ie

National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-70
Revision 2 (Draft)

Sponsored by the Department of Homeland Security

National Checklist Program for IT Products—Guidelines for Checklist Users and Developers (Draft)

Recommendations of the National Institute of Standards and Technology

Stephen D. Quinn
Murugiah Souppaya
Melanie Cook
Karen Scarfone

NIST Special Publication 800-70
Revision 2 (Draft)

National Checklist Program for IT Products—Guidelines for Checklist Users and Developers (Draft)

Recommendations of the National Institute of Standards and Technology

Stephen D. Quinn
Murugiah Souppaya
Melanie Cook
Karen Scarfone

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

December 2010



U.S. Department of Commerce

Gary Locke, Secretary

National Institute of Standards and Technology

Patrick D. Gallagher, Director

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 analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-70 Revision 2 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-70 Rev. 2, 74 pages (Dec. 2010)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

This document is available for download at http://checklists.nist.gov/.

Acknowledgments

The authors, Stephen Quinn and Murugiah Souppaya of the National Institute of Standards and Technology (NIST), and Melanie Cook and Karen Scarfone of G2, Inc. wish to thank John Banghart and Suzanne Lightman of NIST, who reviewed drafts of this document and contributed to its technical content.

The authors acknowledge the individuals who assisted in the development of the original version of SP 800-70, including John Wack of NIST, who was a co-author of that version, and Anthony Harris and Paul M. Johnson of Booz Allen Hamilton, who contributed to the development of the initial draft publication; Timothy Grance, Jeffrey Horlick, Arnold Johnson, Mark Madsen, Edward Roback, Ron Ross, Michael Rubin, Carolyn Schmidt, and Matt Scholl of NIST; Clint Kreitner of the Center for Internet Security; Chase Carpenter, Kurt Dillard, and Jesper Johansson of Microsoft Corporation; Paul Bartock, Trent Pitsenbarger, and Neal Ziring of the National Security Agency; Terry Sherald of the Defense Information Systems Agency; Glenn Brunette of Sun Microsystems; and the following organizations that provided comments: Apple Computer, Inc., the Department of Energy, and Symantec Corporation. The authors also thank the individuals who contributed to Revision 1 of SP 800-70, including Timothy Grance and David Waltermire of NIST, Matt Barrett of G2, Inc., and Paul Cichonski of Booz Allen Hamilton.

The National Institute of Standards and Technology would also like to express its appreciation and thanks to the Department of Homeland Security for its sponsorship and support of the NIST National Checklist Program for IT Products.

Trademark Information

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

All other names are registered trademarks or trademarks of their respective companies.

Table of Contents

Executive Summary **ES-1**

1. Introduction **1-1**

1.1 Authority 1-1
1.2 Purpose and Scope 1-1
1.3 Audience 1-1
1.4 Document Organization 1-1

2. The NIST National Checklist Program **2-1**

2.1 Security Configuration Checklists 2-1
2.2 Benefits of Using Security Checklists 2-2
2.3 Overview of NIST National Checklist Program 2-3
2.3.1 Types of Checklists Listed by National Checklist Program 2-4
2.3.2 Procedures for Users and Developers 2-5

3. Operational Environments for Checklists **3-1**

3.1 Background 3-1
3.2 Standalone Environment 3-2
3.3 Managed Environment 3-4
3.4 Specialized Security-Limited Functionality Custom Environment 3-6
3.5 Legacy Environments 3-8
3.6 United States Government Environments 3-10

4. Checklist Usage **4-1**

4.1 Determining Local Requirements 4-2
4.2 Browsing and Retrieving Checklists 4-3
4.3 Reviewing, Customizing and Documenting, and Testing Checklists 4-9
4.4 Applying Checklists to IT Products 4-10

5. Checklist Development **5-1**

5.1 Background on Security-Related Criteria for Checklists 5-2
5.2 Developer Steps for Creating, Testing, and Submitting Checklists 5-2
5.2.1 Initial Checklist Development 5-3
5.2.2 Checklist Testing 5-4
5.2.3 Checklist Documented 5-5
5.2.4 Checklist Submitted to NIST 5-7
5.3 NIST Steps for Reviewing and Finalizing Checklists for Publication 5-7
5.3.1 NIST Screening of the Checklist Package 5-7
5.3.2 Public Review and Feedback for the Candidate Checklist 5-9
5.3.3 Final Listing on Checklist Repository, Maintenance, and Archival 5-9

Appendix A. References **A-1**

Appendix B. Checklist Description Template **B-1**

Appendix C. Checklist Program Operational Procedures **C-1**

1. Overview and General Considerations C-2
2. Checklist Submission and Screening C-3
3. Candidate Checklist Public Review C-4

4. Final Checklist Listing C-5
5. Final Checklist Update, Archival, and Delisting C-5
6. Record Keeping C-6

Appendix D. Participation and Logo Usage Agreement Form D-1

Appendix E. Additional Requirements for USGCB Baselines E-1

E.1 Developer Steps for Creating, Testing, and Submitting USGCB Baselines E-1
E.2 NIST Steps for Reviewing and Finalizing USGCB Baselines for Publication E-4
E.3 Field Testing Report Template E-5

Appendix F. Acronyms and Abbreviations F-1

Appendix G. Glossary G-1

List of Figures

Figure 3-1: Home Office Standalone Environment Example 3-3
Figure 3-2: Centrally Managed Environment Example 3-5
Figure 3-3: Specialized Security-Limited Functionality Environment Example 3-8
Figure 3-4: Legacy Environment Example 3-9
Figure 3-5: Legacy Workstation Environment 3-9
Figure 4-1: Checklist User Process Overview 4-1
Figure 4-2: NIST Checklist Repository Home Page 4-3
Figure 4-3: NIST Checklist Detail Page 4-4
Figure 5-1: NCP Checklist Development Steps 5-1

