NIST Special Publication 800
NIST SP 800-70r5

National Checklist Program for IT Products

Guidelines for Checklist Users and Developers

Stephen D. Quinn
Blair Heiserman

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-70r5

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-70r5

National Checklist Program for IT Products

Guidelines for Checklist Users and Developers

Stephen D. Quinn
Computer Security Division
Information Technology Laboratory

Blair Heiserman
Information Technology Security & Networking Division
Office of Information Systems Management

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-70r5

May 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Authority
This publication has been developed by NIST in accordance with its statutory responsibilities under the Federal Information Security Modernization Act (FISMA) of 2014, 44 U.S.C. § 3551 et seq., Public Law (P.L.) 113-283. NIST is responsible for developing information security standards and guidelines, including minimum requirements for federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

Publication History
Approved by the NIST Editorial Review Board on 2026-04-23
Supersedes NIST SP 800-70r4 (Feb 2018) https://doi.org/10.6028/NIST.SP.800-70r4

How to Cite this NIST Technical Series Publication
Quinn SD, Heiserman B (2026) National Checklist Program for IT Products: Guidelines for Checklist Users and Developers. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-70r5. https://doi.org/10.6028/NIST.SP.800-70r5

Author ORCID iDs
Stephen Quinn: 0000-0003-1436-684X
Blair Heiserman: 0009-0003-8779-6231

Contact Information

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
checklists@nist.gov

Additional Information

Additional information about this publication is available at https://csrc.nist.gov/pubs/sp/800/70/r5/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

A security configuration checklist is a document that contains instructions, procedures, or machine-readable and executable content to configure an IT product to a specific risk posture for an operational environment, verify that the product has been configured properly, identify unauthorized configuration changes to the product, and/or produce artifacts that show the security posture of the product. Using these checklists can minimize the attack surface, reduce vulnerabilities, lessen the impact of successful attacks, and identify changes that might otherwise go undetected. NIST established the National Checklist Program (NCP) to facilitate the generation of security checklists from authoritative sources, centralize the location of checklists, and make checklists broadly accessible. This publication explains how to use the NCP to find and retrieve checklists and describes the policies, procedures, and general requirements for participation in the NCP.

Keywords

benchmark; change detection; checklist; information security; National Checklist Program (NCP); Security Automation; secure configuration; security configuration checklist; Security Content Automation Protocol (SCAP); software configuration; vulnerability.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Document Conventions

This document was created for current and potential checklist developers and users in both the public and private sectors. Checklist developers include IT vendors, consortia, industry, government organizations, and others in the public and private sector. Checklist users include end users, system administrators, and IT managers within government agencies, corporations, small businesses, and other organizations, as well as private citizens.

It is assumed that readers of this document are familiar with general computer security concepts.

Trademark Information

Patent Disclosure Notice

Table of Contents

List of Tables

List of Figures

Acknowledgments

The authors wish to thank all individuals who have contributed to this and prior revisions of SP 800-70. Specifically, the authors thank the co-authors of the previous revision: Murugiah Souppaya and Melanie Cook, formerly from the National Institute of Standards and Technology (NIST), and Karen Kent of the Trusted Cyber Annex.

Key contributors to this revision include Bob Byers and Bob Gendler from NIST, Karen Kent and Matthew Smith from the Trusted Cyber Annex, Jack Vander Pol and Daniel K. Harris from the Naval Information Warfare Center (NIWC) Atlantic, Greg Witte from Palydin LLC, and Brad Hoehn from Electrosoft.

Executive Summary

A security configuration checklist (also called a lockdown, hardening guide, or benchmark) is a series of instructions or procedures for securely configuring an IT product to a particular risk tolerance for an operational environment, verifying that the product has been configured properly, and/or identifying unauthorized changes to the product. The checklist may be for a commercial, open-source, or government-off-the-shelf (GOTS) IT product.

Checklists can comprise a mix of templates, automated scripts, patch information, Extensible Markup Language (XML) files, and other procedures. Typically, checklists are created by IT vendors for their own products; however, checklists are also created by other organizations, such as academia, consortia, and government agencies. The use of well-written, standardized checklists can markedly reduce the attack surface and vulnerability exposure of IT products.

NIST maintains the National Checklist Repository, a publicly available resource of security configuration checklists for IT products. The repository, https://checklists.nist.gov, contains metadata describing each checklist and links to the website where a checklist is hosted. Having a centralized checklist repository makes it easier for organizations to find current, authoritative versions of security checklists and to determine which ones best meet their needs.

This document is intended for users and developers of security configuration. For checklist users, this document makes recommendations on how they should select checklists from the NIST National Checklist Repository, evaluate and test checklists, and apply them to IT products. For checklist developers, this document sets forth the policies, procedures, and general requirements for participation in the NIST National Checklist Program (NCP).

Major recommendations made in this document for checklist users and developers include the following:

- Organizations should apply checklists to operating systems and applications to reduce the number of weaknesses that can be exploited and to lessen the impact of security breaches, if they occur.
- When selecting checklists, users should carefully consider each checklist's degree of automation, source, use of standards, and other relevant characteristics.
- Checklist users should consider their operational environments when selecting checklists and should customize and test checklists in a non-production environment before applying them to production systems.
- Checklist creators are encouraged to adopt a "catalog of controls" approach for products to facilitate custom checklist reuse.
- IT product vendors are strongly encouraged to develop security configuration checklists for their products and contribute them to the NIST National Checklist Repository.
- Checklists should be incorporated into continuous monitoring and configuration results and deviation monitoring used in automated data feeds for near real-time posture checks.

1. Introduction

There are many threats to users' computers, and new vulnerabilities in IT products (e.g., operating systems, applications) are discovered daily. Patches may not be immediately available for new vulnerabilities, causing the need to rapidly deploy temporary mitigations through product reconfiguration until patches are available. Moreover, restrictive security settings in IT products may be disabled by default to ensure maximum functionality and interoperability for a wide variety of users, which means that many IT products may be immediately vulnerable in their default configuration. It is a complicated, arduous, and time-consuming task to define a reasonable set of security settings for IT products, even for experienced system administrators.

One simple yet effective tool is the security configuration checklist (also called a lockdown, hardening guide, or benchmark). NIST developed the National Checklist Program (NCP) for IT products to facilitate the development of security configuration checklists and meet the requirements of the Cyber Security Research and Development Act of 2002 (Public Law 107-305) (CSRDA) [1].

There is no checklist that can make a system or product completely secure, and using a checklist does not eliminate the need for ongoing security maintenance, such as patch installation. However, using a checklist that emphasizes hardening systems against software flaws (e.g., by applying patches and eliminating unnecessary functionality) and securely configuring systems will typically reduce the number of ways in which the systems can be attacked, resulting in greater levels of product security and protection from future threats. Organizations should use a risk-based approach, and checklists should be tailored by each organization to meet its particular security and operational requirements.

1.1. Purpose and Scope

This document describes the use, benefits, and management of checklists and checklist control catalogs as well as the policies, procedures, and general requirements for participation in the NCP.

1.2. Document Organization

Section 2 provides an overview of checklists and the advantages of the NCP.

Section 3 provides additional details on predefined checklist operational environments that are used in the NCP to help developers create checklists that are consistent with security practices. The material presented in Sec. 3 can also help checklist users select the checklists that best match their own operational environments.

Section 4 contains information for potential checklist users. It describes how to use the NCP to find and retrieve checklists that best match their identified needs. It also provides guidance on how to implement checklists, including how to analyze the specific operating environment and then tailor checklists as applicable.

Section 5 provides guidance for current and prospective checklist developers, including procedures for preparing and submitting a checklist to NIST for inclusion in the checklist repository.

The References section lists the documents cited throughout this document.

Appendix A describes the programmatic and legal requirements that must be satisfied to participate in the NCP.

Appendix B provides the NCP participation and logo usage agreement form.

Appendix C contains steps for automating the NIST Cybersecurity Framework with checklists.

Appendix D lists the acronyms used in this document.

Appendix E presents a glossary of the terms used in this document.

Appendix F provides a log of changes made since the last version of this document was published.

2. NIST National Checklist Program

This section provides an overview of the NCP and the contents of security configuration checklists, which improve the base level of security for an organization.

2.1. Overview of the NCP

Organizational checklists can vary widely in terms of purpose, quality, usability, and the level of security they provide. They may also become outdated as software updates and upgrades are released. Without a central checklist repository, finding security checklists can be difficult. NIST established the NCP to facilitate the development of security checklists for IT products and to make checklists more organized and usable. The NCP's broad goals are to:

- Facilitate the development and sharing of checklists by providing a formal framework for vendors and other checklist developers to submit checklists to NIST
- Provide guidance to developers to help them create standardized, high-quality checklists that conform to common operational environments
- Help developers and users by providing guidelines for documentation and usability
- Encourage software vendors and other parties to develop checklists
- Provide a managed process for reviewing, updating, and maintaining checklists
- Provide an easy-to-use repository of checklist information
- Provide checklist content in a standardized format
- Encourage the use of automation technologies for applying checklists.

Federal agencies are required to use security configuration checklists from the NCP when available. In January 2017, Part 39 of the Federal Acquisition Regulation (FAR) was updated. Paragraph (c) of section 39.101 states,

> In acquiring information technology, agencies shall include the appropriate information technology security policies and requirements, including use of common security configurations available from the National Institute of Standards and Technology's website at https://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated. [2]

2.2. Security Configuration Checklists

A *security configuration checklist*[1] (or simply, "checklist") is a document that contains instructions, procedures, or machine-readable and executable content to configure an IT

[1] This may also be referred to as a lockdown guide, a hardening guide, a security guide, secure configuration, security technical implementation guide (STIG), a baseline, or a benchmark.

product to a specific risk posture for an operational environment, verify that the product has been configured properly, identify unauthorized configuration changes to the product, and/or produce artifacts showing the security posture of the product. The IT product may be commercial, open source, government-off-the-shelf (GOTS), or developed in house.

Using well-written, standardized configuration checklists can reduce the vulnerability exposure of IT products and help secure systems. Checklists can be developed by IT vendors and other organizations with technical competence in IT product security. A security configuration checklist might include any of the following:

- Automated content that sets or verifies security-related settings (e.g., executables, scripts, security templates, SCAP XML files)[2]
- Documentation that guides the checklist user to manually configure an IT product
- Documents that explain the recommended methods to securely install and configure a device
- Rule files that use policy and programmatic documents to define recommended settings for technical controls, such as auditing, authentication mechanisms (e.g., multi-factor authentication, passwords), and firewall security

Not all instructions in a security configuration checklist need to solely address security settings. Checklists can also include administrative practices, such as enabling energy-saving features or specialized security functions (e.g., looking for indicators of compromised artifacts on a host).

Typically, a checklist is deployed as part of enterprise management to apply settings across all systems. When done manually, a system administrator or end user follows the instructions in the checklist to configure or verify that a product or system has implemented the security controls in the checklist. Through enterprise or local implementation, the system administrator may need to include modifications or deviations to the checklist to ensure that the local security policy enables the risk posture needed for the purpose of the device.

