CSAF Writing Bootcamp

Justin Murphy (CISA, US)

Thomas Schmidt (BSI, DE)

Who are we?

Justin Murphy

- Vulnerability Disclosure Analyst @CISA
- Passion for
  - SBOM/VEX
  - CSAF
  - CVD
  - Service
  - International Cooperation

Thomas Schmidt

- Technical ICS Analyst @BSI (usually not into standardization)
- Passion for
  - ICS
  - International Cooperation
  - CVD
  - Capacity building
  - CSAF

Things covered in the workshop

- Introduction into CSAF
- How to write security advisories in CSAF
- Available tools for writing advisories
- VEX and its relation to CSAF
- Hands-on exercises

Introduction into CSAF

- Who has heard of CSAF?
- Who has used CSAF?

BLOG

Transforming the Vulnerability Management Landscape

Released: November 10, 2022

Revised: November 14, 2022

Eric Goldstein, Executive Assistant Director for Cybersecurity

In the current risk environment, organizations of all sizes are challenged to manage the number and complexity of new vulnerabilities. Organizations with mature vulnerability management programs seek more efficient ways to triage and prioritize efforts. Smaller organizations struggle with understanding where to start and how to allocate limited resources. Fortunately, there is a path toward more efficient, automated, prioritized vulnerability management. Working with our partners across government and the private sector, we are excited to outline three critical steps to advance the vulnerability management ecosystem:

- First, we must introduce greater automation into vulnerability management, including by expanding use of the Common Security Advisory Framework (CSAF)
- Second, we must make it easier for organizations to understand whether a given product is impacted by a vulnerability through widespread adoption of Vulnerability Exploitability eXchange (VEX)
- Third, we must help organizations more effectively prioritize vulnerability management resources through use of Stakeholder Specific Vulnerability Categorization (SSVC), including prioritizing vulnerabilities on CISA's Known Exploited Vulnerabilities (KEV) catalog

https://www.cisa.gov/news-events/news/transforming-vulnerability-management-landscape

What is CSAF?

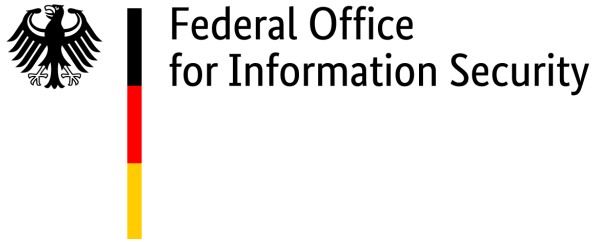
Common Security Advisory Framework

- International, open and free OASIS Standard
- Machine-readable format for security advisories (JSON)
- Standardized way of distribution security advisories
- Build with automation in mind
- Standardized tool set
- Guidance to actionable information
- Successor of CSAF CVRF 1.2

Ready to use!

