TECHNICAL WHITE PAPER:
March 2024

Enabling efficient upstream vulnerability assessment with VEX, SBoMs, and CVE scan results

vmware® by Broadcom

Table of contents

Introduction

While building modern applications, developers often use third-party open source software (OSS) components as building blocks, which brings flexibility and efficiency to their software development processes. On the other hand, however, these applications end up being a highly complex mixture of components and dependencies that are often beyond the control and visibility of security teams and platform engineers. The Log4Shell vulnerability demonstrates how a supply chain vulnerability can put your entire environment at risk if you are not able to quickly identify where the affected dependencies are. So understandably, most businesses are still skeptical about using open source software for mission-critical production use cases as they do not want to fall prey to the ever-increasing software supply chain vulnerabilities.

Software bill of materials (SBoM) is beginning to play an increasingly prominent role in mitigating the supply chain risks posed by OSS. They provide a comprehensive inventory of all ingredients that make up a software application, shedding light on critical details like a software's composition, dependencies, provenance, etc. In the ever-evolving landscape of OSS, this transparency is invaluable. Having a list of components with extensive information about their origin is a good beginning, but oftentimes, SBoMs alone may not be enough to efficiently manage supply chain vulnerabilities.

Common vulnerabilities and exposures (CVE) scan reports are an essential tool for OSS users as they provide a list of vulnerabilities that could expose their products and platforms to known risks. However, the drawback is that the mere act of scanning an OSS application often generates a long list of vulnerabilities, which can be overwhelming, given that not all vulnerabilities in a CVE scan report are always exploitable. Oftentimes, a given vulnerability can be exploited only under certain configurations or circumstances. So, the mention of a vulnerability in your CVE scan report does not automatically mean that your software is at risk. Furthermore, CVE notices are difficult to read without some level of expertise in cybersecurity. Thus, it becomes important to have some kind of support documentation that could help you understand the exploitability of the vulnerabilities in your CVE scan reports. That is exactly what Vulnerability Exploitability eXchange (VEX) documents do.

The National Telecommunications and Information Administration (NTIA) of the U.S. Department of Commerce describes VEX as a companion document that lives side by side with SBOM and CVE scan results. VEX leverages the strategic value of SBoM and CVE scan reports by providing context to vulnerabilities and clarity of potential remediation measures. Thus, SecOps teams can reduce the noise and prioritize their efforts to remediate what matters the most.

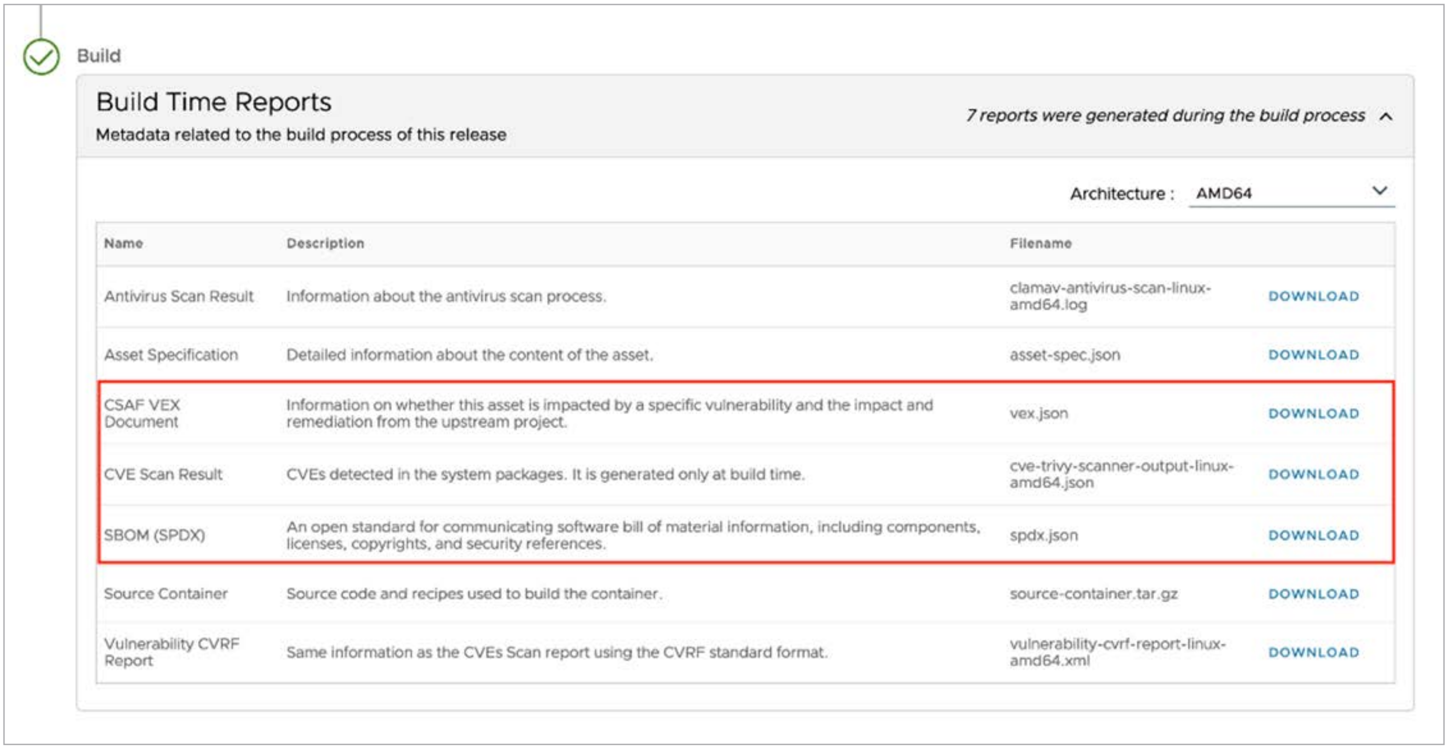
In this whitepaper, you will learn how Tanzu Application Catalog (formerly known as VMware Application Catalog) supports its users with SBoMs, CVE scan reports, and VEX documentation, how they complement each other, and how you can leverage the accurate risk assessment provided by VEX statements to reduce noise in your Trivy CVE report.