Security checklists are intended for a variety of devices and software, including:

- General-purpose operating systems and mobile operating systems
- Common applications, such as email clients, web browsers, word processors, personal firewalls, and antivirus software
- Infrastructure devices, such as software-as-a-service (SaaS) providers, infrastructure-as-a-service (IaaS) providers, platform-as-a-service (PaaS) providers, virtualization platforms, routers, switches, firewalls, virtual private network (VPN) gateways, intrusion detection systems (IDSs), wireless access points, and telecommunication systems
- Application servers, such as Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), proxy and web servers, Simple Mail Transfer Protocol (SMTP), and database servers

[2] More information about SCAP can be found at https://scap.nist.gov/ and SP 800-126.

- Other network devices, such as Internet of Things (IoT) devices, scanners, printers, and copiers
- Artificial intelligence (AI) systems, including hardware and software components comprising the stack

2.3. Benefits of Using Security Checklists

Security checklists help users configure IT products securely. Many product vendors offer secure default settings, but this practice is not universal. Applying checklists to operating systems and applications can reduce the number of exploitable weaknesses and can lessen the impacts of any attacks that occur. Using checklists improves the consistency and predictability of system security, particularly in conjunction with user training and awareness activities and other supporting security controls. Additional benefits associated with using checklists include:

- Providing a base level of security to protect against common local and remote threats (e.g., malware, denial-of-service attacks, unauthorized access, inappropriate usage)
- Verifying the configuration of technical security controls specified by the checklist for system assessments (e.g., confirming compliance with certain Federal Information Security Modernization Act [FISMA] requirements) and understanding the exposure caused by misconfigurations
- Providing artifacts of compliance from the verified implementation of controls at the endpoint to support a near real-time understanding of technical security posture and risk
- Creating catalogs of product technical controls, which significantly reduces the time required to research and develop appropriate security configurations for installed IT products
- Allowing smaller organizations to implement recommended practice security configurations
- Reducing the likelihood of public loss of confidence or embarrassment resulting from a compromise of systems (e.g., a major breach of personally identifiable information [PII])

Using security checklists for security compliance purposes can significantly improve overall levels of security in organizations, but using a checklist cannot make a system or product completely secure. Using checklists that emphasize the hardening of systems will typically result in greater levels of product security and potential protection from future threats (e.g., zero-day vulnerabilities). The use of checklist-based configuration assessments as part of ongoing security monitoring will support protection and detection outcomes. Automated tools should continuously verify that systems remain in a hardened state and should flag any unauthorized

configuration changes. This approach supports a dynamic security posture that is in line with NIST continuous monitoring guidance.[3]

While security configuration checklists may be applied at deployment or during periodic assessments, organizations are encouraged to integrate regular near real-time checklist-based automated verification into continuous monitoring programs wherever feasible. This monitoring should be context-aware to accurately measure risk, allow for actionable remediation, and provide an understanding of risk in the context of the system rather than a flat list of pass/fail items, such as documenting known and approved deviations from checklists. In environments that are characterized by frequent change, the continuous validation of configuration states with context awareness enables the earlier detection of drift, misconfiguration, or exposure.

For cloud and hybrid environments, continuous monitoring should correlate checklist violations across the entire cloud stack (e.g., identity, network, data). Full stack continuous monitoring focuses remediation on critical and exploitable attack paths to identify combinations where a misconfiguration or vulnerability represents a direct risk to sensitive data.

IT vendors that configure their products using checklists that adhere to the FISMA-associated security control requirements[4] will provide more consistency in configuration settings within federal agencies. Applying checklists establishes minimum configuration settings, even if the agencies must tailor the checklists for their specific risk tolerances and operational environments.

2.4. Additional Considerations

The following subsections provide additional information related to checklist mapping, acquisition decisions and selection.

2.4.1. Mapping and Acquisition Considerations

Checklists can verify the configuration state of technical security controls for system assessments, such as confirming compliance with certain FISMA requirements, security frameworks (e.g., the NIST Cybersecurity Framework), and security controls.

FISMA also requires each federal agency to determine minimally acceptable system configuration requirements and ensure compliance with them [3]. Federal agencies and vendors of products for the Federal Government should create and share checklists using the NCP. To facilitate FISMA compliance checking for federal agencies, NIST encourages checklist developers to map the security controls delineated in NIST Special Publication (SP) 800-53 to IT product technical controls.[5]

[3] Additional NIST guidance regarding continuous monitoring may be found in SP 800-137 [8] and the NIST Interagency Report (IR) 8011 series at https://csrc.nist.gov/pubs/ir/8011/v1/final.
[4] See the NIST Cybersecurity and Privacy Reference Tool at https://csrc.nist.gov/projects/cprt/catalog#/cprt/framework/version/SP_800_53_5_2_0/home.
[5] Organizations are also encouraged to include information in their checklists that supports mapping to other sets of requirements, such as the Health Insurance Portability and Accountability Act (HIPAA).

2.4.2. Selecting Checklists

Organizations should consider and prioritize product selection based on the availability of security configuration checklists during their IT product selection processes. Checklists serve as guidance rather than a universal prescription, and organizations are encouraged to tailor checklists to reflect their operational context and risk posture. Any deviation from checklist settings should be thoroughly reviewed for potential risks and accompanied by a documented justification with risk acceptance.

NIST recognizes that some checklists are more automated and standards-based than others. For example, non-automated checklists provide prose-based descriptions of how a person can manually alter a product's configuration, while automated checklists are generally machine-executable. The NIST open-source macOS Security Compliance Project (mSCP)[6] is an example of a well-automated checklist package that focuses on standards-based formats to provide security configuration guidance to organizations more quickly and in a machine-readable and executable format. The mSCP continuously curates and updates automated macOS guidance to address the continuous release cycle of the vendor. The latest macOS security baseline content is maintained and updated on the mSCP GitHub page.

Checklist developers, like the mSCP program, take a programmatic approach to generating and using security configuration baselines. Projects like these can be used to create customized security baselines of technical security controls by leveraging a library of rules that are each mapped to requirements in one or more existing security standards, regulations, or frameworks. This approach provides versioning, consistency, and the generation of checklists in standardized formats that are either native to the platform or in widely accepted security configuration standard formats. Unifying, standardizing, and updating security guidance is simplified and radically accelerated, even as new features and versions of the operating system or application are introduced.

Not all checklist generation requires this degree of programmatic sophistication. Automated checklists may be a Group Policy Object (GPO), stand-alone Open Vulnerability and Assessment Language (OVAL) file, kickstart script, shell script, or simple utility that assesses the security settings on a system. No matter the format, developers of automated checklists are encouraged to use the constructs that are most germane to the platform being configured, be transparent regarding their methods, and assign Common Configuration Enumeration identifiers (CCEs) (i.e., globally unique identifiers) to their individual configuration settings.[7]

Another example of an automated checklist is one that fully adheres to the Security Content Automation Protocol (SCAP). These checklists are often referred to as SCAP content and include mappings between low-level security settings and high-level security requirements. These checklists have undergone syntactic testing to ensure adherence to the SP 800-126 SCAP specification using the NIST SCAP Content Validation Tool (SCAPVal) located at GitHub - usnistgov/scapval.

[6] The mSCP is an open-source project that provides a programmatic approach to generating and using macOS security configuration baselines that adopt this recommendation. More information is available at https://pages.nist.gov/macos_security.
[7] The Common Configuration Enumeration (CCE) list may be reviewed at https://ncp.nist.gov/cce

- **Automated checklists**

  When multiple checklists are available for a particular product, organizations should consider the degree of automation and use of standards for each checklist. Generally, automated checklists can be used more consistently and efficiently than others. There may be other significant differences among checklists; for example, one checklist may include software bundled with an operating system (e.g., web browser, email client), while another addresses the operating system only. Another example is the assumptions on which the checklists are based (e.g., operational environment). A user should identify such differences and determine which checklists seem appropriate and merit further analysis. Checklists can be tailored to suit the risk tolerance of the environment in which they are used.

- **Government Checklist Precedence**

  A checklist's source is particularly important for users from federal civilian agencies, who should first search for government-authorized or mandated checklists (e.g., mandated by Part 39 of the FAR [2]). In general, these users should search for NIST-produced checklists, which are tailored for civilian agency use. If no NIST-produced checklist is available, then agency-produced checklists from the Defense Information Systems Agency (DISA), the National Security Agency (NSA), or the Cybersecurity and Infrastructure Security Agency (CISA) should be used. If formal government-authorized checklists do not exist, organizations are encouraged to use vendor-produced checklists. If vendor-produced checklists are not available, other checklists that are posted on the NCP website may be used. Although government-sourced checklists generally consider the necessary compliance criteria (e.g., use of FIPS-140 based encryption when encryption is necessary, access control criteria), they may need further tailoring to fit the specific operational and risk-based needs of the environment in which they are used.

2.4.3. Checklist Considerations

A checklist should be considered a starting point for an organization. Checklists should be customized to match an organization's risk posture and requirements, including specific deviations with documented justifications, risk acceptance, and compensating controls. Unique factors (e.g., organization-specific requirements, existing security controls) may necessitate changes to an organizational checklist. Organizations should carefully evaluate checklist settings and then make any changes necessary to adapt the settings to the organization's environment, requirements, policies, risk tolerance, and security objectives. This is particularly true for checklists that are intended for an environment with significantly different security needs. For example, producers of checklists in the military can harmonize on the specific types of hardware and software used in their environment, which can allow for more stringent settings. These security configuration settings generally represent a high-water mark rather than a baseline for civilian agencies attempting to use this guidance. The wholesale adoption of a high-water mark in a civilian environment may lead to unintended consequences, and each setting should be specifically tested and validated before widescale deployment. All deviations from

the checklist settings should be documented for future reference and include the reason behind each deviation and the impact of deviating from the setting.

All checklists should be tested in non-production environments before use in production environments. Each checklist in the NIST repository has been tested by its developer, but there are often significant differences between a developer's testing environment and an organization's operational environment. Security control modification can negatively impact a product's functionality, usability, and security controls. Testing and documentation can help organizations address significant issues.

Checklists are significantly more useful when they can run in common operational environments. The NCP has identified several broad and specialized operational environments (e.g., Stand-Alone, Managed), and at least one of the environments should be common to most audiences. Thoroughly identifying and describing these environments will make it easier for users to select the security checklists that are most appropriate for their operating environments and enable developers to better target their checklists to the general security characteristics associated with their operating environments.

Organizations that create content, particularly security baselines, should adopt a risk-based approach for selecting the appropriate settings and defining values that consider the context under which the baseline will be utilized. Checklist content can leverage a library of rules that are each mapped to requirements in one or more existing security standards, regulations, or frameworks to create multiple customized security baselines. This approach provides versioning and consistency of the content. Generally, the technical security settings in products do not drastically change between releases. By pursuing a rules-based approach, rules that remain applicable can be reused and incorporated into guidance for the latest release. This enables quicker adoption of new security features that are not offered in prior versions. Settings that are no longer applicable can be retired by specifying a maximum version.[8]

NIST encourages IT product vendors to develop security configuration checklists for their products, since vendors have the most expertise on possible security configuration settings and their relationships.

Vendors that create security configuration checklists should submit them for inclusion in the National Checklist Repository through the NCP. The NCP provides a process and guidance for developing checklists in a consistent fashion. For checklist developers, steps include the initial development of the checklist, checklist testing, documenting the checklist according to the guidelines of the NCP, and submitting a checklist package to NIST. NIST then screens the checklist according to program requirements and releases the checklist for a 30-day public review. After the public review period and subsequent resolution of issues, the checklist is listed on the NIST checklist repository with its information. Checklist maintenance may potentially be performed by the vendor, resulting in the release of updated checklists. NIST retires or archives checklists as they become outdated or incorrect.

