PCI Merchant API

User Guide

Version 1.2

September 13, 2021

Table of Contents

Get Started

PCI Compliance API

Qualys PCI provides businesses, merchants and online service providers with the easiest, most cost effective and highly automated way to achieve compliance with the Payment Card Industry Data Security Standard (PCI DSS).

Modules supported

PCI Compliance

Authentication

Authentication to your Qualys PCI account with valid credentials is required for making PCI API requests to the Qualys API servers. Learn more about authentication to your Qualys account

Get API Notifications

We recommend you join our Community and subscribe to our API Notifications RSS Feeds for announcements and discussions.

https://community.qualys.com/community/developer/notifications-api

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated Cloud Apps deliver businesses critical security intelligence continuously, enabling them to automate the full spectrum of auditing, compliance and protection for IT systems and web applications on premises, on endpoints and elastic clouds. For more information, please visit www.qualys.com

API Versioning

PCI API supports API versioning. We use version indicator (currently V1) to distinguish major version of our APIs. As per our current policy, when a new major version is released, we will continue to support the previous version. Currently, we have no policy in place to deprecate the previous version of an API.

Notification for upcoming major API updates

We will notify you through blogs and API Release Notes of upcoming new major API releases. In the new major releases, the APIs with version change will have new version-indicator in API endpoint URLs. Minor version updates will not result in a new version indicator in the API URLs and do not require any change in the implementation on your side..

Making API Calls

Curl samples in our API doc

We use curl in our API documentation to show an example how to form REST API calls, and it is not meant to be an actual production example of implementation.

Making Requests with a JSON Payload

PCI API supports GET, POST, PUT, DELETE methods. The API guide mentions the methods supported for the API endpoints. When you create API requests, use the methods supported by the APIs.

The JSON payloads can be compared to a scripting language that allows user to make multiple actions within one single API request, like adding a parameter to an object and updating another parameter.

Note that the parameter names for an API are case-sensitive. You must follow the case of parameter names as specified in the API guide. For example: in the "Launch a new Scan" API request if you specify the "ScanType" parameter as "scantype", then the API will treat the parameter as invalid and its value is ignored.

Sample - Launch a new scan

Let us launch an On Demand scan.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1"
-d @postdatafile.json
"https://pci-api.qualys.com/pci/scan/launch"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "title":"My Scan",
    "bandwidth":"MEDIUM",
```

```
    "scanType": "DNS",
    "splitTarget":"false",
    "targetDns":"www.bbb.com,www.ccc.com,abc.com,www.aaa.com",
    "targetAll":false,
    "launchType":"ONDEMAND"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "id": 9097,
        "status": "Launched scan successfully"
    }
}
```

JSON Output Pagination / Truncation

We support pagination for the JSON output of a search API request. The default page size varies and is based on the page size value configured for the endpoint. You can use the offset and limit parameters to set custom page size.

Qualys user account

Authentication with valid Qualys PCI user account credentials is required for making PCI API requests to the Qualys API servers. These servers are hosted at the Qualys platform, also referred to as the Security Operations Center (SOC), where your account is located. If you need assistance with obtaining a Qualys account, please contact your Qualys account representative.

The application must authenticate using Qualys PCI account credentials (user name and password) as part of the HTTP request. The credentials are transmitted using the "Basic Authentication Scheme" over HTTPS.

For information, see the "Basic Authentication Scheme" section of RFC #2617:

http://www.faqs.org/rfcs/rfc2617.html

The exact method of implementing authentication will vary according to which programming language is used.

The allowed methods, POST and/or GET, for each API request are documented with each API call in this user guide.

Sample request - basic authentication

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1"
"https://pci-api.qualys.com/pci/scan/2185043/details"
```

Scans

List PCI Scans

/pci/scan/list

[GET]

View list of all the scans in your scope.

Permissions required - You must have an active PCI merchant account.

Input Parameters

These elements are optional and act as filters.

