Vulnerability Exploitability eXchange (VEX) – Use Cases

Publication date: April 2022

Abstract

This document provides the recommended minimum data elements of a Vulnerability Exploitability eXchange (VEX) document and offers the reader a set of scenarios with proposed implementations. A VEX document is a form of a security advisory that indicates whether a product or products are affected by a known vulnerability or vulnerabilities. Further work will be needed to build out additional use cases to help users understand how to successfully build VEX documents of varying complexity.

1.0 Introduction

The goal of Vulnerability Exploitability eXchange (VEX) is to allow a software supplier or other parties to assert the status of specific vulnerabilities in a particular product[1]. This document is part of a series of descriptions and guidance documents for VEX.[2]

VEX is a form of a security advisory, similar to those already issued by mature product security teams today. There are a few important improvements for the VEX model over 'traditional' security advisories. First, VEX documents are machine readable, built to support integration into existing and novel security management tools, as well as broader vulnerability tracking platforms. Second, VEX data can support more effective use of Software Bills of Materials (SBOM) data. The ultimate goal of this document is to support greater automation across the vulnerability ecosystem, including disclosure, vulnerability tracking, and remediation.

As a novel, machine-readable mechanism, VEX allows both suppliers and users to focus on vulnerabilities that pose the most immediate risk, while not investing time in searching for or patching vulnerabilities that are not exploitable and therefore have no impact. With that being

[1] The OASIS Common Security Advisory Framework (CSAF) defines 'product' as "any deliverable (e.g., software, hardware, specification, …) which can be referred to with a name. This applies regardless of the origin, the license model, or the mode of distribution of the deliverable." A product comes from a supplier, and can be thought of as referring to any supplied software.

[2] See https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf for an initial overview. More information will be available at https://www.cisa.gov/sbom.

said, the VEX product statuses are not intended to be a discussion-ending declaration but a way to empower consumers to make informed decisions.

This document is meant to give guidance on what constitutes a VEX document. The document offers a set of scenarios that a product supplier may wish to address, starting with simple examples and moving to more complex scenarios. Depending on the supplier and situation, the amount and complexity of information conveyed in a VEX document may vary. It is important to have clarity around how this data should be conveyed and implemented, to support automated consumption of the advisories.

The use cases described below offer several different ways that an organization can structure and organize its published VEX documents. This document does not explicitly offer recommendations about the optimal organization. Suppliers are encouraged to work with their customers and the vendor community to identify the approach that meets respective needs.

2.0 Minimum Data Elements

A VEX document is a binding of product information, vulnerability information, and the relevant status details relating them. Minimum data elements of a VEX document must include the VEX metadata, product details, vulnerability details, and product status.

- **VEX metadata** must include: VEX Format Identifier, Identifier string for the VEX document, Author, Author role, Timestamp.
- **Product details** must include: Product identifier(s) or Product family identifier (e.g., unique identifier or a combination of Supplier name, product name, and version string, as laid out in established SBOM guidance[3]).
- **Vulnerability details** must include: Identifier of the Vulnerability (CVE or other identifier) and vulnerability description (e.g. CVE description).
- **Product Status details** (i.e., status information about a vulnerability in that product) must be from the following list:
    - NOT AFFECTED – No remediation is required regarding this vulnerability.
    - AFFECTED – Actions are recommended to remediate or address this vulnerability.
    - FIXED – These product versions contain a fix for the vulnerability.
    - UNDER INVESTIGATION – It is not yet known whether these product versions are affected by the vulnerability. An update will be provided in a later release.

[3] See documents at https://ntia.gov/SBOM for more information, particularly *Framing Software Component Transparency: Establishing a Common Software Bill of Materials*.

If a status is AFFECTED, the VEX document must have an action statement that tells the product user what to do. If the status is NOT AFFECTED, then a VEX document must have an impact statement to further explain details.

Various other details, such as CVSS scores or links to other resources can be added to a VEX document to add value for the customer. This document introduces the minimum required fields for each use case. Status information in the product status field is introduced in the use case found in section 3.1.1.