Differences between SBoMs, CVE scan reports, and VEX documents

The OSS applications in Tanzu Application Catalog undergo a thorough validation process before they are pushed to customers' private registries. As a result, users receive multiple build time reports through the user interface. These reports provide information on functional and integration tests, as well as crucial metadata related to the asset's build process. (For more details on these reports, please refer to our documentation).

Amongst these build time reports, you will find three key documents:

1. SBoM (in software package data eXchange (SPDX) format) report
2. CVE scan result report
3. Common Security Advisory Framework (CSAF) VEX document



Build Time Reports

Metadata related to the build process of this release

7 reports were generated during the build process

Architecture : AMD64

Name	Description	Filename	
Antivirus Scan Result	Information about the antivirus scan process.	clamav-antivirus-scan-linux-amd64.log	DOWNLOAD
Asset Specification	Detailed information about the content of the asset.	asset-spec.json	DOWNLOAD
CSAF VEX Document	Information on whether this asset is impacted by a specific vulnerability and the impact and remediation from the upstream project.	vex.json	DOWNLOAD
CVE Scan Result	CVEs detected in the system packages. It is generated only at build time.	cve-trivy-scanner-output-linux-amd64.json	DOWNLOAD
SBOM (SPDX)	An open standard for communicating software bill of material information, including components, licenses, copyrights, and security references.	spdx.json	DOWNLOAD
Source Container	Source code and recipes used to build the container.	source-container.tar.gz	DOWNLOAD
Vulnerability CVRF Report	Same information as the CVEs Scan report using the CVRF standard format.	vulnerability-cvrf-report-linux-amd64.xml	DOWNLOAD

The UI of Tanzu Application Catalog shows the Build Time Reports section with the CSAF VEX document, CVE Scan Result ,and SBoM (SPDX) reports highlighted.

Let's look at each of these reports in detail to better understand how they complement each other.