Parameter	Description
sortBy={Scandate|title}	(Optional) Sort the scan list by certain data. One of: "title", "Scandate". If you do not specify any value, then scans will be sorted by scan date.
sortOrder={asc|desc}	(Optional) The sort order, used when the request includes the sortBy parameter. One of: asc (for ascending order) or desc (for descending order). If you do not specify any value, then the scans will be sorted in descending order.
limit={value}	(Optional) The maximum number of records processed for the request, starting at the record number specified by the offset parameter. Limit value must always be greater than “0”. If you specify a value 0 for the parameter, the request will fail. When not specified, default limit is set to 100 scan records. You can specify a value less than or greater than the default.

offset={value}	(Optional) The starting scan record number.

Sample - List scan details

Let us view the list of first latest 100 scans starting at record number 1 with the scans sorted by title.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-
api.qualys.com/pci/scan/list?sortBy=title&offset=1&limit=10"
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "totalCount": 1267,
        "fetchRange": "1-10",
        "scanInfoList": [
            {
                "scanId": 9094,
                "title": "a",
                "status": "Failed",
                "date": "December 18, 2020 at 07:54 AM GMT",
                "scanType": "DNS",
                "compliance": "Fail"
            },
            {
                "scanId": 9093,
                "title": "a",
                "status": "Running",
                "date": "December 18, 2020 at 07:54 AM GMT",
                "scanType": "DNS",
                "compliance": "Fail"
            },
            {
                "scanId": 9092,
                "title": "a",
                "status": "Failed",
                "date": "December 18, 2020 at 07:50 AM GMT",
```

```
                "scanType": "DNS",
                "compliance": "Fail"
            },
            {
                "scanId": 9091,
                "title": "a",
                "status": "Failed",
                "date": "December 18, 2020 at 07:50 AM GMT",
                "scanType": "IP",
                "compliance": "Fail"
            },
            {
                "scanId": 9090,
                "title": "a",
                "status": "Launch Requested",
                "date": "December 18, 2020 at 07:44 AM GMT",
                "scanType": "DNS",
                "compliance": "Fail"
            }
        ]
    }
}
```

Get Scan Details

/pci/scan/{scanId}/details

[GET]

View details for a PCI scan which is in your scope. Want to find a scan ID to use as input? See List PCI Scans.

We display these fields in the API output for finished scans: Title, Started On, Launched By, Duration, ActiveHosts, LaunchType, Bandwidth, Scan Status, Target, scanType, compliance. If the scan is in progress (RUNNING & LAUNCH_REQUESTED), then the API outputs of these scans show these additional parameters along with the parameters mentioned for the finished scans: Total IPs Scanned, Total Hosts, Host Discovery running on, Scan running on, Last Updated.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The element “id” (integer) is required, where “id” identifies the scan ID. Want to find a ID to use as input? See List PCI Scans.

Sample - View details for a scan

Let us view details of scan that has ID 2185043 .

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-api.qualys.com/pci/scan/2185043/details"
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "title": "TestCipher-LB",
        "startedOn": "July 18, 2021 at 09:41 PM GMT",
```

```
        "launchedBy": "John Doe",
        "duration": "00:09:34",
        "activeHosts": 2,
        "launchType": "On Demand",
        "bandwidth": "High",
        "scanStatus": "Canceled",
        "target": "130.35.8.154, 130.35.10.110",
        "scanType": "IP",
        "compliance": "Fail"
    }
}
```


Sample - View details for a running scan

The running scans show some additional parameters in the ScanProgress tag.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json apiVersion: V1"
"https://pci-api.qualys.com/scan/2185075/details"
```

JSON response


```
{
    "responseApiVersion": "LATEST",
    "data": {
        "title": "a",
        "startedOn": "July 18, 2021 at 07:54 AM GMT",
        "launchedBy": "John Doe",
        "duration": "00:00:00",
        "activeHosts": 0,
        "launchType": "On Demand",
        "bandwidth": "Medium",
        "scanStatus": "Running",
        "target": "www.bbb.com,www.ccc.com,abc.com,www.aaa.com",
        "scanType": "DNS",
        "compliance": "Fail",
        "scanProgress": {
            "totalIpsScanned": 0,
            "totalHosts": 0,
            "hostDiscoveryRunningOn": "-",
            "scanRunningOn": "-",
            "lastUpdated": "-"
        }
```