List of Tables

Table 4-1: Checklist Description Fields 4-4
Table 4-2: Checklist Tier Requirement Summary 4-8
Table 5-1: Fields Completed at Initial Checklist Development 5-3
Table 5-2: Fields Completed During Checklist Testing 5-4
Table 5-3: Additional Documentation Fields 5-6
Table B-1: Fields in the Checklist Description Template B-1

Executive Summary

A security configuration checklist (also called a lockdown, hardening guide, or benchmark) is a series of instructions for configuring a product to a particular operational environment. Checklists can comprise templates or automated scripts, patches or patch descriptions, Extensible Markup Language (XML) files, and other procedures. Checklists are intended to be tailored by each organization to meet its particular security and operational requirements. Some checklists also contain instructions for verifying that the product has been configured properly. Typically, checklists are created by IT vendors for their own products; however, checklists are also created by other organizations with the necessary technical competence, such as academia, consortia, and government agencies. The use of well-written, standardized checklists can markedly reduce the vulnerability exposure of IT products. Checklists can be particularly helpful to small organizations and to individuals with limited resources for securing their systems.

NIST maintains the National Checklist Repository, which is a publicly available resource that contains a variety of security configuration checklists for specific IT products or categories of IT products. The repository, which is located at http://checklists.nist.gov/, contains metadata that describes each checklist. Users can browse and search the metadata to locate a particular checklist using a variety of criteria, including the product category, vendor name, and submitting organization. Having a centralized checklist repository makes it easier for organizations to find the 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 checklists. For checklist users, this document makes recommendations for how they should select checklists from the NIST National Checklist Repository, evaluate and test checklists, and apply them to IT products. The document also provides general information to users about threats and fundamental technical security practices for associated operational environments. 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 vulnerabilities that attackers can attempt to exploit and to lessen the impact of successful attacks.

There is no checklist that can make a system or product 100 percent secure, and using checklists does not eliminate the need for ongoing security maintenance, such as patch installation. However, using checklists that emphasize both hardening of systems against software flaws (e.g., by applying patches and eliminating unnecessary functionality) and configuring systems securely 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. Checklists can also be used to verify the configuration of some types of security controls for system assessments, such as confirming compliance with certain Federal Information Security Management Act (FISMA) requirements or other sets of security requirements.

Federal agencies are required to use appropriate security configuration checklists from the NCP when available. In February 2008, revised Part 39 of the Federal Acquisition Regulation (FAR) was published. Paragraph (d) of section 39.101 states, "In acquiring information technology, agencies shall include the appropriate IT security policies and requirements, including use of common security configurations available from the NIST website at http://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated."[1] Also, FISMA (section 3544(b)(2)(D)(iii)) requires each Federal agency to determine minimally acceptable system configuration

[1] http://www.acquisition.gov/far/current/html/FARTOCP39.html

requirements and to ensure compliance with them. Accordingly, Federal agencies, as well as vendors of products for the Federal government, should acquire or implement and share such checklists using the NIST repository. NIST encourages checklist developers to assert mappings to the security controls delineated in NIST SP 800-53 to facilitate FISMA compliance checking for Federal agencies.[2]

Organizations should consider the availability of security configuration checklists during their IT product selection processes.

When selecting checklists, checklist users should carefully consider the degree of automation and the source of each checklist.

NIST has defined four tiers of checklists to assist checklist users in being able to readily identify the major differences among checklists. The tiers range from Tier I checklists, which are prose-based with narrative descriptions of how a person can manually alter a product's configuration, to Tier IV checklists, which have all security settings documented in machine-readable, standardized Security Content Automation Protocol (SCAP) formats; have been validated by NIST or a NIST-recognized authoritative entity for interoperability with SCAP-validated products; and have vetted mappings between low-level security settings (for example, standardized identifiers for individual security configuration issues) and high-level security requirements as represented in security frameworks (for example, SP 800-53 controls for FISMA).

When multiple checklists are available for a particular product, organizations should take into consideration the tier of each checklist. Generally, checklists from higher tiers can be used more consistently and efficiently than checklists at lower tiers. There may be other significant differences among checklists that are not indicated by the tier; 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., environment, threat model). A checklist user should identify such differences and determine which checklist(s) seem appropriate and merit further analysis.

If it is not clear which checklist(s) should be analyzed, users from Federal civilian agencies should first search for government-authorized or mandated checklists. In general, 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) or the National Security Agency (NSA) should be used if available. If formal government-authored checklists do not exist, then organizations are encouraged to use vendor-produced checklists. If vendor-produced checklists are not available, then checklists from other trusted third parties may be used.

Checklist users should customize and test checklists before applying them to production systems.

A checklist that is not mandatory for an organization to adopt should be considered a starting point for an organization to customize. Although the settings are based on sound knowledge of security threats and vulnerabilities, they cannot take into account organization-specific security and operational requirements, existing security controls, and other factors that may necessitate changes. 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 intended for an environment with significantly different security needs.

[2] Organizations are also encouraged to include information in their checklists that supports mapping to other sets of requirements, such as HIPAA.

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.

