Red Hat Security Data API 1.0

Red Hat Security Data API

API Documentation

Last Updated: 2024-02-09

Red Hat Security Data API 1.0 Red Hat Security Data API

API Documentation

Legal Notice

Abstract

The Red Hat Security Data API exposes a list of endpoints to query security data with certain parameters and retrieve CSAF, CVE and OVAL data easily.

Table of Contents

CHAPTER 1. OVERVIEW

Red Hat Product Security is committed to providing tools and security data to help you better understand security threats. This data has been available on our **Security Data** page and will now also be available in a machine-consumable format with the Security Data API. This tool allows customers to programmatically query the API for data that was previously exposed only through files on our Security Data page.

The data provided by the Security Data API is the same as what is found on the Security Data page: Common Security Advisory Framework (CSAF) documents, OVAL v2 (OVAL streams) and CVE data. Using the API the data can be fetched in JSON or XML format.

This effort is a part of Red Hat Product Security's commitment to providing security data to customers in an easy-to-use format.

Please Note: Only one version will be maintained and any changes will be noted in the documentation.

The Security Data API is provided for information and metrics purposes. For any questions or concerns with the API or the data it provides, please contact **Red Hat Product Security**.

Base URL

```
https://access.redhat.com/hydra/rest/securitydata
```

Supported Formats

The API supports JSON, XML, and HTML formats. The format can be specified as an extension to the url like .json or .xml.

Deprecation Notices

The Common Vulnerability Reporting Framework (CVRF) format is now deprecated and no longer supported. See the CVRF compatibility FAQ. Users of this format should migrate to the Common Security Advisory Framework (CSAF) format.

OVAL v1 files are deprecated and no longer available. See the OVAL v1 deprecation announcement for more information. Users of this format should migrate to OVAL v2 (OVAL streams).

CHAPTER 2. CSAF

2.1. LIST ALL CSAFS

Abstract

Provides an index to all recent CSAF documents with a summary of their contents, when no parameter is passed. Returns a convenience object as the response with minimal attributes.

NOTE

See the Explaining Red Hat Errata article for more information on Red Hat Errata (RHSA, RHBA, and RHEA).

JSON

```
GET /csaf.json
```

XML

```
GET /csaf.xml
```

HTML

```
GET /csaf
```

2.2. PARAMETERS

Name	Description	Example
before	Index of CSAF documents before the query date. [ISO 8601 is the expected format]	2016-03-01
after	Index of CSAF documents after the query date. [ISO 8601 is the expected format]	2016-02-01
rhsa_ids	Index of CSAF documents for RHSA_IDs separated by comma	RHSA-2018:2748,RHSA-2018:2791
bug	Index of CSAF documents for Bugzilla Ids	1326598,1084875
cve	Index of CSAF documents for CVEs	CVE-2014-0160,CVE-2016-3990

Name	Description	Example
severity	Index of CSAF documents for severity	low,moderate,important,critical
package	Index of CSAF documents which affect package	samba,thunderbird
page	Index of CSAF documents for page number	Default: 1
per_page	Number of index of CSAF documents to return per page	Default: 1000
created_days_ago	Index of CSAF documents created days ago	10

> By default, search will return the first page of 1000 results, ordered by date. To change the page size use the 'per_page' param, and then iterate through pages using the 'page' param.

NOTE

All the above query parameters can be used in combination with each other to retrieve the desired result.

2.3. RETRIEVE A CSAF

Abstract

CSAF details for the RHSA.

JSON

CSAF documents are in JSON format; the XML view is a representation of the CSAF data in XML format.

```
GET /csaf/<RHSA_ID>.json
```

XML

```
GET /csaf/<RHSA_ID>.xml
```

Notes:

The JSON format for the Common Security Advisory Framework (CSAF) is defined by OASIS, see here for the GitHub repository and here for the schema.