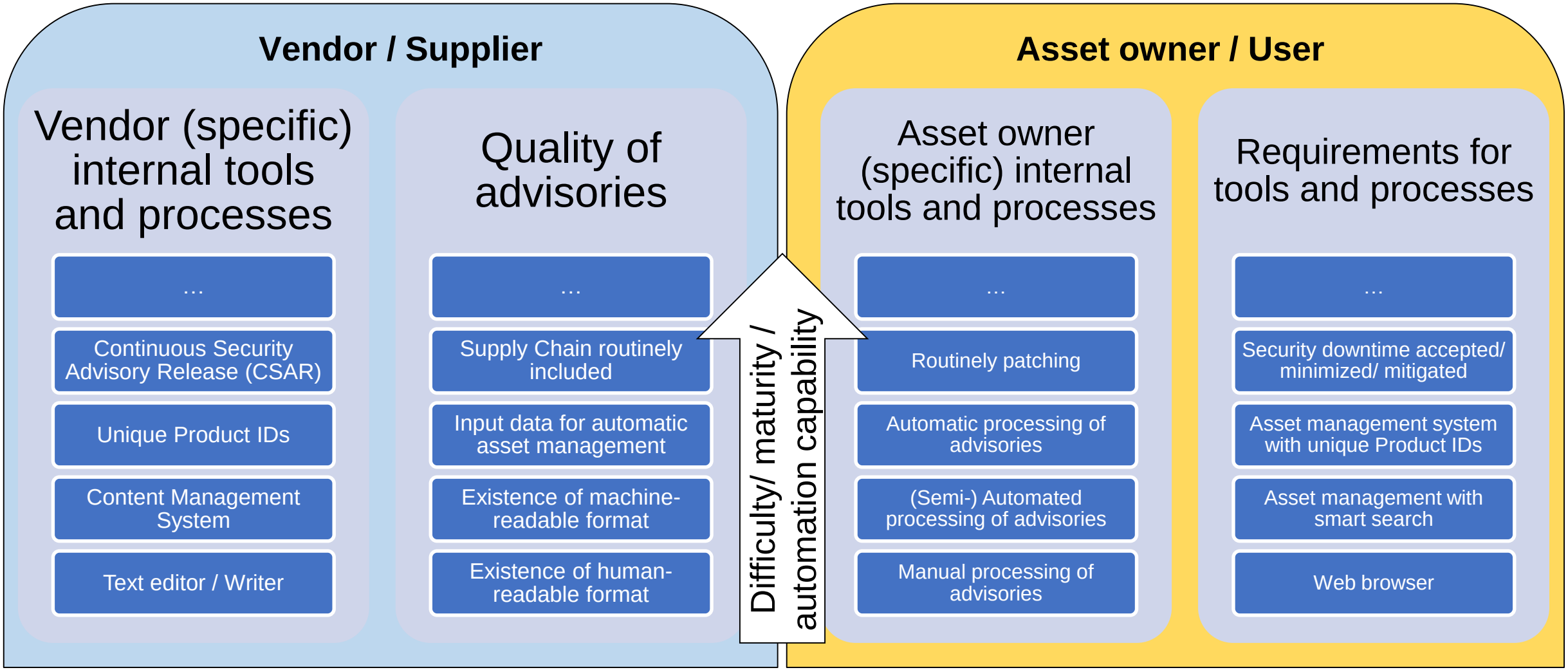
Organizations publishing CSAF

Omicron

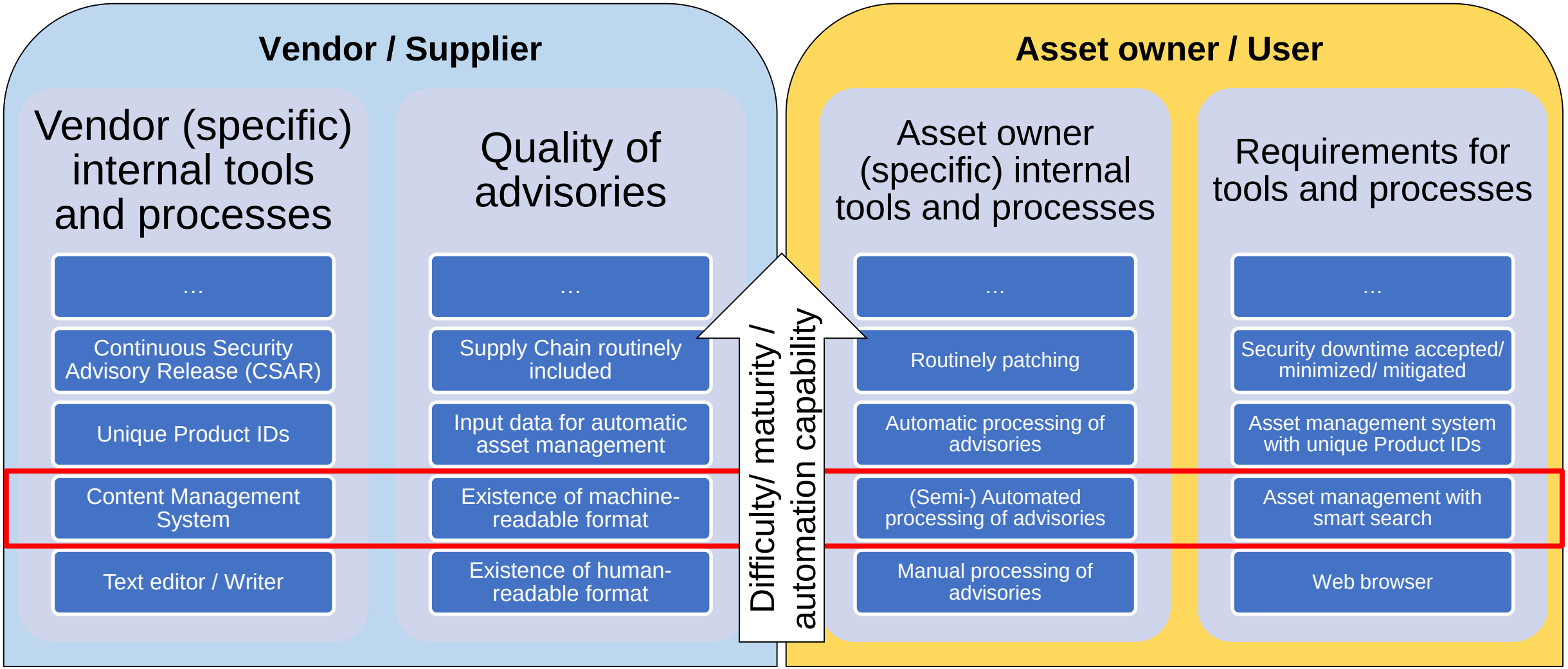


Festo

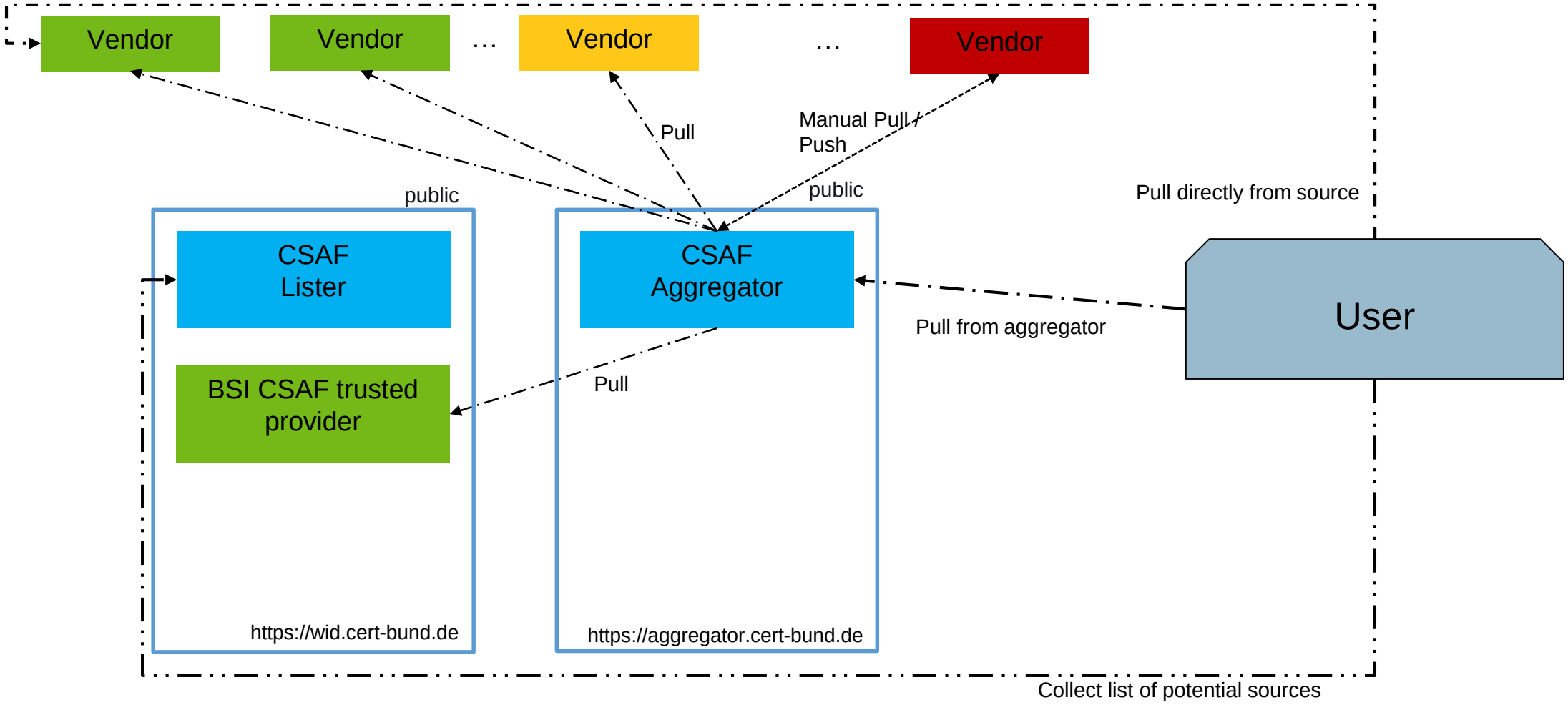
Two sides of the same coin – different maturity stages



Next step: reach stage 2 across parties



Ecosystem


Vendor
Vendor
...
Vendor
...
Vendor
Pull
Manual Pull / Push
public
public
Pull directly from source
CSAF Lister
CSAF Aggregator
User
Pull from aggregator
BSI CSAF trusted provider
Pull
https://wid.cert-bund.de
https://aggregator.cert-bund.de
Collect list of potential sources

Structure of CSAF 2.0 Specification

- Introduction & Design Considerations
- Schema elements
