – Integer

For example, in order to add a new MSSQL database name SQL_DB, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=add-db&HH$Name=SQL_DB&HH$DbType=MSSQL&HH$Host=127.0.0.1&HH$Instance=SQL2005&HH$Username=uname&HH$Password=pword

---

Update VA Database request parameters:

Description	Updates a VA database
Permission	Update DB XML API
Version	4.4.9 or higher

- HH$Id - Integer - ID of the Database to update
- HH$Name - Custom display name for the DB
- HH$Description - String
- HH$DbType - One of these database types
- HH$Host - String
- HH$Port - Integer
- HH$Instance - String
- HH$EnableVA - Boolean - enable/disable VA section (Default: no default value) .
- HH$Username - String (Optional)
- HH$Password - String
- HH$ConnUrl - Full connection URL (for advanced scenarios)
- HH$Properties - Additional properties (name=value pairs on each line)
- HH$OUs - List of OUs (CSV)
- HH$ExcludeUsers - List of users to exclude
- HH$validateCredentials - - Boolean - enable/disable verify connection (Default: false update db) .

  More optional parameters for DAM part only:

- HH$EnableAlternative - Boolean - enable/disable alternative connection.
- HH$AltUsername - String - user name
- HH$AltPassword - String - password
- HH$AltConnectionString - String - connection string
- HH$EnableNetMonitor - Boolean- enable/disable network monitoring
- HH$NetMonitorPorts - String - monitoring ports separated comma list of ports (numeric)
- HH$EnableMemMonitor -Boolean- enable/disable memory monitoring
- HH$NetMonitorServiceNames - List (CSV) of service names - optional & applicable for oracle DB only

Examples: updating database with id #**10000000**

https://127.0.0.1:8443/xmlapi.svc?service=update-db&HH$Id=10000000&HH$EnableAlternative=true&HH$AltUsername=sa&HH$AltPassword=somepassword&HH$AltConnectionString=eatal-MOBL

https://127.0.0.1:8443/xmlapi.svc?service=update-db&HH$Id=10000000&HH$EnableNetMonitor=true&HH$NetMonitorPorts=1443,2443&HH$EnableMemMonitor=true

---

Delete VA Database request parameters:

Description	Deletes a VA database
Permission	Delete DB XML API
Version	4.4.9 or higher

- HH$Id - Integer - ID of the Database to delete

For example deleting database with id #10006000 issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=delete-db&HH$Id=10006000

Add Database to Groups request parameters:

Description	Adds a database to database groups
Permission	Add DB to Group

Version	4.4.9 or higher

- HH$Id - Integer - ID of the database to add to the groups
- HH$Groups - Comma-separated list of group names to add the database to

For example in order to add database with id #10007000 to groups: DB-Group1 and DB-Group2, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=add-db-to-group&HH$Id=10007000&HH$Groups=DB-Group1,DB-Group2

Note:

1. If a db group name contains the comma character (,) then the name must be URL Encoded twice to make sure the comma character will not be considered as a separator. You can use an online URL encoder service such as this: www.urlencoder.org.
   For example: if the db group name is "**test,more%test**" then after encoding it twice you will get "**test%252Cmore%2525test**".

Remove Database From Groups request parameters:

Description	Removes a database from database groups
Permission	Remove DB from Group
Version	4.4.9 or higher

- HH$Id - Integer - ID of the database to remove from the groups
- HH$Groups - Comma-separated list of group names to remove the database from

For example in order to remove database with id #10007000 from groups: DB-Group1 and DB-Group2, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=remove-db-from-group&HH$Id=10007000&HH$Groups=DB-Group1,DB-Group2

Note:

1. If a db group name contains the comma character (,) then the name must be URL Encoded twice to make sure the comma character will not be considered as a separator. You can use an online URL encoder service such as this: www.urlencoder.org.
   For example: if the db group name is "**test,more%test**" then after encoding it twice you will get "**test%252Cmore%2525test**".

