Certificate View API

User Guide

Version 2.3.0

August 2, 2019

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100

Table of Contents

Preface

This user guide is intended for application developers who will use the Qualys Certificate View 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 apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.

Chapter 1 - Welcome

Welcome to Certificate View API.

Get Started

Qualys API Framework - Learn the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

Introduction to Certificate View API Paradigm - Get tips on using the Curl command-line tool to make API requests. Every API request must authenticate using a JSON Web Token (JWT) obtained from the Qualys Authentication API.

Get API Notifications

Subscribe to our API Notifications RSS Feeds for announcements and latest news.

From our Community

Join our Community

API Notifications RSS Feeds

Qualys API Framework

The Qualys Certificate View API uses the following framework.

Request URL

The URL for making API requests respects the following structure:

https://<baseurl>/<module>/<object>/<object_id>/<operation>

where the components are described below.

<baseurl>	The Qualys API server URL that you should use for API requests depends on the platform where your account is located. The base URL for Qualys US Platform 1 is: https:///gateway.qg1.apps.qualys.com
<module>	The API module. For the Certificate View API, the module is: “certview”.
<object>	The module specific object.
<object_id>	(Optional) The module specific object ID, if appropriate.
<operation>	The request operation, such as count.

Base URL to the Qualys API Server

The Qualys API documentation and sample code within it use the API server URL for Qualys US Platform 1: gateway.qg1.apps.qualys.com.

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

Account Login	API Server URL
Qualys US Platform 1	https://gateway.qg1.apps.qualys.com
Qualys US Platform 2	https://gateway.qg2.apps.qualys.com
Qualys US Platform 3	https://gateway.qg3.apps.qualys.com
Qualys EU Platform 1	https://gateway.qg1.apps.qualys.eu
Qualys EU Platform 2	https://gateway.qg2.apps.qualys.eu
Qualys India Platform 1	https://gateway.qg1.apps.qualys.in
Qualys Private Cloud Platform	https://gateway.<customer_base_url>

Introduction to Certificate View API Paradigm

Authentication

You must authenticate to the Qualys Cloud Platform using Qualys account credentials (user name and password) and get the JSON Web Token (JWT) before you can start using the Certificate View APIs. Use the Qualys Authentication API to get the JWT.

For example,

```
curl -X POST https://gateway.qg1.apps.qualys.com/auth -d
"username=value1&password=passwordValue&token=true&permissions=tru
e" -H "ContentType: application/x-www-form-urlencoded
```

where gateway.qg1.apps.qualys.com is the base URL to the Qualys API server where your account is located.

- **username** and **password** are the credentials of the user account for which you want to fetch Certificate View data

- **token** should be true

- **permissions** should be true

- **Content-Type** should be "application/x-www-form-urlencoded"

The Authentication API returns a JSON Web Token (JWT) which you can use for authentication during Certificate View API calls. The token expires in 2 to 3 hours. You must regenerate the token to continue using the Certificate View API.

Using Curl

Curl is a multi-platform command-line tool used to transfer data using multiple protocols. This tool is supported on many systems, including Windows, Unix, Linux and Mac. In this document Curl is used in the examples to build Qualys API requests using the HTTP over SSL (https) protocol, which is required.

Want to learn more? Visit https://curl.haxx.se/

The following Curl options are used according to different situations:

Option	Description
-X “POST”	The POST method is required for all Certificate View API requests.
-H “Authorization: Bearer <token>”	This option is used to provide a custom HTTP request header parameter for authentication. Provide the JSON Web Token (JWT) received from Qualys authentication API in the following format: `Authorization: Bearer <token>` For information about Qualys authentication API, see Authentication.

The sample below shows a typical Curl request using options mentioned above and how they interact with each other.

curl -X POST "https://gateway.qg1.apps.qualys.com/certview/v1/certificates" -H "Authorization: Bearer <token>"

Chapter 2 - Certificate API

Use these API functions to retrieve a list of certificates based on an input filter query. The response contains certificate details including associated host information and SSL/TLS related vulnerabilities and grades.