```
    }
}
```

Launch Scans (On Demand)

/pci/scan/launch

[POST]

Launch PCI scan in the user's account. When you make a request to launch a scan using this API, the service will return a scan reference ID right away and the call will quit without waiting for the complete scan results.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The input parameters for launching a PCI scan are shown below.

Parameter	Description
title={value}	(Required) The scan title. This can be maximum of 256.
bandwidth={value}	(Optional) The bandwidth level you select will affect overall scan performance. Valid values for bandwidth are: High, Medium, Medium-low HTTP impact, Lowest. If no value is set, then MEDIUM will be set as the default bandwidth value. We recommend Medium to get started. High - Scan performance is optimized for high bandwidth usage resulting in the fastest possible scan time. The High bandwidth level will employ multiple scanners and high multiples of concurrent probes. While scans run at the High bandwidth level may be faster to complete, they may overload your network and/or its devices. For network scans, this level is recommended when scanning a large number

	of IP addresses with robust services. Medium - Scan performance is optimized for medium bandwidth usage and fast scan processing of HTTP hosts (web servers). This level is the recommended setting. Medium-low HTTP impact - Scan performance is optimized for medium bandwidth usage and low impact scanning of HTTP hosts (web servers). Low - Scan performance is optimized for low bandwidth usage and low impact scanning of HTTP hosts (web servers). Lowest - Scan performance is optimized for the lowest possible bandwidth usage and low impact scanning of HTTP hosts (web servers).
scanType ={IP|DNS}	(Required) Specify IP to scan by IPs or DNS to scan by DNS.
splitTarget ={true|**false**}	(Optional) Specify this parameter only if scanType is set to DNS. Specify true If you want to scan DNS hosts that resolve to same IP address, use Split Targets option. You can add a maximum of 500 DNS hosts if you want to scan DNS hosts using Split Targets option. Note that your scan time will increase if you select this option. The parameter is set to false if no value is specified. This parameter accepts value 0 for false and value 1 for true.
targetIps = {value}	(Required if scanType = IP) The IP addresses to be scanned. You may enter individual IP addresses and/or ranges. Multiple entries are comma separated. For example, 10.10.10.9-10.10.10.13, 9.9.9.1,20. 20.20.20-20.20.30.11

targetDns = {value}	(Required if scanType = DNS) The DNS hosts to be scanned. Multiple entries are comma separated. Note that Scan by DNS supports scanning DNS hosts that resolve to unique IP addresses. For example, www.bbb.com, www.ccc.com, abc.com,www.aaa.com
targetAll ={true|**false**}	(Optional) Specify true to scan all IPs or DNS hosts based on the value specified in the scanType parameter. To meet PCI compliance, all the IPs or DNS hosts in your account must be scanned and there can be no detected PCI vulnerabilities on any IPs/DNS hosts. This parameter accepts value 0 for false and value 1 for true. If you have a large number of IPs/DNS hosts that must be compliant, you may want to scan a few IPs at a time to help you with the remediation process. Note that if targetAll is true then we will ignore any values that you specify either in targetIps or targetDns parameter.
launchType ={ONDEMAND}	(Optional) Specify "ONDEMAND" to launch a new scan.

Sample - Launch a new scan by IP

Let us launch an On Demand scan to scan the assets by IP address.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1" -d
@postdatafile.json
"https://pci-api.qualys.com/pci/scan/launch"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "title":"My Scan",
    "bandwidth":"MEDIUM",
    "scanType": "IP",
    "splitTarget":"false",
    "targetIps":"10.10.10.9-10.10.10.13,9.9.9.1,20.20.20.20-
20.20.30.11",
    "targetAll":"false",
    "launchType":"ONDEMAND"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "id": 9097,
        "status": "Launched scan successfully"
    }
}
```

Sample - Launch a new scan by DNS hosts

Let us launch an on demand scan to scan the assets by DNS hosts.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1" -d
@postdatafile.json
"https://pci-api.qualys.com/pci/scan/launch"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "title":"My Scan",
    "bandwidth":"MEDIUM",
    "scanType": "DNS",
    "splitTarget":"false",
    "targetDns":"www.bbb.com,www.ccc.com,abc.com,www.aaa.com",
    "targetAll":false,
    "launchType":"ONDEMAND",
```

