From: [redacted]
To:
Subject: Comments on CSF 2.0 public draft and implementation examples draft
Date: Monday, November 6, 2023 8:04:01 PM

Dear NIST team,

Please find my feedback below on the CSF 2.0 public draft and the implementation examples draft (the versions marked August 8th, 2023).

Regards,
Arne

Comments on the CSF 2.0 public draft and implementation examples draft

> NIST seeks feedback on whether this draft revision addresses
> organizations' current and anticipated future cybersecurity challenges,
> is aligned with leading practices and guidance resources, and reflects
> comments received so far. In addition, NIST requests ideas on the best
> way to present the modifications from CSF 1.1 to CSF 2.0 to support
> transition. NIST encourages concrete suggestions for improvements to
> the draft, including revisions to the narrative and Core.

The move to highlight the importance of supply-chain risk management is an important one. In this light it is logical to reference NIST's SP 800-161r1 (in line 562) and NIST's SP 800-218 (in line 606). However, note that 'Guidelines on minimum standards for developer verification of software' (NISTIR 8397) is another important resource that should inform how organisations approach both supply-chain risk management and secure software development. I think it deserves a specific callout in the narrative of the CSF. Consider that the 2022 update to the ISO 27002 controls framework has much greater emphasis on software security compared to the ISO 27002 version from 2013. In contrast, the NIST CSF 2.0 draft contains relatively limited coverage of this topic. Highlighting the key activities contained in NISTIR 8397 would somewhat ameliorate this situation.

I'd like to highlight two concepts used in the CSF draft that are would benefit from further specificity: terms related to 'threat' and the term 'outcomes'. As used in the CSF draft, 'threat(s)' often seems to refer to threat actors, while the extensive literature review by Bodeau et al. -- 'Cyber threat modeling: survey, assessment, and representative framework' -- highlights that the term threat can refer to both threat actors as well as threat events. The Common Criteria define a threat as 'an adverse action performed by a threat agent on an asset'. Also, while Lockheed-Martin's kill chain model published in 2010 as 'Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains' highlights threat actors, their IDDIL/ATC model published in 2019 as 'A threat-driven approach to cyber security' recognises that threats can be 'defined as people or events'. The CSF should not pick sides when it comes to this definition. As is, the draft is partial and thereby neglects the perspective of threats as threat events. This is relevant given that threat modelling of software-intensive systems is generally most productively done by focussing on threat events using frameworks such as STRIDE applied to an abstraction of the system being developed. Some concrete suggestions for the narrative and core: clarify the term 'threats' (e.g. by changing it to 'threat actors and/or events') and besides mentioning cyber threat intelligence also discuss threat modelling. For example, besides the common DFD+STRIDE approach, privacy-related threat modelling can be mentioned in line 646-647, where LINDDUN could be compared and contrasted to security-related threat modelling.

The other term that will benefit from greater specificity is that of 'outcomes'. While in the latter half of the CSF draft there is repeated reference to outcomes as covering any function/category/subcategory level, in the earlier part statements such as 'resources that provide additional guidance on practices and controls that could be used to achieve those outcomes' (lines 7-8) and 'examples of how each outcome can be achieved along with references to additional guidance' (lines 84-85) indicate that the term outcomes specifically refers to the subcategories. While the profiles published by NIST to date take different approaches as to whether and how to include function and category prioritisation, in the end all profiles end up focussing on the level of subcategories. This should be clarified. Personally, I think that when talking about the outcomes of the framework, the focus should be on the subcategories.

Relatedly, there is misalignment between the framework structure represented in figure 1 of the CSF draft (which is also the approach of the CSF 1.1) and the Implementation Examples document which has space for informative references at the category level. In line with CSF 1.1 and the CSF 2.0 draft, the informative references should only be mapped to the level of subcategories and to related implementation examples. Also, while the CSF draft notes that subcategories are non-exhaustive, there should be a note that they are intended as the key reference when it comes to interpretation of what is meant by both categories and functions. This is important because even if presented as a voluntary framework, the CSF 1.1 is being used and the CSF 2.0 will get used as a normative framework.

> NIST seeks feedback on what types of Examples would be most beneficial
> to Framework users, as well as what existing sources of implementation
> guidance might be readily adopted as sources of Examples (such as the
> NICE Framework Tasks, for example). NIST also seeks feedback on how
> often Implementation Examples should be updated and whether and how to
> accept Implementation Examples developed by the community.