[8] The mSCP is an open-source project that provides a programmatic approach to generating and using macOS security configuration baselines that adopt this recommendation.

2.5. Types of Checklists Listed by NCP

The NCP deals with checklists that are tied to *specific* IT products, such as a checklist for a specific operating system version. Some checklists may also incorporate the use of other checklists. For example, a checklist for a database product may reference a checklist for the operating system on which the database product runs. The NCP includes two major groups of checklists:

- **Automated.** An automated checklist is machine-executable and usable by one or more tools that automatically alter or verify settings based on the contents of the checklist. Checklists can be written in product-native scripting languages, such as shell scripts, PowerShell, Group Policy Objects, Jumpstart scripts, XML, and others.
- **Non-automated.** As the name implies, a non-automated checklist is one that is designed to be used manually, such as prose instructions that describe the steps an administrator should take to secure a system or verify its security settings.

Security configuration checklists in the NCP can help organizations comply with FISMA, which requires federal agencies to determine minimally acceptable system configuration requirements and ensure compliance with them. Checklists can also map specific technical control settings to the corresponding SP 800-53 controls and other existing security standards, regulations, or frameworks, which makes the verification of compliance more consistent and efficient. Federal agencies and vendors of products for the Federal Government are encouraged to develop and share such checklists using the NIST repository. This helps reduce silos of effort to identify sufficiently secure settings for IT products that are widely used in the Federal Government, such as common operating systems, servers, and client applications.

The NIST checklist repository (located at https://checklists.nist.gov/) provides information on automated and non-automated checklists that have been developed and screened to meet the requirements of the NCP. The repository hosts copies of checklists, primarily those developed by the Federal Government, and points to other submitted checklist locations. Users can browse checklist descriptions to locate and retrieve a particular checklist using a variety of different fields.

Where feasible, security checklists should be provided in automated formats to enable integration into built pipelines, automated compliance scanners, and continuous monitoring data feeds. Using standardized checklists with mappings to security controls shifts security implementation and validation into CI/CD pipelines and enterprise image builds, ensuring that configuration requirements are verified during the development phase. Automated formats provide rapid and repeatable hardening, which prevent misconfigurations from being deployed to production.

3. Operational Environments for Checklists

To ensure that as many users as possible receive value from checklists, checklist authors should create catalogs of controls from which checklists can be generated for specific operational environments and risk postures. The NCP identifies several broad and specialized operational environments, at least one of which should be common to most audiences. Identifying and describing these environments allows developers to better target their checklists to the general security requirements and risk postures of those environments and allows end users to select the most appropriate checklists.

This section describes the operational environments defined for the NCP and the general threat description and fundamental technical security practices for each environment. The two broad operational environments are referred to as **Stand-Alone** (or small office/home office [SOHO]) and **Managed** (or Enterprise). Two typical **Custom** environments that could be subsets of the broader environments are **Specialized Security-Limited Functionality (SSLF)** and **Legacy**.

IT product users may find it useful to consult this section of the document when initially identifying their own security requirements and needs (see Sec. 4). Developers may find this section useful when building and tailoring security compliance checklists for diverse products while still adhering to the uniform technical security practices and settings associated with the environments (see Sec. 5). Before submitting a checklist to NIST, developers should ensure that they have the most recent version of this document because updates to the criteria for operational environments may occur periodically. The most recent version is available as a separate file at https://checklists.nist.gov/.[9]

3.1. Stand-Alone Environment

The **Stand-Alone** environment describes individually managed devices (e.g., desktops, laptops, smartphones, tablets). It focuses on functionality and typically has the least automation available to apply baselines, making it the least secure environment given the potential to introduce human error. Stand-Alone checklists should be relatively simple for home users or novice system administrators to understand and implement. They are an entry point to defining a configuration for reuse across a handful of systems and improving from the base vendor security posture.

3.2. Managed Environment (Enterprise)

The Managed environment (also referred to as Enterprise) is based on centrally managed IT products and devices (i.e., many devices managed by a single organization), including cloud SaaS providers, IaaS providers, PaaS providers, virtualization platforms, servers, desktops, laptops, smartphones, tablets, and IoT. The Managed environment provides more security but

[9] As new information becomes available, NIST may update the criteria and information for the operational environments as well as other criteria contained in this document.

less functionality than the Stand-Alone environment.[10] It also tends to implement several layers of defense (e.g., firewalls, endpoint detection and response, automated patch management systems, email protections).

Managed environments also include cloud-native and managed execution models, such as serverless functions, short-lived containers, AI-specific infrastructure[11], and fully managed platform services. In these environments, computing resources may be ephemeral, abstracted from the organization, or replaced automatically without human intervention while remaining centrally administered and subject to organizational security policy. Compliance and security practitioners should account for these characteristics by applying and verifying security configuration settings using cloud-native and agentless methods (e.g., inspection of control-plane configurations and service metadata exposed through provider APIs) rather than relying solely on persistent hosts or installed agents. Automated cloud-native checklists should reflect methodologies that are specific to their environments, such as API-based configuration state verification or CI/CD build pipelines. Managed checklists are intended for advanced end users and system administrators. The nature of a typical Managed environment gives administrators centralized control over settings on devices. Authentication, account, and policy management can be administered centrally to maintain a consistent security posture across an organization.

3.3. Custom Environments

A **Custom** environment contains systems in which the functionality and degree of security do not fit the other types of environments. There are typically two types: **Specialized Security-Limited Functionality (SSLF)** and **Legacy.**

3.3.1. Specialized Security-Limited Functionality Environment

SSLF is a Custom environment that is highly restrictive, secure, and usually reserved for systems with the highest threats and associated impacts, such as outward-facing web servers, other publicly accessed systems, and firewalls. It also encompasses computers that contain confidential information (e.g., central repository of personnel records, medical records, or financial information) or that perform vital organizational functions (e.g., accounting, payroll processing, air traffic control). These systems might be targeted by third parties for exploitation or by trusted parties inside of the organization. Because systems in an SSLF environment are at high risk of attack and data exposure, security takes precedence over functionality. The system's data content or mission purpose is of such value that aggressive trade-offs in favor of security outweigh the potential negative consequences to other useful system attributes, such as legacy applications or interoperability with other systems.

[10] Checklists should be customized within Managed environments to meet organizational risk tolerance and requirements (regulatory, security standards and frameworks). An example would be permitting exceptions for groups of users (or devices) with a specific need to deviate from a checklist setting or settings, rather than deviating the setting across the entire enterprise. Tailored deviations, based on documented justification of the needs of a subset of users (or devices) are preferred to preventing users from performing their duties as long as the risk is accepted by an authorizing official.

[11] Checklists are an input that can enable the generation of an AI Bill of Materials (AIBOM) to track model provenance and ensure visibility across the environment.

An SSLF environment could also be a subset of another environment. For example, three desktops in a Managed environment that hold the organization's confidential employee data could be thought of as an SSLF environment within a Managed environment. In addition, a laptop used by a mobile worker might be an SSLF environment in a Stand-Alone environment. An SSLF environment might also be a self-contained environment outside of any other environment, such as a government security installation that processes sensitive data.

SSLF checklists are intended for experienced security specialists and system administrators who understand the impact of implementing strict technical security practices. If home users and other users who do not have security expertise attempt to apply SSLF checklists to their systems, they may experience unwanted limitations on system functionality and potentially cause irreparable system damage.

3.3.2. Legacy Environment

A Legacy environment is another example of a Custom environment and contains older systems or applications that are unable to be secured against modern threats. They often use older, less secure communication mechanisms that cannot be patched but need to be able to communicate with other systems. Non-legacy systems operating in a Legacy environment may need less restrictive security settings so that they can communicate with Legacy systems and applications. They require significant compensating and mitigating controls for ongoing use. Legacy environments may also be subsets of other environments.

4. Checklist Usage

This section describes a high-level process for retrieving and using checklists. Although all checklist users have their own specific requirements, the process described will apply to most situations. This section includes guidance on conducting an initial analysis of local environment threats and risks and lists the potential impacts of such attacks. It also describes a process for selecting and retrieving checklists through the NIST checklist repository and recommends steps for analyzing, tailoring, and applying the checklist. Fig. 1 shows the general process for using checklists.

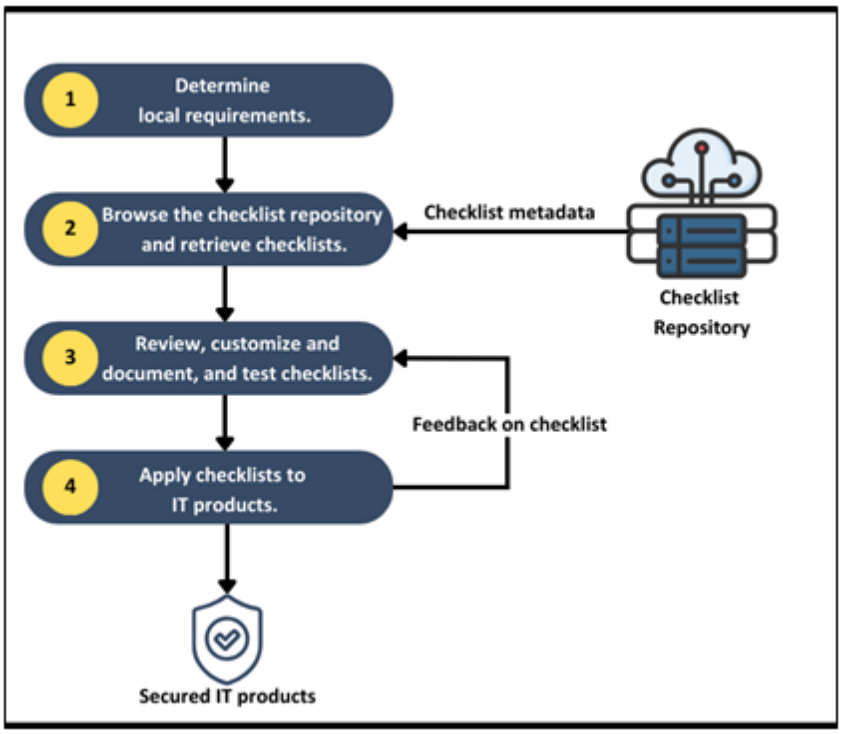

Fig. 1. Checklist user process overview

The general steps involved in acquiring and using checklists are:

1. Users gather their local requirements (e.g., IT products, the operating environment, associated security needs) and then acquire or purchase the IT product that best suits their needs.
2. Users browse the checklist repository to retrieve checklists that match their operational environment and security requirements. If a product is intended to be secure by default, it is still important to check the NIST checklist repository for updates to that checklist.
3. Users review the checklists and select one that best meets their requirements. They then tailor and document the checklist as necessary to consider local policies and functional requirements, test the checklist on a non-production system, and provide feedback to NIST and checklist developers.

4. Users prepare to deploy the checklist (e.g., make configuration or data backups) and then apply the checklist in production.

The following sections detail the activities included in each of these steps.

