[redacted] [redacted]
**Subject:** Feedback for CSF 2.0 Core - Discussion Draft
**Date:** Tuesday, July 18, 2023 10:01:22 AM

---

Dear NIST CSF Editors,

Please find, below, some feedback for the NIST discussion draft, published April 24, 2023.

For context, I am a JPMChase employee that works in the *Cyber and Technology Controls* (CTC) function, which is part of the broader *Enterprise Technology* organization. My Team operates within *Governance, Risk & Controls* (GRC) and we report to our CISO, Pat Opet, who in turn reports to our CIO, Lori Beer.

Our specific responsibility is to build the "foundation" of our internal Cybersecurity and Technology Controls. We have been using the CSF since before 1.1 was finalized, although we now implement it via the *Cyber Risk Institute's* Financial Sector Profile. One of the things that my Team has done has been to "build out" our implementation of the CRI Financial Sector Profile to make it a *Cyber and Technology* control framework [as opposed to purely cyber] on the basis of the fact that sometimes it isn't clear which framework should be applied – and the effort cost of trying to keep 2 discrete structures in lock step is significantly greater than building and maintaining an holistic framework. Happy to share details of that with you, should you be interested.

Please don't hesitate to reach out if you require clarification or if you would like to discuss further. I am based in the UK (Eastern Standard Time +5), but have access to e.g. Zoom or Teams if you would like to discuss in person.

Thank you

Clive

1. **Generic Observation - Threat / Risk Model**
   This observation calls out not an issue with the CSF in or of itself, but makes the observation that a control framework and the supporting catalogue of controls exists for the primary purpose of mitigating threats and risks. In order for a reader to successfully evaluate the fitness-for-purpose of the CSF, it would be necessary to conduct some form of tabletop exercise against a threat model in order to establish that the control framework had the correct elements, structure and scope. This is not the case. Please note, this observation is intended to encourage NIST to consider development of a bespoke "partner" threat model to the CSF, rather than simply adopting an existing alternative such as MITRE ATT&CK. Whilst it may seem obvious to defer to something like MITRE, the challenge NIST faces would be to ensure that the threat model correctly aligns with the intended scope of controls. ATT&CK [and essentially any other existing, well-know threat model] tends to be implicitly tailored for a given purpose and thus may not be a good fit for the CSF.

In a slight alternative to the above, perhaps NIST would consider developing a “CTF” or similar [Cyber Threat Framework], with a corresponding mapping of e.g. MITRE ATT&CK, ISF IRAM2, STRIDE, DREAD, PIZZA, etc., etc., to give symmetry to the CSF and the existing IR mapping. It is only with an established threat framework to refer back to that we can with confidence state that the CSF is both functionally complete and robust.

2. **Generic Observation – Lack of Informative References**
   The shared draft of CSF 2.0 does not include proposed Informative References. At the time of review it was assumed that this was simply to allow focus on the structure, but information shared by the Cyber Risk Institute suggests that NIST might well deprecate this structural element of the Framework. Speaking on behalf of JPMorganChase, those IR’s are among the most important elements of the structure. We use them to “bring in” the actual control language from the industry source materials [in a de-duplicated way] and then employ that content as ingredients to the development of our internal control language. It gives us the confidence that we are building on the collective wisdom of a broad community. If NIST elect to drop IRs, then it becomes that little bit harder to take the CSF forward into an “actionable framework”, since it remains at an altitude that is difficult to convert to measurable controls.

3. **Generic Observation - Lack of Holistic Model**
   The definition and focus of the final draft still excludes references to broader, non-cyber technology practices (for example Operations, Innovation, Quality Management, Program and Project Delivery). Whilst it is entirely reasonable to exclude these from a *Cybersecurity* framework, it would be helpful if the finished version indicated how/where these additional items were intended or conceived to align, especially as this model now includes "Supply Chain" (i.e. 3rd Party Risk, ID.RA-08) and *excludes* these additional *technology* controls.

4. **Generic Observation - Cross-Check/Feedback**
   One design question/consideration that might be worth evaluation is ‘point of application’, which is to say, where in the lifecycle encapsulated within the Framework would we expect to see a control topic implemented for maximum effect? This asks not whether a control is applicable or achievable, but where in the framework the activity might reasonably be expected to occur? For example: V.OC-5 (formerly ID.BE-1/ID.BE-4) provides the requirement that: "Critical outcomes, capabilities and services that the organization relies on are determined and communicated." Observations on this element:-

   a. There may be an argument for moving this to the *Identify* Function on the basis that the above could be written as: "Critical Outcomes, capabilities and services that the organization relies on are ***identified***, documented and communicated." This seems very much like an *identification* activity, not a *governance* activity, unless the focus is upon governance of that overall process, in which case the description is slightly off.

   b. It might be worth thinking about this aspect of the framework as a "through-line" between Functions. For example, if it is so important that we *identify* critical outcomes, capabilities and services, why do we not have a corresponding requirement to *Protect* them, *Monitor* them, or know how to *Recover* them in the event of a degradation? This

general principle - that the most important elements of the framework should be expressed in each *Function*, is generally applicable. It might be helpful to identify and illustrate the model with these logical connections between Functions, creating, in effect a virtualization of the virtuous circle of (Identify Requirement/Design/Implement/Operate/Monitor), so as to enhance the framework with a degree of self-improvement as a feedback loop.