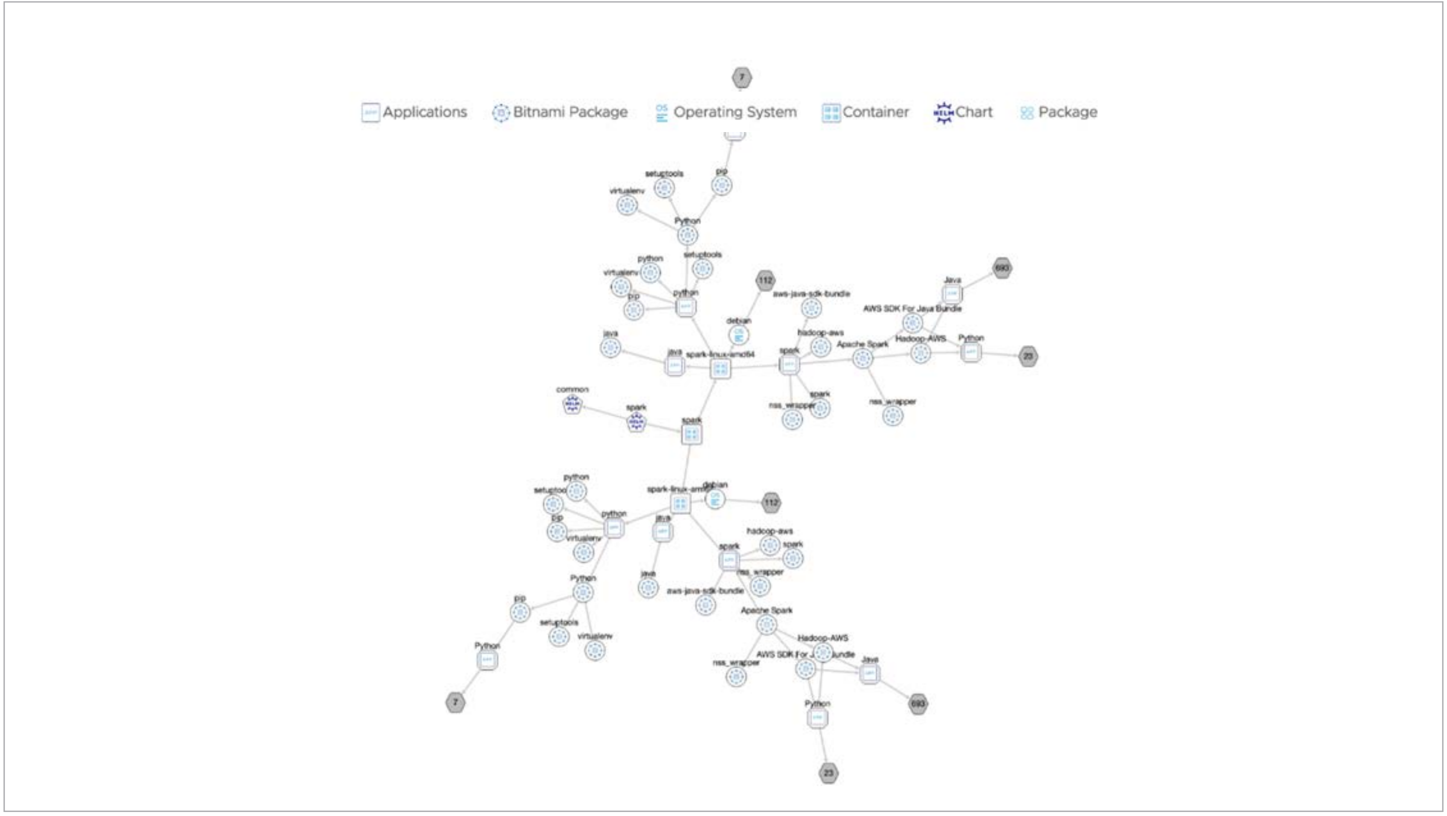
1. SBoM (SPDX)

NTIA defines the SBoM as “a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships.” In April 2023, we announced that all applications in Tanzu Application Catalog will be delivered with a corresponding SBoM report in SPDX format allowing organizations to track application changes and have comprehensive information about author, supplier, and component names, as well as version details.

SPDX format is the de facto standard for SBoM approved by the International Standardization Organization (ISO). It has been utilized for several years by prominent companies like Microsoft, RedHat, Intel, or Synopsys.

This report is available not only for container images, but also for Helm charts and virtual machines. It can be downloaded from the Tanzu Application Catalog and visualized in a graphical format thanks to the SPDX viewer. This allows customers to

- View and inspect relationships between all level packages in an easy way
- Check properties, licenses, and versions into multiple format and resources
- Learn how packages are connected and which packages are included in which resources

