Automation Support for CVE Retrieval

Bob Byers
Computer Security Division
Information Technology Laboratory

Harold Owen
Cocoasystems Inc.

September 27, 2019

National Institute of
Standards and Technology
U.S. Department of Commerce

Contents

Introduction

The National Vulnerability Database (NVD), https://nvd.nist.gov, allows government agencies, software vendors, and researchers to search and view information about vulnerabilities and vulnerable products. In the Fall of 2019, NVD began offering web services to allow computer applications to better access the NVD data. The purpose of this document is to describe how applications can interact with the NVD vulnerability web services, version 1.0.

Readers can sample the vulnerability services by entering one of the following URL into any web browser:

```
https://services.nvd.nist.gov/rest/json/cve/1.0/CVE-2015-5611

https://services.nvd.nist.gov/rest/json/cves/1.0?keyword=cherokee
```

This document is intended for application developers who need to consume the NVD data. It is assumed that the audience is generally familiar with JSON RESTful services. JSON specifies the format of the data returned by the REST service. REST refers to a style of services that allow computers to communicate via HTTP over the Internet.

Section 1 describes the REST parameters that allows you to control and customize which vulnerabilities are returned. The parameters are akin to those found on the NVD public vulnerability search page, https://nvd.nist.gov/vuln/search.

Section 2 describes the response. Each CVE has a text description and reference links. Vulnerabilities that have undergone NVD analysis include CVSS scores, product applicability statements, and more. Readers having experience with JSON may also refer to the response schema:

```
https://csrc.nist.gov/schema/nvd/feed/1.1/nvd_cve_feed_json_1.1.schema
```

The terms vulnerability and CVE are used interchangeably throughout this document. CVE means Common Vulnerability Enumeration, the standard for uniquely identifying vulnerabilities, e.g. CVE-2019-1234. For more information, visit https://cve.mitre.org/.

Additional services are available for retrieving information from the Official CPE Dictionary, https://nvd.nist.gov/products/cpe. Readers interested in these *product* web services should refer to the guide, *Automation Support for CPE Retrieval.*

Legacy Data Feeds

Historically the same information has been available programmatically via data feeds found at https://nvd.nist.gov/vuln/data-feeds. Consumers of these legacy feeds are encouraged to migrate *away* from the feeds and use the new services instead. Despite their popularity, the feeds required downloading and processing large files even when a subset of data would suffice. This was time consuming and a burden to networks. In contrast, the new services allow callers to specify query parameters to filter the response. For instance, you may be interested only in vulnerabilities for a certain time period, for specific products, and so on.

The format of the web service response is almost identical to that of the legacy feeds. Hence, code artifacts that you have for processing feed data may be reusable for the web services.

CVE Requests

NVD offers the following REST services to retrieve vulnerabilities. To retrieve *one specific* vulnerability, use:

HTTP Method:	GET
Content Type:	application/json
URL:	https://services.nvd.nist.gov/rest/json/cve/1.0/<cveId>

Parameters	Type	Description	Required?
cveId	URL path	CVE identifier.	Yes
addOns	URL query	See Include Official CPE Names.	No

The cveId parameter must be provided. The following URL illustrates this service.

```
https://services.nvd.nist.gov/rest/json/cve/1.0/CVE-2015-5611
```

To retrieve a *collection* of the latest vulnerabilities, use:

HTTP Method:	GET
Content Type:	application/json
URL:	https://services.nvd.nist.gov/rest/json/cves/1.0

Parameters	Type	Description	Required?
startIndex	URL query	See Paging Results.	No
resultsPerPage			
pubStartDate		CVE publication or modification date range.	
pubEndDate			
modStartDate			
modEndDate			
includeMatchStringChange		See CPE Name.	
keyword		Free text keyword search.	
isExactMatch		See Keyword Search.	
cweId		CVE by CWE category.	
cvssV2Severity		CVE having base severity score.	
cvssV3Severity			
cvssV2Metrics		CVE having CVSS vectors.	
cvssV3Metrics			
cpeMatchString		CVE product applicability.	
addOns		See Include Official CPE Names.	

All parameters are optional and are intended to limit or *filter* the results. The parameters you use is known collectively as your *search criteria*. The following URL illustrates a request with no search criteria.

```
https://services.nvd.nist.gov/rest/json/cves/1.0
```

Results are ordered by modification date, beginning with the most recently modified CVE.

For brevity, the host https://services.nvd.nist.gov/rest/json/ has been omitted from the remaining examples.