3.0 VEX Use Cases

The use cases below differ in how they address Product (or Product Line), Version, Vulnerability and Status. This table summarizes the use cases.

Use Case	Product		Product Line	Version			Vulnerability		Status	
	Single	Multiple	Multiple	Single	Multiple	All	Single	Multiple	Single	Multiple
3.1.1	x			x			x		x	
3.2.2	x			x				x	x	
3.2.3	x			x				x		x
3.2.4	x				x		x		x	
3.2.5	x					x	x		x	
3.2.6	x				x		x			x
3.3.2		x			x		x			x
3.4.2		x			x			x		x
3.4.3			x					x		x

3.1 Single Product, Single Vulnerability

3.1.1 *Single Product, Single Version, Single Vulnerability, Single Status*

This use case is the equivalent to a simple security advisory with only one vulnerability. The company makes statements about each version of its product in a different VEX document. For a given version of a given product, a particular vulnerability can only have a single status.

Example Company was informed about the security vulnerability Log4j with its associated CVE-2021-44228. The 4 potential VEX statuses are introduced with an example of each:

- **NOT AFFECTED**: Example Company has a product ABC. When the first Log4Shell vulnerability (CVE-2021-44228) was disclosed, the Product Security Incident Response Team (PSIRT) of Example Company released a VEX document stating that product ABC in version 4.2 is not affected. Example Company made this assertion because the class with the vulnerable code was removed before shipping.

    - CSAF example
    - CycloneDX example

  Note: It is required to include an impact statement in order to tell the consumer why the product is not affected. In this example, the statement is that the vulnerable library's "class with the vulnerable code was removed before shipping."

- **AFFECTED**: Example Company has a product DEF which uses a vulnerable version of the Log4j library in DEF's version 1.0. When the first Log4Shell vulnerability (CVE-2021-44228) was disclosed, the PSIRT of Example Company released a VEX document stating that product DEF is affected and customers should update to version 1.1 of product DEF.

    - CSAF example
    - CycloneDX example

  Note: It is required to include an action statement in order to tell consumers what they should do. In this example, the action statement is "customers should update to version 1.1 of product DEF, which fixes the issue."

- **UNDER INVESTIGATION**: Example Company has a product GHI. When the first Log4Shell vulnerability (CVE-2021-44228) was disclosed, the PSIRT of Example Company released a VEX document stating that GHI's version 17.4 is currently being investigated as to whether it is affected by CVE-2021-44228.

    - CSAF example
    - CycloneDX example

  Note: It is expected that the VEX document will be updated with the result of the analysis at some later point in time.

- **FIXED**: Example Company has a product DEF which uses the Log4j library in version 1.0. When the first Log4Shell vulnerability (CVE-2021-44228) was disclosed, the PSIRT of Example Company released an update and a VEX document stating that product DEF version 1.1 is fixed.
  - CSAF example
  - CycloneDX example

  NOTE: This status is mostly used in combination with AFFECTED (i.e., a previous version was affected, the current one is fixed) but can also be used as a standalone status.

3.2 Use Cases for Single Product, Multiple Vulnerabilities

3.2.1 General

A VEX document can reference multiple vulnerabilities. An example of a single product affected by multiple vulnerabilities would be the Ripple20 vulnerabilities.[4] Ripple20 has been described as a set of 19 vulnerabilities in a software library developed by Treck, Inc., and an organization may have to issue an advisory for them at the same time. A company using this software library in a product would need to clarify whether and how these vulnerabilities affect the product. The information provided on Ripple20 vulnerabilities is used in use cases found in sections 3.2.2 and 3.2.3.

3.2.2 *Single Product, Single Version, Multiple Vulnerabilities, Single Status*

In this use case, Example Company has fielded the product ABC with the current version 4.2. The company makes statements about each version of its product in a different VEX document. For a given version of a given product, a particular vulnerability can only have a single status.

When the Ripple20 vulnerabilities were disclosed, customers asked Example Company whether the current version 4.2 of its product ABC was affected. Example Company's PSIRT released a VEX document stating that product ABC version 4.2 is not affected, because it does not use the vulnerable stack.