As a general design principle, we would suggest an evaluation of controls on this basis and offer the above example as a candidate for relocation.

5. **Generic Observation - Design Requirements / Review Against Requirements**
   When a developer asks a user to test code they have written, testing is typically performed "against requirements". In the case of the CSF, we have been asked to review the document 'blind'. We can assess it from the perspective of "does it say the sort of things we want it to say", but it is much harder to assess whether or not it adheres to the design rules by which it has been drafted. This is relevant for a number of reasons… A very significant number of the 1.1. Sub-Categories have been marked "Dropped (moved to FF.CC-nn)", showing that the structural arrangement of the framework has changed… but there are no "design rules" to help us understand how these changes were introduced. The "cover note" (page 2 of the PDF) states, "It reflects changes - some larger, some smaller - across the CSF 1.1 Core through reordering, merging, and otherwise modifying the cybersecurity outcomes of the Framework." The challenge here is for us to understand what the desired outcome was for "reordering, merging, and otherwise modifying the cybersecurity outcomes".

   By way of concrete example… consider the new sub-Category ID.AM-08, which we see includes the content from the following legacy sub-Categories:-

   a. PR.DS-3: Assets are formally managed throughout removal, transfers, and disposition
   b. PR.IP-2: A System Development Life Cycle to manage systems is implemented
   c. PR.MA-1: Maintenance and repair of organizational assets are performed and logged, with approved and controlled tools
   d. PR.MA-2: Remote maintenance of organizational assets is approved, logged, and performed in a manner that prevents unauthorized access

   In this case, NIST have chosen to merge an *asset lifecycle* control with a *system development lifecycle* control and a *technology maintenance* control - three *very* different activities. It simply isn't clear from the provided documentation as to the process by which these elements were selected for merge. For example, if *Maintenance* is part of an asset lifecycle, why isn't the management of its vulnerabilities? Without a better understanding of the "design requirements" (the review process by which merger candidates were evaluated) it isn't objectively possible to see if NIST are following a structured approach or if these changes are merely arbitrary in nature.

6. **Generic Observation - Inconsistent Edits (RS.CO-2 & RC.RP-01)**

In some cases - as with the above referenced examples - edits appear to be contradictory. For example, RS.CO-2 has been changed from "Incidents are reported consistent with established criteria" until it reads, "Internal and external stakeholders are notified of incidents, as required by law, regulation, or policy". There is nothing in *2.0 proposal* statement that is not implicitly present in the *1.1final* published statement.

By contrast, RC.RP-01 has been changed from "Recovery plan is executed during or after a cybersecurity incident" until it reads, "The incident recovery plan is executed".

The first example expands the previous text, potentially being more specific, but adds no new value. The second example contracts the original text, but doesn't necessarily improve clarity given that most readers would likely not conceive of invoking a recovery plan except in response to a qualifying event.

There is no discernible pattern to explain how these two sub-Categories have been edited - in "opposite directions". This makes it much, much more difficult to evaluate the validity of the changes as a whole, since they are so clearly inconsistent.

7. **Generic Observation - Positive Control Monitoring (Gap)**
   In the *Detect* function, (perhaps the most appropriate Categories would be either *Continuous Monitoring* or *Detection Processes*), there is presently no Sub-Category designed to *test the effectiveness of deployed controls on an on-going basis.* GV.RM-08 discusses the effectiveness of the overall *strategy*; ID.IM-03 discusses the need to apply lessons learned as improvement opportunities (implicitly after something has not performed to design/expectation), but there is nothing 'in the middle'. There might be a case for a *Detect* function entry that requires adopting organizations to design, for every deployed cybersecurity and technology control, the means to monitor that control to ensure that it is performing as intended.

   This can be significant, because if we limit our detection processes to "anomalous events", then implicitly we're waiting until something has gone wrong before we look at it [which might be too late]. Monitoring expected performance for deviations has the potential to act as a cross-check for our ability to correctly identify anomalous events.