Paging Results

By default, the `/cves` service returns the most recent 20 CVE. The CVE returned can be thought of as a logical *page* of results. You can control which page of results is returned using the startIndex and resultsPerPage parameters. The startIndex parameter determines the first CVE in the response page. The index is zero-based, meaning the first CVE is at index zero.

The resultsPerPage parameter specifies the page size. The actual number of CVE in the page may be fewer depending on how many CVE matched your search criteria. For network considerations, the maximum allowable page size is limited to 5000 CVE.

The total number of results that match your search criteria is indicated in each response. From the total results and your page size, your application can compute the number of requests needed to retrieve all results. See In the examples that follow, a single colon (:) on a line by itself indicates where data has been omitted for clarity.

Total Results, below.

The following URL illustrates how to retrieve the second page of results.

```
cves/1.0?startIndex=20&resultsPerPage=20
```

Since the default page size is 20, the resultsPerPage is unnecessary in the example.

Retrieving All CVE

Presently NVD contains more than 120,000 vulnerabilities relating to thousands of vendor products. If your goal is to fetch all records, then multiple consecutive requests are required. For example, 120+ requests for 1,000 results per page. However, NIST firewall rules in place to prevent denial of service attacks on NVD can thwart your application. To avoid this, it is recommended that your application *sleeps* for several seconds between requests in order that your legitimate requests are not denied.

In addition, applications are discouraged from repeatedly requesting all the records every day. Rather, download all of them initially, then use *date range* parameters to retrieve new and recently modified records since your last request. See CVE by Date Range.

CVE by Date Range

Two pairs of optional date parameters allow you to retrieve vulnerabilities based on when they were added to or modified in NVD, respectively. Date parameters are in the form:

```
yyyy-MM-dd'T'HH:mm:ss:SSS z
```

The pubStartDate and pubEndDate parameters specify the set of CVE that were added to NVD (i.e., *published*) during the period. The modStartDate and modEndDate parameters specify CVE that were subsequently modified.

It is not necessary to provide both start and end dates if your goal is to retrieve all CVE *after* a certain date, or *up to* a certain date.

The following URL illustrates how to retrieve CVE modified after the start of 2019, EST.

```
cves/1.0?modStartDate=2019-01-01T00:00:00:000 UTC-05:00
```

CPE Name Changes

The modStartDate and modEndDate parameters filter results based on the modification date of the vulnerability record. However, the modification of product names by NIST in the Official CPE Dictionary is not said to *modify* related CVE. Hence, these parameters do not normally influence the search results when CPE names are modified or added.

If your goal is to retrieve vulnerabilities where CPE names changed during the time period, use includeMatchStringChange=true. This returns vulnerabilities where either the vulnerabilities *or* the associated product names were modified.

```
cves/1.0?modStartDate=2019-01-01 T00:00:00:000 UTC-
05:00&includeMatchStringChange=true
```

Keyword Search

The keyword parameter allows your application to retrieve records where a word or phrase is found in the vulnerability description or reference links.

```
cves/1.0?keyword=apple
```

Exact Match

If the keyword is a phrase, i.e., contains more than one term, then the isExactMatch parameter may be used to influence the response. Use isExactMatch=true to retrieve records matching the exact phrase. Otherwise, the results contain any record having any of the terms.

```
cves/1.0?keyword=denial+of+service&isExactMatch=true
```

CWE Search

CWE refers to the classification of vulnerabilities at https://cwe.mitre.org/. NIST staff associate one or more CWE to each vulnerability during the analysis process. In the following example, CWE-20 means vulnerabilities caused by Improper Input Validation. To filter search results based on CWE, use the cweId parameter.

```
cves/1.0?cweId=CWE-20
```

CVSS Search

CVSS refers to the scoring system used by NIST to assess the severity of vulnerabilities, https://www.first.org/cvss/. NVD provides base scores using the CVSS version 2 and, more recently, version 3.x.

Two pairs of parameters allow you to filter vulnerabilities based on CVSS base scores. Use either the cvssV2Severity or cvssV3Severity parameter to find vulnerabilities having a LOW, MEDIUM, or HIGH version 2 or 3.x score, respectively. For CVSS V3.x, cvssV3Severity=CRITICAL is also supported.

```
cves/1.0?cvssV2Severity=HIGH
```

If your application supports CVSS *vector strings*, use the cvssV2Metric or cvssV3Metrics parameter to find vulnerabilities having those score metrics.

```
cves/1.0?cvssV3Metrics=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
```

Partial vector strings are supported, such as:

```
cves/1.0?cvssV3Metrics=C:H/A:N
```

Product Applicability (CPE)

NVD analysts identify which product or products are affected by each vulnerability. The set of associated products is known as the *applicability statement* of the CVE. NVD uses the Common Platform Enumeration (CPE), version 2.3, to convey product vendors, names, versions, etc. For more information, see https://cpe.mitre.org/.