Before applying a checklist that will be used to alter product settings, users should first test it on non-critical systems, preferably in a controlled non-operational environment. 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, and some of these differences may affect checklist deployment. In some cases, a security control modification can have a negative impact on a product's functionality and usability, or on other products or security controls. Consequently, 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.

Checklist users should take their operational environments into account when selecting checklists, and checklist developers should target their checklists to one or more operational environments.

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

NIST strongly encourages IT product vendors to develop security configuration checklists for their products and contribute them to the NIST National Checklist Repository.

NIST encourages IT product vendors to develop security configuration checklists for their products, since the vendors have the most expertise on the possible security configuration settings and the best understanding of how the settings relate to and affect each other.

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 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 screens the checklist according to program requirements and then releases the checklist for public review, which typically lasts 30 to 60 days. After the public review period and subsequent resolution of issues, the checklist is listed on the NIST checklist repository with its metadata. NIST periodically asks checklist developers to review their checklists and to provide updates as necessary. NIST retires or archives checklists as they become outdated or incorrect.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347, and also under the Cyber Security Act, which tasks NIST to "develop, and revise as necessary, a checklist setting forth settings and option selections that minimize the security risks associated with each computer hardware or software system that is, or is likely to become widely used within the Federal Government."

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III [3].

This guideline has been prepared for use by federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict 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.

1.2 Purpose and Scope

This document describes security configuration checklists and their benefits, and explains how to use the NIST National Checklist Program (NCP) to find and retrieve checklists. The document also describes the policies, procedures, and general requirements for participation in the NCP.

1.3 Audience

This document was created for current and potential checklist developers and users in both the public and private sectors. Checklist developers include information technology (IT) vendors, consortia, industry, government organizations, and others in the public and private sectors. 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.

1.4 Document Organization

Section 2 contains an overview of checklists and describes the advantages of the NIST NCP and how it works. It contains cross-references to other sections of this document that provide greater detail.

Section 3 provides additional details on pre-defined checklist operational environments, threat discussions, and fundamental technical security practices that are used in the NCP to help developers create checklists that are consistent with security practices. The material presented in Section 3 can also

help checklist users better understand the fundamental security practices and 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 the identified needs. It also contains 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. This guidance contains information on the procedures for preparing and submitting a checklist to NIST for inclusion in the checklist repository.

Appendix A lists reference sources that were used to develop this document.

Appendix B describes the checklist description fields of the template used to catalogue checklists in the NIST repository.

Appendix C contains the programmatic and legal requirements that must be satisfied to participate in the NCP.

Appendix D contains the NCP participation and logo usage agreement form.

Appendix E details additional requirements that United States Government Configuration Baseline (USGCB) checklists must meet.

Appendix F contains a list of acronyms used in this document.

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

2. The NIST National Checklist Program

Maintaining secure networks and hosts continues to increase in importance. Widespread electronic attacks on all computer systems have become commonplace. There are many threats to users' computers, ranging from remotely launched exploitations of network services to malicious code spread through emails, malicious websites, and downloads of infected files. Vulnerabilities in IT products (e.g., operating systems and applications) are discovered almost daily, and many ready-to-use exploitation techniques are widely available on the Internet. Also, because IT products often are intended for a wide variety of audiences, restrictive security controls are usually not enabled by default, which means that many IT products are immediately vulnerable in their out-of-the-box configuration.

Complicating this situation is that today's systems and products can be complex to administer and difficult to secure. For example, the personal computer systems of today are far more complicated and sophisticated than yesterday's systems, and many if not most users and administrators cannot be expected to manage them securely without assistance. It is a complicated, arduous, and time-consuming task even for experienced system administrators to know what a reasonable set of security settings is for many different IT products. However, security is important to all audiences, from individual home users to large enterprise end users, because all systems face threats. In some cases, home and telecommuter user systems may benefit from the same strong security controls that are usually found in larger organizations because they face common threats via use of the Internet.

Although the solutions to IT security are complex, one simple yet effective tool is the security configuration checklist. To facilitate development of security configuration checklists and to meet the requirements of the Cyber Security Research and Development Act of 2002 (Public Law 107-305) (CSRDA), NIST developed the National Checklist Program (NCP) for IT Products. This section contains an overview of the NCP. It begins by describing the contents of checklists and giving examples of the types of IT products for which checklists are often created. It next explains the benefits of using security configuration checklists, such as improving the base level of security for an organization. It also explains the goals and benefits of the NCP, which include increasing the quality, usability, and availability of checklists. This section also provides an overview of the procedures for checklist users and developers, as well as a summary of FISMA-related guidance pertaining to use of configuration checklists.

2.1 Security Configuration Checklists

A *security configuration checklist* (also referred to as a lockdown guide, hardening guide, security guide, security technical implementation guide [STIG], or benchmark)[3] is essentially a document that contains instructions or procedures for configuring an IT product to an operational environment. Some checklists also contain instructions or procedures for verifying that the product has been configured properly. Using well-written, standardized checklists can reduce the vulnerability exposure of IT products and be particularly helpful to small organizations and individuals in securing their systems. Checklists can be developed not only by IT vendors, but also by other organizations with technical competence in IT product security. A security configuration checklist might include any of the following:

[3] From this point on in this document, the term *checklist* (used according to CSRDA terminology) is used to describe a security configuration checklist or what other literature may refer to as a lockdown guide, hardening guide, or benchmark configuration.

- Configuration files that automatically set or verify various security settings (e.g., executables, security templates that modify settings, Security Content Automation Protocol (SCAP) XML files, and scripts).[4]
- Documentation (e.g., text file) that guides the checklist user to manually configure an IT product
- Documents that explain the recommended methods to securely install and configure a device
- Policy documents that set forth guidelines for such things as auditing, authentication mechanisms (e.g., passwords), and perimeter security.