- Profiles
- Additional Conventions
- Tests
  - Mandatory
  - Optional
  - Informative
- Distributing CSAF documents
- Safety, Security, and Data Protection Considerations
- Conformance

Where to find more information?

https://csaf.io

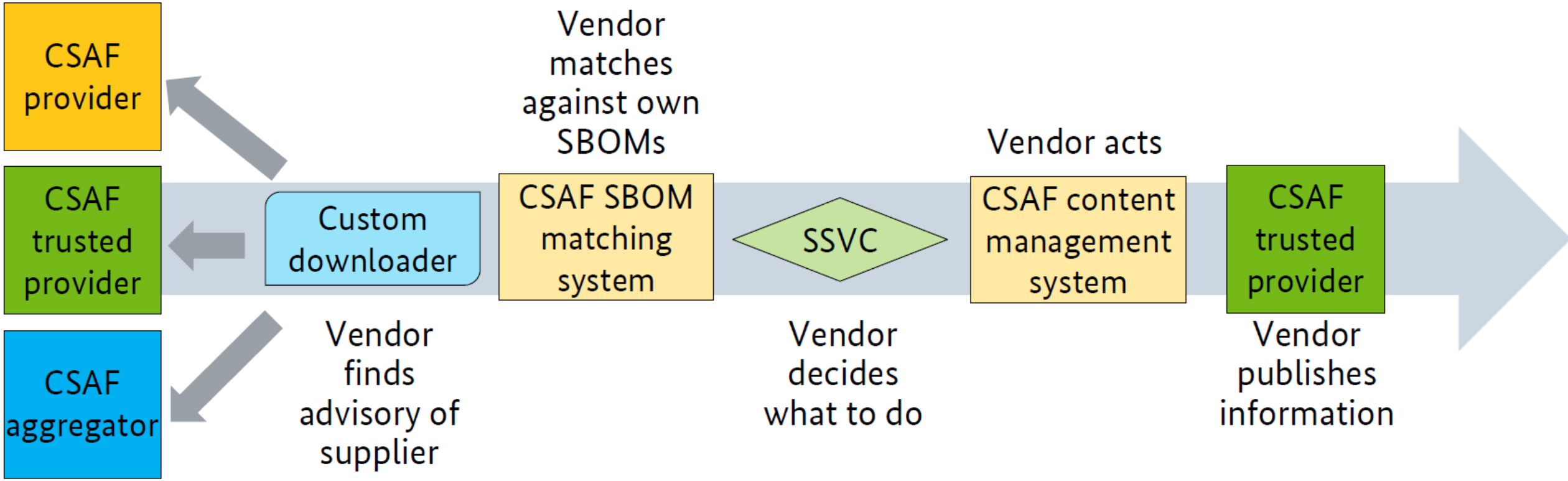
- OASIS TC: CSAF website: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=csaf
- CSAF GitHub: https://github.com/oasis-tcs/csaf
- CSAF 2.0 JSON Schema: https://docs.oasis-open.org/csaf/csaf/v2.0/csaf_json_schema.json
- CSAF 2.0 Prose: https://docs.oasis-open.org/csaf/csaf/v2.0/csaf-v2.0.html
- CSAF 2.0 Examples: https://github.com/oasis-tcs/csaf/tree/master/csaf_2.0/examples