List Certview Certificates

/certview/v1/certificates

[POST]

Input Parameters

filter (String)	Filter the events list by providing a query using Qualys syntax. Refer to the "How to Search" topic in the online help for assistance with creating your query. For example - expiryGroup: Expired Refer to the list of tokens you can use to build the query: Search tokens
pageNumber (Integer)	The page to be returned. Starts from zero.
pageSize (Integer)	The number of records per page to be included in the response. Default is 10.
sort (String)	Sort the results using a Qualys token. For example - [{"lastFound":"desc"}]
Authorization (String)	(Required) Authorization token to authenticate to the Qualys Cloud Platform. Prepend token with "Bearer" and one space. For example - Bearer authToken

Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

Sample with all parameters defined

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/certificates" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{ \"filter\" : \"expiryGroup:Expired\", \"pageNumber\": 0,
\"pageSize\" : 50, \"sort\" :\"[{\\\"lastFound\\\" :
\\\"asc\\\"}]\"}" -H "Authorization: Bearer <JWT Token>"
```

Response:

```
[{
    "keySize": 1024,
```

```
    "subject": {
      "organization": "Qualys, Inc.",
      "locality": "Foster City",
      "name": "wayne.qualys-demo.com",
      "state": "ca",
      "country": "US",
      "organizationUnit": [
        "QA"
      ]
    },
    "validFrom": 1545782400000,
    "signatureAlgorithm": "SHA256withRSA",
    "issuer": {
      "organization": "DigiCert Inc",
      "organizationUnit": [],
      "name": "DigiCert Test SHA2 Intermediate CA-1",
      "country": "US",
      "state": "",
      "certhash":
"a52d05988b61a33d6ac3edb449eb47150fa5b7a26c7dfc4e61f905ca36e165ee"
,
      "locality": ""
    },
    "rootIssuer": null,
    "instanceCount": 1,
    "dn": "CN=wayne.qualys-demo.com, OU=QA, O=\"Qualys, Inc.\",
L=Foster City, ST=ca, C=US",
    "certhash":
"6ebb35565f886d2a120388cc773be5d54aaeec701cf78a16abce7175ed6648d6"
,
    "assets": [
      {
        "netbiosName": "SYS_10_113_196_229",
        "assetId": "38187ae6-f364-43c6-9fe2-476ee438c43a",
        "name": "10.113.196.229",
        "hostId": null,
        "hostname": null,
        "operatingSystem": "Ubuntu / Tiny Core Linux / Linux
2.6.x",
        "tags": [
          {
            "name": "Business Units",
            "uuid": "fe90f3de-c1fa-474f-89b2-9de7ba67816b"
          },
          {
            "name": "prahalad_1",
```

```
            "uuid": "d6b28108-454f-492d-8319-cfdb32bc0e32"
          },
          {
            "name": "Test",
            "uuid": "cadd8477-ce01-4d81-a86c-dc909304ed24"
          },
          {
            "name": "Cloud Agent",
            "uuid": "0bd92635-e857-4ff9-812e-77297607abcb"
          },
          {
            "name": "Test_Asset_Grp",
            "uuid": "c96ceb01-1bf6-41a1-8129-184601ab42b3"
          },
          {
            "name": "new1_grp",
            "uuid": "e753e7d7-1860-44c8-8936-eea4784f411f"
          }
        ],
        "certificates": null,
        "hostInstances": [
          {
            "severity": null,
            "protocol": "tcp",
            "sslProtocols": [
              "TLSv1.2"
            ],
            "gradingHighlights": null,
            "port": 443,
            "grade": "T",
            "status": null,
            "category": null,
            "service": "http",
            "fqdn": "",
            "gradingDetails": null,
            "gradeSummary": null,
            "trusted": null,
            "vulnerabilities": [
              {
                "title": "SSL Certificate - Information",
                "severity": 1,
                "qid": 86002
              },
              {
                "title": "SSL/TLS Server supports
TLS_FALLBACK_SCSV",
```