8. **Generic Observation - Lack of Segregation as a Control Principle (Gap)**
   It might be beneficial to have a generic requirement for organizations to identify and enforce all areas where operational segregation is required. Not specifically from an access permissions perspective, but also: from the principle of not allowing developers to approve their own changes; from a discussion of toxic combinations at a role-based level. Probably aligns best with PR.PS-01, but is not explicitly mentioned…

9. **Generic Observation - “Bottom Up” as opposed to “Top Down”**
   Without specific knowledge of the process taken to develop the framework, a review of the document as drafted rather suggests that the sub-Categories (and Categories) have been selected based on a top-down rationalisation of the structure of the 1.1Final edition, as opposed to a detailed examination of the candidate *Informative References* and from there

the delivery of a “bottom up” build of a framework. The advantage of the bottom-up approach is that it gives the assurance that all the relevant parts of the underlying Informative References are either included, or that known/intended framework gaps are left (allowing adopters to fill in if required).

The “bottom up” approach is more effective when organisations need to take the framework and actually *apply it,* because of course that application requires the development of more detailed expressions of control expectations, which may typically be derived from industry materials referenced via the *Informative References*. Developing a framework in isolation from the content of those *Informative References* invites a fundamental disconnect.

Bottom-up is inherently more effective than top-down.

10. **Risk Appetite and Risk Tolerance (GV.RM-03 )**
"Risk appetite and risk tolerance statements are determined and communicated based on the organization's business environment (formerly ID.RM-2 and ID.RM-3)… Agree with the principle being expressed here, but not the form of words, which I think is too imprecise. First, as written there is no clear explanation as to why we need both "appetite" and "tolerance" - how are these different? Secondly, the requirement needs to be a little more rigorous, specifically expressing the need for risk appetite to be expressed in a way that is deterministic - i.e. that can be independently and objectively tested and that can derive an explicit outcome of the nature "The organization is within risk tolerance" or "The organization has exceeded risk tolerance in [this/these] area[s]…" It isn't enough to just state an appetite: it has to be measurable and actively managed/monitored.

11. **Sensitive data and corresponding metadata are inventoried and tracked (ID.AM-07)**
This potentially spans two quite distinct activities: The identification and registration of sensitive data is clear an *Identify* function activity, but the tracking is almost certainly a *Detect* function activity.

12. **Separating System *Development* from System *Lifecycle* (ID.AM-08)**
(This is an excellent example of why TCF 2.0 makes sense for JPMC… because in the new NIST CSF, activities that would traditionally be aligned with the *Technology Development* CDA are here merged with those aligned to the *Technology Asset Management* CDA, in a likely confusing way.)

This Sub-Category seeks to merge the *technology ownership lifecycle* with the *system development lifecycle*. The former needs to emphasise elements such as inventory management, maintenance support, clearing/purging data and disposal; the latter needs to emphasise design, architecture, testing and change. By bringing these two elements together, the framework risks conflating two large/significant areas of control activity in to a single entity: a best this could deliver an unwieldy result; at worst it could degrade the level of focus applied, resulting in weaker overall coverage.

Additionally, the inclusion of both PR.MA-1 and PR.MA-2 add to the complexity of this control

topic, since the association emphasizes the relationship between maintenance and the lifecycle, whereas from a cyber control framework perspective, it is probably more important to associate maintenance with ensuring that vulnerabilities are patched and/or that technology is kept on current releases in order to ensure on-going vendor support.

13. **Exceptions to Security Measures (ID.RA-10)**
    Nice to see the inclusion of "non-compliance exceptions" in the framework, but to be fully effective this needs to be more formally linked to ID.RA-08, so that the exceptions are also assessed for risk. A simple wording change to the text (for example to, "Exceptions to security measures are reviewed, tracked, assessed for risk, prioritized and compensated for"… Without a risk dimension, rigorous application of the framework could result in a very large number of NCE's being identified, which then become "noise" that have to be managed, masking the "real risks" by virtue of their numbers.

14. **Management of the Data Lifecycle (PR.DS-09)**
    Might be worth considering adding "storage" or "retention" to the identified lifecycle stages, since organizations are increasingly applying more and more sophisticated knowledge management techniques to data, which in turn means that the lifespan of data is extending and that data may be kept in stored or archival form for significant periods.

15. **Software is patched, updated, replaced (PR.PS-02)**
    This sub-Category conflates two important and different objectives: a requirement to update/replace software to (for example) ensure that it remains within vendor support programs; but also to eliminate vulnerabilities (see old version, which explicitly calls out vulnerabilities). The former practices should be planned, managed and carefully controlled with (relatively) slow-moving, managed practices. The latter must necessarily be much more responsive and may require emergency patching practices. In other words, the surrounding controls for these tasks are necessarily very different, so merging them introduces the risk that standardization on one set of practices could either hamper emergency patching, or reduce the control maturity applied to BAU maintenance.