> NIST seeks input on: concrete improvements to the Examples; whether the
> Examples are written at an appropriate level of specificity and helpful
> for a diverse range of organizations; what other types of Examples
> would be most beneficial to Framework users; what existing sources of
> implementation guidance might be readily adopted as sources of Examples
> (such as the NICE Framework Tasks); how often Examples should be
> updated; and whether and how to accept Examples developed by the
> community.

The master task list of the NICE framework is an excellent source of inspiration for implementation examples. I deem the task statements to be the most useful part of the NICE framework (so much so that I think that those building an academic curriculum or a professional role profile should do away with the rest of the NICE framework and focus on the specific tasks and associated deliverables they deem to be most relevant, complementing such a list with a body of knowledge distilled from sources such as CyBOK). The general guidance in section 2.1 of SP 800-181r1 and the more detailed guidance in section 3.1 and section 3.2 of the 'Task, knowledge, and skill statements authoring guide for workforce frameworks' provide a great basis for how to write task statements, and in turn they are also a great basis for defining what makes a good implementation example. These criteria should be applied to the proposed examples. Looking only at the Govern function and only at the requirement to 'begin with the activity being executed', we see that the following examples don't align with the NICE recommendations as they read more like status descriptions: GV.RM-01 Ex3; GV.SC-09 Ex1 and Ex5; GV.RR-01 Ex1, Ex2, and Ex3; GV.RR-04 Ex2. A similar analysis should be performed for all functions and for all NICE principles. Note that these NICE principles are also great sources to refer to for those looking to create their own task statements or implementation examples. In doing so, an important caveat that should be mentioned to the reader is that mappings of the CSF to and from NICE shouldn't be done in a dogmatic way. Given the overlap between various NICE task statements, as well as the extensibility of the NICE framework to different contexts, any mention or use of task statements as inspiration for implementation examples should highlight that these are not the 'one and only' mapping.

In light of the above discussion, there are some points of improvement to Appendix A of the CSF draft. This appendix highlights 'target roles and responsibilities' (line 761) which is further described in the 'roles and responsibilities' paragraph (lines 786-792). Unfortunately, by referring to roles and responsibilities, the key point of tasks as a useful focal point and appropriate unit of analysis risks getting lost in all the other bells and whistles of the NICE framework. I suggest changing the 'target roles and responsibilities' column in the 'notional organizational profile template' table to 'target tasks' and changing the paragraph on roles and responsibilities (lines 786-792) to a short summary of the NICE guidance referred to above (i.e. section 2.1 of NIST SP 800-181r1 and section 3.1 and section 3.2 of 'Task, knowledge, and skill statements authoring guide for workforce frameworks').

The level of abstraction in the implementation examples should be in line with that of the tasks given in the NICE framework. From a global perspective, this seems to be the case for most implementation examples. As to update frequency, this could be a rolling release process, whereby the important factors are to be critical about preventing excessive overlap with existing task statements and in ensuring easy referencing of examples. Changes to wording in task statements should be fine, as long as the gist remains the same and as long as there is sufficient clarification that the task statements are intended to be interpreted from a 'broad church' perspective (i.e. an ~80% similarity score implies a match with a task description). Such an approach is useful as it makes it easier to align on implementation examples and related NICE task definitions as ILOs (intended learning objectives) in cyber-security education. This should also encourage linking theory and practice. In order to ensure alignment with the NICE framework, and to prevent a proliferation of examples, if any suggestions are accepted from the community, they should go through the NICE framework process, where overlap and general applicability can be checked. This should also serve as a way of ensuring that the NICE framework and cyber-security education in general remain relevant to practice.

As to concrete improvements to the examples, besides reviewing them to align with the NICE framework principles for task statements, I also think it is important to consider the inclusion of examples that cover the intersection of the fields of supply-chain risk management and secure software development. Specifically, the guidelines in NISTIR 8397 on the verification of software are a fertile ground for inspiration. Potential example directions include: threat modelling a system, writing security-related test cases, and performing static and dynamic analysis. Such activities can significantly improve the quality of components, products, and services that make up the complex supply chains described in the CSF draft. (Note that while ID.RA-05 Ex1 includes a mention of threat modelling, many of the other subcategories of ID.RA relate to threat modelling in some way. Also note that line 598 of the CSF draft contains a reference to the non-existent PR.PS-07, which should probably say PR.PS-06.)

A final minor comment: For the background colour used to identify the function areas in the online tools an in the PDF, at least some colours seem to fail the WCAG AAA color contrast check for normal text.