4.1. Determining Local Requirements

Organizations usually conduct a requirements analysis before selecting and purchasing a particular IT product. Such an analysis would include identifying the needs of the organization (i.e., what the product must do) and the security requirements for the product (e.g., relevant security policies or regulations). It is best to assess requirements early in the process of incorporating security into IT operations, regardless of size.

When planning security, it is essential to first define the threats that must be mitigated. Organizations that use checklists should conduct risk assessments to identify the specific threats against their systems, determine the effectiveness of existing security controls in counteracting those threats, and perform risk mitigation to decide whether any additional measures should be implemented, as discussed in SP 800-37r2 [4]. Performing risk assessments and mitigation helps organizations better understand their needs and decide whether they need to modify or enhance selected checklists. Tailoring for specific users or devices may be required to facilitate work across the organization, such as different settings for developers, kiosks, and specialty devices.

The risk mitigation methodology includes the following simple steps:

- **Identify functional needs.** What must the product do? Identifying the end user's requirements (e.g., remote access for telecommuters, a web server to make internal information available to employees) is necessary to ensure that the security solution and controls allow the system to meet its functional requirements.
- **Identify threats and vulnerabilities.** A threat is the potential for a particular threat source to successfully exploit an information system vulnerability. A vulnerability is a weakness that can be intentionally exploited or accidentally triggered. The goal of this step is to identify potential threat sources that are applicable to the IT product or system as well as vulnerabilities that could be exploited by the potential threat sources.
- **Identify security needs.** The goal of this step is to determine the controls needed to minimize or eliminate the likelihood (or probability) of a threat exploiting a product or system vulnerability. It answers the question, "What security features must the product provide?" Armed with this information, the organization can make wiser choices about which IT product best meets its needs.

NIST has written several documents and guides to help federal agencies select, acquire, and use information security products. The National Vulnerability Database (NVD) is an application of the Cybersecurity and Privacy Platform (CPP) that provides a search engine for identified application, system, and vendor vulnerabilities and information on patches or fixes that are available to correct the vulnerabilities. It is available at https://nvd.nist.gov.

4.2. Browsing and Retrieving Checklists

After determining local requirements and identifying an IT product, a checklist user is ready to browse the NIST checklist repository. The checklists are categorized by content type (i.e., degree of automation and standardization) and authority (i.e., the organization responsible for producing the original security configuration guidance represented by the checklist). Users can browse the checklists based on the content type, IT product, or authority and through a keyword search of the checklist name and summary for user-specified terms. The search results show the detailed checklist information and link to content and any supporting resources associated with the checklist. Selecting a particular checklist will show a description template that includes extensive information to help users decide whether the checklist will suit their specific purposes.

Some checklists address more than one application or operating system, such as several products from a single organization. To help users navigate the site from the checklist detail page, there is a Checklist Group link that represents the grouping of checklists based on a common source material. For example, the Defense Information Systems Agency (DISA) Office 365 Checklist contains configuration settings for multiple products, including desktop publishing and email applications. The NCP decomposes the checklist information according to these individual targets but keeps them conveniently linked to the same source document via the Checklist Group.

In some cases, multiple checklists are available for a particular version of a product. Such checklists are often similar, but they have important differences, such as the degree of automation provided, the intended audience (e.g., providing general recommendations versus complying with federal agency-specific requirements), and the checklist purpose (e.g., reconfiguring a product versus identifying a successful compromise of the product). To help users identify major differences among checklists, NIST has categorized checklists by content type, such as:

- **Prose.** Prose checklists provide narrative descriptions of how a person can manually alter a product's configuration.
- **Automated.** Automated checklists document their security settings in a standard or proprietary machine-readable format, such as shell scripts, Group Policy Objects (GPOs), SCAP content, kickstart files, .inf files, executables, and XCCDF files.

  SCAP is a standards-based example of an automated checklist. SCAP checklists adhere to the SCAP specification detailed in SP 800-126 for documenting security settings in machine-readable standardized SCAP formats. SCAP content that is available on the National Checklist Program repository has been evaluated with the NIST SCAP Content Validation Tool (SCAPVal).[12] This evaluation ensures that the checklist conforms to the SCAP specification. The SCAPVal tool does not evaluate the checklist for logic errors,

[12] SCAPVal is available for download for each SCAP version on the SCAP specification website at https://scap.nist.gov/revision/. This tool validates the correctness of the SCAP data stream according to the SCAP version specified in the corresponding version of SP 800-126 at the website listed above.

such as the use of an “equal to” operator when “equal to or greater than” should have been used.

An example of a comprehensive checklist-generating package is the mSCP content hosted on the NIST NCP, which uses a *catalog of controls* approach to checklist generation. This approach is used by the mSCP to curate a library of rules with each rule mapped to requirements in one or more existing security standards, regulations, or frameworks. The rule library can be used to create multiple customized security baselines by leveraging the technical security controls and settings that do not drastically change from release to release. By pursuing a rules-based approach, rules that remain applicable can be reused and incorporated into guidance for the newest product version. This enables quicker adoption of new security features that are not offered in prior versions. mSCP allows a user to create a checklist in various machine-readable formats, including shell scripts and SCAP based on the policies and requirements selected.

When multiple checklists are available for a particular product, organizations should consider the degree of automation and standards in each checklist. Generally, automated checklists or checklists generated from a technical rules library can be used more consistently and efficiently than others. There may be other significant differences among checklists. For example, one checklist may include software bundled with an operating system (e.g., web browser and email client), while another checklist addresses that operating system only. Another example is the assumptions on which the checklists are based (e.g., operational environment and risk tolerance). A checklist user should identify such differences and determine which checklists seem appropriate and merit further analysis. Checklists should also be evaluated as a starting point for tailoring to match their operating environment and risk tolerance (e.g., a civilian government baseline versus a military baseline).

Users from federal civilian agencies should first search for government-authorized or mandated checklists, such as NIST-produced checklists that are tailored for civilian agencies. If no NIST-produced checklist is available, then agency-produced checklists from DISA, the National Security Agency (NSA), or CISA should be used, if available. If formal government-authorized checklists do not exist, organizations are encouraged to use vendor-produced checklists. If vendor-produced checklists are not available, other checklists that are posted on the NCP website may be used.

Organizations often submit checklists with associated alphanumeric version identifiers (e.g., R1.2.0). Unfortunately, these identifiers do not have universal meanings. Some organizations may change the version number when new checks are added, old technology is deleted, patches are added, or simply based on a review date. Conversely, other organizations may update their checklist and not change the version numbers. To clarify updates to checklists, NCP uses Checklist Revisions to indicate that something has changed, even if the version identifier did not change. For example, if the organization does not change the version number on the document, but the content has been updated (e.g., patches were added for a given month), the current checklist will be listed as archived and the checklist with the updated patch content will show as the current checklist. Likewise, if the submitting organization updates the version

identifier, then the NCP will list the current checklist as archived and link to the new checklist. From the checklist detail page, a user can navigate to the checklist history via the "Archived Revisions" link.

4.3. Reviewing, Customizing, Documenting, and Testing Checklists

Checklist users should download all documentation for the checklist and review it carefully. The documentation should explain any required preparatory activities, such as backing up a system. Because a checklist may not exactly match a user's specific requirements, reviewing a checklist is useful in determining whether the checklist may need to be tailored[13] and whether the system or product will require further changes after applying the checklist.

The user's review can identify the impact on an organization's current policies and practices if a given security checklist is used. An organization may determine that some aspects of the checklist do not conform to certain organization-specific security and operational needs and requirements. Organizations should carefully evaluate the checklist settings and give them considerable weight, then make any changes necessary to adapt the settings to the organization's environment, requirements, policies, and security objectives. This is particularly true for checklists that are intended for an environment with significantly different security needs. 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*. Because the checklist may be used many times within the organization, the checklist itself might need to be modified. This is especially likely if the checklist includes a script or template to be applied to systems.

At this point, all deviations from the settings in the checklist should be documented for future reference. The documentation should include the justification behind each deviation, including the impact of retaining the setting and the impact of deviating from the setting. This documentation helps in managing changes to the checklist over the life cycle of the product being secured. Feedback on the checklist can be sent to NIST and the checklist developers. Feedback is especially important to developers in gauging whether the checklist is well-written and the settings are applicable to the targeted environment.

Before applying a checklist that will be used to alter product settings, users should first test it on non-production systems, preferably in a controlled non-operational or virtual environment. Each checklist in the NIST checklist repository has been tested by its developer, but there are often significant differences between a developer's testing environment and an organization's operational environment that may affect checklist deployment and impact systems. The testing configuration of the IT product should match the deployment configuration. Security control modification can have a negative impact on a product's functionality and usability, other products, and interactions with other security controls. For example, installing a patch could inadvertently break another patch, or enabling a firewall could inadvertently block software

[13] If multiple checklists are available for the same product, the checklist user should compare the settings or steps in the selected checklist to see which settings or steps differ to determine if these alternate recommendations should be used.

from updating or disrupt patch management software. It is important to perform testing to determine the impact on system security, functionality, and usability; to document the results of testing; and to take appropriate steps to address any significant issues. Section 4.4 provides recommendations for performing backups and other suggestions to prevent or recover from potential damage or unwanted effects that could occur when applying an untested checklist.

Before using a checklist to verify product settings without altering them, users should test. If the checklist is automated, users should also test the tool or tools that will be used with the checklist to ensure that they do not inadvertently disrupt the functionality of the system or alter the configuration of the product. Checklist testing should be performed to identify discrepancies between the expected and actual settings, which could indicate errors in the checklist, such as environment-specific characteristics for which the checklist was not modified.

4.4. Applying Checklists to IT Products

A checklist can be applied to an IT product in one of two ways: modifying the product's settings or verifying the existing settings.

- Setting Modification
    - Even after reviewing and testing a checklist, users should handle deployment carefully to minimize any issues that might arise from applying the checklist.
    - For users who are unable to test a checklist in a non-operational environment (e.g., home users), it is important to carefully review the checklist documentation and determine whether an initial backup is required. The *Rollback Capability* field in the checklist description will indicate whether the results of applying the checklist can be reversed to return the product to its original configuration. Regardless of this setting, it is strongly recommended that a user back up the IT product's configuration before installing the checklist recommendations.
    - At a minimum, users should back up all critical data files in their computing environment. If possible, the user should make a full backup of the system to ensure that it can be restored to its pre-checklist state if necessary.[14] Large organizations should also follow this procedure. If possible, use a test system to validate the impact of the checklist in the operational environment. After all pilot deployment groups are complete with a cooling off period to assess residual impact reports, deploy the settings enterprise-wide.
    - Cloud environments or build pipelines can also leverage checklists to rapidly establish secure workloads, create prebuilt and secured images or containers, build security into pipeline execution, or apply checklist settings when building and monitoring code.

[14] Making a full backup is recommended before making any major system change, not only when implementing a checklist.

- Setting Verification
    - Even after reviewing and testing a checklist, users should handle verification carefully to ensure that product settings are not inadvertently altered.
    - Continuously monitoring settings using automated compliance scanners, cloud-native APIs, and continuous monitoring data feeds can provide near real-time visibility into enterprise and cloud environments.