- CSAF example
- CycloneDX example

[4] See https://www.cisa.gov/uscert/ncas/current-activity/2020/06/16/ripple20-vulnerabilities-affecting-treck-ip-stacks for more information.

3.2.3 *Single Product, Single Version, Multiple Vulnerabilities, Multiple Statuses*

In this use case, Example Company has fielded product GHI. The company makes statements about each version of its product in a different VEX document. For a given version of a given product, a particular vulnerability instance can only have a single status. However, other instances of the same or different vulnerabilities may have different statuses.

Product GHI's TCP/IP-Stack is based on Treck's stack, with some custom implementation and modifications. When Ripple20 was disclosed, Example Company's PSIRT released a VEX document stating that product GHI's version 17.4 is not affected by some of the Ripple20 vulnerabilities (namely CVE-2020-11897, CVE-2020-11902, CVE-2020-11899, CVE-2020-11905, CVE-2020-11906, CVE-2020-11913), but that it is affected by certain others (CVE-2020-11898, CVE-2020-11907, CVE-2020-11909, CVE-2020-11910, CVE-2020-11911). Moreover, others are still under investigation (CVE-2020-11896, CVE-2020-11900, CVE-2020-11904, CVE-2020-11903, CVE-2020-11908) and some are fixed already (CVE-2020-11901, CVE-2020-11912, CVE-2020-11914).

- ○ CSAF example
- ○ CycloneDX example

3.2.4 *Single Product, Multiple versions, Single Vulnerability, Single Status*

In this use case, Example Company has fielded product ABC and provided updates or otherwise updated it over time, so that there are multiple versions of the software. Different types of software or suppliers of software track versions and distributions differently.

The company makes statements about each version of its product in a single VEX document.

The new Log4j vulnerability, with associated CVE-2021-44228, has been identified in a component of product ABC. Example Company prepares a VEX document to inform customers that the vulnerability is exploitable (status: KNOWN_AFFECTED) in product ABC's versions 2.4, 2.6, and all versions between and including 2.9 through version 4.1. There are 90 separate product versions within that range, including minor versions and bug fixes.

- ○ CSAF example
- ○ CycloneDX example

NOTE: Multiple versions can be communicated in two ways within a VEX document. Each version can be called out in enumeration (e.g., 2.9, 3.0) or a range (e.g., "vers:generic/>=2.9|<=4.1"). An additional example of a range could be "all versions before 2.0 are AFFECTED." Both range and enumeration are valid approaches.

However, complex version ranges can often be more complex to parse. To reduce the effort on the user's side, it is recommended to explicitly enumerate versions wherever possible. The version range specifier tries to tackle the problem but can only solve that for known version schemes.

3.2.5 *Single Products, All Versions, Single Vulnerability, Single Status*

In this use case, Example Company has fielded product XYZ. The company makes statements about all versions of product XYZ in a single VEX document.

When the Log4j vulnerability with associated CVE-2021-44228 was disclosed, the Example Company's PSIRT released a VEX document stating that all of XYZ's versions are currently being investigated (status: UNDER INVESTIGATION) as to whether they are affected by CVE-2021-44228.

- CSAF example
- CycloneDX example

NOTE: A VEX document author might state that all versions have a certain status. It might be useful to convey a message such as "XYZ is not affected as it is a Rust program - it does not use Java at all and therefore cannot be affected." However, a misperception can be created by doing so as this includes future versions. If the company chooses to reimplement XYZ and use Java in a later version, this can become problematic if the company does not update the VEX document as the statement is not bound to the current latest version when the statement was made. In combination with the status under investigation, the author must be cautious. All versions means all versions and therefore can also include versions which are already End of Life.

3.2.6 *Single Product, Multiple versions, Single Vulnerability, Multiple Statuses*

In this use case, Example Company has fielded product ABC and provided updates, or otherwise updated it over time, so that there are multiple versions of the software. The new Log4j vulnerability, with associated CVE-2021-44228, has been identified in a component of product ABC.