Not all instructions in a security configuration checklist address security settings. Checklists can also include administrative practices that improve an IT product's security. Often, successful attacks on systems result from poor administrative practices, such as not changing default passwords or not applying vendor patches.

Typically, a system administrator or end user follows the instructions in the checklist to configure a product or system to the level of security implemented in the checklist or to verify that a product or system is already configured properly. The system administrator may need to modify the checklist to incorporate the local security policy.

Examples of the types of devices and software for which security checklists are intended are as follows:

- General-purpose operating systems
- Common desktop applications such as email clients, web browsers, word processors, personal firewalls, and antivirus software
- Infrastructure devices such as routers, firewalls, virtual private network (VPN) gateways, intrusion detection systems (IDS), wireless access points, and telecommunication systems
- Application servers such as Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), web, Simple Mail Transfer Protocol (SMTP), and database servers
- Other network devices such as mobile devices, scanners, printers, copiers, and faxes.

2.2 Benefits of Using Security Checklists

Security configuration checklists, when developed correctly, can help users configure IT products so that they have more protection than the installed out-of-the-box defaults provide. Applying checklists to operating systems and applications can reduce the number of vulnerabilities that attackers can attempt to exploit and lessen the impact of successful attacks. 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 the following:

- Provides a base level of security to protect against common and dangerous local and remote threats (e.g., viruses and worms, denial-of-service attacks, unauthorized access, and inappropriate usage)

[4] More information about SCAP can be found at http://scap.nist.gov/ and NIST Special Publication 800-126, *The Technical Specification for the Security Content Automation Protocol (SCAP)* [21].

- Verify the configuration of certain technical security controls for system assessments, such as confirming compliance with certain FISMA requirements or other sets of requirements, and understanding the exposure caused by misconfigurations
- Significantly reduces the time required to research and develop appropriate security configurations for installed IT products
- Allows smaller organizations to leverage outside resources to implement recommended practice security configurations
- Reduces the likelihood of public loss of confidence or embarrassment resulting from a compromise of publicly accessible systems.

Although using security configuration checklists can significantly improve overall levels of security in organizations, using a checklist cannot make a system or a product 100 percent secure. However, using checklists that emphasize hardening of systems against the hidden software flaws will typically result in greater levels of product security and protection from future threats (e.g., zero-day vulnerabilities). IT vendors that configure their products using checklists that adhere to the FISMA-associated security control requirements will not only provide more consistency in configuration settings within the federal agencies but also provide a much more cost-effective method for establishing and verifying the minimum configuration settings, even if the agencies must modify the original checklists provided by checklist developers to fine-tune the configuration settings for their specific applications and operational environments.

2.3 Overview of NIST National Checklist Program

Many organizations have created checklists; however, these checklists vary widely in terms of quality and usability, and they may become outdated as software updates and upgrades are released. Without a central checklist repository, finding security checklists can be difficult. In addition, checklists may differ significantly from one another in terms of the level of security provided. Also, it may be difficult to determine if the checklist is current or how the checklist should be implemented.

To facilitate development of security configuration checklists for IT products and to make checklists more organized and usable, NIST established the NCP. The goals of the NCP are to—

- Facilitate 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 making checklists better documented and more usable
- Encourage software vendors and other parties to develop checklists
- Provide a managed process for the review, update, and maintenance of checklists
- Provide an easy-to-use repository of checklists
- Provide checklist content in a standardized format
- Encourage the use of automation technologies for checklist application.

Federal agencies are required to use appropriate security configuration checklists from the NCP when available. In February 2008, revised Part 39 of the Federal Acquisition Regulation (FAR) was published. Paragraph (d) of section 39.101 states, "In acquiring information technology, agencies shall include the appropriate IT security policies and requirements, including use of common security configurations available from the NIST website at http://checklists.nist.gov. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated."[5]

2.3.1 Types of Checklists Listed by National Checklist Program

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

- **Automated.** An automated checklist is one that is used through one or more tools that automatically alter or verify settings based on the contents of the checklist. Many checklists are written in Extensible Markup Language (XML), and there are special tools that can use the contents of the XML files to check and alter system settings.[6] For example, the Security Content Automation Protocol (SCAP) is commonly used to express checklist content in a standardized way that can be processed by tools that support SCAP.
- **Non-Automated.** As the name implies, a non-automated checklist is one 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.

Security configuration checklists in the NCP can help organizations meet FISMA requirements. FISMA requires each agency to determine minimally acceptable system configuration requirements and to ensure compliance with them. Checklists can also map specific technical control settings to the corresponding NIST SP 800-53 controls, which can make the verification of compliance more consistent and efficient. Accordingly, federal agencies, as well as vendors of products for the federal government, are encouraged to acquire or develop and to share such checklists using the NIST repository. The development and sharing of checklists can reduce what would otherwise be a "reinvention of the wheel" 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 http://checklists.nist.gov/) contains automated and non-automated checklists that have been developed and screened to meet the requirements of the NCP. Users can browse checklist descriptions to locate and retrieve a particular checklist using a variety of different fields, including such fields as product category, vendor name, and submitting organization. A mailing list for the checklist program is available at http://nvd.nist.gov/home.cfm?emaillist.