- CSAF author guide: https://secvisogram.github.io/secvisogram-documentation/
- Secvisogram online editor: https://secvisogram.github.io

Tools developed by the community

- CSAF producer: https://github.com/secvisogram/secvisogram or https://github.com/mfd2007/yace
- CSAF content management system: https://github.com/secvisogram/secvisogram + https://github.com/secvisogram/csaf-cms-backend *(WIP)*
- CSAF trusted provider: https://github.com/csaf-poc/csaf_distribution
- CSAF aggregator: https://github.com/csaf-poc/csaf_distribution *(WIP)*
- Provider checker: https://github.com/csaf-poc/csaf_distribution *(WIP)*
- CSAF management system: *open for commercial and Open Source tools*
- CSAF asset matching system: *open for commercial and Open Source tools*
- CSAF modifier: *custom implementation*
- CSAF downloader: https://github.com/csaf-poc/csaf_distribution
- CSAF full validator: https://github.com/secvisogram/csaf-validator-service

Supply Chain



Prerequisites fulfilled? Set up the following

- Laptop with WLAN
- Connect to workshop WLAN
- Access webservers
- Familiar with Linux shell

Material at **https://files.test**

yace

- Simple online editor to create CSAF documents
- Limited functionality
- https://github.com/mfd2007/yace

Exercise: Simple CSAF advisory

- https://yace.test
- https://files.test/Exercise_01
- Instructions as Markdown file

Product Tree

- Hierarchical structure to convey products
- Usually: use `branches` with categories `vendor`, `product_name`, `product_version`
- `product_identification_helper` used for better comparison

Revision History

- Integer vs semantic version!
- New entry for each public revision
- `legacy_version` to link to “non-conforming” versions from human-readable advisories
- `current_release_date` set from date of last item in `revision_history`
- `version` must be updated as well

Document vs Vulnerabilities

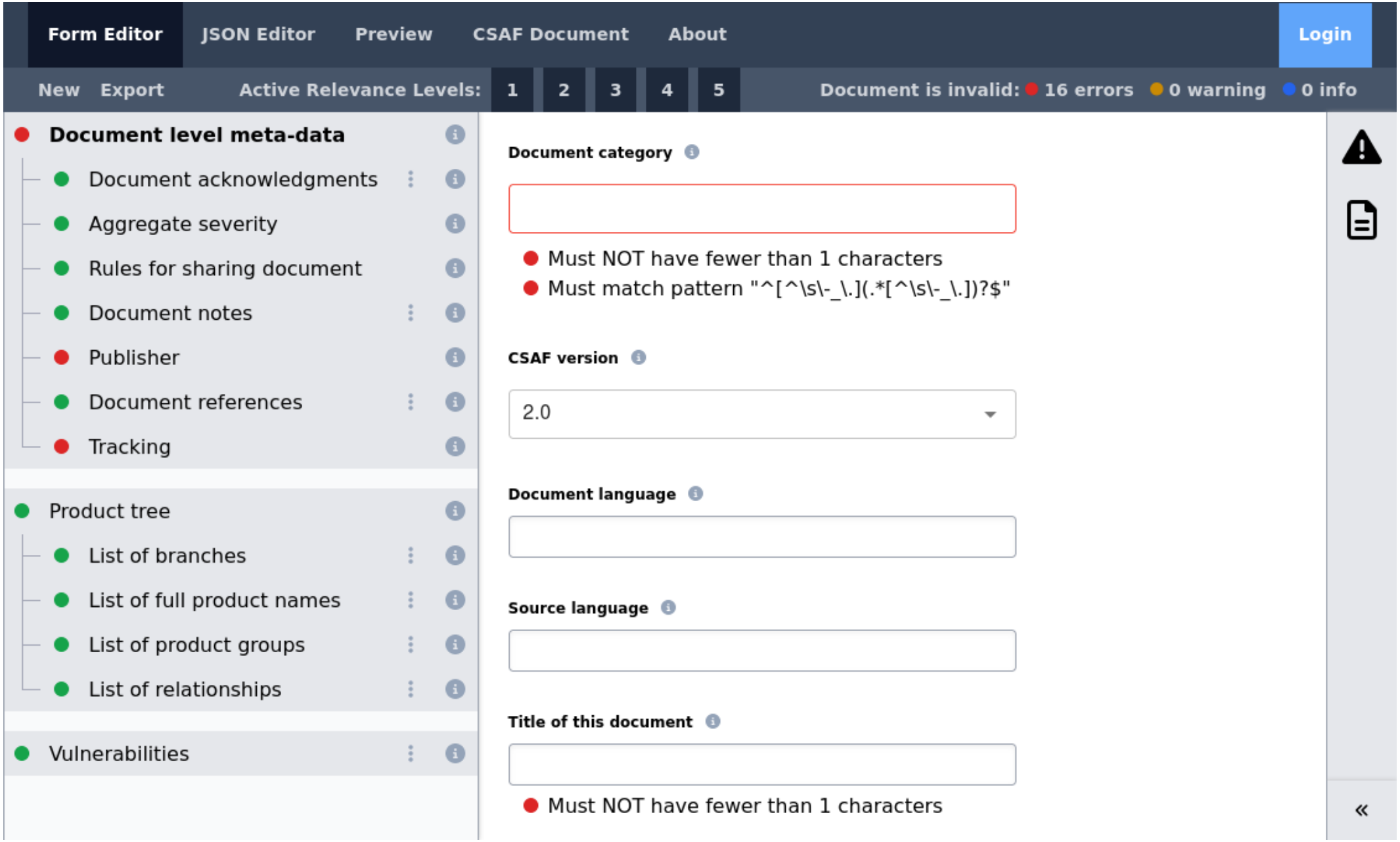
- Different elements (`acknowledgments`, `notes`, `references`) exist in both
- Document: applies to the whole document
  - Legal disclaimer, Product descriptions,…
- Vulnerabilities: applies only to this specific vulnerability
  - Vulnerability description