```
}
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "id": 9098,
        "status": "Launched scan successfully"
    }
}
```

Cancel Scan

/pci/scan/{scanId}/cancel

[PUT]

Cancel a scan in a RUNNING, PAUSED status which is in your scope.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The element “id” (integer) is required, where “id” identifies the scan. Want to find a ID to use as input? See List PCI Scans.

Sample - Cancel an unfinished scan

Cancel the unfinished scan that has the ID 2183053.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X PUT -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-api.qualys.com/pci/scan/2183053/cancel"
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "id": 9093,
        "status": "Scan cancellation request is successful."
    }
}
```

Retrieve Scan Status

/merchant/scan/{scanId}/status

[GET]

Retrieve the status of a scan which is in the user's scope. Possible scan statuses that you may see for a scan: Launch Requested, Running, Finished, Cancelling, Importing, Paused, No Host Alive, Failed.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The element "id" (integer) is required, where "id" identifies the scan. Want to find a ID to use as input? See List PCI Scans.

Sample - View scan status

View status for the scan with the ID 2183053.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-api.qualys.com/merchant/scan/2183053/status"
```

JSON response

```
{
    "responseApiVersion": "LATEST",
    "data": {
        "id": 2183053,
        "status": "Canceled"
    }
}
```

Manage Hosts

List Hosts

/pci/asset/list

[GET]

View details of IP/DNS/Virtual Hosts that you have added to your account. Only the host assets which is in your scope will be listed.

Permissions required - You must have an active PCI merchant account.

Input Parameters

Parameter	Description
assetType={ALL|IP|DNS|VIRTUALHOST}	(Optional) Specify the type of asset that you want to list. Default value is ALL which will list all the assets in your account.

Sample - List host details

Let us view the assets in your account.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-api.qualys.com/pci/asset/list?"
```

JSON response

```
{
"responseApiVersion": "LATEST - V1",
```

```
"data": {
"ip": "1.1.1.1,3.3.2.1,3.3.2.4,4.4.4.1-4.4.4.2,9.9.9.1-
9.9.9.10,20.20.20.20-20.20.30.20",
"dns":
"www.bbb.com,www.ccc.com,abc.com,www.aaa.com,cmacbb.com,cmamcb.com,cmb
aoo.com,cmbbca.com,cmbboc.com,cmbcca.com,cmcbcb.com,cmcbcc.com,cmccab.
com,cmcccm.com,cmcoab.com,cmcobo.com,cmmabo.com,cmmbco.com",
"virtualHost":
"WWW.ABC1.COM:10:10.10.10.10/home/web/html/app/param1=test&param2=1234
5+category,isc.iq:80:20.20.20.20,WWW.AB1C.COM:10:30.30.30.30/studentwe
b,WWW.ABC.COM:10:40.40.40.40,WWW.A1BC.COM:10:50.50.50.50"
}
}
```

Add Hosts

/pci/asset/add

[POST]

Add IP, DNS, and Virtual Hosts to your account.

Note that for PCI Express accounts, the capability to remove IP hosts is not permitted if the PCI account is linked to a VM (Vulnerability Management) account.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The input parameters for adding host assets are shown below.

Parameter	Description
assetType={value}	(Required) Specify the type of asset that you want to add. The valid values are: IP, DNS, and VIRTUALHOST. The values are case-insensitive.
assets	(Required) Depending on the value you specified for assetType, you need to provide either the IP address, DNS host, or Virtual Host. When adding multiple IPs/DNS/Virtual hosts, separate each one with a comma. If "assetType" is IP, then specify the IP addresses that you want to scan and report on. Your IP assets are the internet-facing IP addresses and or ranges that must be scanned for PCI compliance. If you have domains that are host in-scope PCI infrastructure, these domains must be added to your account. You can specify maximum number of IP hosts equal to the purchased IP count. Examples of valid IP/Range: 192.168.0.200,192.168.0.87-192.168.0.92. The API

returns an error if 1) the number of IP added exceeds the purchased IP count, 2) the IPs that you are trying to add already exist in your account, 3) the IP is restricted, duplicate or invalid.

