

Public Comment on NIST SP 800-70r5 ipd

by Mike Sanderson

January 16, 2026

Introduction

My comments concern my lifelong use of the Mac platform including as a software developer for Apple's iPhone since 2013. I am aware of the importance of having a secure development environment, because without a secure development environment, the software produced is not secure. I downloaded the MacOS 26 (Tahoe) checklists from mSCP and DISA; the underlying steps were largely the same.

I found the checklists provided by mSCP for MacOS 26 to be substantially different and substantially less useful than the text in NIST SP 800-70r5 ipd described them.

- Confusion in documents between checklists that are “machine readable” and “machine executable” as these are two distinct categories distinct from prose, with very different implications
- Lack of discussion of “Hybrid” checklists, which most of the mSCP checklists are.
- Overwhelming complexity and presentation of the MacOS 26 mSCP checklists from NCP Repository.

More About me:

I am a writer and application software developer, and since 2013 have worked on iPhone apps, working on projects with high security requirements. This includes overseeing technical development for a launch partner for Apple Wallet NFC in 2015, working on a secure messaging app with high-security requirements, handling sensitive user health data from HealthKit, and doing independent security research. I am based in New York City and have delivered multiple presentations to professional development groups since 2016. My comments come from the software development perspective, including extensive experience with API design and use.

“Automated” Checklists - Machine-Readable vs. Machine Executable

There is a significant difference between machine-readable and machine-executable.

This can be seen in the use of terms throughout, somewhat interchangeably. :

- “machine-readable” (648, 667-668, 918, 1509, 1576)
- “machine-executable” (217-218, 464)

- “machine-consuamable” (220,221)
- “machine-automated” (385)

The terms “machine-consuamable” and “machine-automated” are ambiguous and should not be used to distinguish types of checklists, though in a specific context they might make sense.

There is a significant difference between machine-readable and machine-executable. While technically if executable is can be considered readable, and even prose is technically machine-readable, generally:

- “machine-readable” means that it is in some standard format, with validations, that a tool expecting it could use, but might contain data, including, strings, references, or executable commands.
- “machine-executable” means commands that can be executed directly a machine that can execute commands. as part of its shell. Its shell is provided with the command, the most common being “bash” or “zsh”.

Generally, the goal is to provide fully machine-readable checklists, in a standard format, which have prose fields that explain each item.

Hybrid Checklists

Checklists can be hybrid. A prose checklist could include in the prose specific commands that could be executed. This would still be a prose checklist, unless the command is isolated in its own field; that is, if the command is in the same text box as instructions such as “Execute the following command:” it is a prose checklist that provides executable commands, but is not an automated checklist.

About “machine-readable,” technically, anything, even a text file or PDF. However, machine-readable generally would mean the data is in a standard data structure such as xml, json, or yml, and information is included in standard forms in specific fields. If some fields do not include standard data, but instead include prose or unstructured lists, then the data is partially machine readable.

Both of the checklists returned for macOS 26 (Tahoe), including the checklist from mSCP, are partially machine readable. Even where xml or xlsx files are returned, there is prose mixed in with commands, multiple values included in a single field, which could be marginally acceptable but there is not

The mSCP does include a more robust, fully-machine-readable format in their GitHub repository under “rules”. The structured data for each rule appears to (finally) not include prose in the provided executable commands for the “check” field, although the “fix” field still has prose.

These github rules also have a field for “mobileconfig” and “mobileconfig_info”, which provides structured information for the standard Apple platform configuration file, originally for iOS but now also for mac, a XML-based “Property List” file with the extension mobileconfig. A general tool would have to have knowledge of this file type in order to make use of these mSCP rules.

The mSCP, however, does provide scripts in its scripts folder, and the utilities folder within it, that could be circulated much more widely — if the format used by mSCP was a more widely understood format.

Overwhelming Complexity of the Checklist

As a computer programmer who has worked on the Mac platform my entire life, I struggled to figure out how I might be able to use these. As described in 495-497 and 510-512, I should be in the core of Stand-Alone (or small office/home office).

When I downloaded the mSCP, it was a Zip file that then expanded into 7 subfolders. There is a readme, but it is one simplistic paragraph and then The first question is the meaning of the 7 subfolders. Only after going through them, would it be possible to realize that each holds the complete checklist multiple times — something that could have been explained at the onset. The user would then have to figure out which format and which specific checklist they want to start with. All of these steps are elided in the document NIST SP 800-70r5 .

The premise is that someone in a stand-alone environment wants a final checklist of all possible hardening steps (or rather, that they want all possible combinations). Yes, providing a final checklist of all possible hardening steps is useful, but the premise of downloading this final checklist and then editing it is a task of unreasonable complexity.

What individuals need to do is a description of various kinds of security needs, and then be provided checklist items that provide hardening steps for those needs.

Concrete Example of FIPS 140

For an example provided in FIPS 140, which is the intersection of the above issues. Take 714-717:

Organizations should tailor the checklists to reflect local rules, regulations, and mandates; for example, federal civilian agencies would need to ensure that checklists reflect compliance with encryption requirements in Federal Information Processing Standards Publication (FIPS) 140, Security Requirements for Cryptographic Modules.

The only way to identify items dealing with FIPS in the MacOS 26 checklists provided by mSCP is to do a plain text search for “FIPS”. Two items clearly apply. If these items were tagged as

relating to FIPS, that would be an example of how including machine-readable fields could allow users with that need to be able to use the checklists to identify checklist items.

The bespoke mSCP rules format has a “tags” field, but it does not include anything related to FIPS 140, nor is it clear what that standard “tag” for FIPS 140 would be. No reputable automated tool would use a plain-text search of user-facing prose to present security information, but if the information was included in a machine-readable field in a standard format, it would be fine.

The goal should be to provide checklists that are fully tagged and organized in a standard machine-readable format, one in which tools or automated systems can identify the significance of checklist items and present it back to the user.

Conclusion

The use of checklists for hardening is crucial, and the National Checklist Program should be a powerful way to share expertise. However, what is provided is currently overwhelming, and to be useful for Stand-Alone users (and many others) there need to be intermediate processes to help different users understand how to select checklist items that relate to their security needs. There also needs to be a clearer distinction among prose checklists, machine-readable checklists, and machine-executable commands, and for both prose, machine-executable commands, and structured data like tags and formats to be included in machine-readable fields.

—Mike Sanderson
Brooklyn, New York
January 16, 2026