16. **Generic Observation - Patch Windows (Gap) [follow-on from PR.PS-02]**
    There is an argument to be made for requiring organizations to have formally declared "patch windows", even for technology designed/implemented to run 24x7. The issue here is that if you have technology designed for always-on operation (e.g. an on-line retail banking platform), then in the event that a vulnerability is identified that has the potential to put the platform at risk, you want to get that vulnerability patched as quickly as possible. One critical step in this process is to think in terms of pre-scheduled maintenance windows - periods of time where the system users known in advance that the system may be taken down for emergency maintenance (e.g. 02:00-04:00 on Sunday mornings). When in place, this will lead to shortened exposure times.

17. **Log records are generated (PR.PS-04)**
    The language has changed in a way that degrades the effectiveness of this sub-Category - in the prior version the objective was to capture "log records", which implicitly *does not* attempt

to predict which records might relate to a cyber event. The new version narrows this to "Log records … for cybersecurity events". The issue here is that when investigating/triaging/recovering from a cyber event, log records from all sorts of unrelated sources might be able to help piece together "what happened, when". Narrowing the scope as proposed introduces a risk that useful/necessary log data may not be collected, hampering potential recovery. (In simple terms, you may not know which logs you need until *after* the event, by which time it might be too late.)

18. **Backups of "platform software" (PR.PS-06)**
    As written this excludes e.g. configuration files from consideration, yet the functionality of many software applications today can be significantly altered by changes to configuration files [not least of which - extreme example - might be a configuration file containing license keys to unlock the software…] "… and associated files" would be a nice addition, since it can be crafted to express the need to archive documentation alongside the code it describes, which in turn can lead to the documentation being maintained.

19. **Malicious Code Detection (DE.CM-01)**
    All references to the identification of malicious code are based around the *Detect* Function. However, there are opportunities for this to be present at earlier phases in the lifecycle, including product acquisition, configuration (PR.PS-01) and system development (ID.AM-08). Inclusion or reference here would allow for *Protective* (i.e. *preventative*) measures, since, for example, scanning code before production deployment helps reduce the risk of malicious content being deployed to an operational state. Like ID.AM-08, this sub-Category has become needlessly broad.

20. **Personnel Activity and Technology Usage Monitoring (DE.CM-03)**
    The change to the language of the description implies that "only the activity of personnel" will be monitored for potential cybersecurity events, where in reality we would expect to monitor personnel, third parties, clients/customers and in fact any actor accessing organizational technology. The original language was cleaner.

21. **Computing hardware and software and their data are monitored (DE.CM-09)**
    This is another sub-Category which seeks to merge multiple legacy sub-Categories to a single entity, but in this case it includes PR.DS-8, "Integrity checking mechanisms are used to verify hardware integrity". Unlike all the other proposed merge candidates for DE.CM-09, PR.DS-8 is primarily concerned with *hardware reliability*, not *nefarious activity*. This control asks adopting organizations to use features such as checksums (e.g. CRCs, Cyclic Redundancy Checks), memory "

22. **Integrity of Backups and other Restoration Assets (RC.RP-03)**
    It isn't clear how this sub-Category adds material value to the framework. First, there is already a requirement to regularly test backups, expressed in PR.DS-11. Secondly, as written RC.RP-03 explicitly states an expectation that the integrity of backups is verified *before using them for restoration.* The problem is, this point in time (for verification) is too late. If we wait until immediately prior to attempting a restoration and the integrity test fails, what then? It's

too late to take another backup...

23. **Post-Incident Operational Norms (RC.RP-04)**
    Reading this Sub-Category definition in isolation it is almost impossible to discern precisely what it is asking for. Is it a point-in-time recovery position? Is it the restoration of "BAU operational capacities and throughputs"? Is it the application of risk management practices and organizational priorities as factors when considering how to scope/determine post-incident capabilities? Is it something else? Is it a combination of things?

    If there is a desire to express some element of "post incident" practices - which seems entirely reasonable, then it might be more helpful to break this down slightly (or, consider the adoption of lower-level qualifiers... such as *Diagnostic Statements*) to help clarify the intent.

24. **Public Relations (RC.CO-01)**
    This is neither a cybersecurity nor a technology control. I would suggest that it is either dropped in its entirety (my preference), or instead be subsumed in to RC.CO-3.

25. **Reputation Repair (RC.CO-02)**
    This is neither a cybersecurity nor a technology control. I would suggest that it is either dropped in its entirety (my preference), or instead be subsumed in to RC.CO-3.