After initially applying a checklist, an organization may need to acquire and apply revised versions of the checklist in the future. Depending on the product being secured, a checklist may be updated periodically based on a set schedule or updated as needed, frequently or infrequently. An organization that acquires an updated checklist would perform the same steps already described in this section while taking advantage of knowledge gained and documented from applying previous versions of the checklist. Checklist versions should be compared against each other to find the delta in the configuration settings. Assess the risk of those changes to the existing deployment and any sets of small group deviations applied within the environment.

4.5. Providing Feedback on Checklists

NIST welcomes all "bug" reports, comments, and suggestions from checklist users regarding individual checklists or the repository itself. Such feedback should be directed to checklists@nist.gov.[15]

Some of the questions that checklist users may want to consider when evaluating a checklist include the following:

- Documentation
    - Does it explain the security objectives?
    - Does it contain a complete, clear, and concise description of the checklist settings?
- Recommended Practices
    - Are the checklist settings consistent with recommended practices?
    - Do the checklist settings consider recent vulnerabilities?
- Impact of Settings
    - Has the checklist developer tested the checklist settings on the product in an operationally realistic environment and determined that the application of the checklist meets the security objectives of the checklist?
    - Do any of the checklist settings cause the product to become inoperable or unstable?

[15] Checklist users who want to publish their own version of a checklist may act in a checklist developer role and submit it to the NIST checklist repository, provided that there are no intellectual property restrictions on the original checklist.

    - Do any of the checklist settings reduce product functionality? If so, is this documented?

- Ease of Implementation
    - Is the checklist straightforward to apply?
    - Are the instructions concise, sound, and complete?
    - Is the required skill level identified?
    - Are there procedures to verify that the installation was successful?
    - Is there guidance for uninstalling the checklist or restoring the product to the state before installation?
    - If the checklist cannot be rolled back, does the documentation recommend other preparatory measures, such as backups?
- Assistance
    - Is checklist-related help available?
    - Is there a repository or website for the checklist?
    - Does the documentation contain information for troubleshooting if errors occur or if the checklist settings cause the product to operate incorrectly?
    - Is assistance available for qualified users of the product?
    - If the checklist developer is NOT the IT product's vendor, does the documentation indicate whether the checklist has been sponsored or endorsed by the IT product's vendor?

5. Checklist Development

This section describes the general process for developing security configuration checklists and submitting them to the NCP. It includes an overview of the process that NIST will follow to screen the checklist submissions, publish them in its repository, and update or archive the checklist. The appendices of this document provide administrative requirements for participation in the NCP. Before submitting a checklist to NIST, developers should ensure that they have the most recent version of this document, which is available as a separate file at https://nvd.nist.gov/ncp/participation.

The checklist life cycle comprises the following steps:

1. **Initial Checklist Development:** The developer[16] becomes familiar with the procedures and requirements of the checklist program and develops an initial version of the checklist, including selection of a target environment.
2. **Checklist Testing:** Test the checklist in the target environment, and correct any problems with the checklist.
3. **Checklist Documentation:** Document the checklist according to the NCP guidelines.
4. **Checklist Submitted to NIST:** Submit the checklist and documentation package to NIST for screening and public review.
5. **NIST Screening:** NIST screens the checklist package's information and addresses any issues with the developer prior to public review.
6. **Public Review and Feedback:** NIST holds a 30-day public review of the candidate checklist. The developer addresses comments as necessary.
7. **Final Listing on Checklist Repository:** NIST lists the checklist on the repository as final and announces its availability.
8. **Checklist Maintenance and Archival:** Anyone can provide feedback on the checklist throughout its life. The developer updates the checklist periodically, as necessary. The checklist is archived when it is no longer being maintained or is no longer needed.

Each step should be carried out to ensure that the checklist is accurate, tested, and documented during its development and subsequent publication, update, or archival. The following sections describe considerations for each step.

5.1. Developer Steps for Creating, Testing, and Submitting Checklists

The first four steps in the development methodology listed above involve the developer creating, testing, documenting, and submitting checklists. Sections 5.2 through 5.5 describe each of these steps in greater detail.

[16] For simplicity, the rest of this document uses the term "developer" to refer to the entities developing a checklist.

5.2. Initial Checklist Development

During initial checklist development, a developer familiarizes themselves with the requirements of the checklist program and all procedures involved during the checklist life cycle, as described throughout this section. The developer agrees to the requirements for participation in the NCP before continuing to develop the checklist. The participation requirements are described in this document but presented in administrative and programmatic terms in Appendix A, which is intended for those in organizations who must formally agree to NCP requirements. The participation agreement is contained in Appendix B.[17] After agreeing to the NCP requirements, the developer identifies which operational environments (see Sec. 3) the checklist should be implemented for and builds the checklist accordingly. The output of this step is an initial checklist for the product.

NIST recognizes that detailed checklist development cannot be covered extensively in this document. Developers may find publications on commonly accepted technical security principles and practices, as catalogued in SP 800-53 [5] and SP 800-160 [6], to be helpful when developing a checklist. Additional examples of checklists are available in the National Checklist Program repository. The mSCP has also documented their process for developing checklists for macOS and iOS at https://pages.nist.gov/macos_security/.

In terms of vulnerability coverage, security objectives should consider the most up-to-date vulnerabilities and generally be consistent with recognized sources of vulnerability-related information, including the NIST's NVD and CISA's Known Exploited Vulnerabilities (KEV).[18]

Developers of checklists for products that are used by the Federal Government should consult FISMA-associated security control requirements. SP 800-53 [5] provides a catalog of security controls with assigned ratings that match the low, moderate, and high impact levels specified in FIPS 199 [7]. Developers of IT products used in federal information systems are encouraged to help federal agencies meet the requirements in FISMA by creating checklists that can be used in a variety of operational environments or for information systems with differing impact levels, as described in FIPS 199 and SP 800-53.

5.3. Checklist Testing

Before a checklist is submitted to NIST, it should be fully tested in the target environment and platform. The checklist should be tested with a variety of applications and hardware platforms to fully understand the impact of applying it. Ideally, at least some testing should be performed in a production or mirrored production environment. The testing data does not need to be submitted to NIST, but the developer should retain the data for review.

Selecting the most appropriate set of security controls can be a daunting task because many security controls limit system functionality and usability. In some cases, a security control can negatively impact other security controls. For example, installing a patch could inadvertently break another patch. It is important to perform testing for all security controls to determine

[17] The latest updates to these sections and this document are available at https://nvd.nist.gov/ncp/participation. This updated material should be consulted before formally agreeing to participate in the program.
[18] The NVD is at https://nvd.nist.gov/. CISA's website is https://www.cisa.gov/.

what impact they have on system security, functionality, and usability and to document potential impacts of applying a given setting. Users of the checklist will be able to take appropriate steps to address any significant issues caused by well-documented settings.

5.4. Checklist Documented

The quality of checklist documentation often makes a major difference in the checklist's effectiveness. The checklist documentation should clearly explain how to use the checklist with concise, sound, and complete instructions. The skill level required to use the checklist should be identified, as well as the targeted environment. The documentation should also explain the significance of individual settings, including any changes to product functionality. If applicable, the documentation should also include procedures to verify that the checklist installation is successful, as well as guidance for uninstalling the checklist or restoring the product to its state before installation of the checklist. If it is not possible to roll back checklist settings, the checklist documentation should recommend alternative procedures (e.g., backups, system restoration) as applicable.

The testing methodology, such as how the checklist was tested and what platforms were used, should be documented. The checklist documentation should also contain information for troubleshooting if errors occur or if the checklist settings limit the product's functionality, which may cause the product to not operate as desired. Ideally, assistance should be available for (registered) users of the product if there are problems.

Checklist developers are encouraged to contact NIST at cce@nist.gov to be assigned a set of CCE identifiers (i.e., globally unique identifiers) for their configuration settings. Although CCE is often associated with SCAP content, it can also be used apart to ensure globally unique identification for individual security settings in a checklist. See Appendix C regarding the use of CCEs to demonstrate connected paths from requirements to actual settings on the IT product.

Checklist developers must complete an online checklist description form for each checklist.[19] Table 1 shows the fields in the checklist description form that developers are to complete.

Table 1. Checklist description form fields

Field Name	Description
Checklist Name	The name of the checklist.
Version	The version or release number of the checklist.
Publication Date	The date when the actual checklist document was published in the format MM/DD/YYYY.
Product Category	The main product category of the IT product (e.g., firewall, IDS, operating system, web server).
Target	The set of specific IT systems or applications for which a checklist has been created.
CPE Name	The CPE representation of a specific Target.

[19] An offline version of the checklist description form can be downloaded from the NCP Participation Materials site on the checklist repository at https://nvd.nist.gov/ncp/participation.

Field Name	Description
Checklist Role	The primary use or function of the IT product as described by the checklist (e.g., client desktop host, web server, bastion host, network border protection, intrusion detection).
Checklist Summary	The purpose of the checklist and its settings.
Known Issues	Summary of issues that may arise after application of the checklist to help users pinpoint any functional and operational problems caused by the checklist.
Audience	The intended audience that should be able to install, test, and use the checklist, including suggested minimum skills and knowledge required.
Target Operational Environment	The IT product's operational environment (i.e., Stand-Alone, Managed, or Custom) with descriptions (e.g., Specialized Security-Limited Functionality, Legacy). Generally only applicable for security compliance/vulnerability checklists.
Checklist Type	The type of checklist (e.g., Compliance, Vulnerability, Specialized).
Checklist Installation Tools	The functional tools required to use the checklist to configure the system if they are not included with the checklist.
FIPS 140-2/140-3 Compliance	Whether the product can operate in a FIPS 140-2/140-3 validated mode (yes or no).
Compliance	Whether the checklist or controls within the checklist are consistent with various regulations and standards, such as HIPAA, Gramm-Leach-Bliley Act (GLBA), FISMA (e.g., mappings to SP 800-53 controls), ISO 27001, Sarbanes-Oxley, the Department of Defense (DoD), Federal Risk and Authorization Management Program (FedRAMP), Control Objectives for Information and Related Technologies (COBIT), and the NIST Cybersecurity Framework (CSF).
Authority	The organization responsible for producing the original security configuration guidance represented by the checklist. Authorities are ranked according to their "Authority Type." On the NCP website, authorities are grouped with their authority types with the syntax *Authority Type*: *Authority*.
Author	The organization responsible for creating the checklist in its current format. In most cases, an organization will represent both the author and authority of a checklist, but this is not always true. For example, if an organization produces validated SCAP content for a NIST publication, the organization that created the SCAP content will be listed as the author, but NIST will remain the authority.
Rollback Capability	Whether the changes in product configuration made by applying the checklist can be rolled back and, if so, how to roll back the changes.
Testing Information	Platforms on which the checklist was tested. Can include additional testing-related information, such as a summary of the testing procedures used. Should specify any operational testing performed in production or mirrored production environments.
Comments, Warnings, Miscellaneous	Any additional information that the checklist developer wishes to convey to users.
Disclaimer	Legal notice pertaining to the checklist.
Product Support	Whether the vendor will accept support calls from users who have applied this checklist on their IT product; warranty for the IT product has not been affected. Required to use the NCP logo if the submitter is the product vendor. If the submitter is not the product vendor, the submitter should describe any agreement that they may have with the product vendor.
Point of Contact	An email address, website, or repository where questions, comments, suggestions, and problem reports can be sent in reference to the checklist. The point of contact should be a location that the checklist developer monitors for reported issues.
Sponsor	States the name of the IT product manufacturer organization and individuals who sponsor the submitted checklist if it is submitted by a third-party entity.