Note categories

- *Summary*: two to three sentences, barely any technical detail
- *Description*: one to three paragraphs with (possibly) some technical detail
- *Details*: detailed description, exceeding the length of a `description` and go deeper into technical details

Secvisogram

- First open source tool conforming CSAF producer
- Structure based on the JSON of the standard
- Different views to reduce complexity
- Backend option available
- Templates possible
- https://github.com/secvisogram/secvisogram/


Form Editor
JSON Editor
Preview
CSAF Document
About
Login
New
Export
Active Relevance Levels:
1
2
3
4
5
Document is invalid:
16 errors
0 warning
0 info
Document level meta-data
Document acknowledgments
Aggregate severity
Rules for sharing document
Document notes
Publisher
Document references
Tracking
Product tree
List of branches
List of full product names
List of product groups
List of relationships
Vulnerabilities
Document category
Must NOT have fewer than 1 characters
Must match pattern "^[^\s\-_\.](.*[^\s\-_\.])?$"
CSAF version
2.0
Document language
Source language
Title of this document
Must NOT have fewer than 1 characters

CSAF CMS Backend

- Automatically assign IDs
- Track revision history and version
- Track changes
- Use workflows
- Use company logos
- Use templates
- *Publish from there automatically (WIP)*
- https://github.com/secvisogram/csaf-cms-backend/

Exercise: Update advisory

- https://secvisogram.test
- https://files.test/Exercise_02
- Instructions as Markdown file

Product Tree

- Hierarchical structure to convey products
- Usually: use `branches` with categories `vendor`, `product_name`, `product_version`
- `product_identification_helper` used for better comparison

Exercise: Add product

- https://secvisogram.test
- https://files.test/Exercise_03
- Instructions as Markdown file

Product version ranges

- vers vs vls
- Use only if
  - Only insufficient data is available
  - Enumeration is hard and vers is well-defined
- Can’t be used with CPE
- Careful with implications of `vers:all/*`

Exercise: Add product with version range

- https://secvisogram.test
- https://files.test/Exercise_04
- Instructions as Markdown file

Hardware and Software

- Separate hardware from software
- Bind them via `relationships`

Exercise: Describe hardware and software

- https://secvisogram.test
- https://files.test/Exercise_05
- Instructions as Markdown file

Product status

- `first_affected`
- `known_affected`
- `last_affected`
- `first_fixed`
- `fixed`
- `known_not_affected`
- `under_investigation`
- `recommended`

`first_affected`, `known_affected`, `last_affected` → affected

`first_fixed`, `fixed` → fixed

`known_not_affected` → not affected

`under_investigation` → under investigation

affected, fixed, not affected, under investigation → contradicting groups

https://github.com/lunasec-io/lunasec/blob/998c69decc0894a214efa035854b48b1af18eb6e/docs/static/img/log4shell-logo.png

THE LINUX FOUNDATION PROJECTS

Zephyr

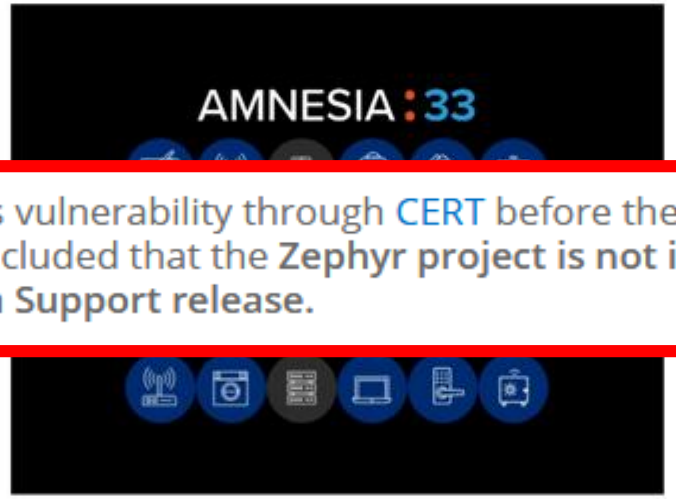
Resources Community Learn Members Contact Become a Member