Add VA Scan request parameters:

Description	Adds a VA scan
Permission	Add Scan XML API
Version	4.4.9 or higher

- HH$Name - String
- HH$Description - String
- HH$Schedule - Schedule in crontab format
- HH$DatabasesAdd – Add comma-separated list of DB IDs
- HH$DatabasesRemove – Remove comma-separated list of DB IDs
- HH$GroupsAdd – Add comma-separated list of DB group names
- HH$GroupsRemove – Remove comma-separated list of DB group names
- HH$DatabaseExceptionsAdd – Add comma-separated list of DB IDs to exclude from the scan
- HH$DatabaseExceptionsRemove – Remove comma-separated list of DB IDs to exclude from the scan
- HH$TagsAdd – Add comma-separated list of categories (tags)
- HH$TagsRemove – Remove comma-separated list of categories (tags)
- HH$OmmittedRulesAdd – Add comma-separated list of rule IDs to exclude from the scan
- HH$OmmittedRulesRemove – Remove comma-separated list of rule IDs to exclude from the scan
- HH$actions - Comma-separated list of actions: (Available actions: LOG, EMAIL, SYSLOG, WINLOG, FILELOG, AUTO_RESOLVE)
- HH$winLevel - Winlog level
- HH$fileLevel - File log level
- HH$sysLevel - Syslog level

- HH$resolve - Name of resolution (in case AUTO_RESOLVE action was set)
- HH$email - Email address (in case EMAIL action was set)
- HH$Severity - Comma-separated list of severities to check (HIGH, MEDIUM, LOW, NOTICE, INFO)
- HH$Enabled - Boolean
- HH$StorePassword - Boolean = Store weak passwords
- HH$RebuildPassword - Boolean - rebuild the password cache
- HH$OUs - List of comma-separated OUs
- HH$Server - The server that this scan belongs to (relevant when the server is configured as a cluster)

For example in order to add new scan on database with id #10007000 on category Audit, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=add-scan&HH$Name=newScan&HH$DatabasesAdd=10007000&HH$TagsAdd=Audit

---

Update VA Scan request parameters:

Description	Updates a VA scan
Permission	Update Scan XML API
Version	4.4.9 or higher

- HH$Id - ID of the scan to update
- HH$Name - String
- HH$Description - String
- HH$Schedule - Schedule in crontab format
- HH$DatabasesAdd – Add comma-separated list of DB IDs
- HH$DatabasesRemove – Remove comma-separated list of DB IDs
- HH$GroupsAdd – Add comma-separated list of DB group names
- HH$GroupsRemove – Remove comma-separated list of DB group names
- HH$DatabaseExceptionsAdd – Add comma-separated list of DB IDs to exclude from the scan
- HH$DatabaseExceptionsRemove – Remove comma-separated list of DB IDs to exclude from the scan
- HH$TagsAdd – Add comma-separated list of categories (tags)
- HH$TagsRemove – Remove comma-separated list of categories (tags)
- HH$OmmittedRulesAdd – Add comma-separated list of rule IDs to exclude from the scan
- HH$OmmittedRulesRemove – Remove comma-separated list of rule IDs to exclude from the scan
- HH$actions - Comma-separated list of actions: (Available actions: LOG, EMAIL, SYSLOG, WINLOG, FILELOG, AUTO_RESOLVE)
- HH$winLevel - Winlog level
- HH$fileLevel - File log level
- HH$sysLevel - Syslog level
- HH$resolve - Name of resolution (in case AUTO_RESOLVE action was set)
- HH$email - Email address (in case EMAIL action was set)
- HH$Severity - Comma-separated list of severities to check (HIGH, MEDIUM, LOW, NOTICE, INFO)
- HH$Enabled - Boolean
- HH$StorePassword - Boolean = Store weak passwords
- HH$RebuildPassword - Boolean - rebuild the password cache
- HH$OUs - Comma-separated list of OUs
- HH$Server - The server that this scan belongs to (relevant when the Server is configured as a cluster)

For example in order to update scan with id 10009000 add database with id #10006000, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=update-scan&HH$Id=10009000&HH$DatabasesAdd=10006000,10007000

Delete VA Scan request parameters:

Description	Deletes a VA scan
Permission	Delete Scan XML API
Version	4.4.9 or higher

- HH$Id - ID of the scan to delete

For example in order to delete scan with id #10008000, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=Delete-scan&HH$Id=10008000

Start VA Scan request parameters:

Description	Starts a run of a VA scan
Permission	Run Scan XML API
Version	4.4.9 or higher

- HH$Id - ID of the scan to start

For example in order to start scan with id #10008000, issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=start-scan&HH$Id=10008000

Rule Objects request parameters:

Description	Performs operations on Rule Objects
Permission	Object XML API
Version	*set*, *delete* - 4.4.9 or higher list - 4.4.9-P2 or higher object-api-restricted - 4.6.3 (52890) or higher

- object-name - Name of the rule object to modify
- action - One of **set**, **delete, list**. If **set** is used, the value-append, value-delete or value-set must be defined
  set - Creates the object if it does not exist or manipulates an existing object. When choosing the **set** action, in case rule-object does not exist , action-type should be selected.
    - object-type - String - Mandatory - Rule object type.common values: statement,ip,object
      provide one of value-set,value-append,value-delete:
    - value-set - Sets values for the rule object. If the rule object does not exist, it gets created.
        - object-description - Optional description
    - value-append - Appends the specified value/s to the rule object. Multiple values may be specified in a comma-separated list.
    - value-delete - Delete all specified value/s from the rule object. Multiple values may be specified in a comma-separated list.
    - object-api-restricted - true / false- default [false], Mandatory when creating new rule object which is API restricted (i.e manipulated only via XML API
        - db-ids- Mandatory - List of database ids (DAM) . Multiple values may be specified in a comma-separated list. for example: db-ids=10200000,10201000
        - value-set/value-append/value-delete - Mandatory - provide Group of values (i.e see definition in this section)
            - object-default-value - Alternative/default value for the case of missing mapping otherwise rule is ignored.

              - Optional only on value-set - Multiple values may be specified in a comma-separated list (for example: object-default-value=windows,linux,mac)
              - in order to reset value provide empty value (i.e object-default-value=)

Notice

Interaction with such an API restricted object involve providing mapping of keys (via parameter: db-ids) & values (via parameters: value-set/value-append/value-delete).

therefore **order** does really matter here and also **number** of db-ids should be equal to number of groups of values.
Keep your values non empty. for set operation in order to reset values you may use a simple couple of brackets (i.e ())
For example: The parameter line: db-ids=12600000,12601000&value-set=(WIN,LINUX)(MAC1,MAC2)
is actually mapping 12600000 => WIN,LINUX and 12601000 => MAC1,MAC2

Group of values

collection of multiple elements,each one is surrounded with brackets ()
whereas each element is constructed from Multiple non empty values specified as comma-separated list.
Format: (V1,V2,...Vn)(W1,W2,...Wm)
Examples:

a. (WIN1)(WIN2)

b. (WIN1,WIN2,WIN3)(WIN1)(WIN2)

Examples for rule object

Note: only one value action (append, set or delete) is allowed per a single request

List the rule object:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=list

Find a rule object:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&object-name=nameOfRuleObject

Add a new rule object with a value list (‘1.1.1.1,1.1.2.2), type (‘ip’) and a description (‘Black List IPs’), issue the request:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-name=blacklist_ip&value-set=1.1.1.1,1.1.2.2&object-type=ip&object-description=Black%20List%20IPs

Delete a specific value (‘1.1.2.2’) from a rule object:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-name=blacklist_ip&value-delete=1.1.2.2

Delete completely a rule object use:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=delete&object-name=blacklist_ip

Examples for API restricted

Create a simple API restricted rule object.

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-type=STATEMENT&object-name=name&object-api-restricted=true&db-ids=12600000&value-set=(val1)

Create an API restricted rule object with a default value, which behaves like map between 3 databases to single value each:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-type=STATEMENT&object-name=name&object-api-restricted=true&db-ids=12600000,12601000,12602000&value-set=(val1)(val2)(val3)&object-default-value=d1

Delete values from API restricted rule object:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-name=name&db-ids=12600000,12601000,12602000&value-delete=(val1)(val2)(val3)

Append values for API restricted rule object:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-name=name&db-ids=12600000,12601000,12602000&value-append=(val0,val1)(val0,val2)(val0,val3)