---

5 http://www.acquisition.gov/far/current/html/FARTOCP39.html

6 The Extensible Checklist Configuration Description Format (XCCDF) is an XML-based format for automating tool usage and eliminating interpretation issues. The XCCDF XML format can be used for both technical checklists (e.g., operating systems, software applications, and hardware configurations) and non-technical checklists (e.g., physical security for IT systems). More information on XCCDF is available from NIST Interagency Report (IR) 7275 Revision 3, *Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.1.4,* which is available for download at http://nvd.nist.gov/scap/xccdf/docs/xccdf-spec-1.1.4-20071102.pdf. Another XML-based format for checklists is the Open Vulnerability and Assessment Language (OVAL), which is used to exchange technical details about how to check for the presence of vulnerabilities and configuration issues on systems. More information on OVAL is available at http://oval.mitre.org/.

2.3.2 Procedures for Users and Developers

The general steps involved in acquiring and using checklists are simple and straightforward—

1. Users gather their local requirements (e.g., IT products, the operating environment, and 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 the user's operational environment and security requirements. If a product is intended to be secure out-of-the-box (e.g., it was secured by the vendor using a security configuration checklist), it is still important to check the NIST checklist repository for updates to that checklist.
3. Users review the checklists and select the checklist that best meets their requirements, then tailor and document the checklist as necessary to take into account local policies and functional requirements, test the checklist, and provide feedback to NIST and checklist developers.
4. Users prepare to deploy the checklist, such as making configuration or data backups, and then apply the checklist in production.

Section 4 provides more details on the activities and considerations associated with each step. The checklist description fields, used when browsing checklists, are summarized in Appendix B.

For checklist developers, the process includes two stages. The first stage involves actions by only the developer; the second stage involves interactions among NIST, the developer, and public reviewers. The first stage consists of four steps—

1. The developer becomes familiar with the procedures and requirements of the NCP and completes an agreement to participate in the program.
2. The developer creates, tests, and refines the checklist.
3. The developer documents the checklist according to the guidelines of the NCP.
4. The developer prepares a checklist submission package and submits it to NIST.

In stage two, NIST performs the remaining four steps, with interaction from the developer and public reviewers—

5. NIST screens the checklist according to program requirements and addresses any issues with the developer.
6. A public review of the checklist is conducted, which typically lasts 30 to 60 days. Comments submitted during the review are addressed as applicable by the developer and NIST.
7. NIST posts the checklist on the repository and announces its presence.
8. Periodic updates are made to the checklist and the issue of checklist archival is addressed.

3. Operational Environments for Checklists

Checklists are significantly more useful if they can be associated with generic operational environments. However, it is difficult and sometimes impossible to specify these environments in detail; they must by necessity be general so that they are useful to a wide range of audiences. 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 associated with the environments, and allows end users to more easily select the checklists that are most appropriate for their environments.

This section describes the operational environments defined for the NCP, and the general threat description and fundamental technical security practice for each environment. The two broad operational environments are referred to as **Standalone** (or Small Office/Home Office [SOHO]) and **Managed** (or Enterprise). Three typical **Custom** environments, which could be subsets of the broader environments, are **Specialized Security-Limited Functionality (SSLF), Legacy,** and **United States Government**.

Users of IT products may find it useful to consult this section of the document when initially identifying their own security requirements and needs (outlined in detail in Section 4). Developers may find this section useful when building checklists because tailoring checklist development to these environments and their policies will enable developers to create checklists for diverse products but still adhere to the general uniform technical security practices and settings associated with the environments. This is discussed in detail in Section 5. Before submitting a checklist to NIST, developers should ensure 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 http://checklists.nist.gov/.[7]

3.1 Background

When planning security, it is essential to first define the threats that must be mitigated. Knowledge of potential threats is important to understanding the reasons behind the various fundamental technical security practices presented in this document.

The threat discussions for each environment represent the major threat categories that were considered when selecting the environments and their associated fundamental security practices. Many threats against data and resources are possible because of mistakes—either software flaws and weak configuration settings in operating system and application software that create exploitable vulnerabilities, or errors made by end users and administrators. Threats may involve intentional actors (e.g., attacker who wants to access information on a system) or unintentional actors (e.g., administrator who forgets to disable user accounts of a former employee.) Threats can be local, such as a disgruntled employee, or remote, such as an attacker in another geographical area. Organizations that use checklists should conduct risk assessments to identify the specific threats against their systems and determine the effectiveness of existing security controls in counteracting the threats; they then should perform risk mitigation to decide what additional measures (if any) should be implemented, as discussed in the NIST *Guide for Applying the Risk Management Framework to Federal Information Systems* [6]. Performing risk assessments and mitigation helps organizations better understand their needs and decide whether or not they need to modify or enhance selected checklists.

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

The checklist environment fundamental technical security practices are based on commonly accepted technical security principles and practices, catalogued in various NIST Special Publications (SP) [13] and other sources such as the Department of Defense (DoD) *Information Assurance Technical Framework* [24]. In particular, NIST SP 800-27, *Engineering Principles for Information Technology Security (A Baseline for Achieving Security)* [4], contains a set of engineering principles for system security that provide a foundation upon which a more consistent and structured approach to the design, development, and implementation of IT security capabilities can be constructed. Section 5.1 contains a detailed discussion of the security-related criteria recommended for developers when building checklists.

3.2 Standalone Environment

The **Standalone** environment, also referred to as **Small Office/Home Office (SOHO)**, describes small, informal computer installations that are used for home or business purposes. This environment encompasses a variety of small-scale environments and devices, such as laptops, mobile devices, home computers, and remote systems (e.g., telecommuting systems and small branch offices). For technical and business (economic) reasons, SOHO systems are generally not managed remotely. Figure 3-1 shows a typical Standalone network architecture.

