Qualys Cloud Platform (VM, PC) v10.x

API Release Notes

Version 10.2

June 9, 2020

This new version of the Qualys Cloud Platform (VM, PC) includes improvements to the Qualys API. You'll find all the details in our user guides, available at the time of release. Just log in to your Qualys account and go to Help > Resources.

What's New

Qualys API Server URL

The Qualys API URL you should use for API requests depends on the Qualys platform where your account is located.

Click here to identify your Qualys platform and get the API URL

This documentation uses the API server URL for Qualys US Platform 1 (https://qualysapi.qualys.com) in sample API requests. If you're on another platform, please replace this URL with the appropriate server URL for your account.

API Support for vCenter - ESXi Mapping

APIs affected	/api/2.0/fo/auth/vcenter/vcenter_mapping/
New or Updated API	New
DTD or XSD changes	New

We have added API support for vCenter - ESXi mapping. Now you'll be able to list, import and purge vCenter - ESXi mapping data.

A new API endpoint **/api/2.0/fo/auth/vcenter/vcenter_mapping/** is added.

A new DTD file **vcenter_esxi_map_list_output.dtd** is added to list mapping in xml format.

Input Parameters

The following table shows input parameters used for listing, importing and purging vCenter - ESXi mapping data.

Parameter	Description
echo_request={0|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
action={action}	(Required) One action (list, import or purge) required for the request.
id_min={value}	(Optional to list) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.
id_max={value}	(Optional to list) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.
output_format={XML|**CSV**}	(Optional to list) Specifies the format of the mapping list output. When not specified, the output format is CSV. A valid value is XML or CSV.
truncation_limit={value}	(Optional to list) Specifies the maximum number records listed per request.
vcenter_ip={value}	(Optional to list) Specifies the IP address of the vCenter.
esxi_ip={value}	(Optional to list) Specifies the IP address of the ESXi server.
network_id={1|**0**}	(Optional) By default, the parameter is set to 0. If this parameter is not provided, it will be Global Default Network.

Parameter	Description
csv_data={value}	(Required to import and purge) The CSV data file containing the vCenter - ESXi mapping records that you want to add/purge. This parameter or xml_data must be specified. The parameters csv_data and xml_data cannot be specified in the same request.
xml_data={value}	(Required to import and purge) The XML data file containing the vCenter - ESXi mapping records that you want to add/purge. This parameter or csv_data must be specified. The parameters csv_data and xml_data cannot be specified in the same request.

Sample - List vCenter - ESXi Mapping in CSV Format

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?ac
tion=list"
```

OR

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?ac
tion=list&output_format=csv"
```

CSV output:

```
----BEGIN_RESPONSE_BODY_CSV
vCenter IP,ESXi IP,Mapping Data Source
"11.11.11.11","30.30.30.23","File"
"10.10.10.10","10.10.10.12","File"
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
"Status Message"
"Finished"
----END_RESPONSE_FOOTER_CSV
```

Sample - List vCenter - ESXi Mapping in XML Format

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl'
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/?ac
tion=list&output_format=xml"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VCENTER_ESXI_MAP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/vce
nter_esxi_map_list_output.dtd">
<VCENTER_ESXI_MAP_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2020-05-22T16:49:40Z</DATETIME>
        <VCENTER_ESXI_MAP_LIST>
            <VCENTER_ESXI_MAP>
                <VCENTER_IP>11.11.11.11</VCENTER_IP>
                <ESXI_IP>30.30.30.23</ESXI_IP>
                <MAPPING_DATA_SOURCE>File</MAPPING_DATA_SOURCE>
            </VCENTER_ESXI_MAP>
            <VCENTER_ESXI_MAP>
                <VCENTER_IP>10.10.10.10</VCENTER_IP>
                <ESXI_IP>10.10.10.12</ESXI_IP>
                <MAPPING_DATA_SOURCE>File</MAPPING_DATA_SOURCE>
            </VCENTER_ESXI_MAP>
        </VCENTER_ESXI_MAP_LIST>
    </RESPONSE>
</VCENTER_ESXI_MAP_LIST_OUTPUT>
```

New DTD:

DTD: <platform API server>/api/2.0/fo/auth/vcenter/vcenter_mapping/vcenter_esxi_map_list_output.dtd