Set values for db-ids & Reset values for specific db-id:

https://127.0.0.1:8443/xmlapi.svc?service=ruleobject&action=set&object-name=name&object-type=STATEMENT&db-ids=12600000,12601000,12602000&value-set=(val0,val1)(val0,val2)()

Resolve Alerts request parameters:

Description	Resolves Alerts
Permission	Alert XML API
Version	4.4.9 or higher

QUERY parameters are used to select which alerts will be resolved:

- HH$ExecutionTimeFrom - Date (format: dd MMM yyyy HH:mm:ss)
- HH$ResolveReason - String
- HH$Resolves - CSV Resolve ids for the resolution field
- HH$Id - long
- HH$Agents - CSV Agent ids
- HH$TagName - String
- HH$DbGroupName - String

- HH$ExecutionTimeTo - Date
- HH$Databases - CSV Database ids
- HH$Operation - String
- HH$OsUser - String
- HH$ResolvedBy - User id
- HH$Severities - CSV ActionSeverity names (INFO, NOTICE, LOW, MEDIUM, HIGH)
- HH$SourceHost - String
- HH$SourceIP - String
- HH$Rules - CSV Rule ids
- HH$ResolveNames - CSV Resolve name string
- HH$RuleName - String
- HH$QuarantineId - String
- HH$ReleaseTimeAfter - Date
- HH$ReleaseTimeBefore - Date
- HH$ExecUser - String
- HH$DatabaseId - database Long
- HH$ExecProgram - String
- HH$Clientid - String
- HH$Module - String
- HH$ModifyDateFrom - Date
- HH$ModifyDateTo - Date
- HH$Sid – Integer
- HH$TimeBackPeriod – Long

SET parameters are used to pass the values to be set for the alerts subset selected by the above parameters:

- HH$$resolveReason - String
- HH$$resolveName - Resolve name string (Resolved, Unresolved, False Alarm, …)

For example, to resolve all the unresolved alerts received during the past five minutes, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=resolve-alerts&HH$ResolveNames=Unresolved&HH$TimeBackPeriod=300000&HH$$resolveName=Resolved

---

Resolve Results request parameters:

Description	Resolves VA Results
Permission	VA Result XML API
Version	4.4.9 or higher

QUERY parameters are used to select which results will be resolved:

- HH$ExecutionTimeFrom - Date (format: dd MMM yyyy HH:mm:ss)
- HH$ResolveReason - String
- HH$Resolves - CSV Resolve ids
- HH$Id - long
- HH$TagName - String
- HH$DbGroupName - String
- HH$ExecutionTimeTo - Date
- HH$Databases - CSV Database ids
- HH$ResolvedBy - User id
- HH$Severities - CSV ActionSeverity names (INFO, NOTICE, LOW, MEDIUM, HIGH)
- HH$RuleName - Tests that generated the result
- HH$ScanNames - Scan names that generated the result
- HH$Categories - CSV VA result categories to include in the response
- HH$SqlFix - VA results containing the specified SQL fix.

SET parameters are used to pass the values to be set for the results subset selected by the above parameters:

- HH$$resolveReason - String
- HH$$resolveName - Resolve name string (Resolved, Unresolved, False Alarm, …)

For example, to resolve all the results received during the past 24 hours, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=resolve-results&HH$TimeBackPeriod=86400000&HH$$resolveName=Resolved&HH$$resolveReason=DONE

Sensor Restart request parameters:

Description	Restarts a sensor
Permission	Sensor Restart XML API
Version	4.4.9-P1 or higher

- HH$Id – csv list of longs

For example, to restart sensors 10100000 and 1020000, issue the following request:

https://127.0.0.1:8443/xmlapi.svc?service=sensor-restart&HH$Id=10100000,10200000

Rules request parameters:

Description	Manage custom rules (add/update/delete) or list rules
Permission	Object XML API
Version	list - 4.4.9 or higher add, update, delete - 4.6.3 or higher

Parameters:

- HH$Id – Id of the selected rule. Valid for the operations; *list*, *update*, *delete*
- HH$Name – select a rule by name or give the rule a name during *add* action
- HH$NewName – the new rule name when updating the rule. Valid only for the *update* operation. Optional but cannot be empty.
- HH$RuleType – filter rules by type: custom/predefined, default=custom (optional param). This parameter only applies when listing rule(s).
- HH$Operation - the operation to perform: *list, add, delete, update. Default=list. All rule types (custom, predefined) can be listed, but only custom rules can be managed.*
- HH$Expression – the rule expression that is sent to the sensor
- HH$Exceptions – list of exception expressions separated by a logical operator that can be either 'OR' (OR operator) or 'AND' (AND operator). An example of three expressions:
  object = '888-888' OR action CONTAINS '8888' OR cmdtype NOT IN ('update') AND client_ip = 1.2.3.4 AND client_host_name LIKE 'host'
- HH$InstallOnDBs – Install this rule on a list of databases by Id
- HH$InstallOnDBGroups – Install this rule on a list of database groups by name
- HH$InstallOnDBsExclude – Exclude installation of this rule on a list of databases by Id
- HH$Tags – a list of tags
- HH$Comment – the comment field
- HH$Enable – enable rule: true/false, default = true
- HH$RuleAction – the action to perform when a rule is matched. The format of this parameter is JSON-like as follows:

```
{
  allow_rule : {
     global_allow,
  }
|
  send_alert :
  {
     severity: INFO|NOTICE|LOW|MEDIUM|HIGH[1],
    to_archive |
    console : {
      snmp_trap, terminate_session:{quarantine_user_for:10}
    },
    syslog: TRACE|DEBUG|INFO|WARN|ERROR|FATAL,
    event_log:TRACE|DEBUG|INFO|WARN|ERROR|FATAL,
    log_file: TRACE|DEBUG|INFO|WARN|ERROR|FATAL,
    email:
```

```
    {
      severity: INFO|NOTICE|LOW|MEDIUM|HIGH,
      addresses: “list_of_email_addresses”[2]
    },
    stop_processing_rules
  }
}
```

- HH$EditRoles - the roles that are granted edit permission for this rule
- HH$AdvancedOptions - advanced rule options, currently contain just the Monitoring Source. The format is JSON-like, for example: {monitoring_source:AUTO}
  The valid values for monitoring_source are: AUTO, ALL, MEMORY, NETWORK
- HH$AdvActionScript - action script
- HH$AdvSensitiveRegex - Mask sensitive data regular expression
- HH$AdvLimitAlertsSec - Limit alerts per second, value can be one of 1,5,10,100,1000 or -1 which mean Unlimited. Default is 1000.
- HH$AdvLimitAlertsSession - Limit alerts per session, value can be one of 1,5,10,100,1000 or -1 which mean Unlimited. Default is -1.
- HH$AdvMinRowsForAlert - Minimum rows for alert
- HH$AdvApplyActionsRuleTrigger - A list of 2 numbers: N,S that means: apply actions when rule triggers N times in S seconds.
- HH$AdvAutoResolve - Automatically resolve to one of the values: FalseAlarm, Resolved
- HH$AdvIgnoreSigned - true or false, default is false.

[1] Action Severities (INFO,NOTICE,LOW,MEDIUM,HIGH) and Log Levels (TRACE,DEBUG,INFO,WARN,ERROR,FATAL) **should be in upper case**.
[2] **list_of_email_addresses** is a list of email addresses separated by a semicolon. The entire list should enclosed by double quotes.

Examples

list	List all custom rules: **https://127.0.0.1:8443/xmlapi.svc?service=rules&HH$RuleType=custom** List all predefined rules: **https://127.0.0.1:8443/xmlapi.svc?service=rules&HH$RuleType=predefined** Get specific rule with id=210: **https://127.0.0.1:8443/xmlapi.svc?service=rules&HH$Id=210** Get specific rule with name=MyRule: **https://127.0.0.1:8443/xmlapi.svc?service=rules&HH$Name=MyRule**