Field Name	Description
Licensing	States the license agreement (e.g., the checklist is copyrighted, open source, Creative Commons, General Public License [GPL], free software, shareware).
Automated Content	A link to machine-readable content that represents the configuration guidance.
Supporting Resource	A link to any supporting information or content related to the guidance. This field can hold data ranging from an English prose representation of the actual guidance to configuration scripts that apply guidance-specific settings on a target.
Dependency/Requirement	Indicates that another checklist or guide is required to properly use and implement the current checklist.
References	Any supporting references chosen by the developer that were used to produce the checklist or checklist documentation.

The developer needs to complete the fields as indicated to accurately describe the checklist and minimize user confusion about what the checklist accomplishes.

In summary, well-structured checklist documentation includes the following, as appropriate:

- Statement of the security objectives, including the expected behavior of the product after applying the checklist
- The intended audience (e.g., end user, system administrator) and the level of technical skill required to use the checklist
- Explanation of the checklist settings, including each setting's effect on the operation of the product and any functionality that the settings enable or disable
- Backup procedures or any other initial steps required before applying the checklist
- As appropriate, step-by-step instructions for applying the checklist (e.g., screen shots, illustrated procedures) and verifying that the installation is successful
- Troubleshooting instructions or other information and references

5.5. Checklist Submitted to NIST

After the checklist developer has completed, tested, and documented the checklist, they may submit the package of materials to NIST. The package includes the following:

- Checklist and configuration files, templates, and scripts
- Completed checklist description
- Checklist documentation
- Identification of the developer point of contact
- Signed participation agreement

The participation agreement and other requirements are outlined in detail in Appendix A, which also includes the appropriate NIST contact information.

Checklist packages are submitted to NIST through the NCP Submission website. The website allows checklist developers to view the checklists they have submitted, see tasks that have

been assigned to them (e.g., fixing errors on a previously submitted checklist), update existing checklists, and perform other actions. NIST also provides web services for submitting, fetching, and maintaining checklists. To request access to the NCP Submission website or associated web services, email checklists@nist.gov.

5.6. NIST Steps for Reviewing and Finalizing Checklists for Publication

The NIST process for screening and publishing a checklist, which corresponds to steps 5 through 8 in the checklist life cycle, is described in the following sections.

5.7. NIST Screening of the Checklist Package

This step involves determining whether the checklist materials are ready for public review. NIST screens the checklist information for completeness and accuracy and ensures that checklist content is well-formed if it is SCAP-expressed. NIST may contact the developer with questions about the submitted materials during the screening period.

5.8. Public Review and Feedback for the Candidate Checklist

After the checklist package has been screened and the developer has addressed any issues, NIST will post it as a candidate draft and announce it for public review for a period of 30 days. This allows the public to review and test the checklist and to provide the checklist developers and NIST with comments and feedback. Information from comments and feedback may be incorporated in a revision of the checklist to improve its quality. When a candidate checklist has completed the review process, its information is added to the checklist repository.

A checklist reviewer emails checklists@nist.gov to provide comments on the reviewer's test environment, procedures, and other relevant information. Depending on the review, the checklist developer may need to respond to comments. NIST may also consult independent expert reviewers as appropriate. Typical reasons for using independent reviewers include:

- NIST may decide that it does not have the expertise to determine whether the comments have been addressed satisfactorily.
- NIST may disagree with the proposed issue resolutions and seek reviews from third parties to get additional perspectives.

At the end of the public review period, NIST will give the developer 30 days to respond to comments.

5.9. Final Listing on Checklist Repository

After any outstanding issues are addressed, NIST lists the final checklist on the repository and announces its availability. At this time, the developer (e.g., IT product vendor) may be eligible to use the checklist logo on the IT product's promotional material if they plan to provide assistance for the checklist. Requirements for use of the logo are described in Appendix B.

5.10. Checklist Maintenance and Archival

Throughout a checklist's life cycle, anyone can provide comments or ask questions regarding the checklist by emailing checklists@nist.gov, and NIST will pass feedback to the checklist developer. Depending on the product and how frequently updates occur, NIST may maintain an email list for the associated checklists. Users who subscribe to the email list can receive announcements of updates or other issues connected with a checklist. The selected checklist's description (on the checklist repository) will contain instructions for subscribing to the email list.

After the final checklist is listed, NIST will periodically review the checklist to determine whether it is still relevant or requires changes. NIST will make an announcement if the developer decides to update the checklist at any time. If the revised checklist contains major changes, it will be treated as a new submission and will be required to undergo the same review process as a new submission.

At the discretion of NIST or the developer, the checklist can be removed from the repository or marked as an archive. Typical reasons for such actions would be that the product is no longer supported or is obsolete or if the developer no longer wishes to provide support for the checklist.

References

[1] Cyber Security Research and Development Act of 2002, Public Law 107-305, 116 Stat. 2367. Available at http://www.gpo.gov/fdsys/pkg/PLAW-107publ305/pdf/PLAW-107publ305.pdf
[2] Part 39 of the Federal Acquisition Regulation (FAR). Available at https://www.acquisition.gov/far/part-39
[3] Federal Information Security Modernization Act (FISMA) of 2014, Public Law 113-283. Available at https://www.congress.gov/113/plaws/publ283/PLAW-113publ283.pdf
[4] Joint Task Force (2018) Risk Management Framework for Information Systems and Organizations: A Security Life Cycle Approach for Security and Privacy. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Special Publication (SP) NIST SP 800-37r2. https://doi.org/10.6028/NIST.SP.800-37r2
[5] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Special Publication (SP) NIST SP 800-53r5. https://doi.org/10.6028/NIST.SP.800-53r5
[6] Ross R, Winstead M, McEvilley M (2022) Engineering Trustworthy Secure Systems. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Special Publication (SP) NIST SP 800-160v1r1. https://doi.org/10.6028/NIST.SP.800-160v1r1
[7] National Institute of Standards and Technology (2004) Standards for Security Categorization of Federal Information and Information Systems. (Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) NIST FIPS 199. https://doi.org/10.6028/NIST.FIPS.199
[8] Dempsy K, Chawla NS, Johnson A, Johnston R, Jones AC, Orebaugh A, Scholl M, Stine K (2011) Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Special Publication (SP) NIST SP 800-137. https://doi.org/10.6028/NIST.SP.800-137

Appendix A. Checklist Program Operational Procedures

Operational Procedures

for

The NIST National Checklist Program

for Information Technology Products

Version 1.4

This document sets forth the policies, procedures, and general requirements for the NIST National Checklist Program for Information Technology Products. This document is intended for those individuals in organizations who need to formally agree to the program's requirements.

This document is organized as follows:

- Section 1 — General considerations for the NIST National Checklist Program
- Section 2 — Procedures for the initial screening of a checklist prior to public review
- Section 3 — Procedures for the public review of a candidate checklist
- Section 4 — Final acceptance procedures
- Section 5 — Maintenance and delisting procedures
- Section 6 — Record keeping

The following terminology is used in this appendix:

- *Candidate* is a checklist that has been screened and approved by NIST for public review.
- *FCL* refers to the final checklist list, which is the listing of all final checklists on the NIST repository.
- *Final* is a checklist that has completed public review, has had all issues addressed by the checklist developer and NIST, and has been approved for listing on the repository according to the procedures of this section.
- *Checklist* refers to a checklist for a specific product and version.
- *Checklist Developer* or *Developer* is an individual or organization that develops and owns a checklist and submits it to the National Checklist Program.
- *Independent Qualified Reviewers* are tasked by NIST with making a recommendation to NIST regarding the public review or listing of the checklist. They work independently of

other reviewers and are considered experts in the technology represented by the checklist.

- *Logo* refers to the NIST National Checklist Program logo.
- *National Checklist Program*, *Program*, or *NCP* is used in place of the NIST National Checklist Program for Information Technology Products.
- *NIST Checklist Repository* or *Repository* refers to the website that maintains the checklists, the descriptions of the checklists, and other information regarding the National Checklist Program.
- *Public Reviewer* is any member of the general public who reviews a candidate checklist and sends comments to NIST.
- *Operational Environments* refer to the operational environments outlined in this document.

References to documents that form a basis for the requirements of this program are as follows:

- FIPS 199, *Standards for Security Categorization of Federal Information and Information Systems*, https://doi.org/10.6028/NIST.FIPS.199
- SP 800-53r5, *Security and Privacy Controls for Information Systems and Organizations*, September 2020 (updated December 20, 2020), https://doi.org/10.6028/NIST.SP.800-53r5
- SP 800-70r5, *National Checklist Program for IT Products — Guidelines for Checklist Users and Developers,* https://doi.org/10.6028/NIST.SP.800-70r5
- SP 800-126, *Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.4*, available from https://scap.nist.gov
- SP 800-126A, *SCAP 1.4 Component Specification Version Updates: An Annex to NIST Special Publication 800-126r4*, available from https://scap.nist.gov
- SP 800-128, *Guide for Security-Focused Configuration Management of Information Systems*, https://doi.org/10.6028/NIST.SP.800-128
- SP 800-160v1r1, *Engineering Trustworthy Secure Systems,* November 2022, https://doi.org/10.6028/NIST.SP.800-160v1r1
- SP 800-219r1, *Automated Secure Configuration Guidance from the macOS Security Compliance Project (mSCP)*, https://doi.org/10.6028/NIST.SP.800-219r1

A.1. Overview and General Considerations

This section focuses on general considerations for all parts of the National Checklist Program.

a) **Checklist Life Cycle Overview:**

1. Checklist developers inquire about the program and download a submission package. The developer subsequently contacts NIST with a tested checklist,

supporting information, and a signed agreement to the requirements of the NCP. Checklist submission requirements and procedures are discussed in Sec. 2.

2. NIST verifies that all information is complete and performs a high-level screening of the checklist package. Checklists that meet the requirements for listing receive further consideration and are referred to as "candidate checklists." Section 2 discusses screening criteria and procedures.
3. NIST lists the candidate checklist on the repository for public review for a period of 30 days, as discussed in Sec. 3.
4. NIST forwards comments from public reviewers to the developer. The developer addresses the issues as appropriate, and the checklist is listed on the FCL, as discussed in Sec. 4.
5. NIST periodically reviews each final checklist to determine whether its listing should continue, be updated, or be archived, as discussed in Sec. 5.

b) **Intellectual Property Rights:** Developers retain intellectual property rights to their checklists.

c) **Confidential Information:** NIST does not anticipate the need to receive confidential information from checklist developers. If it becomes necessary to disclose confidential information to NIST, NIST and the developer must enter into a separate confidentiality agreement prior to such disclosure.

d) **Independent Qualified Reviewers:** NIST may decide to seek technical advice from independent qualified experts who will review checklist submissions to determine whether they meet program requirements. The reviewers are tasked with making a recommendation to NIST regarding a subsequent public review or final listing of the checklist. Typical but non-exclusive reasons for using independent reviewers include the following:

1. NIST does not possess the expertise to determine whether issues have been addressed satisfactorily.
2. NIST disagrees with proposed issue resolutions.

e) **Terminating Consideration of a Checklist Submission:** NIST or the developer may terminate the consideration of checklist submissions at any time. If NIST terminates consideration, the points of contact are asked to respond within 10 business days. Typical but not exclusive of the reasons for terminating the consideration of checklist submissions include the following:

1. The submission package does not meet the screening criteria.
2. The developer fails to address issues raised at other times.
3. The developer violates the terms and conditions of participation in the program.