If "assetType" is DNS, then specify the DNS hosts that you want to scan and report on. Example DNS hosts: www.example.com or example.com.

If "assetType" is VIRTUALHOST, then specify the virtual host addresses that you want to scan and report on. A virtual host is a single machine that acts like multiple systems, hosting more than one domain. For example, an ISP could use one server with IP address 194.55.109.1 to host two Web sites on the same port: www.merchantA.com and www.merchantB.com. You can add up to the maximum number of Virtual hosts permitted in your account settings (From UI, go to Account > Settings).

Supported formats: FQDN:Port:IP, FQDN:Port:IP/Path. For example:

www.merchantA.com:2020:194.55.109.1
www.merchantB.com:2020:194.55.109.1/path2
www.merchantC.com:8080:194.55.109.1

A valid virtual host should consist of the IP address of the virtual host, the port number to be associated with the hosted domain, and the domain name (FQDN) to be hosted by the IP address. Only Path to subdirectory is optional. A virtual host path should have maximum 512 characters, the site value should have minimum 4 characters and maximum 256, the IP address and Site must be valid, and port must be numeric with length 1-5.

The API returns an error if 1) the number of virtual hosts exceeds the purchased virtual host count, 2) the virtual hosts that you are trying to add already exist in your account, and 3) the virtual host is restricted, duplicate or invalid.

Sample - Add an IP address

Let us add an IP address.

API request

```
curl -u "USERNAME:PASSWD" -X POST -H "content-type: application/json"
-H "apiVersion:V1" -d @postdatafile.json
"https://pci-api.qualys.com/pci/asset/add"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"IP",
    "assets": "65.67.89.99"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "1 IP added successfully"
}
```

Sample - Add a DNS host

Let us add a DNS host.

API request

```
curl -u "USERNAME:PASSWD" -X POST -H "content-type: application/json"
-H "apiVersion:V1" -d @postdatafile.json
"https://pci-api.qualys.com/pci/asset/add"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"DNS",
    "assets": "dns1.com"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "1 DNS added successfully"
}
```

Sample - Add a virtual host

Let us add a virtual host.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1" -d
@postdatafile.json
"https://pci-api.qualys.com/pci/asset/add"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"VIRTUALHOST",
    "assets":
"WWW.ABC1.COM:10:10.10.10.10/home/web/html/app/param1=test&param2=1234
5+category"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "1 VirtualHost added successfully"
}
```

Remove Hosts

/pci/asset/delete

[DELETE]

Delete IP, DNS, and Virtual Hosts from your account.

Note that for PCI Express accounts, the capability to remove IP hosts is not permitted if the PCI account is linked to a VM account.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The input parameters for removing host assets are shown below.

Parameter	Description
assetType={value}	(Required) Specify the type of asset that you want to remove. The valid values are: IP, DNS, and VIRTUALHOST. The values are case-insensitive.
assets	(Required) Depending on the value you specified for assetType, you need to provide either the IP address, DNS host, or Virtual Host. When removing multiple IPs/DNS/Virtual hosts, separate each one with a comma. If "assetType" is IP, then you can remove the IP only when these conditions are true: 1) no scan is currently running on the IP within your account, 2) the IP is not included in the target of any scheduled scans within your account, and 3) the IP is not included in a scan currently being imported from the VM module to your account. Trial user must submit a request for the removal of IPs from UI. In this case you may submit a request for

the removal of IPs that are currently in your account and a support representative will process your request. If you are using Express PCI, this workflow is not available. Please contact your account manager.

The API returns an error if 1) the IPs to be removed are being used in running or paused scans, 2) the IPs to be removed are being used in the future scheduled scans, 3) you are a Trial user, 4) the IPs that you are trying to remove are not in your account or the specified IPs are invalid.

If "assetType" is DNS, then specify the address of DNS hosts that you want to remove. The API returns an error if the DNS hosts that you are trying to remove are not in your account or the specified DNS hosts are invalid.

If "assetType" is VIRTUALHOST, then specify the virtual host addresses that you want to remove. A valid virtual host should consist of the IP address of the virtual host, the port number to be associated with the hosted domain, and the domain name (FQDN) to be hosted by the IP address. Only Path to subdirectory is optional. A virtual host path should have maximum 512 characters, the site value should have minimum 4 characters and maximum 256, IP address and Site must be valid, and port must be numeric with length 1-5. The API returns an error if the Virtual hosts that you are trying to remove are not in your account or the specified virtual hosts are invalid.

Sample - Remove IP addresses

Let us remove IP addresses by specifying an IP range.

API request