The Standalone environment assumes the following end-user audiences and operational settings:

- Home users with standalone systems, generally with dial-up or high-speed access to the Internet, possibly using wired or wireless home networks, and possibly sharing resources across the networks
- Telecommuters using standalone systems who work from a home office
- Small businesses, typically with small networks of standalone desktop systems and small office servers protected from direct Internet access by a firewall, but possibly including some small centrally managed networks of desktop systems and products, and typically not maintaining publicly accessible servers
- Other small organizations with similar functions.

Standalone environments are typically the least secured. The individuals who perform system administrator duties on Standalone systems are assumed to be less knowledgeable about security, which often results in environments that are less secure than they should be because the focus is on functionality. In some cases, there may be no network-based security controls such as firewalls, so Standalone systems may be directly exposed to external attacks. Standalone environments are frequently targeted for exploitation—not necessarily to acquire information, but instead to attack other computers or incidentally as collateral damage from the propagation of a worm.

Standalone checklists should be relatively simple to understand and implement by home users or novice system administrators in small organizations.

Because the primary threats in Standalone environments are external and because Standalone devices generally have less restrictive security policies than Managed or Specialized Security-Limited Functionality systems, they tend to be most vulnerable to attacks from remote threats. Local threats are often less significant because few people typically have local access to Standalone systems; however, it is still important to protect against local and other threats. Standalone systems typically are exposed to attacks against network services and by malicious payloads (e.g., viruses and worms). These attacks are most likely to affect availability (e.g., crashing the system, consuming all network bandwidth, breaking functionality), but they also may affect integrity (e.g., infecting data files) and confidentiality (e.g., providing remote access to sensitive data and emailing data files to others).

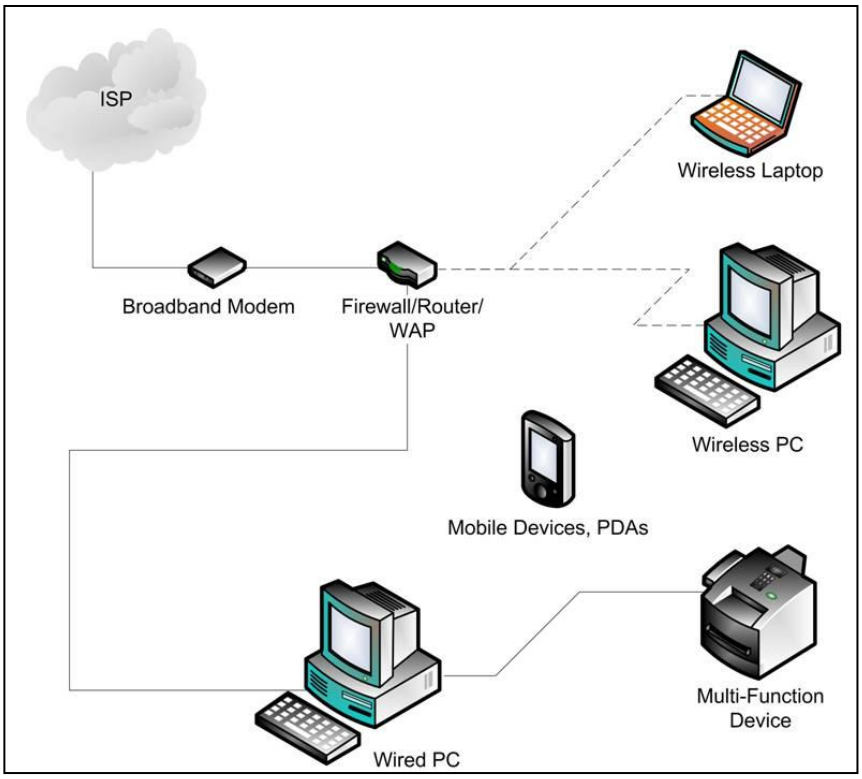

Figure 3-1: Home Office Standalone Environment Example

The fundamental technical security practices for the Standalone environment include protecting IT systems and products from the common out-of-the-box configuration vulnerabilities, blocking external access to the network, and restricting local access when possible. The adoption of inexpensive, hardware-based firewall routers and personal firewalls can help secure Standalone environments. Another key to Standalone security is strengthening the hosts on the Standalone network by patching vulnerabilities and altering settings to restrict unneeded services and applications. Some commonly accepted security practices for Standalone environments include the following:

- Use of small hardware firewall appliances at Internet connections to block inbound connections and to filter outbound traffic, if feasible
- Use of personal firewall products on Standalone systems
- Application (e.g., antivirus software, web browser, and email client) and operating system updates patches applied regularly
- Web and email clients configured to filter and block traffic/messages that could contain malicious content
- Unnecessary applications disabled (e.g., personal web servers, Simple Network Management Protocol [SNMP], messaging)
- Encryption used for wireless network traffic and as appropriate for other traffic
- Restrictions on which systems/users can connect to wired and wireless local area networks (LAN)

- Restrictions on user privileges
- Restrictions on sharing resources such as directories or printers
- Backup and recovery procedures
- Physical security procedures.