Blog News Top News

Zephyr Security Update on Amnesia:33

December 16, 2020

Written by David Brown, on behalf of the Zephyr Security Team



On December 8, 2020, Forescout released a report containing numerous vulnerabilities found in various embedded TCP/IP stacks, known as AMNESIA:33. These vulnerabilities, across multiple network implementations, concern various memory and overflow errors, some of which are readily exploitable.

The Zephyr project received notification of this vulnerability through CERT before the publication date. We analyzed these vulnerabilities, and any affected code, and concluded that the **Zephyr project is not impacted by any of these vulnerabilities, neither in the current releases, nor in any Long Term Support release.**

Despite being collected under a single name, this report describes 33 vulnerabilities that are largely unrelated to one another. The report is the result of an analysis of 4 TCP/IP implementations that are commonly used in embedded systems: uIP, uIP in Contiki-OS, PicoTCP, and Fnet. Of these implementations, only the code in Fnet has ever been used in Zephyr.

The Zephyr LTS release 1.14 contains an implementation of the TCP stack from Fnet. Of the vulnerabilities reported in Fnet, 2, CVE-2020-17468, and CVE-2020-17469, are in the IPv6 Fnet code, one, CVE-2020-17467, affects Link-local Multicast Name Resolution LLMNR), and 2, CVE-2020-24383, and CVE-2020-17470 affect DNS functionality. None of the affected code has been used in the Zephyr project, while 1.14 does use the Fnet TCP, it does not use the affected IPv6, DNS or LLMNR code.

For current releases, including the current 2.4.0, this code has been replaced by a Zephyr-specific implementation.

The Zephyr project takes security seriously, for more information on our processes involving security, including how to report vulnerabilities can be found on our Security page.

The Zephyr project received notification of this vulnerability through CERT before the publication date. We analyzed these vulnerabilities, and any affected code, and concluded that the **Zephyr project is not impacted by any of these vulnerabilities, neither in the current releases, nor in any Long Term Support release.**

Not affected but…

- …still costumers call / write tickets
- …nobody reads the press release
- Huge effort for hotlines / support

Already investigating but…

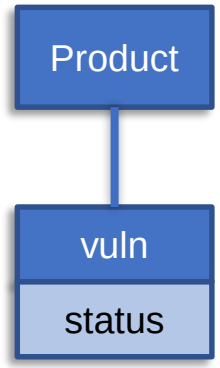
- …still costumers call / write tickets
- …nobody reads the forum entry
- Huge effort for hotlines / support

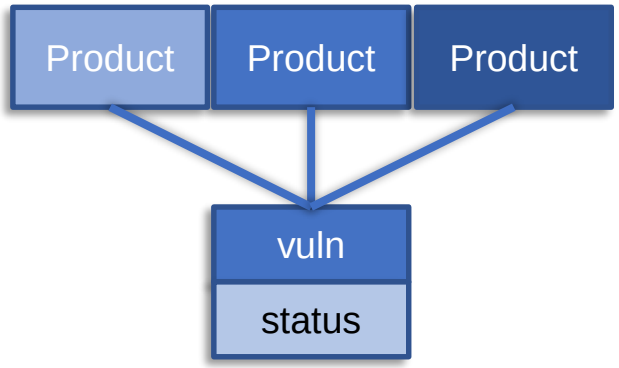
VEX

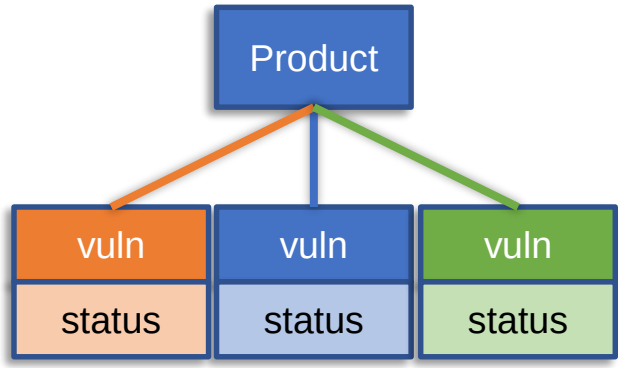
- Vulnerability Exploitability eXchange
- Communicate product status explicit
- Machine-readable to address scalability

- No a specification but implementations in different standards
  → CSAF profile

“Modality” of VEX – Use Cases







Benefits

- Save money!
- Reduce human workload!
- Spread the word about the good work your PSIRT does!
- Use established tools and distribution