```
curl -H "X-Requested-With: test"-u "USERNAME:PASSWD" -X DELETE -H
"content-type: application/json" -H "apiVersion:V1"
-d @postdatafile.json
```

```
"https://pci-api.qualys.com/pci/asset/delete"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"IP",
    "assets": "10.10.10.9-10.10.10.13,9.9.9.1"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "6 IP removed successfully"
}
```

Sample - Remove a DNS host

Let us remove a DNS host.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X DELETE -H
"content-type: application/json" -H "apiVersion:V1" -d
@postdatafile.json
"https://pci-api.qualys.com/pci/asset/delete"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"DNS",
    "assets": "dns1.com"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "1 DNS Host removed successfully"
}
```

Sample - Remove a virtual host

Let us remove a virtual host.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X DELETE -H
"content-type: application/json" -H "apiVersion:V1" -d
@postdatafile.json
"https://pci-api.qualys.com/pci/asset/delete"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
{
    "assetType":"VIRTUALHOST",
    "assets":
"WWW.ABC1.COM:10:10.10.10.10/home/web/html/app/param1=test&param2=1234
5+category"
}
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "1 VirtualHost removed successfully"
}
```

Vulnerabilities

List Vulnerabilities

/pci/vuln/list

[GET]

Lists the current vulnerabilities of a recent scan. Search for the vulnerabilities with failed compliance status and with rejected and expired false positive request. In the API output, we show ID(which will be further used to raise False Positive request), Title, PCI compliant status, severity, host IP address, DNS host, last scanned date and false positive status.

Permissions required - You must have an active PCI merchant account.

Input Parameters

The input parameters to filter the vulnerabilities.

Parameter	Description
sortBy={title|severity|ip|scandate}	(Optional) Specify how you want to organize the results. One of: "host IP address", "scan date", "vulnerability's title", "severity". If you do not specify any value, then vulnerabilities will be sorted by host IP address.
sortOrder={asc|desc}	(Optional) The sort order, used when the request includes the sortBy parameter. One of: asc (for ascending order) or desc (for descending order). If you do not specify any value, then vulnerabilities will be sorted in descending order.

limit={value}	(Optional) The maximum number of records processed for the request, starting at the record number specified by the offset parameter. Limit value must always be greater than “0”. If you specify a value 0 for the parameter, the request will fail. When not specified, default limit is set to 100 scan records. You can specify a value less than or greater than the default. Maximum limit value that you can specify is 1000.
offset={value}	(Optional) The starting scan record number.
ip={value}	(Optional) Show vulnerabilities for only certain IP addresses. One or more IPs/ranges may be specified. Multiple entries are comma separated. A host IP range is specified with a hyphen (for example, 10.10.10.44-10.10.10.90).
dns={value}	(Optional) Show vulnerabilities for a DNS host. For example, 205-189-240-1.bogus.tld.
title={value}	(Optional) Show vulnerabilities for a vulnerability title. You can search vulnerability title by keywords.
qid={value}	(Optional) Show vulnerabilities for a vulnerability ID (QID). For example, 115731.
severity={POTENTIAL_HIGH|POTENTIAL_MED|POTENTIAL_LOW|	(Optional) Show vulnerabilities for certain severity. You can

CONFIRMED_HIGH|CONFIRMED_MED|CONFIRMED_LOW}	search for potential and confirmed vulnerability with high, medium and low severity. Multiple entries are comma separated. To filter vulnerabilities for multiple severities, use comma to separate the values: "severity=CONFIRMED_MED, POTENTIAL_MED, POTENTIAL_HIGH".
falsePositive={Requested, Rejected, Expired}	(Optional) Show vulnerabilities for which false positive request status is submitted, rejected and expired. Multiple entries are comma separated. To filter vulnerabilities for multiple false positive values, use comma to separate the values: "falsePositive=Requested, Rejected, Expired".
pciFailVulns={0|1/true|false}	(Optional) When set to true or 1, we will show you the vulnerabilities with failed PCI Compliance status. When set to false or 0, we will show all the vulnerabilities irrespective of their compliance status. Default value of the parameter is false.
fpSubmissionList={0|1/true|false}	(Optional) Specify true or 1 to show a sample JSON body with all the IDs for PCI fail vulnerabilities for which you can submit false positive requests. This means you do not have to copy each ID from the API output to prepare a JSON

payload to submit false positive request. The pciFailVulns parameter must be set to true if this parameter is true.

Sample - List vulnerabilities

Let us list vulnerabilities with severity as confirmed medium and potential medium and PCI compliance status as fail.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-
api.qualys.com/pci/vuln/list?limit=10&sortOrder=desc&sortBy=severity&o
ffset=1&limit=10&severity=CONFIRMED_MED,
POTENTIAL_MED,POTENTIAL_HIGH&pciFailVulns=true";
```