add	Add a new rule with: • Name: big-rule • Expression: statement contains ‘credit’ • Rule Action: {send_alert: {severity: MEDIUM, console: {}}} • Exception list: cmdtype NOT IN ('update') AND client_ip = 1.2.3.4 AND client_host_name LIKE 'host' • Install On Db Groups: All DBMSs • Install On DBs: 10500000 • Exclude DBs: 10300000,10301000 • Tags: t1,t3,t5 • Comment: this is just a comment • Roles: Read_Only • Advanced Options: {monitoring_source:AUTO} • Action script: select * from dual • Mask sensitive data regex: (\d\d\d\d\d)+ • Limit alerts per second: 100 • Limit alerts per session: Unlimited • Minimum rows for alert: 666 • Apply actions when rule triggers 9 times in 2 seconds. • Auto resolve: False Alarm • Ignore Signed: true For sake of simplicity, we'll list the parameters and their values in the list form. Parameters needs to be urlencoded. HH$Operation=add HH$Name=big-rule HH$Expression=statement%20contains%20%27credit%27 HH$RuleAction=%7Bsend_alert%3A%20%7Bseverity%3A%20MEDIUM%2C%20console%3A%20%7B%7D%7D%7D HH$Exceptions=cmdtype%20NOT%20IN%20(%27update%27)%20%24AND%24%20client_ip%20%3D%201.2.3.4%20AND%20client_ HH$InstallOnDBGroups=All%20DBMSs HH$InstallOnDBs=10500000 HH$InstallOnDBsExclude=10300000,10301000 HH$Tags=t1,t3,t5 HH$Comment=this%20is%20just%20a%20comment HH$EditRoles=Read_Only HH$AdvancedOptions=%7Bmonitoring_source%3AAUTO%7D HH$AdvActionScript=select%20%2A%20from%20dual HH$AdvSensitiveRegex=%28%5Cd%5Cd%5Cd%5Cd%5Cd%29%2B HH$AdvLimitAlertsSec=10 HH$AdvLimitAlertsSession=-1 HH$AdvMinRowsForAlert=666 HH$AdvApplyActionsRuleTrigger=9,2 HH$AdvAutoResolve=FalseAlarm HH$AdvIgnoreSigned=true
delete	Delete the rule with the name: big-rule https://localhost:8443/xmlapi.svc?service=rules&HH$Operation=delete&HH$Name=big-rule

update	Update the rule with the name: big-rule and set new values. Note that the Exclude DBs value is erased • New name: bigger-rule • Rule Action: {send_alert: {severity: INFO, to_archive}} • Exclude DBs: (none) • Tags: t8 HH$Operation=update HH$Name=big-rule HH$RuleAction=%20%7Bsend_alert%3A%20%7Bseverity%3A%20INFO%2C%20to_archive%7D%7D HH$InstallOnDBsExclude= HH$Tags=t8

Notes:

a. You can use the **HH$Id** parameter and specify the Rule Id. However, please note that rules use revisions and the Id might change with a new revision. So the best option to identify a rule is by it's name.
b. The output of the *add* or *update* operations is the rule that was added or updated.
c. When updating the rule and the user wishes to reset the value of a specific parameter, then the value of that parameter should be left empty.

Database Groups request parameters:

Description	Performs operations on a database group (add, remove, update, list)
Permission	DBMS XML API
Version	4.4.9-P2 or higher

- HH$Name - The name of the database group
- HH$Action - The action to perform on the Database Group, one of: *list, add, remove, update*
- HH$Description - The description of the database group
- HH$NewName - The new name of the database group when updating it

For example, to list all the database groups, use:

https://127.0.0.1:8443/xmlapi.svc?service=dbgroups

To list all the databases that are attached to the db group named: **my_dbgroup**, use:

https://127.0.0.1:8443/xmlapi.svc?service=dbgroups&HH$Name=my_dbgroup

To add a new database group named: **db_group1**, use:

https://127.0.0.1:8443/xmlapi.svc?service=dbgroups&HH$Action=add&HH$Name=db_group1&HH$Description=This%20Group%20is%20mine

To delete the database group name: **db_group3**, use:

https://127.0.0.1:8443/xmlapi.svc?service=dbgroups&HH$Action=delete&HH$Name=db_group3

To update the database group name: **db_group9** and change the name to a new value: **db_group101**, use:

https://127.0.0.1:8443/xmlapi.svc?service=dbgroups&HH$Action=update&HH$Name=db_group3&HH$NewName=db_group101

Notes:

1. Database Group names are unique.
2. If the **HH$Action** parameter is not specified, the list action is used as default.
3. If the **HH$Name** is missing when listing database groups then all the db groups are listed

---