```
                "severity": 1,
                "qid": 38610
              },
              {
                "title": "SSL Server Information Retrieval",
                "severity": 1,
                "qid": 38116
              },
              {
                "title": "Deprecated Public Key Length",
                "severity": 2,
                "qid": 38598
              },
              {
                "title": "SSL Certificate - Expired",
                "severity": 2,
                "qid": 38167
              },
              {
              "title": "SSL Certificate - Subject Common Name Does
Not Match Server FQDN",
                "severity": 2,
                "qid": 38170
              },
              {
               "title": "SSL Certificate - Signature Verification
Failed Vulnerability",
                "severity": 2,
                "qid": 38173
              },
              {
             "title": "TLS Secure Renegotiation Extension Support
Information",
                "severity": 1,
                "qid": 42350
              }
            ],
            "vulnCount": 8,
            "qids": null,
            "network": null
          }
        ],
        "created": 1545211730000,
        "updated": 1558527287000,
        "vulnCount": null,
        "assetInterfaces": [
```

```
        {
          "hostname": "",
          "address": "10.113.196.229"
        }
      ],
      "gradeCell": null,
      "certificateCount": 0
    }
  ],
  "selfSigned": false,
  "validTo": 1548936000000,
  "hostInstances": null,
  "issuerCategory": "DigiCert Test SHA2 Intermediate CA-1",
  "serialNumber": null,
  "subjectAlternativeNames": {
    "IP Address": null,
    "DNS Name": [
      "wayne.qualys-demo.com",
      "bruce-wayne.qualys-demo.com"
    ]
  },
  "lastFound": 1558527287000,
  "extendedValidation": false
}]
```

Chapter 3 - Analyze Certificate API

Use these API functions to analyze information based on host or IP.

Use this API to retrieve the list of endpoints that are associated with an FQDN in the CertView inventory

Analyze Certificate Information

/certview/v1/analyze

[POST]

Input Parameters

host (String)	(Required) Host on which scan is executed, it can be IP For example - www.ssllabs.com, 10.10.10.10
Authorization (String)	(Required) Authorization token to authenticate to the Qualys Cloud Platform. Prepend token with "Bearer" and one space. For example - Bearer authToken

Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

Sample - Host is IP

Request:

```
curl -X POST "gateway.qg1.apps.qualys.com/certview/v1/analyze" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{\"host\":\"64.41.200.100\"}" -H "Authorization: Bearer <JWT
Token>"
```

Response:

```
{
    "host": "64.41.200.100",
    "endpoints": [
        {
            "ipAddress": "64.41.200.100",
            "port": 443,
            "service": "http",
            "grade": "A+",
            "gradeTrustIgnored": "A+",
            "hasWarnings": false,
            "exceptional": true
        }
```

```
    ]
}
```

Sample - Host is FQDN

Request:

```
curl -X POST "gateway.qg1.apps.qualys.com/certview/v1/analyze" -H
"Accept: application/json" -H "Content-Type: application/json" -d
"{\"host\":\"www.ssllabs.com\"}" -H "Authorization: Bearer <JWT
Token>"
```

Response:


```
{
  "host": "www.ssllabs.com",
  "endpoints": [
    {
      "ipAddress": "64.41.200.100",
      "port": 443,
      "service": "http",
      "grade": "A+",
      "gradeTrustIgnored": "A+",
      "hasWarnings": false,
      "isExceptional": true
    },
    {
      "ipAddress": "64.41.200.100",
      "port": 5443,
      "service": "rdp",
      "grade": "A+",
      "gradeTrustIgnored": "A+",
      "hasWarnings": false,
      "isExceptional": true
    },
    {
      "ipAddress": "64.41.200.111",
      "port": 443,
      "service": "http",
      "grade": "A+",
      "gradeTrustIgnored": "A+",
      "hasWarnings": false,
      "isExceptional": true
    }
  ]
}
```

Chapter 4 - Endpoint API

Use these API function to retrieve detailed endpoint information.

List Endpoints

/certview/v1/getEndpointData

[POST]

Input Parameters

Authorization (String)	(Required) Authorization token to authenticate to the Qualys Cloud Platform. Prepend token with "Bearer" and one space. For example - Bearer authToken
ip (String)	(Required) Host IP for which the endpoint details are required.
port (Integer)	Used to filter the endpoint details based on port. In Certview scan, we can scan multiple ports as certificates can be found on multiple ports. Define the port number to filter the endpoint data based on port.
fqdn (String)	Used to filter the endpoint details based on FQDN. Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the service or protocol parameter is specified.
service (String)	Used to filter the endpoint details based on service. Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the fqdn or protocol parameter is specified.
protocol (String)	Used to filter the endpoint details based on protocol. Note: For filtering based on fqdn, port is required parameter. Also, this field is required if the service or protocol parameter is specified.

Permissions

- User must be a Super User or must have the CERTVIEW.API.ACCESS permission.

Sample with all parameters defined

Request:

```
curl -X POST
"https://gateway.qg1.apps.qualys.com/certview/v1/getEndpointData"
-H "Accept: application/json" -H "Content-Type: application/json"
-d "{ \"ip\": \"64.41.200.100\", \"port\": 443, \"fqdn\": \"\",
\"service\": \"tcp\"}" -H "Authorization: Bearer <JWT Token>"
```

Response:

```
[
    {
        "ipAddress": "64.41.200.100",
        "port": 443,
        "service": "http",
        "grade": "A+",
        "gradeTrustIgnored": "A+",
        "hasWarnings": false,
        "isExceptional": true,
        "details": {
            "certChains": [
                {
                    "certIds": [

"4bda9e40d19260d636042a0e6ad5222a024f05f7001d61220f17e6632428f1d6"
,

"154c433c491929c5ef686e838e323664a00e6a0d822ccc958fb4dab03e49a08f"
,

"4348a0e9444c78cb265e058d5e8944b4d84f9662bd26db257f8934a443c70161"
                    ],
                    "trustPaths": [
                        {
                            "certIds": [

"4bda9e40d19260d636042a0e6ad5222a024f05f7001d61220f17e6632428f1d6"
,

"154c433c491929c5ef686e838e323664a00e6a0d822ccc958fb4dab03e49a08f"
,

"4348a0e9444c78cb265e058d5e8944b4d84f9662bd26db257f8934a443c70161"
                            ],
                            "trust": [
                                {
```

```
                        "rootStore": "Mozilla",
                        "isTrusted": true
                    }
                ]
            }
        ],
        "noSni": false
    }
],
"protocols": [
    {
        "id": 771,
        "name": "TLS",
        "version": "1.2"
    },
    {
        "id": 770,
        "name": "TLS",
        "version": "1.1"
    },
    {
        "id": 769,
        "name": "TLS",
        "version": "1.0"
    }
],
"suites": [
    {
        "protocol": 771,
        "list": [
            {
                "name": "DHE-RSA-AES128-SHA",
                "cipherStrength": 128,
                "kxType": "DH"
            },
            {
                "name": "DHE-RSA-AES256-SHA",
                "cipherStrength": 256,
                "kxType": "DH"
            },
            {
                "id": 103,
                "name": "DHE-RSA-AES128-SHA256",
                "cipherStrength": 128,
                "kxType": "DH"
```