JSON response


```
{
    "responseApiVersion": "LATEST - V1",
    "data": {
        "totalCount": 7306,
        "fetchRange": "1-10",
        "merchantVulnList": [
            {
                "id": 636136,
                "qid": 110033,
                "title": "Microsoft Office 2003 SP2 Missing",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.220",
                "dns": "10-10-2-220.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 617316,
```

```
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.109",
                "dns": "10-10-2-109.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 614872,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.92",
                "dns": "10-10-2-92.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 614400,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.88",
                "dns": "10-10-2-88.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 614392,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.88",
                "dns": "10-10-2-88.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
```

```
            },
            {
                "id": 611910,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.74",
                "dns": "10-10-2-74.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 611891,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.74",
                "dns": "10-10-2-74.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 608954,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.52",
                "dns": "10-10-2-52.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 607120,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.39",
```

```
                "dns": "10-10-2-39.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            },
            {
                "id": 607109,
                "qid": 86473,
                "title": "Web Server HTTP Trace/Track Method Support
Cross-Site Tracing Vulnerability",
                "pciCompliant": "Fail",
                "severity": "Confirmed Medium",
                "ip": "10.10.2.39",
                "dns": "10-10-2-39.bogus.tld",
                "dateLastScanned": "10/07/14",
                "fpStatus": "NA"
            }
        ]
    }
}
```

Get Vulnerability Details

pci/vuln/{id}/details

[GET]

View detailed information about a vulnerability. The API output fields are described in the PCI Online Help in the "Vulnerability Details" topic.

Input Parameters

The element “id” (integer) is required, where “id” identifies the record. Want to find a ID to use as input? See List Vulnerabilities.

Sample - View details for the vulnerability

Let us view details of a vulnerability that has ID 6316136.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X GET -H
"content-type: application/json" -H "apiVersion: V1"
"https://pci-api.qualys.com/pci/vuln/636136/details"
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": {
        "title": "Statd Format Bug Vulnerability",
        "ip": "10.10.1.65",
        "dns": "N/A",
        "qid": 66040,
        "severity": "Potential Low",
        "cvssBase": "0",
        "cvssTemporal": null,
        "pciCompliant": "Fail",
        "category": "RPC",
        "port": null,
        "service": "RPC",
        "protocol": null,
        "fpStatus": "NA",
```