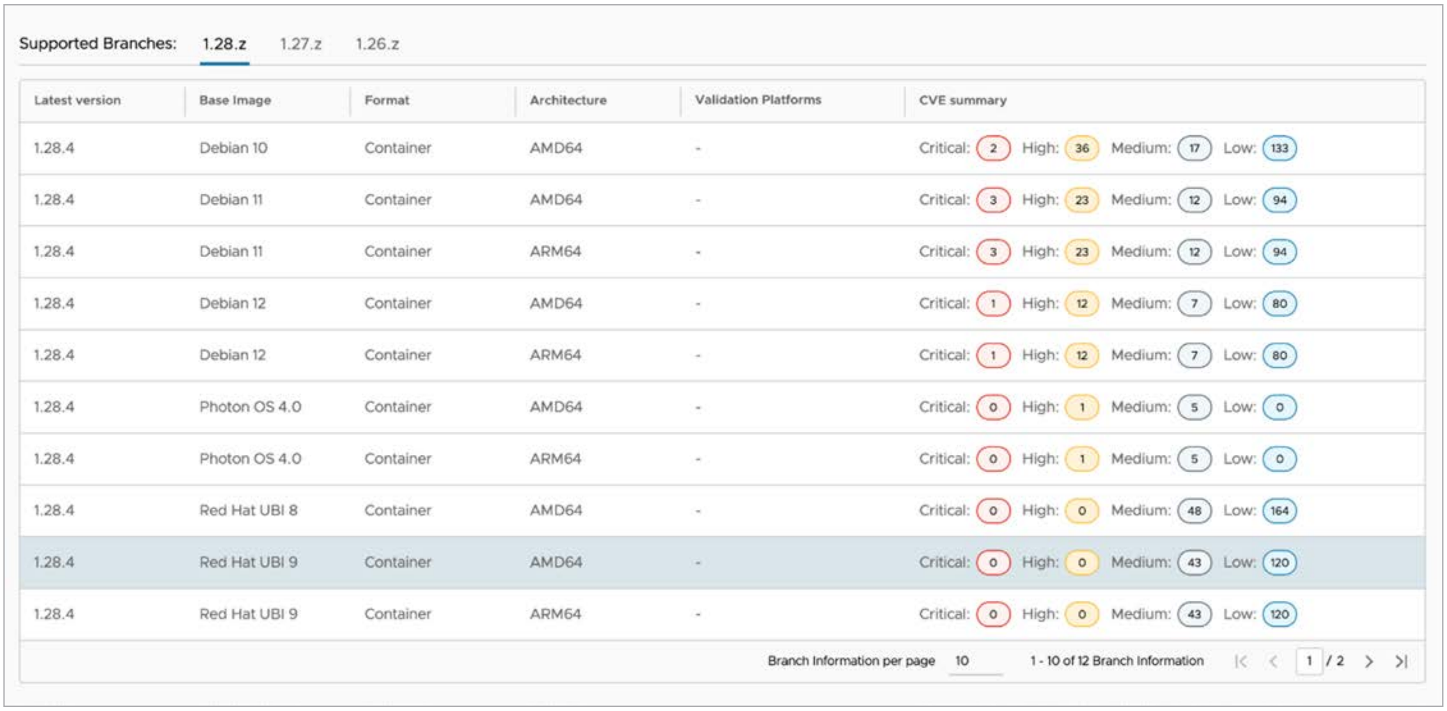


Apache Spark SBoM report displayed as a graphic view.

2. CVE scan result

The applications in Tanzu Application Catalog are continuously scanned using Trivy and Grype scanners to detect vulnerabilities. As a result, we provide customers with a report that contains a list of the latest detected security vulnerabilities for a given component, a reference to their CVE-ID number, a description of the issue and which versions of the software are affected, the Common Vulnerability Scoring System (CVSS) score, and more. This report is delivered in the Common Vulnerability Reporting Framework (CVRF) standard format as well.

To quickly identify the number of vulnerabilities and their severity in each package, customers can navigate to the Tanzu Application Catalog Library section. To view, click on an application and check the CVE summary column before adding applications to their catalogs. This feature is only available for logged in users.



Supported Branches: 1.28.z 1.27.z 1.26.z

Latest version	Base Image	Format	Architecture	Validation Platforms	CVE summary
1.28.4	Debian 10	Container	AMD64	-	Critical: 2 High: 36 Medium: 17 Low: 133
1.28.4	Debian 11	Container	AMD64	-	Critical: 3 High: 23 Medium: 12 Low: 94
1.28.4	Debian 11	Container	ARM64	-	Critical: 3 High: 23 Medium: 12 Low: 94
1.28.4	Debian 12	Container	AMD64	-	Critical: 1 High: 12 Medium: 7 Low: 80
1.28.4	Debian 12	Container	ARM64	-	Critical: 1 High: 12 Medium: 7 Low: 80
1.28.4	Photon OS 4.0	Container	AMD64	-	Critical: 0 High: 1 Medium: 5 Low: 0
1.28.4	Photon OS 4.0	Container	ARM64	-	Critical: 0 High: 1 Medium: 5 Low: 0
1.28.4	Red Hat UBI 8	Container	AMD64	-	Critical: 0 High: 0 Medium: 48 Low: 164
1.28.4	Red Hat UBI 9	Container	AMD64	-	Critical: 0 High: 0 Medium: 43 Low: 120
1.28.4	Red Hat UBI 9	Container	ARM64	-	Critical: 0 High: 0 Medium: 43 Low: 120

Branch Information per page 10 1 - 10 of 12 Branch Information 1 / 2

CVE summary accessible from the Tanzu Application Catalog Library for logged in users.

As mentioned earlier, SBoM and CVE scan results aim to paint a whole picture of your OSS supply chain but they do not provide context on the true risk you face from upstream vulnerabilities.