Example Company concludes that some versions of product ABC are affected by CVE-2021-44228 (see use case found in section 3.2.4). Example Company also concludes that there are some versions of product ABC that are not impacted by CVE-2021-44228. Example Company prepares a VEX document to communicate that the vulnerability is exploitable (status: KNOWN_AFFECTED) in versions 2.4, 2.6, and all versions from and including 2.9 to 4.1, but also not exploitable (status: KNOWN_NOT_AFFECTED) in versions 1.0 to 2.3, 2.5, and 2.7 to 2.8.

- CSAF example
- CycloneDX example

3.3 Use Cases for Multiple Products, Single Vulnerability

3.3.1 General

A VEX document can refer to multiple products. An example of multiple products affected by a single vulnerability is the Log4j vulnerability, CVE-2021-44228, released in 2021. A company utilizing the Log4j software library may choose to create a VEX document containing all of its affected products rather than one VEX document for each product.

Naming software products is an ongoing problem and this document does not propose to resolve this issue.[5] For a given product with an identifier, different versions may be affected differently. However, there are multiple practices, including multiple "best practices" for describing the version of a given product. This paper describes several approaches to addressing these best practices below. Note that this problem expands beyond the challenge of security advisories, but if a single model is not realistic, then a scalable machine-readable approach to advisories will require some predictability.

3.3.2 *Multiple Products, Multiple Versions, Single Vulnerability, Multiple Statuses*

In this use case, Example Company has fielded products ABC and JKL and provided updates over time, so that there are multiple versions software. The new Log4j vulnerability, with associated CVE-2021-44228, has been identified in components of products ABC and JKL.

To make it easier for its customers to download and distribute VEX material, Example Company decides to communicate the exploitability status of products ABC and JKL in a single VEX document as opposed to creating two separate VEX documents. Example Company prepares a VEX document to communicate the exploitability of CVE-2021-44228 in product ABC as before (see use case found in section 3.2.6).

The VEX document also includes the assertion that the vulnerability disclosed in CVE-2021-44228 is exploitable (status: KNOWN_AFFECTED) in versions 4.5 to 5.0 of product JKL. Moreover, CVE-2021-44228 is not exploitable (status: KNOWN_NOT_AFFECTED) in product JKL versions 1.0 to 4.4. The mitigation for all versions of product JKL is to upgrade to the new version 5.1, in which the status of CVE-2021-44228 is FIXED.

[5] More information on the software identification challenge, along with guidance, is available at: https://ntia.gov/files/ntia/publications/ntia_sbom_software_identity-2021mar30.pdf.

- CSAF example
- CycloneDX example

3.4 Multiple Products, Multiple Vulnerabilities

3.4.1 General

An example of multiple products, multiple vulnerabilities can be based on the subsequent Log4j vulnerability disclosures (CVE-2021-45046, CVE-2021-45105, CVE-2021-44832) weeks after the initial Apache Log4j CVE-2021-44228 was released. A company utilizing the Log4j software library might develop one VEX document that addresses the status of all of these vulnerabilities in a number of products, as opposed to creating a separate VEX document for every new CVE. This is because one VEX document containing vulnerability exploit information for all Log4j vulnerabilities might be easier for downstream consumers to receive, process, and distribute than a stream of asynchronous VEX document releases with each document being produced for a different Log4j vulnerability.

A supplier might also choose to provide VEX documents that are focused on particular products or product lines. If the supplier follows this practice, a user that only has a few of the supplier's products would not have to go through every VEX document produced by the supplier, on the off chance that it might provide information about one of their products.

3.4.2 *Multiple Products, Multiple Versions, Multiple Vulnerabilities, Multiple Statuses*

In this use case, the new Log4j vulnerability CVE-2021-44228, and the subsequent vulnerability CVE-2021-45105[6] have been identified in components of products ABC and JKL.