For more information about the CSAF/VEX data provided by Red Hat see: https://www.redhat.com/en/blog/csaf-vex-documents-now-generally-available

Sample Query URLs

https://access.redhat.com/hydra/rest/securitydata/securitydata/csaf
https://access.redhat.com/hydra/rest/securitydata/securitydata/csaf.xml
https://access.redhat.com/hydra/rest/securitydata/securitydata/csaf.json

https://access.redhat.com/hydra/rest/securitydata/csaf.json?after=2023-09-01
https://access.redhat.com/hydra/rest/securitydata/csaf.json?created_days_ago=10
https://access.redhat.com/hydra/rest/securitydata/csaf.json?cve=CVE-2023-1829,CVE-2023-3090,CVE-2023-3390
https://access.redhat.com/hydra/rest/securitydata/csaf.json?rhsa_ids=RHSA-2022:6155,RHSA-2023:2378
https://access.redhat.com/hydra/rest/securitydata/csaf.json?bug=2053532,2153399
https://access.redhat.com/hydra/rest/securitydata/csaf.json?severity=important&created_days_ago=30
https://access.redhat.com/hydra/rest/securitydata/csaf.json?package=thunderbird

https://access.redhat.com/hydra/rest/securitydata/csaf/RHSA-2022:6155
https://access.redhat.com/hydra/rest/securitydata/csaf/RHSA-2022:6155.xml
https://access.redhat.com/hydra/rest/securitydata/csaf/RHSA-2022:6155.json

CHAPTER 3. CVE

3.1. LIST ALL CVES

Abstract

List all the recent CVEs when no parameter is passed. Returns a convenience object as response with very minimum attributes.

JSON

```
GET /cve.json
```

XML

```
GET /cve.xml
```

HTML

```
GET /cve
```

3.2. PARAMETERS

Name	Description	Example
before	CVEs before the query date. [ISO 8601 is the expected format]	2016-03-01
after	CVEs after the query date. [ISO 8601 is the expected format]	2016-02-01
ids	CVEs for Ids separated by comma	CVE-2017-8797,CVE-2014-0161
bug	CVEs for Bugzilla Ids	1326598,1084875
advisory	CVEs for advisory	RHSA-2016:0614,RHSA-2016:0610
severity	CVEs for severity	low,moderate,important
package	CVEs which affect the package	samba,thunderbird
product	CVEs which affect the product. The parameter supports Perl compatible regular expressions.	linux 7,openstack
cwe	CVEs with CWE	295,300

Name	Description	Example
cvss_score	CVEs with CVSS score greater than or equal to this value	7.0
cvss3_score	CVEs with CVSSv3 score greater than or equal to this value	7.0
page	CVEs for page number	Default: 1
per_page	Number of CVEs to return per page	Default: 1000
created_days_ago	Index of CVEs definitions created days ago	10
include_package_state	CVEs with package_state information	true, false

> By default, search will return the first page of 1000 results, ordered by date. To change the page size use the 'per_page' param, and then iterate through pages using the 'page' param.

NOTE

All the above query parameters can be used in combination with each other to retrieve the desired result.

3.3. RETRIEVE A CVE

Abstract

Retrieve full CVE details.

Path

```
GET /cve/<CVE>.json
```

Example: /cve/CVE-2016-3706.json

Returns a JSON representation of the CVE data for CVE-2016-3706.

3.4. CVE FORMAT

Abstract

Unlike CSAF or OVAL, the CVE representation is not a standard. Notes on what fields may exist and what they mean follow.

Name	Description	Additional Information
ThreatSeverity	The Severity of the flaw.	See this document for more information.
PublicDate	When the flaw became public.	ISO 8601 format.
Bugzilla	Id, URL, and Description of the bug in Red Hat's Bugzilla.	
CVSS	CVSSv2 score and metrics.	The 'status' attribute may have a value of 'draft' or 'verified', indicating how far along the investigation of the flaw has progressed. See this document for more information.
CVSS3	CVSSv3 score and metrics.	The 'status' attribute may have a value of 'draft' or 'verified', indicating how far along the investigation of the flaw has progressed. See this document for more information.
CWE	The CWE chain for this flaw.	See the mitre.org description and our list of possible cwe values.
Details	Details about the flaw, possibly from Red Hat or Mitre.	
Statement	A statement from Red Hat about the issue.	
References	Links to more information about the issue.	
Acknowledgements	People or organizations that are being recognized.	
Mitigation	A way to fix or reduce the problem without updated software.	