3. CSAF VEX document

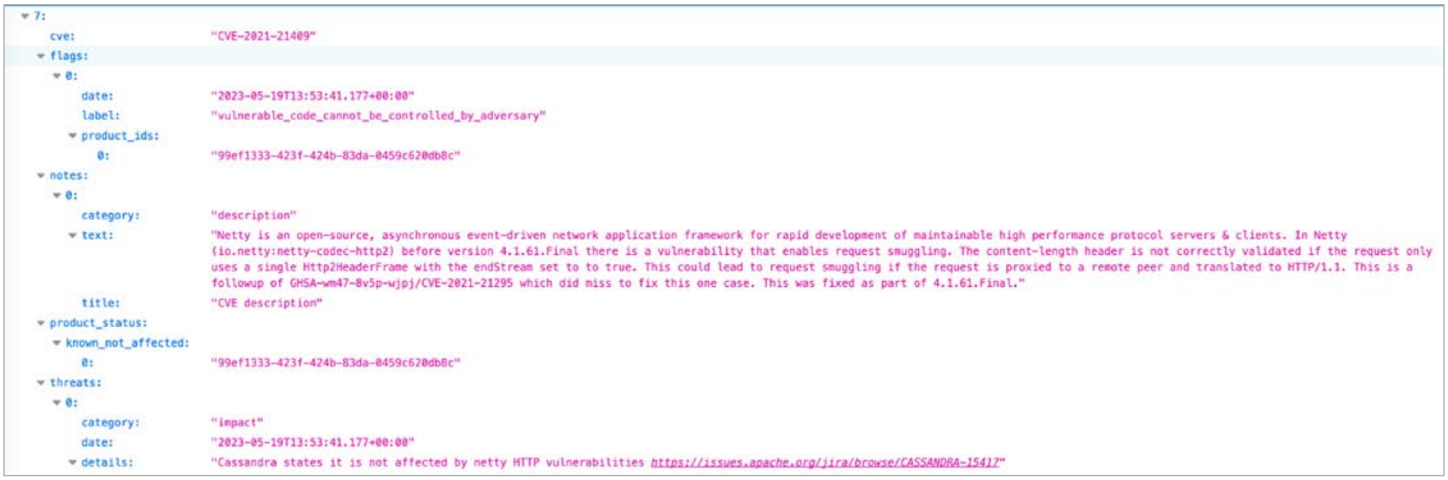
A VEX document is a machine-readable type of security advisory developed by the NTIA. The primary use case of VEX, as defined by the NTIA is to provide users (e.g., operators, developers, and services providers) additional information on whether a product is impacted by a specific vulnerability in an included component and, if affected, whether there are actions recommended to remediate.

The VEX documents provided by Tanzu Application Catalog follow CISA recommendations referred to in the use case "3.2.3 Single Product, Single Version, Multiple Vulnerabilities, Multiple Statuses", as you will see in the example below.

With SBoM and CVE scan results, customers have visibility into product changes and associated risks. However, it is a laborious task to investigate every vulnerability individually. Due to the elevated number of vulnerabilities that may be reported for a given OSS, without much context, security teams can be overwhelmed with data without any idea on how to prioritize those findings.

Some experts define VEX documents as a negative security advisory specifically designed to enumerate vulnerabilities that do not impact a product. From this perspective, VEX plays a pivotal role in boosting the efficiency of security teams by reducing the time spent investigating non-exploitable vulnerabilities that do not affect a given software product.