```
<!-- QUALYS VCENTER_ESXI_MAP_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT VCENTER_ESXI_MAP_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>
<!ELEMENT RESPONSE (DATETIME, VCENTER_ESXI_MAP_LIST?, WARNING?)>
<!ELEMENT VCENTER_ESXI_MAP_LIST (VCENTER_ESXI_MAP+)>
<!ELEMENT VCENTER_ESXI_MAP (VCENTER_IP, ESXI_IP, MAPPING_DATA_SOURCE?)>
<!ELEMENT VCENTER_IP (#PCDATA)>
<!ELEMENT ESXI_IP (#PCDATA)>
<!ELEMENT MAPPING_DATA_SOURCE (#PCDATA)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!-- EOF -->
```

Sample - Import vCenter - ESXi Mapping

You'll be able to import vCenter - ESXi mapping in the CSV and XML format. You can provide CSV or XML data in API call or in the file.

CSV Data in API Call

Following is the sample API request when you want to import mapping using CSV data in API call.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary
"action=import&csv_data=vCenter IP,ESXi
IP%0A10.10.10.10,10.10.10.11%0A10.10.10.10,10.10.10.12"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

XML Data in API Call

Following is the sample API request when you want to import mapping using XML data in API call.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary
"action=import&xml_data=<VCENTER_ESXI_MAP_LIST><VCENTER_ESXI_MAP><VCENTER
_IP>11.11.11.11</VCENTER_IP><ESXI_IP>22.22.22.22</ESXI_IP></VCENTER_ESXI_
MAP><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.12</VCENTER_IP><ESXI_IP>22.22.
22.23</ESXI_IP></VCENTER_ESXI_MAP></VCENTER_ESXI_MAP_LIST>"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

CSV Data in File

Following is the sample API request when you want to import the mapping using a file containing CSV data. In the sample request, **add.csv** is a CSV data file.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@add.csv"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

Sample content of **add.csv** file:

```
action=import&csv_data=
vCenter IP,ESXi IP
10.10.10.10,20.20.20.20
10.10.10.10,20.20.20.21
10.10.10.10,20.20.20.22
11.11.11.11,30.30.30.23
12.12.12.12,40.40.40.24
```

XML Data in File

Following is the sample API request when you want to import the mapping using a file containing XML data. In the sample request, **add.xml** is a XML data file.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@add.xml"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

Sample content of **add.xml** file:

```
action=import&xml_data=
<?xml version="1.0" encoding="UTF-8" ?>
<VCENTER_ESXI_MAP_LIST>
    <VCENTER_ESXI_MAP>
        <VCENTER_IP>10.10.10.10</VCENTER_IP>
```

```
            <ESXI_IP>20.20.20.21</ESXI_IP>
        </VCENTER_ESXI_MAP>
        <VCENTER_ESXI_MAP>
            <VCENTER_IP>10.10.10.10</VCENTER_IP>
            <ESXI_IP>20.20.20.22</ESXI_IP>
        </VCENTER_ESXI_MAP>
    </VCENTER_ESXI_MAP_LIST>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-07T10:57:23Z</DATETIME>
    <TEXT>Successfully imported 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Sample - Purge vCenter - ESXi Mapping

You'll be able to purge vCenter - ESXi mapping in the CSV and XML format. You can provide CSV or XML data in API call or in the file.