Minimum fields – VEX document metadata

```
{
  "document": {
    "category": "csaf_vex",
    "csaf_version": "2.0",
    "publisher": {
      "category": "vendor",
      "name": "Example Company ProductCERT",
      "namespace": "https://psirt.example.com"
    },
    "title": "Example VEX Document Use Case 001 - Not Affected",
    "tracking": {
      "current_release_date": "2023-03-03T11:00:00.000Z",
      "id": "EVD-2023-001",
      "initial_release_date": "2023-03-03T11:00:00.000Z",
      "generator": {
        "engine": {
          "name": "Secvisogram",
          "version": "2.1.0"
        }
      },
      "revision_history": [
        {
          "date": "2023-03-03T11:00:00.000Z",
          "number": "1",
          "summary": "Initial version."
        }
      ],
      "status": "final",
      "version": "1"
    }
  },
```

VEX format identifier

Author [author] and Author role [author_role]

Timestamp last updated [doc_time_last_updated]
Document ID [doc_id]
Timestamp first issued [doc_time_first_issued]

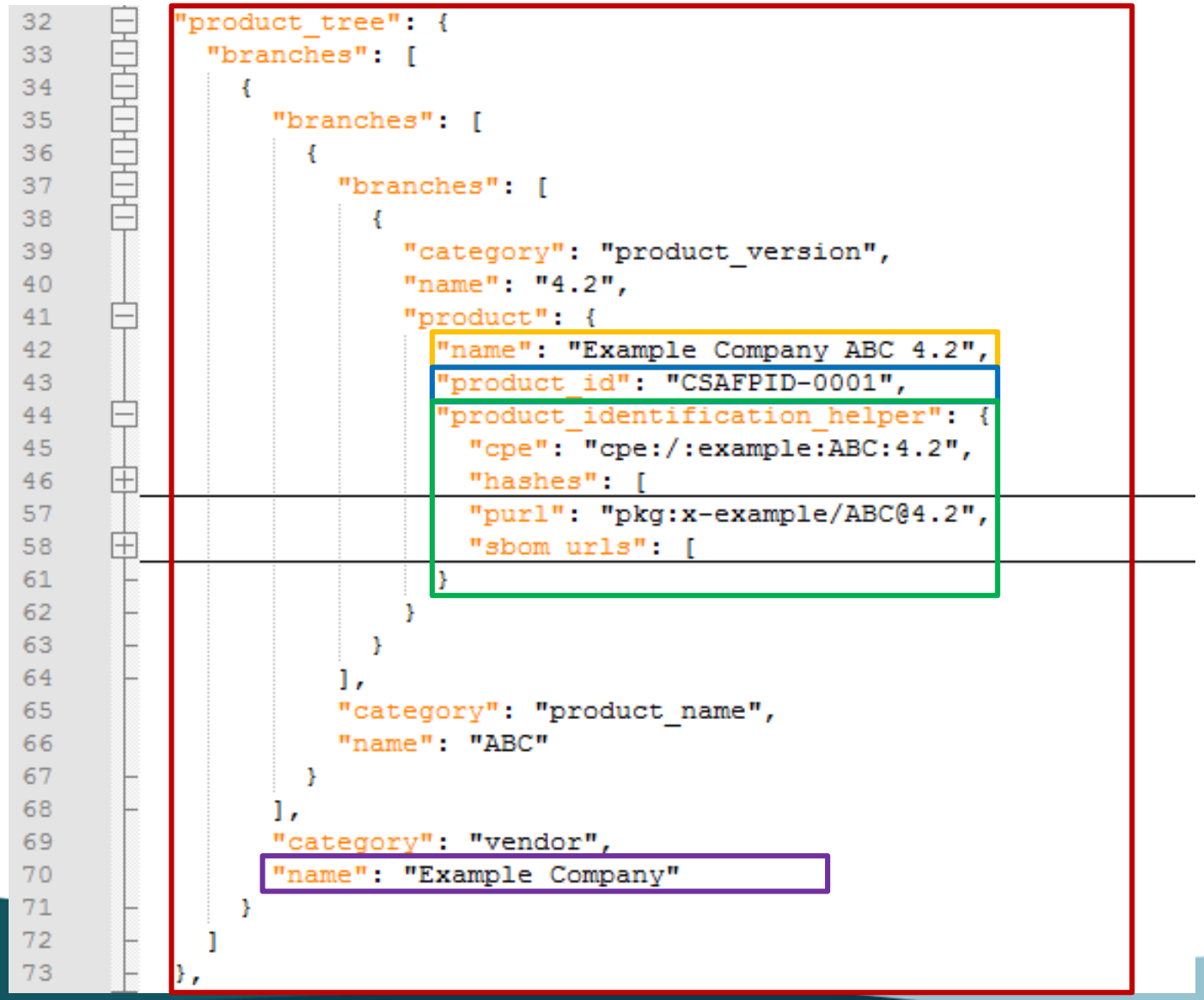
Tooling [tooling]

Document version [doc_version]

Minimum fields – VEX statement metadata

- Statement ID [statement_id]: given through CSAF structure (e.g. /vulnerabilities[2]/product_status/known_affected)
- Inherited from document:
  - Statement version [statement_version]
  - Timestamp first issued [statement_time_first_issued]
  - Timestamp last updated [statement_time_last_updated]