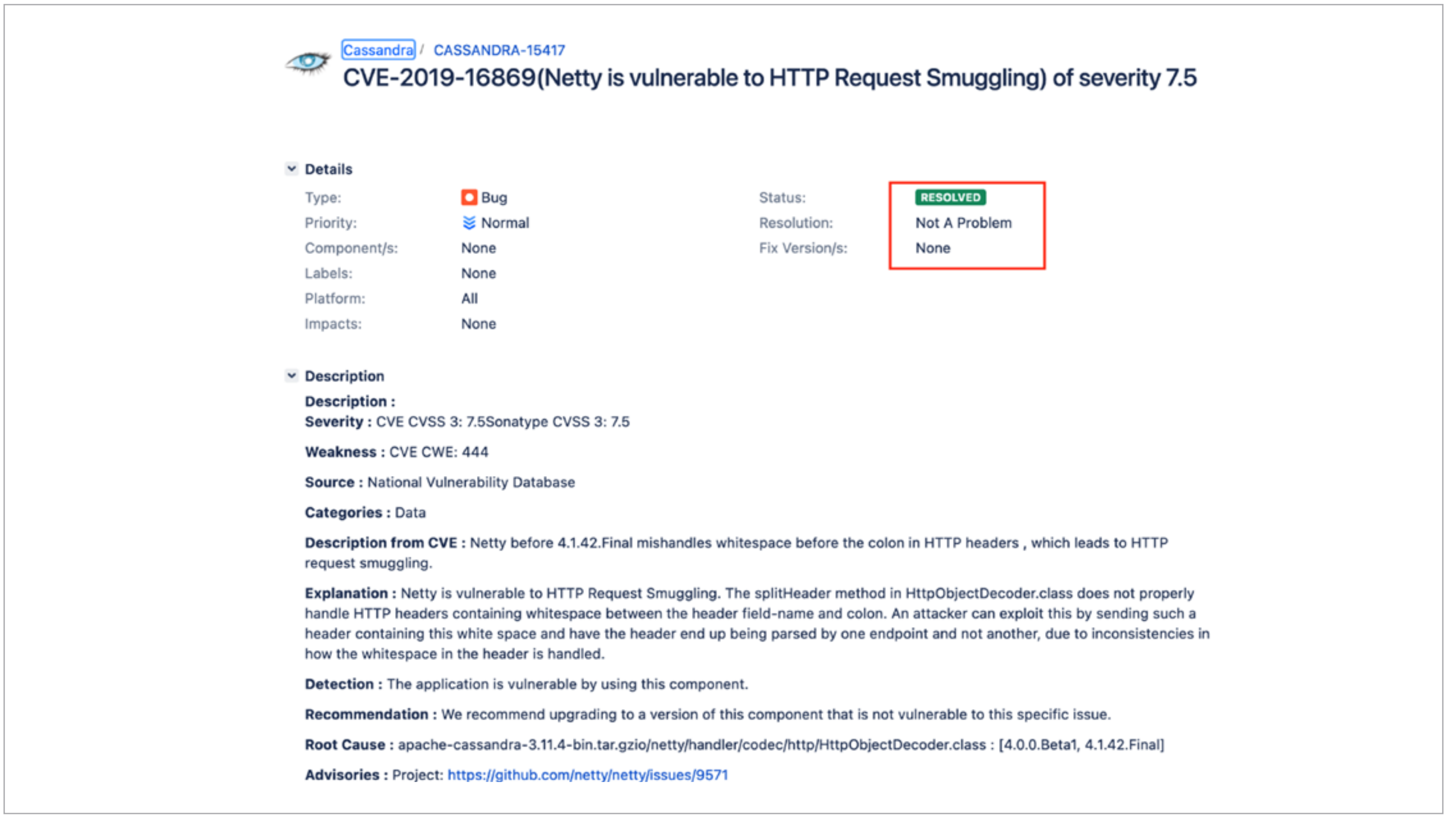
Let's use the following vulnerability discovered in an Apache Cassandra container as an example to understand how VEX documentation can be used to reduce CVE noise.



```
▾ 7:
    cve:                    "CVE-2021-21409"
  ▾ flags:
    ▾ 0:
        date:               "2023-05-19T13:53:41.177+00:00"
        label:              "vulnerable_code_cannot_be_controlled_by_adversary"
      ▾ product_ids:
          0:                "99ef1333-423f-424b-83da-0459c620db8c"
  ▾ notes:
    ▾ 0:
        category:           "description"
      ▾ text:               "Netty is an open-source, asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. In Netty
                            (io.netty:netty-codec-http2) before version 4.1.61.Final there is a vulnerability that enables request smuggling. The content-length header is not correctly validated if the request only
                            uses a single Http2HeaderFrame with the endStream set to to true. This could lead to request smuggling if the request is proxied to a remote peer and translated to HTTP/1.1. This is a
                            followup of GHSA-wm47-8v5p-wjpj/CVE-2021-21295 which did miss to fix this one case. This was fixed as part of 4.1.61.Final."
        title:              "CVE description"
  ▾ product_status:
    ▾ known_not_affected:
        0:                  "99ef1333-423f-424b-83da-0459c620db8c"
  ▾ threats:
    ▾ 0:
        category:           "impact"
        date:               "2023-05-19T13:53:41.177+00:00"
      ▾ details:            "Cassandra states it is not affected by netty HTTP vulnerabilities https://issues.apache.org/jira/browse/CASSANDRA-15417"
```

Apache Cassandra VEX document showing CVE-2021-21489 details.

In the software supply chain, what usually happens is that an upstream supplier of the software (e.g., Netty) becomes aware of a vulnerability and starts analyzing it, after which they release a patch and advisory. The vendor of the software, Apache, who has Netty as one of the components of Cassandra, analyzes the vulnerability, finds that the vulnerability doesn't meet the conditions to be exploitable, and so they neither release a patch nor an advisory, and close the issue in their project, marking it as "unaffected."