CSV Data in API Call

Following is the sample API request when you want to purge mapping using CSV data in API call.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary
"action=purge&csv_data=vCenter IP,ESXi
IP%0A10.10.10.10,10.10.10.11%0A10.10.10.10,10.10.10.12"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

XML Data in API Call

Following is the sample API request when you want to purge mapping using XML data in API call.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-With: curl' --data-binary
"action=purge&xml_data=<VCENTER_ESXI_MAP_LIST><VCENTER_ESXI_MAP><VCENTER_
IP>11.11.11.11</VCENTER_IP><ESXI_IP>22.22.22.22</ESXI_IP></VCENTER_ESXI_M
AP><VCENTER_ESXI_MAP><VCENTER_IP>11.11.11.12</VCENTER_IP><ESXI_IP>22.22.2
2.23</ESXI_IP></VCENTER_ESXI_MAP></VCENTER_ESXI_MAP_LIST>"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

CSV Data in File

Following is the sample API request when you want to purge the mapping using a file containing CSV data. In the sample request, **purge.csv** is a CSV data file.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@purge.csv"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

Sample content of **purge.csv** file:

```
action=purge&csv_data=
vCenter IP,ESXi IP
10.10.10.10,20.20.20.20
10.10.10.10,20.20.20.21
10.10.10.10,20.20.20.22
11.11.11.11,30.30.30.23
12.12.12.12,40.40.40.24
```

XML Data in File

Following is the sample API request when you want to purge the mapping using a file containing XML data. In the sample request, **purge.xml** is a XML data file.

API request:

```
curl -u "USERNAME:PASSWORD" -H 'X-Requested-with: curl' --data-binary
"@purge.xml"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vcenter/vcenter_mapping/"
```

Sample content of **purge.xml** file:

```
action=purge&xml_data=
<?xml version="1.0" encoding="UTF-8" ?>
<VCENTER_ESXI_MAP_LIST>
    <VCENTER_ESXI_MAP>
        <VCENTER_IP>10.10.10.10</VCENTER_IP>
        <ESXI_IP>20.20.20.21</ESXI_IP>
    </VCENTER_ESXI_MAP>
    <VCENTER_ESXI_MAP>
        <VCENTER_IP>10.10.10.10</VCENTER_IP>
        <ESXI_IP>20.20.20.22</ESXI_IP>
    </VCENTER_ESXI_MAP>
</VCENTER_ESXI_MAP_LIST>
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-05-07T10:57:23Z</DATETIME>
    <TEXT>Successfully purged 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Add Target Type to Unix Authentication Record

APIs affected	/api/2.0/fo/auth/unix/
New or Updated API	Updated
DTD or XSD changes	Yes

You can now provide a target type while creating or updating the Unix (SSH2) authentication record. We have added a new TARGET_TYPE parameter that allows you to define the type of target for a Unix auth record.

Parameter	Description
target_type={value}	(Optional) Specify the target type. You can choose from the following values: - A10 - HP_COMWARE - CISCO_ASA_WITH_FIREPOWE - auto (default)

Sample - Create Unix auth record with target type set to HP_COMWARE

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&title=u
x-target-
type&username=root&ips=10.11.42.114&login_type=basic&password=root&target
_type=HP_COMWARE
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2020-05-26T21:17:17Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>149016</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Sample - Update Unix auth record with target type CISCO_ASA_WITH_FIREPOWE

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=update&ids=1490
16&target_type=CISCO_ASA_WITH_FIREPOWE
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2020-05-26T21:34:18Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Updated</TEXT>
                <ID_SET>
                    <ID>149016</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Sample - List Unix auth record with to view updated target type

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWORD"
https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=list&ids=149016
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_UNIX_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/auth_unix_list_output.
dtd">
<AUTH_UNIX_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2020-05-26T21:35:23Z</DATETIME>
        <AUTH_UNIX_LIST>
            <AUTH_UNIX>
                <ID>149016</ID>
                <TITLE>
                    <![CDATA[ux-target-type]]>
                </TITLE>
                <USERNAME>
```

```
                    <![CDATA[root]]>
                </USERNAME>
                <SKIP_PASSWORD>0</SKIP_PASSWORD>
                <CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
                <TARGET_TYPE>
                    <![CDATA[Cisco Adaptive Security Appliance with
FirePower]]>
                </TARGET_TYPE>
                <IP_SET>
                    <IP>10.11.42.114</IP>
                </IP_SET>
                <NETWORK_ID>0</NETWORK_ID>
                <CREATED>
                    <DATETIME>2020-05-26T21:17:17Z</DATETIME>
                    <BY>username</BY>
                </CREATED>
                <LAST_MODIFIED>
                    <DATETIME>2020-05-26T21:34:18Z</DATETIME>
                </LAST_MODIFIED>
            </AUTH_UNIX>
        </AUTH_UNIX_LIST>
    </RESPONSE>
</AUTH_UNIX_LIST_OUTPUT>
```