Two weeks after CVE-2021-44228, the initial Log4j CVE is released, an additional Log4j vulnerability with associated CVE-2021-45105 is released. Example Company produces a VEX document stating that the vulnerability disclosed in CVE-2021-44228 is exploitable (status: KNOWN_AFFECTED) in products ABC versions 2.4, 2.6, 2.9 to 4.1, as well as JKL versions 4.5 to 5.0. The VEX document also states that the vulnerability disclosed in CVE-2021-45105 is exploitable (status: KNOWN_AFFECTED) in the same versions of products ABC and JKL. JKL version 5.1 is also affected by CVE-2021-45105 and therefore the mitigation is now to upgrade to version 5.2. Other mitigations remain the same.

- CSAF example
- CycloneDX example

[6] The authors acknowledge that CVE-2021-45046 was discovered before CVE-2021-45105. For the examples, CVE-2021-45105 will be used.

3.4.3 *Multiple Product Lines, Multiple Vulnerabilities, Multiple Statuses*

There are instances where vendors group products into product lines. For example, MicroLogix 1400 is a product line of programmable logic controllers offered by Rockwell Automation, Inc., that includes products such as 1766-L32BWA and 1766-L32BXBA. Rather than call out each of those product numbers individually, some vendors may choose to state the exploitability of the entire product line, for one or more vulnerabilities.

Example Company has a product line of Ethernet switches: PROD_ALPHA, which includes products MNO, PQR, and STU, as well as a product line of remote terminal units, PROD_BETA, which includes products VWX and YZA. When CVE-2021-44228 and CVE-2021-45105 are released for the Log4j vulnerabilities, Example Company produces a VEX document stating that these vulnerabilities are not exploitable (status: KNOWN_NOT_AFFECTED) in any of the products within product line PROD_ALPHA, but are exploitable (status: KNOWN_AFFECTED) in all products within PROD_BETA. Example Company decides to communicate that these entire product lines are affected/not affected as opposed to communicating each product individually.

- CSAF example

NOTE: Although this use case is valid, caution should be taken. Automated systems perform poorly when they need to infer what products are within a product line unless that information is made machine processable and is available elsewhere. Moreover, users may not know exactly which products are in which product line.

4.0 Next steps for VEX

Further work will continue to refine the VEX specification, offer practical guidance, and discuss implementations. One area of interest is to specify more detail for machine-readable information about status. The community will also better explore the operationalization of VEX, and work with the implementing standards. More work is also needed to understand the transport and data management side of VEX implementation and use.

More information about SBOM, broadly, is available at https://cisa.gov/SBOM. To learn more about SBOM and VEX, and join the VEX working discussion, please reach out to SBOM@cisa.dhs.gov.

Appendix A: About this document

This document was a product of the VEX Working Group, which grew out of the NTIA Multistakeholder Process and the Framing Working Group, initially beginning work in 2020. That work continued into 2022, facilitated by CISA.

Participants included:

Allan Friedman, CISA
Bruce Lowenthal, Oracle Corporation
Bryan Cowan, Fortress Information Security
Charlie Hart, Hitachi America Ltd.
Derek Kruszewski, aDolus Technology Inc.
Dmitry Raidman, Cybeats
Duncan Sparrell, sFractal Consulting
Ed Heierman, Abbott
Eliot Lear, Cisco Systems
Dr. Hans-Martin von Stockhausen, Siemens Healthineers
Jeremiah Stoddard, INL
Jim Jacobson, Siemens Healthineers
Josh Bressers, Anchore
Justin Murphy, CISA
Kate Stewart, The Linux Foundation
Matthew Paulsen, Juniper Networks, Inc.
Michael Hoover, INL
Nisha Kumar, Oracle
Rich Steenwyk, GE Healthcare
Steve Springett, OWASP
Thomas Schmidt, Federal Office for Information Security (BSI) Germany
Timothy Klett, INL
Timothy Walsh CISSP, Mayo Clinic
Tom Alrich, Tom Alrich LLC
Tony Turner, Fortress Information Security

Others participated, but do not wish to be named. Input into this document and the broader VEX effort included feedback from multiple presentations, with a particular appreciation for feedback from the Healthcare SBOM PoC community.