Cassandra / CASSANDRA-15417

CVE-2019-16869(Netty is vulnerable to HTTP Request Smuggling) of severity 7.5

Details

Type:	Bug	Status:	RESOLVED
Priority:	Normal	Resolution:	Not A Problem
Component/s:	None	Fix Version/s:	None
Labels:	None		
Platform:	All		
Impacts:	None		

Description

Description :
Severity : CVE CVSS 3: 7.5Sonatype CVSS 3: 7.5

Weakness : CVE CWE: 444

Source : National Vulnerability Database

Categories : Data

Description from CVE : Netty before 4.1.42.Final mishandles whitespace before the colon in HTTP headers , which leads to HTTP request smuggling.

Explanation : Netty is vulnerable to HTTP Request Smuggling. The splitHeader method in HttpObjectDecoder.class does not properly handle HTTP headers containing whitespace between the header field-name and colon. An attacker can exploit this by sending such a header containing this white space and have the header end up being parsed by one endpoint and not another, due to inconsistencies in how the whitespace in the header is handled.

Detection : The application is vulnerable by using this component.

Recommendation : We recommend upgrading to a version of this component that is not vulnerable to this specific issue.

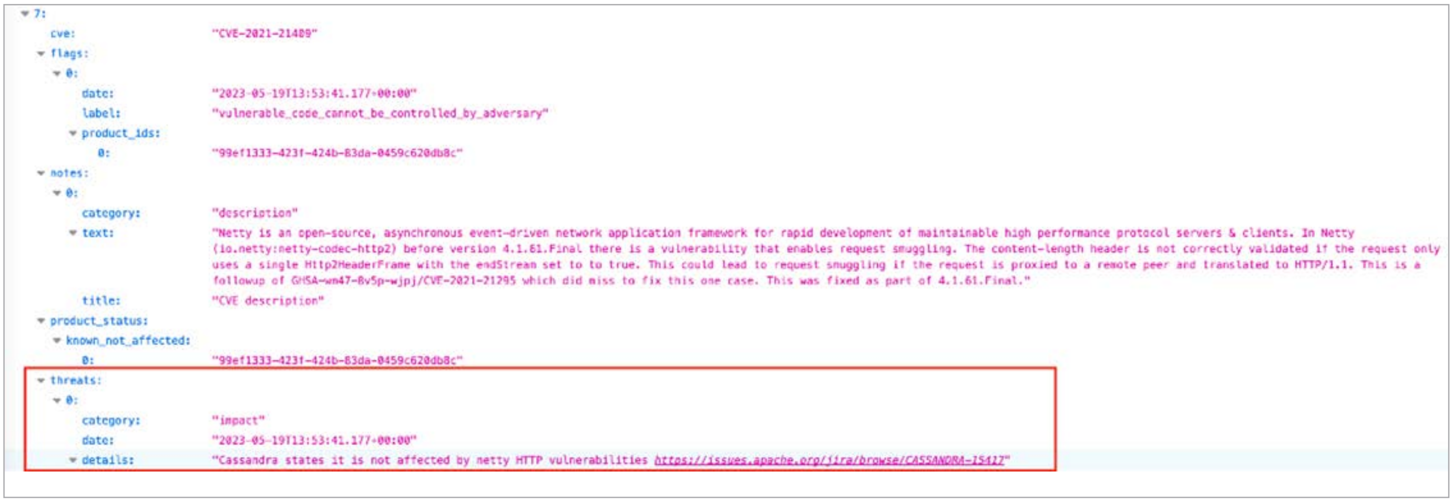
Root Cause : apache-cassandra-3.11.4-bin.tar.gzio/netty/handler/codec/http/HttpObjectDecoder.class : [4.0.0.Beta1, 4.1.42.Final]

Advisories : Project: https://github.com/netty/netty/issues/9571

The resolution for the above-mentioned CVE provided in Apache Cassandra's official ticketing system.

However, vulnerability scanners still flag that vulnerability as critical. For users it would be a toilsome task to go back and forth between the SBoM file to check if that vulnerability affects any of the components of the application they use, check its criticality on the CVE scan result, and navigate through the upstream project issues dashboard trying to understand if they might be affected or not by that vulnerability.