Use the cpeMatchString parameter to filter vulnerabilities based on product. It is beyond the scope of this document to explain the CPE syntax. However, a few examples are given here for illustration.

To find vulnerabilities for Microsoft Windows 10, use:

```
cves/1.0?cpeMatchString=cpe:2.3:o:microsoft:windows_10
```

To find vulnerabilities for version 1511 use:

```
cves/1.0?cpeMatchString=cpe:2.3:o:microsoft:windows_10:1511
```

To find all vulnerabilities associated with any Microsoft product, use:

```
cves/1.0?cpeMatchString=cpe:2.3:*:microsoft
```

Include Official CPE Names

By default, the response includes all CPE applicability statements associated with the vulnerability. It is important to note that applicability statements are *not* CPE names. Rather, they are CPE *match strings* that may be used in searching the Official CPE Dictionary.

Both the /cve and /cves services support an optional parameter addOns=dictionaryCpes allows you to control whether matching CPE names from the Official Dictionary are included in the response.

```
cve/1.0/CVE-2017-1029?addOns=dictionaryCpes

cves/1.0?addOns=dictionaryCpes
```

If this parameter is omitted, then only the applicability statements are returned. See Notice that totalResults indicates the total number of vulnerabilities that match your search criteria. This is useful in computing the number of requests needed to retrieve all matching pages. See Paging Results.

The result element matches the NVD Legacy Data Feeds as specified in the nvd_cve_feed_json_1.1.schema.

The CVE_Items element is the array of vulnerabilities (page of results), omitted here.

Vulnerability

At the high-level, each vulnerability (from the CVE_Items array) can have the following elements.

Element	Description	Required?
cve	CVE ID, description, reference links, CWE.	Yes
configurations	CPE applicability statements and optional CPE names.	No
impact	CVSS severity scores.	No
publishedDate	CVE publication date.	No
lastModifiedDate	CVE modified date.	No

The following illustrates the JSON structure of the vulnerability.

```
{
  "cve" : {
:
  },
  "configurations" : {
:
  },
  "impact" : {
:
    }
  },
  "publishedDate" : "2019-09-06T19:15Z",
  "lastModifiedDate" : "2019-09-10T14:25Z"
},
```

While the schema requires only the cve element, in practice each vulnerability has a publishedDate and lastModifiedDate.

The configurations and impact are provided only after the vulnerability has undergone analysis by NIST.

The remaining sections describe the main features of the cve, configurations, and impact elements.

CVE Element

The cve element conforms to CVE_JSON_4.0_min_1.1.schema. It contains the CVE identifier, description, reference links, and problem type (CWE).

CVE Identifier

The following illustrates part of a cve element. Notice the CVE identifier found within the CVE_data_meta element.

```
  "cve":{
    "data_type":"CVE",
      "data_format":"MITRE",
      "data_version":"4.0",
      "CVE_data_meta":{
        "ID":"CVE-2019-1010218",
        "ASSIGNER":"cve@mitre.org"
        },
:
  },
```

CVE Description

An example of a vulnerability description is shown here. In rare occasions the description_data element can contain multiple values.

```
  "cve":{
:
    "description":{
      "description_data":[{
        "lang":"en",
        "value":"Cherokee Webserver Latest Cherokee Web server Upto Version
1.2.103 (Current stable) is affected by: Buffer Overflow - CWE-120. The
impact is: Crash. The component is: Main cherokee command. The attack vector
is: Overwrite argv[0] to an insane length with execl. The fixed version is:
There's no fix yet."
      }]
    }
  },
```

CVE References

All vulnerabilities have at least one Internet link that provides additional information about the vulnerability. The references element contains those links. Note that NIST categorizes link using the tag elements, e.g., Third Party Advisory.


```
  "cve":{
:
      "references":{
        "reference_data":[{
          "url":"https://i.imgur.com/PWCCyir.png",
            "name":"https://i.imgur.com/PWCCyir.png",
            "refsource":"MISC",
            "tags":["Exploit","Third Party Advisory"]
        }]
      },
:
  },
```