DTD update:

DTD: <platform API server>/api/2.0/fo/auth/unix/auth_unix_list_output.dtd

```
<!-- QUALYS AUTH_UNIX_LIST_OUTPUT DTD -->
<!-- $Revision$ -->
<!ELEMENT AUTH_UNIX_LIST_OUTPUT (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>

...
<!ELEMENT RESPONSE (DATETIME, (AUTH_UNIX_LIST|ID_SET)?, WARNING_LIST?,
GLOSSARY?)>
<!ELEMENT AUTH_UNIX_LIST (AUTH_UNIX+)>

<!ELEMENT AUTH_UNIX (ID, TITLE, USERNAME, SKIP_PASSWORD?,
CLEARTEXT_PASSWORD?, TARGET_TYPE?, (ROOT_TOOL?|ROOT_TOOL_INFO_LIST?),
((RSA_PRIVATE_KEY?, DSA_PRIVATE_KEY?)|PRIVATE_KEY_CERTIFICATE_LIST?),
PORT?, IP_SET, LOGIN_TYPE?, DIGITAL_VAULT?, NETWORK_ID?, CREATED,
LAST_MODIFIED, COMMENTS?, USE_AGENTLESS_TRACKING?,
AGENTLESS_TRACKING_PATH?, QUALYS_SHELL?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT SKIP_PASSWORD (#PCDATA)>
<!ELEMENT CLEARTEXT_PASSWORD (#PCDATA)>
<!ELEMENT TARGET_TYPE (#PCDATA)>
<!ELEMENT ROOT_TOOL (#PCDATA)>
<!ELEMENT ROOT_TOOL_INFO_LIST (ROOT_TOOL_INFO)*>
<!ELEMENT RSA_PRIVATE_KEY EMPTY>
<!ELEMENT DSA_PRIVATE_KEY EMPTY>
<!ELEMENT PRIVATE_KEY_CERTIFICATE_LIST (PRIVATE_KEY_CERTIFICATE)*>
<!ELEMENT PORT (#PCDATA)>

...

<!ELEMENT GLOSSARY (USER_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>
<!-- EOF -->
```

Schedule EC2 Scans using API

APIs affected	/api/2.0/fo/schedule/scan/compliance/
New or Updated API	Updated
DTD or XSD changes	No

We have now added the capability to schedule or update scheduled EC2 scans using the compliance API.

Sample - Create EC2 scan

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/compliance?
action=create&scan_title=API_Schedule_EC2_PC&target_from=tags&tag_set_by=
name&tag_include_selector=any&tag_set_include=Auth&connector_name=AWS+Con
nector&ec2_endpoint=us-east-
1&active=0&occurrence=daily&start_date=05/21/2020&start_hour=20&start_min
ute=30&time_zone_code=IN&option_title=Initial+PC+Options&frequency_days=3
64&end_after=1&observe_dst=no&iscanner_name=EC2_Scanner"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2020-06-07T22:09:26Z</DATETIME>
    <TEXT>New compliance scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>279256</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Cloud Perimeter Scan API to Create/Update Scan Job with No Targets

APIs affected	/api/2.0/fo/scan/cloud/perimeter/job/
New or Updated API	No
DTD or XSD changes	No

We will now allow you to create/update a cloud perimeter scan job through Cloud Perimeter Scan API even if no scan targets are resolved from the provided details.

At the time of scan, if no scan targets are resolved from the provided details, the scan will not be launched, and we add the error in the Activity log and Run history of the schedule scan job.

In addition to this, we have added new validations to validate the values specified in input parameters: vpc_id and tag_set_by={id|name}. The validations are:

- For vpc_id, we will check if the specified vpc_id exists for the selected connector.