Minimum fields – Product details



```
"product_tree": {
  "branches": [
    {
      "branches": [
        {
          "branches": [
            {
              "category": "product_version",
              "name": "4.2",
              "product": {
                "name": "Example Company ABC 4.2",
                "product_id": "CSAFPID-0001",
                "product_identification_helper": {
                  "cpe": "cpe:/:example:ABC:4.2",
                  "hashes": [
                  "purl": "pkg:x-example/ABC@4.2",
                  "sbom urls": [
                  }
            }
          ],
          "category": "product_name",
          "name": "ABC"
        }
      ],
      "category": "vendor",
      "name": "Example Company"
    }
  ]
},
```

Product details with Product identifier [product_id]

Human-readable product name

Product Identifier for reference within document

Different identification methods

Supplier [supplier]

Product statuses

- Not Affected → impact statement
- Affected → action statement
- Fixed
- Under Investigation

Minimum fields – Vulnerability and Product status

```
74  "vulnerabilities": [
75    {
76      "cve": "CVE-2021-44228",
77      "flags": [
78        {
79          "date": "2023-03-03T11:00:00.000Z",
80          "label": "vulnerable_code_not_present",
81          "product_ids": [
82            "CSAFPID-0001"
83          ]
84        }
85      ],
86      "notes": [
87        {
88          "category": "description",
89          "text": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in co
                    LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary
                    2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionali
                    and does not affect log4net, log4cxx, or other Apache Logging Services projects.",
90          "title": "CVE description"
91        },
92        {
93          "category": "general",
94          "text": "All status determination are done by thorough reviews and analysis.",
95          "title": "Status determination"
96        }
97      ],
```

Vulnerability identifier [vul_id]

Justification [justification]

Description [vul_description]

Status notes [status_notes]

Minimum fields – Vulnerability and Product status

```
98      "product_status": {
99        "known_not_affected": [
100         "CSAFPID-0001"
101       ]
102     },
103     "threats": [
104       {
105         "category": "impact",
106         "date": "2023-03-03T11:00:00.000Z",
107         "details": "Class with vulnerable code was removed before shipping.",
108         "product_ids": [
109           "CSAFPID-0001"
110         ]
111       }
112     ]
113   }
114 ]
115 }
```

Status [status]

Impact statement [impact_statement] with Timestamp of impact statement [impact_statement_time]

How to link to an SBOM?

Product identification helpers:

- Retrievable SBOM

```
"sbom_urls": {
    //...
  "items": {
    "https://example.com/location-to-sbom"
  }
}
```

How to link to an SBOM component?

- CycloneDX

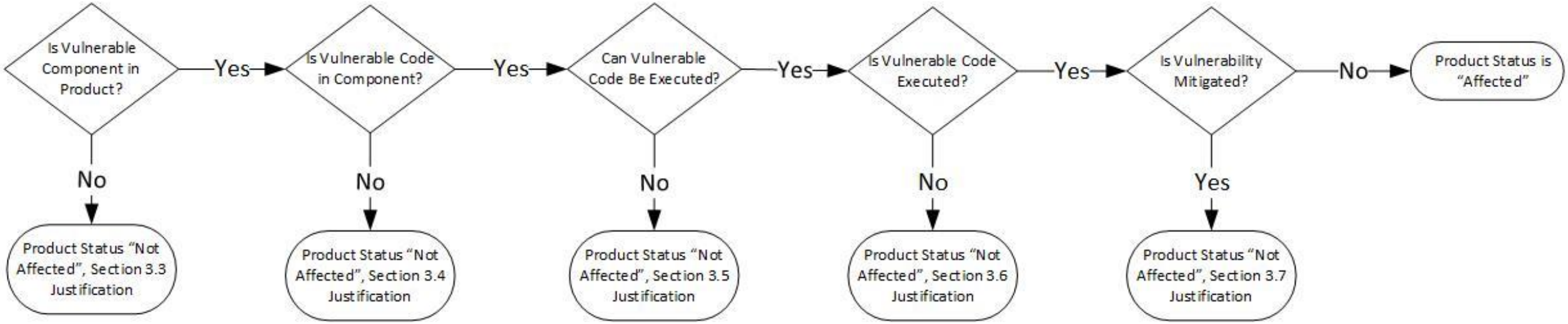
```
"x_generic_uris": [
  {
    "namespace": "https://cyclonedx.org/capabilities/bomlink/",
    "uri": "urn:cdx:411dafd2-c29f-491a-97d7-e97de5bc2289/1#pkg:maven/org.jboss.logging/jboss-logging@3.4.1.Final?type=jar"
  }
]
```

- SPDX

```
"x_generic_uris": [
  {
    "namespace": "https://spdx.github.io/spdx-spec/document-creation-information/#65-spdx-document-namespace-field",
    "uri": "https://swinslow.net/spdx-examples/example4/main-bin-v2#SPDXRef-libc"
  }
]
```

Status Justifications

- `Component_not_present`
- `Vulnerable_code_not_present`
- `Vulnerable_code_cannot_be_controlled_by_adversary`
- `Vulnerable_code_not_in_execute_path`
- `Inline_mitigations_already_exist`



Exercise: Create a VEX

- https://secvisogram.test
- https://files.test/Exercise_06
- Instructions as Markdown file

Next steps

- Publish CSAF on your website
- Integrate it into your workflow
- Become a CSAF trusted provider
- Submit yourself to a CSAF lister / aggregator: https://wid.cert-bund.de/portal/wid/csaf/submit
- **Spread the word! #oCSAF**

Thank you!

https://csaf.io