CVE Problem Type (CWE)

All CWE assigned to a vulnerability are found in the problem_type element. In some cases there are more than one CWE.


```
  "cve":{
:
    "problemtype":{
      "problemtype_data":[{
        "description":[{
          "lang":"en",
          "value":"CWE-119"
        }]
      }]
    },
:
  },
```

Configurations and

```
"configurations" : {
      "CVE_data_version" : "4.0",
      "nodes" : [ {
        "operator" : "OR",
        "cpe_match" : [ {
          "vulnerable" : true,
          "cpe23Uri" :
"cpe:2.3:a:imapfilter_project:imapfilter:*:*:*:*:*:*:*:*",
          "versionEndIncluding" : "2.6.12"
        } ]
      } ]
    },
```

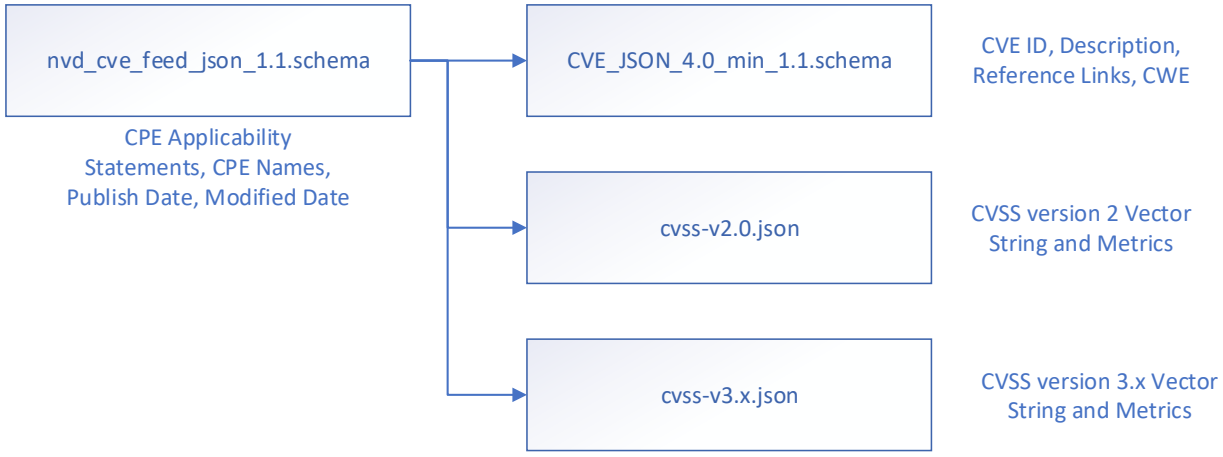
Other possible elements are versionEndExcluding, versionStartIncluding, and versionStartExcluding.

CPE Names in the response, described below.

For the /cves service, if you are fetching vulnerabilities based on the modification date range, and are including matching CPE names, consider using the optional includeMatchStringChange=true parameter described above in the CPE Name Changes section. This ensures vulnerabilities are returned when official names have changed even if the vulnerabilities themselves have not changed.

CVE Response

This section describes the response returned by the vulnerability services. It is worth pointing out that the response is based on four (4) JSON schema that were developed independently as part of three separate initiatives. Hence developers may notice stylistic differences in data element names. The following diagram shows that the main *feed* schema is dependent on the other three.



In the examples that follow, a single colon (:) on a line by itself indicates where data has been omitted for clarity.

Total Results

The following example illustrates the high-level response to a /cves/1.0 service request. Since the /cve (singular) service returns only one specific vulnerability, this does *not* apply.

```
{
  "resultsPerPage":20,
  "startIndex":0,
  "totalResults":15,
  "result": {
    "CVE_data_type":"CVE",
      "CVE_data_format":"MITRE",
      "CVE_data_version":"4.0",
      "CVE_data_timestamp":"2019-09-16T15:56Z",
      "CVE_Items":[
:
      ]
  }
}
```

Notice that totalResults indicates the total number of vulnerabilities that match your search criteria. This is useful in computing the number of requests needed to retrieve all matching pages. See Paging Results.

The result element matches the NVD Legacy Data Feeds as specified in the nvd_cve_feed_json_1.1.schema.

The CVE_Items element is the array of vulnerabilities (page of results), omitted here.

Vulnerability

At the high-level, each vulnerability (from the CVE_Items array) can have the following elements.

