15 Best Practices for a good Security Advisory in CSAF format

A security advisory in CSAF format should be a well formatted and well understandable source of information to make things clear and not to raise a lot of questions. To reach this goal, the following rules should be applied (Rules apply to a security advisory, please use the csaf_security_advisory profile).

Regarding the document subsection of a CSAF document

1. For property tlp the TLP label should be set to WHITE in order to have no limits in distributing the advisory (/document/distribution/tlp/label).

2. The summary of a revision_history entry should be used to describe shortly, clearly and human readable what has been changed in regard to the previous revision. This is an enabler for a fast decision whether a new revision matters or not (/document/tracking/revision_history[]/summary).

3. The assignment of document ids should be consistent throughout an organization. The document id is used to build the filename and uniquely identify the document (/document/tracking/id). Together with the publisher namespace, it identifies a document globally unique.

4. The filename must follow the rules, defined in the CSAF standard (section 5.1).

5. The canonical URL in /document/references makes it possible to automatically retrieve the latest version of that CSAF document.

6. The information to identify a publisher of a CSAF document should not be changed during a document lifecycle. Exceptions would be major events such as a company name changes (/document/publisher).

Regarding the product_tree subsection of a CSAF document

7. Provide product information as accurate and detailed as possible, using the /product_tree/branches including the category, vendor, product_name and product_version.

8. Product versions should be enumerated by using product_version wherever possible as matching products from an asset database or SBOM against a product_version_range element can be complex, non-deterministic or error prone. If the issuing party doesn’t have enough information to enumerate products by version, the use of a product_version_range is acceptable.

9. Provide detailed information to enable a user/customer to properly identify a product in use. Use the product_identification_helper to convey that information (/product_tree/*/product/product_identification_helper).
10. Separation of hard- and software (firmware) is useful. Make clear how to identify the product itself and how to identify the installed software version, currently used by the product. Make use of relationship objects to convey this information.

Regarding the vulnerabilities subsection of a CSAF document

11. Make clear which products are affected and which are fixed or not_affected (/vulnerabilities[]/product_status). If you list “not affected” products, consider using the profile CSAF VEX. However, it is recommended to provide at least a short statement in the details field of /vulnerabilities[]/threats, why that product is not affected.

12. Provide CVSS V3.1 scores (/vulnerabilities[]/scores[]).

13. Provide a CVE tracking number(/vulnerabilities[]/cve).

14. Provide proper information about the mitigation possibilities through /vulnerabilities[]/remediations. Use e.g. no_fix_planned if a product is end of life and none_available if the fix is currently being developed.

15. A vulnerability should have at least a short description which could be used for a summary (/vulnerabilities[]/notes). This can be the CVE description (with title CVE description and category description) or a vulnerability summary (with title Vulnerability summary and category summary).