Tanzu Application Catalog does that work for them. As a distributor of Cassandra, it creates a VEX document that provides a list of vulnerabilities that don't affect the product as well as those that may affect it along with context and actions to minimize their impact.



```
7:
    cve:                  "CVE-2021-21409"
    flags:
        0:
            date:         "2023-05-19T13:53:41.177+00:00"
            label:        "vulnerable_code_cannot_be_controlled_by_adversary"
            product_ids:
                0:        "99ef1333-423f-424b-83da-0459c620db8c"
    notes:
        0:
            category:     "description"
            text:         "Netty is an open-source, asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers & clients. In Netty
                          (io.netty:netty-codec-http2) before version 4.1.61.Final there is a vulnerability that enables request smuggling. The content-length header is not correctly validated if the request only
                          uses a single Http2HeaderFrame with the endStream set to to true. This could lead to request smuggling if the request is proxied to a remote peer and translated to HTTP/1.1. This is a
                          followup of GHSA-wm47-8v5p-wjpj/CVE-2021-21295 which did miss to fix this one case. This was fixed as part of 4.1.61.Final."
            title:        "CVE description"
    product_status:
        known_not_affected:
            0:            "99ef1333-423f-424b-83da-0459c620db8c"
    threats:
        0:
            category:     "impact"
            date:         "2023-05-19T13:53:41.177+00:00"
            details:      "Cassandra states it is not affected by netty HTTP vulnerabilities https://issues.apache.org/jira/browse/CASSANDRA-15417"
```

VEX statement from Tanzu Application Catalog describing the resolution agreed by the upstream project highlighted.

This provides SecOps teams with the information to efficiently filter out false positives and enables them to focus on vulnerabilities that require immediate remediation.

This is a process that continuously repeats itself, so we can assert that VEX documents are dynamic while SBoM are more static documents. SBoM provides a fixed snapshot of application components, dependencies, licenses that are not meant to change and VEX relates to vulnerabilities, so it is constantly changing and evolving.

VEX documents are available in the Common Security Advisory Framework (CSAF) standard, a comprehensive format that was built with automation in mind, and widespread adopted by experts in the industry such as Cisco, RedHat, Oracle or even national CERTs.

Moreover, as part of our workflow, our team takes a proactive approach by reporting those vulnerabilities detected by our CVE scans that don't have an existing ticket or issue in the upstream project. When the vendor addresses the vulnerability in the upstream code and the project maintainers release an updated, patched, and tested version, Tanzu Application Catalog then builds a new release based on this improved code. This ensures that the software will function securely and seamlessly in production environments.

Reducing noise is your Trivy CVE report with VEX from Tanzu Application Catalog

We are committed to promoting the adoption of the CSAF VEX format among third-party scanners and tools such as Trivy, working directly with them and with the CSAF team as well. As a result of this continuous collaboration among us, Trivy, in its latest versions v0.49.2, supports CSAF VEX documentation. This enables Trivy to filter the CVEs they report based on the data present in VEX documents. So, users of Tanzu Application Catalog get to combine the power of SBoM, VEX documentation and CVE scan reports, and easily filter out the false-positive CVEs in the CVE scan reports.

After getting SPDX SBoM and VEX documents from Tanzu Application Catalog, and installing Trivy, you need to execute the following command:

```
./trivy sbom  path/to/spdx.json --vex path/to/vex.json
```

Now, Trivy scans the SPDX SBoM and VEX document you downloaded from Tanzu Application Catalog and generates a more precise CVE report filtering out all CVEs marked as not exploitable. Thus, your teams working on vulnerability management can easily avoid wasting their precious time and efforts on false positive CVEs.

We have plans to work with the Trivy team to make VEX documents auto-discoverable. In the future, you won't need to find and feed the VEX document to Trivy, but Trivy will automatically find and load the VEX document and provide accurate results out of the box.

What's Next?

As you have learned, VEX documents serve as the perfect companion to SBoM and CVE scan results in helping you mitigate risks posed by upstream vulnerabilities and reduce noise from false positives. VEX documents enrich SBoMs with vulnerability details for each component and consolidate the CVE scan results to a focused list containing only those that require action.

With VEX, SBoMs, and CVE scan results, Tanzu Application Catalog provides as a centralized source of truth where you can, not only get customizable, trusted and verified OSS applications and components to build applications, but also get all the information required to efficiently manage and assess the vulnerabilities that may pose risk to your software supply chain.

Future versions of Tanzu Application Catalog will include even more functionality to enable users to manage application metadata. As a centralized database of SBoMs, CVEs, and VEX information, users will have the ability to perform catalog wide component searches and visualize component dependencies.

Learn more

As OSS plays an indispensable role in the software development processes of many businesses today, more and more enterprises are partnering with Tanzu Application Catalog to enable their developers to adopt OSS in a more secure and sustainable manner.

Get more information about Tanzu Application Catalog with these resources:

- Product webpage
- Technical product documentation
- Whitepaper: Security Measures in Tanzu Application Catalog
- A current list of applications supported by Tanzu Application Catalog

The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Item No: bc-vmw-Tanzu-whitepaper-tech-usletter_031324-V3 3/24