Element	Description	Required?
cve	CVE ID, description, reference links, CWE.	Yes
configurations	CPE applicability statements and optional CPE names.	No
impact	CVSS severity scores.	No
publishedDate	CVE publication date.	No
lastModifiedDate	CVE modified date.	No

The following illustrates the JSON structure of the vulnerability.

```
{
  "cve" : {
:
  },
  "configurations" : {
:
  },
  "impact" : {
:
    }
  },
  "publishedDate" : "2019-09-06T19:15Z",
  "lastModifiedDate" : "2019-09-10T14:25Z"
},
```

While the schema requires only the cve element, in practice each vulnerability has a publishedDate and lastModifiedDate.

The configurations and impact are provided only after the vulnerability has undergone analysis by NIST.

The remaining sections describe the main features of the cve, configurations, and impact elements.

CVE Element

The cve element conforms to CVE_JSON_4.0_min_1.1.schema. It contains the CVE identifier, description, reference links, and problem type (CWE).

CVE Identifier

The following illustrates part of a cve element. Notice the CVE identifier found within the CVE_data_meta element.

```
  "cve":{
    "data_type":"CVE",
      "data_format":"MITRE",
      "data_version":"4.0",
      "CVE_data_meta":{
        "ID":"CVE-2019-1010218",
        "ASSIGNER":"cve@mitre.org"
        },
:
  },
```

CVE Description

An example of a vulnerability description is shown here. In rare occasions the description_data element can contain multiple values.

```
  "cve":{
:
    "description":{
      "description_data":[{
        "lang":"en",
        "value":"Cherokee Webserver Latest Cherokee Web server Upto Version
1.2.103 (Current stable) is affected by: Buffer Overflow - CWE-120. The
impact is: Crash. The component is: Main cherokee command. The attack vector
is: Overwrite argv[0] to an insane length with execl. The fixed version is:
There's no fix yet."
      }]
    }
  },
```

CVE References

All vulnerabilities have at least one Internet link that provides additional information about the vulnerability. The references element contains those links. Note that NIST categorizes link using the tag elements, e.g., Third Party Advisory.

```
  "cve":{
:
      "references":{
        "reference_data":[{
          "url":"https://i.imgur.com/PWCCyir.png",
            "name":"https://i.imgur.com/PWCCyir.png",
            "refsource":"MISC",
            "tags":["Exploit","Third Party Advisory"]
        }]
      },
:
  },
```

CVE Problem Type (CWE)

All CWE assigned to a vulnerability are found in the problem_type element. In some cases there are more than one CWE.

```
  "cve":{
:
    "problemtype":{
      "problemtype_data":[{
        "description":[{
          "lang":"en",
          "value":"CWE-119"
        }]
      }]
    },
:
  },
```

Configurations Element

The configurations element has the CVE *applicability statements* that convey which product, or products, are associated with the vulnerability according to analysis by NIST. Recall that each CPE shown here is a *match string* that can be used to search the Official CPE Dictionary.

The configurations element conforms to the nvd_cve_feed_json_1.1.schema.

Configurations are a *tree*, or hierarchical data structure consisting of nodes where each node contains CPE match string or child nodes. (A node will never contain both CPEs and child nodes, and is never empty.)

Each node has either an OR- or an AND-operator (and in rare cases a NEGATE flag) to covey the logical relationship of the CPE or child nodes within. For example, if the vulnerability exists only when *both* CPE products are present, the operator is "AND". If the vulnerability exists if *either* CPE is present, then the operator is "OR".

The following is an example of a configuration with two child nodes, each having one product.

```
"configurations":{
  "CVE_data_version":"4.0",
  "nodes":[
    {
        "operator":"AND",
        "children":[
          {
              "operator":"OR",
              "cpe_match":[
                {
                    "vulnerable":true,
                    "cpe23Uri":"cpe:2.3:o:tesla:model_3_firmware:-
:*:*:*:*:*:*:*"
                }]
          },{
              "operator":"OR",
              "cpe_match":[
              {
                "vulnerable":false,
                "cpe23Uri":"cpe:2.3:h:tesla:model_3:-:*:*:*:*:*:*:*"
              }]
            }
        ]
  }]
},
```

Notice that the first product is marked as vulnerable, but the second is not. (The vulnerability is said to exist only if the firmware in this example is *running on* the hardware.)

Configurations vary in complexity, partly due to their recursive nature. Some vulnerabilities have one node with one CPE, while others have more than one configuration, i.e., more than one root node element. Nodes may contain one CPE match string or dozens.