- For tag_set_by={id|name}, we will check if the tag ids or tag names are valid.

If all the validations are satisfied, we will allow you to create/update cloud perimeter scan job even with no targets.

We are already validating these input parameters:

- The specified connector_uuid or connector_name exists for your Qualys subscription. If not, then API request returns an error message “Invalid connector_name or connector_uuid provided”.

-The specified region_code exists for the selected connector. If not, then API request returns an error message “Invalid region_code provided”.

- The specified vpc_id is of correct format. That is, it starts with 'vpc-.*'. If not, then API request returns an error message “Invalid vpc id specified”.

Updates to Option Profile Info DTD

We added new elements (in bold) to the Option Profile Info DTD (option_profile_info.dtd) for future use.

DTD: <platform>/api/2.0/fo/subscription/option_profile/option_profile_info.dtd

```
<!ELEMENT OPTION_PROFILES (OPTION_PROFILE)*>
<!ELEMENT OPTION_PROFILE (BASIC_INFO, SCAN, MAP?, ADDITIONAL)>
<!ELEMENT BASIC_INFO (ID, GROUP_NAME, GROUP_TYPE, USER_ID, UNIT_ID,
SUBSCRIPTION_ID, IS_DEFAULT?, IS_GLOBAL?, IS_OFFLINE_SYNCABLE?,
UPDATE_DATE?)>
<!ELEMENT ID (#PCDATA)>
...
<!ELEMENT PERFORMANCE (PARALLEL_SCALING?, OVERALL_PERFORMANCE,
HOSTS_TO_SCAN, PROCESSES_TO_RUN, PACKET_DELAY,
PORT_SCANNING_AND_HOST_DISCOVERY, EXTERNAL_SCANNERS_TO_USE?,
HOST_CGI_CHECKS?, MAX_CGI_CHECKS?, MAX_TARGETS_PER_SLICE?,
MAX_NUMBER_OF_TARGETS?, CONF_SCAN_LIMITED_CONNECTIVITY?,
SKIP_PRE_SCANNING?)>
<!ELEMENT PARALLEL_SCALING (#PCDATA)>
<!ELEMENT OVERALL_PERFORMANCE (#PCDATA)>
<!ELEMENT HOSTS_TO_SCAN (EXTERNAL_SCANNERS, SCANNER_APPLIANCES)>
<!ELEMENT EXTERNAL_SCANNERS (#PCDATA)>
<!ELEMENT SCANNER_APPLIANCES (#PCDATA)>
<!ELEMENT PROCESSES_TO_RUN (TOTAL_PROCESSES, HTTP_PROCESSES)>
<!ELEMENT TOTAL_PROCESSES (#PCDATA)>
<!ELEMENT HTTP_PROCESSES (#PCDATA)>
<!ELEMENT PACKET_DELAY (#PCDATA)>
<!ELEMENT PORT_SCANNING_AND_HOST_DISCOVERY (#PCDATA)>
<!ELEMENT EXTERNAL_SCANNERS_TO_USE (#PCDATA)>
<!ELEMENT HOST_CGI_CHECKS (#PCDATA)>
<!ELEMENT MAX_CGI_CHECKS (#PCDATA)>
<!ELEMENT MAX_TARGETS_PER_SLICE (#PCDATA)>
<!ELEMENT MAX_NUMBER_OF_TARGETS (#PCDATA)>
<!ELEMENT CONF_SCAN_LIMITED_CONNECTIVITY (#PCDATA)>
<!ELEMENT SKIP_PRE_SCANNING (#PCDATA)>
<!ELEMENT LOAD_BALANCER_DETECTION (#PCDATA)>
...
<!ELEMENT IGNORE_ALL_TCP_RST (#PCDATA)>
<!ELEMENT IGNORE_FIREWALL_GENERATED_TCP_SYN_ACK (#PCDATA)>
<!ELEMENT NOT_SEND_TCP_ACK_OR_SYN_ACK_DURING_HOST_DISCOVERY (#PCDATA)>
```