A.2. Checklist Submission and Screening

This section outlines the procedures and requirements for submitting checklists to NIST and the process by which NIST determines whether checklists are suitable for public review. When checklists meet the screening criteria, they receive further consideration in a public review and are referred to as "candidate checklists." NIST then follows the subsequent procedures.

a) **Notification of Checklist Program Requirements:** NIST maintains a complete set of information for developers on the repository. The information outlines the requirements for participation in the program and describes materials and time frames.

b) **Materials Required From the Developer:**

1. Contact information for an individual from the submitting organization who will serve as the point of contact for questions and comments pertaining to the checklist, as well as contact information for a backup or deputy point of contact. The information must include a postal address, a direct telephone number, and an email address.
2. The checklist, documentation, and description template.
3. The participation agreement, which must be printed, signed, and sent to NIST. NIST accepts emailed PDF copies of the participation agreement, facsimiles, or copies via regular mail.
4. Participation fees. Currently, there is no fee for checklist developers, though NIST reserves the right to charge fees for participation in the future. Fees are not retroactive.

c) **Preliminary Screening Checklist Contents:** NIST performs a preliminary screening to verify that checklist packages meet basic program requirements. NIST will not typically perform an in-depth analysis of the content of the checklist (e.g., its reflection of recommended security and engineering practices), though NIST reserves the right to do so.

A.3. Candidate Checklist Public Review

NIST follows the subsequent procedures when listing candidate checklists for public review:

a) **Public Review Period:** NIST lists candidate checklists for a 30-day public comment period. NIST reserves the right to extend the review cycle, particularly for long or complicated checklists. NIST provides the following disclaimer in conjunction with candidate checklists:

> *NIST does not guarantee or warrant the checklist's accuracy or completeness. NIST is not responsible for loss, damage, or problems that may be caused by using the checklist.*

b) **Accepting Comments from Reviewers:** Public reviewers email checklists@nist.gov to provide comments about the test environment, procedures, and other relevant information. The contents of these emails are considered public records.

c) **Maintaining Records:** NIST may maintain copies of correspondence and feedback between the public and developers by creating a unique email address for each checklist. If so, NIST will archive the information.

d) **Addressing Comments:** After the end of the public review period, the developer has 30 days to respond to comments.

A.4. Final Checklist Listing

After NIST determines that a checklist and the associated developers have met all requirements for final listing, NIST lists checklists in the FCL and refers to them as "final checklists." NIST then follows the subsequent procedures:

a) **Finalizing Checklists:** NIST lists the checklist in the FCL. NIST may send announcements to various email lists maintained by NIST or other organizations.

b) **Handling Comments:** NIST continues to accept comments about final checklists by maintaining a central email address on the repository at checklists@nist.gov. NIST lists the procedures to be followed when contacting the developer as well as the contact information for the developer (e.g., email address, URL). If the point of contact changes, NIST must be notified immediately.

A.5. Final Checklist Update, Archival, and Delisting

NIST follows these procedures for periodic update, archival, and delisting of final checklists:

a) **Periodic Reviews:** NIST periodically reviews each checklist to identify changes in its status. NIST may contact developers to determine whether there are changes in the status of a checklist. Developers have 30 days to respond and indicate whether checklists should be updated, archived, or delisted.

b) **Updates:** NIST may indicate on the FCL when checklists are under review. Developers have 60 days after the review to submit the updated material to NIST. Depending on the magnitude of updates, NIST may screen the checklist and schedule a public review.

c) **Archival:** A checklist may be archived for a variety of reasons. For example, a developer may no longer want to provide support for the checklist, or a product may no longer be supported. At the discretion of the developer or NIST, the checklist can remain in the repository but be reclassified as an archive.

d) **Delisting:** When delisting occurs (e.g., when a developer fails to respond to inquiries from NIST about the status of a checklist), NIST removes the checklist from the FCL. NIST may send announcements to various email lists maintained by NIST or other organizations.

A.6. Record Keeping

NIST maintains information associated with the program and requires that participants in the checklist program also maintain certain records, as follows:

a) **NIST Records:** After a checklist has been submitted to NIST, while a checklist is listed on the FCL as a final or archived checklist, and for three years after the most recent update to the checklist, NIST will maintain the following:

   1. The checklist description template, as listed on the repository
   2. The checklist and its description, as listed on the repository
   3. All comments submitted as part of the public review
   4. All comments submitted to NIST regarding the checklist

b) **Developer Records:** After a checklist has been submitted to NIST and while a checklist is listed on the FCL as a final or archived checklist, the developer will maintain the following:

   1. The checklist description template, as listed on the repository
   2. The checklist and checklist description, as listed on the repository
   3. Test reports and other evidence of checklist testing

Appendix B. Participation and Logo Usage Agreement Form

This appendix contains the terms and requirements for participation in the NIST National Checklist Program (NCP) and for use of the NIST National Checklist Program logo. Prior to submission of a checklist to NIST, developers should ensure that they have the most recent version of this appendix, which is available as a separate file at https://nvd.nist.gov/ncp/participation.

Participation and Logo Usage Agreement Form

for

The NIST National Checklist Program for Information Technology Products

Version 1.5

The phrase "NIST National Checklist Program for Information Technology Products" and the NIST National Checklist Program logo are intended for use in association with specific versions of information technology (IT) products for which a checklist has been created and has met the requirements of the National Institute of Standards and Technology (NIST) National Checklist Program for Information Technology Products for final listing on its checklist repository. You may participate in the NIST National Checklist Program and use the phrase and logo provided that you agree in writing to the following terms and conditions:

1. You will follow the rules and requirements of the program as outlined in the NIST Operational Procedures for the NIST National Checklist Program (Appendix A of NIST SP 800-70 Revision 5).
2. You will respond to comments and issues raised by a public review of your checklist submission within 30 days of the end of the public review period. Any comments from reviewers and your responses may be made publicly available.
3. You agree to maintain the checklist and provide a response within 10 business days to requests from NIST for information or assistance with regard to the contents of the checklist.
4. You agree to maintain checklist-related records according to the requirements of the NIST National Checklist Program, as listed in Appendix A of NIST SP 800-70 Revision 5, item 6.b.
5. You will hold NIST harmless in any subsequent litigation involving the checklist submission.

6. You may terminate your participation in the NIST National Checklist Program at any time. You will provide two business weeks' notice to NIST of your intention to terminate participation. NIST may terminate its consideration of a checklist submission or your participation in the NIST National Checklist Program at any time. NIST will contact you two business weeks prior to its intention to terminate your participation. You may appeal the rejection and provide supporting evidence within one business week.
7. You may not use the name of NIST or the Department of Commerce on any advertisement, product, or service that is directly or indirectly related to this agreement other than attribution of the content source. By accepting this agreement, NIST does not directly or indirectly endorse any product or service provided or to be provided by you, your successors, assignees, or licensees. You may not in any way imply that this agreement is an endorsement of any such product or service. You may not combine use of the logo with other marks, phrases, or logos in such a way that would imply endorsement by NIST.
8. The phrase "NIST National Checklist Program for Information Technology Products" and the NIST National Checklist Program logo are Registered Marks of NIST, which retains exclusive rights to their use. NIST reserves the right to control the use of the phrase "NIST National Checklist Program for Information Technology Products" and the NIST National Checklist Program logo.
9. Your permission to advertise participation in the NIST National Checklist Program and use of the logo is conditional on and limited to those products and the specific product versions for which a checklist is made currently available by NIST through the NIST National Checklist Program on its Final Checklist List.
10. Your permission to advertise participation in the NIST National Checklist Program and use of the logo is conditional on and limited to those checklist developers who provide assistance to users of the checklist regarding proper use of the checklist and that the warranty for the product and the specific product versions is not changed by use of the checklist.
11. Your use of the logo on product reports, letterhead, brochures, marketing material, and product packaging must be accompanied by the following: "TM: a Registered Mark of NIST, which does not imply product endorsement by NIST or the U.S. Government."
12. The dimensional requirements for the size, placement, color, and other aspects of the logo are specified in NIST SP 800-70 Revision 5.
13. NIST reserves the right to charge a participation fee in the future. No fee is required at present. No fees will be made retroactive.
14. NIST may terminate the NIST National Checklist Program at its discretion. NIST may terminate your participation in the program for any violation of the terms and conditions of the program or for statutory or regulatory reasons.

By signature below, the developer agrees to the terms and conditions contained herein.

Organization or company name:

Name and title of organization authorized person:

Signature:

Date:

Appendix C. Automating NIST CSF 2.0

This overview illustrates how the NCP facilitates traceable and automatable connections among CSF 2.0 outcomes, SP 800-53r5 controls, CCE items, and executable technical checks, such as SCAP (XCCDF/OVAL), PowerShell, Intune, shell/Ansible, GPO templates, and mSCP for macOS. This mapped pathway enables evidence-ready conformance from policy objectives to individual system settings and checks. Although the CSF 2.0 is used as an example in this appendix, any high-level framework can be substituted, such as those mapped to the CSF 2.0 on the NIST Online Informative References (OLIR) program page.

CCE provides stable, globally unique identifiers for low-level configuration settings so that individual checklist rules can be referenced unambiguously across tools and documents. In the NCP context, automated checklists are encouraged to include CCE identifiers. CCE allows for mappings between low-level settings and high-level requirements to be explicit, enabling repeatable verification and evidence generation. CCEs can be obtained from NIST by sending a request to cce@nist.gov. See https://ncp.nist.gov/cce for more details regarding CCE usage in checklists in the NCP.

C.1. How the Path Connects Policy to Automation

- Organizational risk strategy objectives and policy are expressed through CSF outcome statements that are addressed with security and privacy controls (i.e., from SP 800-53r5).
- Those controls are implemented and verified through system-specific identifiers (i.e., CCEs) as device-specific configuration statements that automated scanners can interpret, which ensures that checklist rules can be referenced consistently across tools and documents.
- Automated checklists incorporate CCE IDs expressed in SCAP to link policies, system settings, and high-level requirements with clear, repeatable verification.

C.2. Implementation and Traceability

- CSF 2.0 Subcategories (e.g., PR.AA-01) record desired outcomes (e.g., through a CSF Profile).
- OLIR maps these outcomes to SP 800-53r5 controls or enhancements (e.g., AC-02, AC-14, IA-02) that are then mapped to CCE identifiers, anchoring requirements to concrete configuration objects.

- SCAP content or native scripts (e.g., PowerShell, Intune, shell, Ansible, GPO, mSCP) enable automation.
- Evidence can be collected and the authoritative NCP checklist can be cited for provenance and reporting.
- CCE IDs serve as the link between controls and technical checks, whether automation is SCAP-based or uses native scripting.