NIST and other security publications can be consulted for additional guidance in security practices related to Standalone environments. Users may find the guidance on system administration for Microsoft Windows systems [15], telecommuting [11], and wireless network security [12], [18] particularly useful. NIST has a variety of security-related SPs and general security guidance available on its computer security website.[8]

3.3 Managed Environment

The **Managed** environment, also referred to as **Enterprise**, typically contains large organizational systems with defined suites of hardware and software configurations, usually consisting of centrally managed IT products (e.g., workstations and servers) protected from direct Internet access by firewalls and other network security devices. Figure 3-2 shows a typical Enterprise network architecture. For example, it would include networked printers and multi-function devices, managed workstations, and internal servers.

The Managed environment audience generally includes medium to large businesses, large governmental agencies, and organizations requiring managed telecommuting systems and remote offices. Managed checklists are intended for advanced end users and system administrators in a medium to large organization. Managed environments typically have a group of individuals dedicated to supporting users and providing security. The combination of structure and skilled staff allows security practices to be implemented during initial system deployment and during ongoing support and maintenance. The managed nature of typical Managed environments gives administrators centralized control over various settings on workstations, servers, and other types of devices, as well as the sharing of resources (e.g., file servers and printers). The enterprise enables only the services needed for normal business operations, with other possible avenues of exploit removed or disabled. Authentication, account, and policy management can also be administered centrally to maintain a consistent security posture across an organization.

Remote and local threats to Managed networks could have significant impacts on systems and applications. Managed organizations often have systems with permanent, well-known IP addresses and name spaces with high visibility on the Internet. Most systems on Managed networks are inward-facing—protected from direct exposure to the Internet by firewalls—but penetrations of those systems through other means could allow an intruder to gain access to internal networks. For example, viruses and worms could spread across homogenous networks in a short time. Also, in Managed environments, the insider threat is generally greater than in a Standalone environment because the Managed environment has a larger number of users.

The Managed environment is more restrictive and provides less functionality than the Standalone environment. However, Managed environments typically have better control over the flow of various types of traffic, such as filtering traffic based on protocols and ports at the enterprise's connections with external networks. Because of the supported and largely homogeneous nature of the Managed environment, it is typically easier to use more functionally restrictive settings in Managed environments than in Standalone environments. Managed environments also tend to implement several layers of defense

[8] The NIST computer security website is located at http://csrc.nist.gov/.

(e.g., firewalls, antivirus servers, IDSs, patch management systems, and email filtering), which provides greater protection for systems.