Name	Description	Additional Information
AffectedRelease	A released Erratum that fixes the flaw for a particular product.	Contains product name and CPE, and Erratum link, type, and release date. Optionally also includes "Package" information that describes the name and version of the src.rpm that fixes the issue (will not exist if multiple src.rpms are in the same Erratum).
PackageState	Information about a package / product where no fix has been released yet.	Contains product name and CPE, package (src.rpm) name, and fix state, which is one of ['Affected','Fix deferred','New','Not affected','Will not fix'].
UpstreamFix	The version of the upstream project that fixes the flaw.	

CHAPTER 4. OVALSTREAMS

4.1. LIST ALL OVAL STREAMS

Abstract

Provides an index to all OVAL stream files from where they can be downloaded. When no parameter is passed, returns a list of all OVAL stream files.

JSON

```
GET oval/ovalstreams.json
```

XML

```
GET oval/ovalstreams.xml
```

HTML

```
GET oval/ovalstreams
```

4.2. PARAMETERS

Name	Description	Example
after	Index of OVAL stream files modified after the query date. Expected format: ISO 8601.	2016-02-01
label	Index of OVAL stream files for a product version label.	jboss-eap-6

> By default, returned results are ordered by date.

NOTE

All the above query parameters can be used in combination with each other to retrieve the desired result.

4.3. RETRIEVE AN OVAL STREAM

Abstract

Returns the OVAL stream data for a product identified by base name.

JSON

OVAL stream files are in XML format; the JSON view is a representation of the OVAL data in JSON format.

```
GET oval/ovalstreams/<BASE>.json
```

Example: oval/ovalstreams/RHEL7.json

Returns a JSON representation of the OVAL streams for Red Hat Enterprise Linux 7.

XML

```
GET oval/ovalstreams/<BASE>.xml
```

NOTE

For more information about the OVAL format see the FAQ.

CHAPTER 5. EXAMPLE SCRIPT

```
#!/usr/bin/env python

import sys
import requests
from datetime import datetime, timedelta

API_HOST = 'https://access.redhat.com/hydra/rest/securitydata'

PROXIES = {}

# uncomment lines below to specify proxy server
# HTTPS_PROXY = "http://yourproxy.example.com:8000"
# PROXIES = { "https" : HTTPS_PROXY }

def get_data(query):

    full_query = API_HOST + query
    r = requests.get(full_query, proxies=PROXIES)

    if r.status_code != 200:
        print('ERROR: Invalid request; returned {} for the following '
              'query:\n{}'.format(r.status_code, full_query))
        sys.exit(1)

    if not r.json():
        print('No data returned with the following query:')
        print(full_query)
        sys.exit(0)

    return r.json()


# Get a list of issues and their impacts for RHSA-2022:1988
endpoint = '/cve.json'
params = 'advisory=RHSA-2022:1988'

data = get_data(endpoint + '?' + params)

for cve in data:
    print(cve['CVE'], cve['severity'])


print('-----')
# Get a list of kernel advisories for the last 30 days and display the
# packages that they provided.
endpoint = '/csaf.json'
date = datetime.now() - timedelta(days=30)
params = 'package=kernel&after=' + str(date.date())

data = get_data(endpoint + '?' + params)

kernel_advisories = []
for advisory in data:
    print(advisory['RHSA'], advisory['severity'], advisory['released_on'])
    print('-', '\n- '.join(advisory['released_packages']))
    kernel_advisories.append(advisory['RHSA'])


print('-----')
# From the list of advisories saved in the previous example (as
# `kernel_advisories`), get a list of affected products for each advisory.
endpoint = '/csaf/'

for advisory in kernel_advisories:
    data = get_data(endpoint + advisory + '.json')
    print(advisory)

    for product_branch in data['product_tree']['branches']:
        for inner_branch in product_branch['branches'][0]['branches']:
            print('-', inner_branch['name'])
```