```
},
{
    "id": 107,
    "name": "DHE-RSA-AES256-SHA256",
    "cipherStrength": 256,
    "kxType": "DH"
},
{
    "id": 158,
    "name": "DHE-RSA-AES128-GCM-SHA256",
    "cipherStrength": 128,
    "kxType": "DH"
},
{
    "id": 159,
    "name": "DHE-RSA-AES256-GCM-SHA384",
    "cipherStrength": 256,
    "kxType": "DH"
},
{
    "id": 49171,
    "name": "ECDHE-RSA-AES128-SHA",
    "cipherStrength": 128,
    "kxType": "ECDH"
},
{
    "id": 49172,
    "name": "ECDHE-RSA-AES256-SHA",
    "cipherStrength": 256,
    "kxType": "ECDH"
},
{
    "id": 49191,
    "name": "ECDHE-RSA-AES128-SHA256",
    "cipherStrength": 128,
    "kxType": "ECDH"
},
{
    "id": 49192,
    "name": "ECDHE-RSA-AES256-SHA384",
    "cipherStrength": 256,
    "kxType": "ECDH"
},
{
    "id": 49199,
```

```
                "name": "ECDHE-RSA-AES128-GCM-SHA256",
                "cipherStrength": 128,
                "kxType": "ECDH"
            },
            {
                "id": 49200,
                "name": "ECDHE-RSA-AES256-GCM-SHA384",
                "cipherStrength": 256,
                "kxType": "ECDH"
            }
        ]
    },
    {
        "protocol": 770,
        "list": [
            {
                "name": "DHE-RSA-AES128-SHA",
                "cipherStrength": 128,
                "kxType": "DH"
            },
            {
                "name": "DHE-RSA-AES256-SHA",
                "cipherStrength": 256,
                "kxType": "DH"
            },
            {
                "name": "ECDHE-RSA-AES128-SHA",
                "cipherStrength": 128,
                "kxType": "ECDH"
            },
            {
                "name": "ECDHE-RSA-AES256-SHA",
                "cipherStrength": 256,
                "kxType": "ECDH"
            }
        ]
    },
    {
        "protocol": 769,
        "list": [
            {
                "id": 51,
                "name": "DHE-RSA-AES128-SHA",
                "cipherStrength": 128,
                "kxType": "DH"
```

```
                },
                {
                    "id": 57,
                    "name": "DHE-RSA-AES256-SHA",
                    "cipherStrength": 256,
                    "kxType": "DH"
                },
                {
                    "id": 49171,
                    "name": "ECDHE-RSA-AES128-SHA",
                    "cipherStrength": 128,
                    "kxType": "ECDH"
                },
                {
                    "id": 49172,
                    "name": "ECDHE-RSA-AES256-SHA",
                    "cipherStrength": 256,
                    "kxType": "ECDH"
                }
            ]
        }
    ],
    "vulnBeast": true,
    "renegSupport": 2,
    "compressionMethods": 0,
    "supportsRc4": false,
    "rc4WithModern": false,
    "rc4Only": false,
    "forwardSecrecy": 4,
    "supportsAead": true,
    "protocolIntolerance": 40,
    "heartbleed": false,
    "heartbeat": true,
    "openSslCcs": 1,
    "openSSLLuckyMinus20": 1,
    "ticketbleed": 1,
    "bleichenbacher": 1,
    "poodle": false,
    "poodleTls": 1,
    "fallbackScsv": true,
    "freak": false,
    "hasSct": 1,
    "logjam": false,
    "drownVulnerable": false,
    "zombiePoodle": 1,
```

```
            "goldenDoodle": 1,
            "supportsCBC": true,
            "zeroLengthPaddingOracle": 1,
            "sleepingPoodle": 1
        }
    }
]
```

Appendix A - Error Messages

This appendix describes the types of error messages returned from Certificate View API requests.

Error Code	Resolution
400	Provide an input parameter in correct syntax.
401	Make sure the user has appropriate permissions to call the Certificates API.
422	Provide a valid input parameter.
500	The server encountered an unexpected condition which prevented it from fulfilling the request. Please try again later.