```
        "bugTraqList": [
            {
                "url": "http://www.securityfocus.com/bid/1480",
                "urlText": "1480"
            }
        ],
        "cveList": [
            {
                "url": "http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-2000-0666",
                "urlText": "CVE-2000-0666"
            },
            {
                "url": "http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-2000-0800",
                "urlText": "CVE-2000-0800"
            }
        ],
        "vendorReferenceList": [
            {
                "url": "http://rhn.redhat.com/errata/RHSA-2000-
043.html",
                "urlText": "RHSA-2000-043"
            }
        ],
        "dateLastUpdate": "June 05, 2009 at 12:00 AM GMT",
        "threat": "The rpc.statd program, which is part of the nfs-
utils packages, is distributed with a number of popular Linux
distributions. The rpc.statd server is an RPC server that implements
the Network Status and Monitor RPC protocol. It's a component of the
Network File System (NFS) architecture. \n<P>\nrpc.statd contains a
format string vulnerability when calling the syslog() function. This
vulnerability allows remote users to execute code as root. The logging
code in rpc.statd uses the syslog() function to pass user-supplied
data as the format string. A malicious user can construct a format
string that injects executable code into the process address space and
overwrites a function's return address, forcing the program to execute
the code.\n<P>\nrpc.statd requires root privileges for opening it's
network socket, but fails to drop these privileges later on.
Therefore, code injected by the malicious user will execute with root
privileges.\n<P>\nDebian, Red Hat and Connectiva have all released
advisories on this matter. Presumably, any Linux distribution that
runs the statd process is vulnerable, unless already patched for the
problem.",
```

```
        "impact": "If successfully exploited, unauthorized users can
execute remote commands as root.\n",
        "solution": "For Red Hat Linux:\nUpgrade to the latest version
of nfs-utils (0.1.9.1 or later), as listed in <A TARGET=\"_blank\"
HREF=\"http://rhn.redhat.com/errata/RHSA-2000-043.html\">RHSA-
2000:043-02</A>.\n<BR>\nFor Debian Linux:\nUpgrade to the latest
version of nfs-utils (0.1.9.1 or later), as listed in <A
TARGET=\"_blank\"
HREF=\"http://www.debian.org/security/2000/20000719a\">Debian Security
Advisory 20000719a</A>.\n<BR>\nFor other distributions:\nContact your
vendor for upgrade or patch information.\n\n\n",
        "patch": "<P> <A HREF=\"http://rhn.redhat.com/errata/RHSA-
2000-043.html%22%3ERHSA-2000:043-02\" TARGET=\"_blank\">RHSA-2000:043-
03: Red Hat Linux 6.2</A>",
        "result": "N/A"
    }
}
```

Submit False Positive Request

pci/falsePositive/create

[POST]

Submit the false positive request for vulnerabilities listed in List Vulnerabilities API Output using the ID received from the JSON response. You can submit maximum 1000 false positives in a request. Submission of false positive in approved or pending state is not allowed. After successful false positive submission, we will send you email notification. For Qualys Partner, we will send email notification to only customer. For Non-Qualys Partner, we will send email notification to both Partner Support (if Support email is configured for Partner in the PCI Admin module) and customer.

Each approved false positive is valid for 90 days. After 90 days, the approved false positive will expire automatically. The next time you run a network scan after a false positive expires, if the QID is detected on the host, you will fail PCI compliance.

Permissions required - You must have an active PCI merchant account.

Input Parameters

Parameter	Description
id={value}	(Required) The element “id” (integer) is required, where “id” identifies the vulnerability for which you want submit false positive requests. The ID is validated with your account to ensure that ID exists in your account. Multiple entries are comma separated. For example, "1282052,1276964,1276901"
comment={value}	(Required) The reason for your false positive request for each vulnerability.
sameComment={true|false}	(Optional) Specify true to apply same

	comment to multiple false positive submissions which are passed in the id parameter as comma-separated list of ids. Specify false for single false positive submission.

Sample - Submit a false positive request

Submit false positive request for vulnerabilities with multiple IDs.

API request

```
curl -H "X-Requested-With: test" -u "USERNAME:PASSWD" -X POST -H
"content-type: application/json" -H "apiVersion:V1"
-d @postdatafile.json
"https://pci-api.qualys.com/pci/falsePositive/create"
Note: “postdatafile.json” contains the request POST data.
```

Request POST data

```
[{
  "id":"1282052,1276964,1276901",
  "comment":"same comment for all false positive
1282052,1276964,1276901",
  "sameComment":"true"
},
{  "id":"1284130",
   "comment":" 1284130 single comment for single false positive",
   "sameComment":"false"
},
{
   "id":"1136846",
   "comment":"1136846 single comment for single false positive"
}]
```

JSON response

```
{
    "responseApiVersion": "LATEST - V1",
    "data": "5 False Positive requests submitted successfully"
}
```