Version Ranges

In some cases, the CPE match string indicates a range of product versions. Notice in the following example that the version is *not* specified in the cpe23Uri element; instead, the versionEndIncluding indicates the last vulnerable version.

```
    "configurations" : {
      "CVE_data_version" : "4.0",
      "nodes" : [ {
        "operator" : "OR",
        "cpe_match" : [ {
          "vulnerable" : true,
          "cpe23Uri" :
"cpe:2.3:a:imapfilter_project:imapfilter:*:*:*:*:*:*:*:*",
          "versionEndIncluding" : "2.6.12"
        } ]
      } ]
    },
```

Other possible elements are versionEndExcluding, versionStartIncluding, and versionStartExcluding.

CPE Names

Recall that the vulnerability service has an optional query parameter, addOns=dictionaryCpes, described above. When the request has this parameter, the response returns official CPE names for each CPE match string in the configuration, in so far as they are present in the Official CPE Dictionary.

The following example shows matching CPE names for a match string.

```
"configurations":{
  "CVE_data_version":"4.0",
  "nodes":[{
    "operator":"OR",
      "negate":false,
      "cpe_match":[{
        "vulnerable":true,
        "cpe23Uri":"cpe:2.3:a:elementor:elementor:*:*:*:*:*:*:*:*",
        "versionEndExcluding":"1.8.0",
        "cpe_name":[{
          "cpe23Uri":"cpe:2.3:a:elementor:elementor:-:*:*:*:*:*:*:*",
            "lastModifiedDate":"2019-09-10T15:38Z"},
            {"cpe23Uri":"cpe:2.3:a:elementor:elementor:0.1.0:*:*:*:*:*:*:*",
            "lastModifiedDate":"2019-09-10T15:38Z"},
            {"cpe23Uri":"cpe:2.3:a:elementor:elementor:0.1.1:*:*:*:*:*:*:*",
            "lastModifiedDate":"2019-09-10T15:38Z"},
            {"cpe23Uri":"cpe:2.3:a:elementor:elementor:0.1.2:*:*:*:*:*:*:*",
            "lastModifiedDate":"2019-09-10T15:38Z"},
            {"cpe23Uri":"cpe:2.3:a:elementor:elementor:0.1.3:*:*:*:*:*:*:*",
            "lastModifiedDate":"2019-09-10T15:38Z"},
:
```

Since configurations can be large, and the number of matches can be many, applications are cautioned from using this parameter for requests that return large numbers of vulnerabilities.

Impact Element

The impact element provides the CVSS severity scores for the vulnerability if it has been analyzed by NIST.

The cvssV3 and cvssV2 elements within the impact element conform to the cvss-v3.x.json and cvss-v2.0.json schemas, respectively. Additional elements provided by NIST conform to the parent nvd_cve_feed_json_1.1.schema such as the exploitability and impact sub-scores.

CVSS V3.x

The following is an example CVSS V3.0 score returned by the vulnerability service. In more recent cases the version is 3.1.


```
"impact":{
  "baseMetricV3":{
    "cvssV3":{
        "version":"3.0",
        "vectorString":"CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H",
        "attackVector":"NETWORK",
        "attackComplexity":"LOW",
        "privilegesRequired":"NONE",
        "userInteraction":"REQUIRED",
        "scope":"UNCHANGED",
        "confidentialityImpact":"HIGH",
        "integrityImpact":"HIGH",
        "availabilityImpact":"HIGH",
        "baseScore":8.8,
        "baseSeverity":"HIGH"
        },
      "exploitabilityScore":2.8,
      "impactScore":5.9
  },
:
}
```

CVSS V2.0

The following is an example CVSS V2.0 score returned by the vulnerability service. Note that, in addition to the exploitability and impact sub-scores, the V2.0 score has additional metadata, including the severity.


```
"impact":{
:
  "baseMetricV2":{
    "cvssV2":{
      "version":"2.0",
      "vectorString":"AV:N/AC:M/Au:N/C:P/I:P/A:P",
      "accessVector":"NETWORK",
      "accessComplexity":"MEDIUM",
      "authentication":"NONE",
      "confidentialityImpact":"PARTIAL",
      "integrityImpact":"PARTIAL",
      "availabilityImpact":"PARTIAL",
      "baseScore":6.8
    },
    "severity":"MEDIUM",
    "exploitabilityScore":8.6,
    "impactScore":6.4,
    "obtainAllPrivilege":false,
    "obtainUserPrivilege":false,
    "obtainOtherPrivilege":false,
    "userInteractionRequired":true
  }
}
```