C.3. Checklist Development Guidance

In your NCP checklist, clearly state:

- Content Type (i.e., Prose, Automated, SCAP) and how CCEs are referenced
- SCAP Content data streams, enabling validation with SCAPVal
- Supporting Resources, linking native scripts that maintain CCE mapping
- Regulatory/Framework Mappings using OLIR to connect CSF, SP 800-53, and CCE for full traceability

C.4. Operational Environment Tailoring and Considerations

- Adjust CCE-coded rules for each environment, either Stand-Alone (e.g., simple, rollback-friendly remediation) or Managed/Enterprise, integrated with centralized policy and layered security.
- CCE anchors remain constant, but profile parameters adapt to the environment.
- Where SCAP can be used, XCCDF integration incorporates CCE IDs in rules using the element and system URL for consistency in rule logic and downstream automation. When SCAP is unavailable, embed CCE IDs in native automation to maintain traceability (e.g., PowerShell/Intune, mSCP).
- Metadata traceability is provided by referencing both CSF/SP 800-53 and CCE in checklist metadata, including relevant automation approaches.
- These methods support multiple platforms, including Linux (e.g., shells, Ansible), macOS (e.g., mSCP compliance scripts mapped to CCE rules), and Windows (e.g., GPO backups, Intune JSON, PowerShell).
- Use SCAPVal to validate SCAP streams and submit checklists to the NCP repository for visibility and reuse.

C.5. Checklist Submission and Maintenance

- Ensure that CCE identifiers are consistently present in rule metadata and automation artifacts.

- Follow NCP's public review process (typically 30 days), maintain versions, archive outdated content, and keep contact information up to date for handling CCE-related updates and errata.

C.6. Appendix References

- CCE Program: https://ncp.nist.gov/cce
- NCP Checklist Repository: https://checklists.nist.gov/
- SCAP and SCAPVal resources: https://scap.nist.gov/
- NIST CSF 2.0 (NIST CSWP 29), Feb 26, 2024, https://doi.org/10.6028/NIST.CSWP.29
- NIST OLIR Program (overview, catalog, and templates): https://csrc.nist.gov/projects/olir
- SP 800-53r5 (current release, incl. 2025 updates), https://doi.org/10.6028/NIST.SP.800-53r5
- CCE Program (current lists and schema): https://ncp.nist.gov/cce
- NCP Checklist Repository: https://checklists.nist.gov/
- NIST IR 7275r4, *Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2*, https://csrc.nist.gov/pubs/ir/7275/r4/upd1/final
- NIST macOS Security Compliance Project (mSCP): https://pages.nist.gov/macos_security/
- CSF 2.0 news and resources: https://www.nist.gov/news-events/news/2024/02/nist-releases-version-20-landmark-cybersecurity-framework

Appendix D. List of Symbols, Abbreviations, and Acronyms

Selected acronyms and abbreviations used in the guide are defined below.

AIC
Architecture and Infrastructure Committee

CCB
Change Control Board

CCE
Common Configuration Enumeration

CMVP
Cryptographic Module Validation Program

CISA
Cybersecurity and Infrastructure Security Agency

COBIT
Control Objectives for Information and Related Technologies

CPE
Common Platform Enumeration

CSRDA
Cyber Security Research and Development Act of 2002

CVE
Common Vulnerabilities and Exposures

CVSS
Common Vulnerability Scoring System

DHCP
Dynamic Host Configuration Protocol

DHS
Department of Homeland Security

DISA
Defense Information Systems Agency

DNS
Domain Name System

DoD
Department of Defense

FAQ
Frequently Asked Questions

FCL
Final Checklist List

FedRAMP
Federal Risk and Authorization Management Program

FIPS
Federal Information Processing Standards

FISMA
Federal Information Security Modernization Act

GLBA
Gramm-Leach-Bliley Act

GPL
General Public License

GPO
Group Policy Object

HIPAA
Health Insurance Portability and Accountability Act

IA
Information Assurance

IATF
Information Assurance Technical Framework

IDS
Intrusion Detection System

IP
Internet Protocol

IR
Interagency Report

IT
Information Technology

ITL
Information Technology Laboratory

MSCP
macOS Security Compliance Project

NCP
National Checklist Program

NIST
National Institute of Standards and Technology

NSA
National Security Agency

NVD
National Vulnerability Database

OCIL
Open Checklist Interactive Language

OMB
Office of Management and Budget

OVAL
Open Vulnerability and Assessment Language

SCAP
Security Content Automation Protocol

SCAPVAL
Security Content Automation Protocol Validation Tool

SMTP
Simple Mail Transfer Protocol

SNMP
Simple Network Management Protocol

SP
Special Publication

SSLF
Specialized Security-Limited Functionality

STIG
Security Technical Implementation Guide

TIS
Technology Infrastructure Subcommittee

VPN
Virtual Private Network

XCCDF
Extensible Configuration Checklist Description Format

XML
Extensible Markup Language

Appendix E. Glossary

Selected terms used in this guide are defined below.

audience
The intended audience that should be able to install, test, and use the checklist, including suggested minimum skills and knowledge required to correctly use the checklist.

author
The organization responsible for creating the checklist in its current format. In most cases, an organization will represent both the author and authority of a checklist, but this is not always true. For example, if an organization produces validated SCAP content for a NIST publication, the organization that created the SCAP content will be listed as the author, but NIST will remain the authority.

authority
The organization responsible for producing the original security configuration guidance represented by the checklist.

authority type
The type of organization that is the authority for the checklist. The three types are governmental authority, software vendor, and third party (e.g., security organizations).

automated checklist
A checklist that is used through one or more tools that automatically alter or verify settings based on the contents of the checklist. Automated checklists document their security settings in a machine-readable format, either standard or proprietary.

candidate checklist
A checklist that has been screened and approved by NIST for public review.

checklist
A document that contains instructions, procedures, or machine-readable and executable content to configure an IT product to a specific risk posture for an operational environment, verify that the product has been configured properly, identify unauthorized configuration changes to the product, and/or produce artifacts that show the security posture of the product. Also referred to as a security configuration checklist, lockdown guide, hardening guide, security guide, secure configuration, security technical implementation guide (STIG), or benchmark.

checklist developer
An individual or organization that develops and owns a checklist and submits it to the National Checklist Program.

checklist group
Represents the grouping of checklists based on a common source material. Commonly used if an organization packages multiple sets of product guidance under the same name.

checklist revision
Represents a change to the checklist content that does not affect the underlying rule/value configuration guidance put forth by the content. A scenario that would require a checklist revision is when automated content is created for a prose checklist. This revision would change the checklist's content type from prose to automated content. A new checklist revision would be created to accommodate this change, while still maintaining the prose checklist revision for interested parties.

checklist role
The primary use or function of the IT product as described by the checklist (e.g., client desktop host, web server, bastion host, network border protection, intrusion detection).

checklist type
The type of checklist, such as compliance, vulnerability, or specialized.

content type
The form of the checklist content in terms of the degree of automation and standardization. Examples include prose, automated, and SCAP content.

custom environment
An environment that contains systems in which the functionality and degree of security do not fit the other types of environments.

final checklist
A checklist that has completed public review, has had all issues addressed by the checklist developer and NIST, and has been approved by NIST for listing on the repository.

final checklist list (FCL)
The listing of all final checklists on the NIST repository.

independent qualified reviewer
A reviewer tasked by NIST with making a recommendation regarding public review or listing of the checklist.

legacy environment
A custom environment that contains older systems or applications that may need to be secured to meet today's threats but often use older, less secure communication mechanisms and need to be able to communicate with other systems.

logo
The NIST National Checklist Program logo.

managed environment
An environment comprising centrally managed IT products.

NIST checklist repository
The website that maintains the checklists, the descriptions of the checklists, and other information regarding the National Checklist Program. Also known as the repository.

non-automated checklist
A checklist that is designed to be used manually, such as English prose instructions that describe the steps an administrator should take to secure a system or to verify its security settings.

operational environment
The type of environment in which the checklist is intended to be applied. Types of operational environments are Stand-Alone, Managed, and Custom, including Specialized Security-Limited Functionality and Legacy.

product category
The main product category of the IT product (e.g., firewall, operating system, web server).

prose checklist
A checklist that provides a narrative descriptions of how a person can manually alter a product's configuration.

public reviewer
A member of the general public who reviews a candidate checklist and sends comments to NIST.

review status
The status of the checklist within the internal NCP review process. Possible status options are Candidate, Final, Archived, or Under Review. A status of "Final" signifies that NCP has reviewed the checklist and accepted it for publication within the program.

SCAP content checklist
An automated checklist that adheres to the SCAP specification in SP 800-126 for documenting security settings in machine-readable standardized SCAP formats.

Specialized Security-Limited Functionality (SSLF) Environment
A custom environment that is highly restrictive and secure. It is usually reserved for systems that have the highest threats and associated impacts.

Stand-Alone Environment
An environment that contains individually managed devices (e.g., desktops, laptops, smartphones, tablets).

target
The set of specific IT systems or applications for which a checklist has been created.

target operational environment
The IT product's operational environment (i.e., Stand-Alone, Managed, or Custom) with descriptions (e.g., Specialized Security-Limited Functionality, Legacy). Generally only applicable for security compliance/vulnerability checklists.

Appendix F. Change Log

In May 2026, the following changes were made to this report:

- Revised the Abstract to expand the definition of a checklist to include machine-readable and executable content and the production of artifacts showing the security posture of the product.
- Expanded the Section 1 (Introduction) to incorporate guidance previously located in the Executive Summary, including detailed subsections on mapping and acquisition considerations, checklist selection, government checklist precedence, and checklist tailoring considerations. Introduced risk-based framing and risk tolerance language throughout.
- Revised Executive Summary to reflect a streamlined set of major recommendations, including a new recommendation that checklist creators adopt a "catalog of controls" approach and a new recommendation that checklists be incorporated into continuous monitoring and automated data feeds for near real-time posture assessment.
- Reordered Section 2 and added discussion of the NIST macOS Security Compliance Project (mSCP) as an example model for programmatic, standards-based checklist generation. Also added guidance encouraging developers to assign Common Configuration Enumeration (CCE) identifiers to individual configuration settings.
- Revised Section 3 to consolidate Custom Environments (SSLF and Legacy) under a new parent subsection (Sec. 3.3). Removed Section 3.5 (United States Government Environment), reflecting the retirement of the USGCB program.
- Updated Section 4.2 to add the Cybersecurity and Infrastructure Security Agency (CISA) as an authoritative source of government checklists alongside DISA and NSA. Added guidance noting that government-sourced checklists, particularly military checklists, may represent a high-water mark that requires tailoring before adoption in civilian environments.
- Revised Section 4.3 to strengthen language around risk-based tailoring, documented justifications, risk acceptance, and compensating controls when deviating from checklist settings.
- Revised Section 5 to flatten the two-tier subsection structure (formerly Secs. 5.1.x and 5.2.x) into top-level subsections. Updated references from SP 800-53 Revision 4 to SP 800-53 Revision 5 and updated other cited NIST publications throughout.
- Removed Appendix D (Additional Requirements for USGCB Baselines) in its entirety, including the Field Testing Report template, reflecting the retirement of the United States Government Configuration Baseline program.
- Added a new Appendix C (Automating NIST CSF 2.0), covering how checklists connect to the NIST Cybersecurity Framework 2.0, including guidance on policy-to-automation pathways, implementation and traceability, checklist development guidance,

operational environment tailoring, and checklist submission and maintenance considerations.

- Renumbered appendices based on the changes described above and updated the Glossary to reflect revised definitions from this document.
- Provided other minor editorial changes and updated URLs throughout the document.