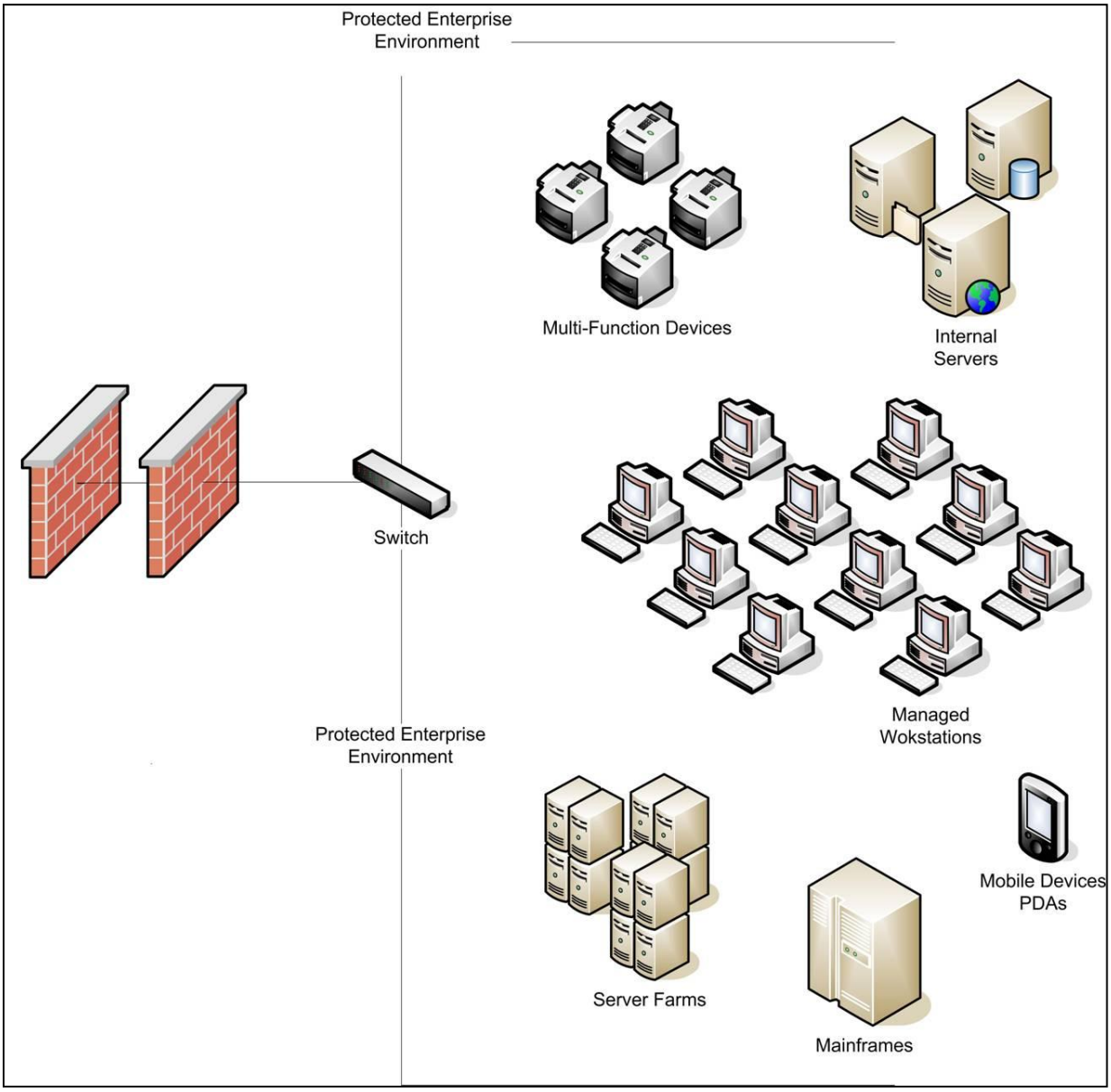

Figure 3-2: Centrally Managed Environment Example

In the Managed environment, systems are typically susceptible to local and remote threats. Local attacks, such as unauthorized use of another user's workstation, most often lead to a loss of confidentiality (e.g., unauthorized access to data), but also may lead to a loss of integrity (e.g., data modification) or availability (e.g., theft of a system). Remote threats may be posed not only by attackers outside the organization, but also by local users who are attacking other local systems across the organization's network. Most security breaches caused by remote threats involve malicious payloads sent by external parties, such as viruses and worms acquired from emails or infected websites. Threats against network-based applications tend to affect a smaller number of systems and may be caused by internal or external

parties. Both malicious payloads and network application attacks are most likely to affect availability (e.g., crashing the system, consuming all network bandwidth, and breaking functionality), but also may affect integrity (e.g., infecting data files) or confidentiality (e.g., providing remote access to sensitive data). Data disclosure threats tend to come from internal parties who are monitoring traffic on local networks, and they primarily affect confidentiality.

Some commonly accepted security practices for Managed environments are as follows:

- Segmented internal networks with internal firewalls and other defense-in-depth techniques
- Centralized management of systems with highly restricted local user access
- Centralized management of security-related applications such as antivirus software
- Automated installation of system and application patches and updates
- Restricted access to printer and multi-function devices and their features
- Centralized systems for log monitoring
- Centralized backup and recovery facilities.

Security publications can be consulted for additional guidance in security practices related to Managed environments. NIST has produced a variety of SPs that are particularly useful for the Managed operational environment. Relevant publications available from the NIST security website include guidance for system administration of Microsoft Windows systems [15], wireless network security [12], [18], active content and mobile code [5], security patches [7], firewalls [8], information security testing [19], and incident handling [14], [17].

3.4 Specialized Security-Limited Functionality Custom Environment

A **Custom** environment contains systems in which the functionality and degree of security do not fit the other types of environments. **Specialized Security-Limited Functionality (SSLF)** is a typical Custom environment that is highly restrictive and secure; it is usually reserved for systems that have the highest threats and associated impacts. Typical examples of such systems are outward-facing web, email, and DNS servers, other publicly accessed systems, and firewalls. It also encompasses computers that contain confidential information (e.g., central repository of personnel records, medical records, and financial information) or that perform vital organizational functions (e.g., accounting, payroll processing, and air traffic control). These systems might be targeted by third parties for exploitation, but also might be targeted by trusted parties inside the organization. Because systems in an SSLF environment are at high risk of attack or data exposure, security takes precedence over functionality. The systems' data content or mission purpose is of such value that aggressive tradeoffs in favor of security outweigh the potential negative consequences to other useful system attributes such as legacy applications or interoperability with other systems.

An SSLF environment could 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 (e.g., organization management) might be an SSLF environment in a Standalone environment. An SSLF environment might also be a self-contained environment outside any other environment, such as a government security installation processing sensitive data.

SSLF checklists are intended for experienced security specialists and seasoned 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 typically experience unwanted limitations on system functionality and cause possibly irreparable system damage.

Systems in the SSLF environment face the same threats as systems in Managed environments. Most recommendations for systems in SSLF environments are intended to thwart external threats; SSLF systems may be directly connected to the Internet, and as in the Managed environment, may have permanent, well-known IP addresses and name spaces with high visibility. Systems may be subject to automated intrusions and denial-of-service attacks as well as to manual intrusions, and penetration of firewalls and servers could lead to local attacks and intrusions. In addition, the threat of local attacks may be high if the systems are connected to large networks with many users; conversely, the threat of local attacks may be less if the systems are connected to smaller networks. Because of the risks and possible consequences of a compromise, this environment usually has the most functionally restrictive and secure configuration. The suggested configuration provides the greatest protection, with considerable tradeoffs to ease of use, functionality, and remote system management.

It is difficult to specify technical security practices except in general terms because many disparate types of systems and applications could, depending on how they are used, qualify as SSLF. However, it is likely that the following general practices and controls will be applicable:

- Systems should generally process as few types of data as possible (e.g., do not combine multiple server applications on the same system).
- Systems should be stripped of all unnecessary services and applications.
- If possible, host-based firewall applications should be used.
- Systems should have as few users as possible.
- The strongest possible authentication should be used (e.g., authentication token, biometrics, and smart cards).
- Remote administration or access should be restricted; if used, connections should be encrypted.
- Security-related operating system and application patches and updates should be tested and applied as soon as possible.
- Systems should be placed behind firewalls and other network security devices that restrict access and filter unnecessary protocols.
- Intrusion detection logs and other logs should be monitored frequently.
- Vulnerability assessment tools should be run against the systems frequently.
- System administrators should be highly skilled in the appropriate technologies.

NIST and other organizations have recommended security practices for firewalls [8], web servers [9], and email servers [10]. The publications mentioned previously for the Standalone and Managed environments also should be consulted for detailed recommendations.

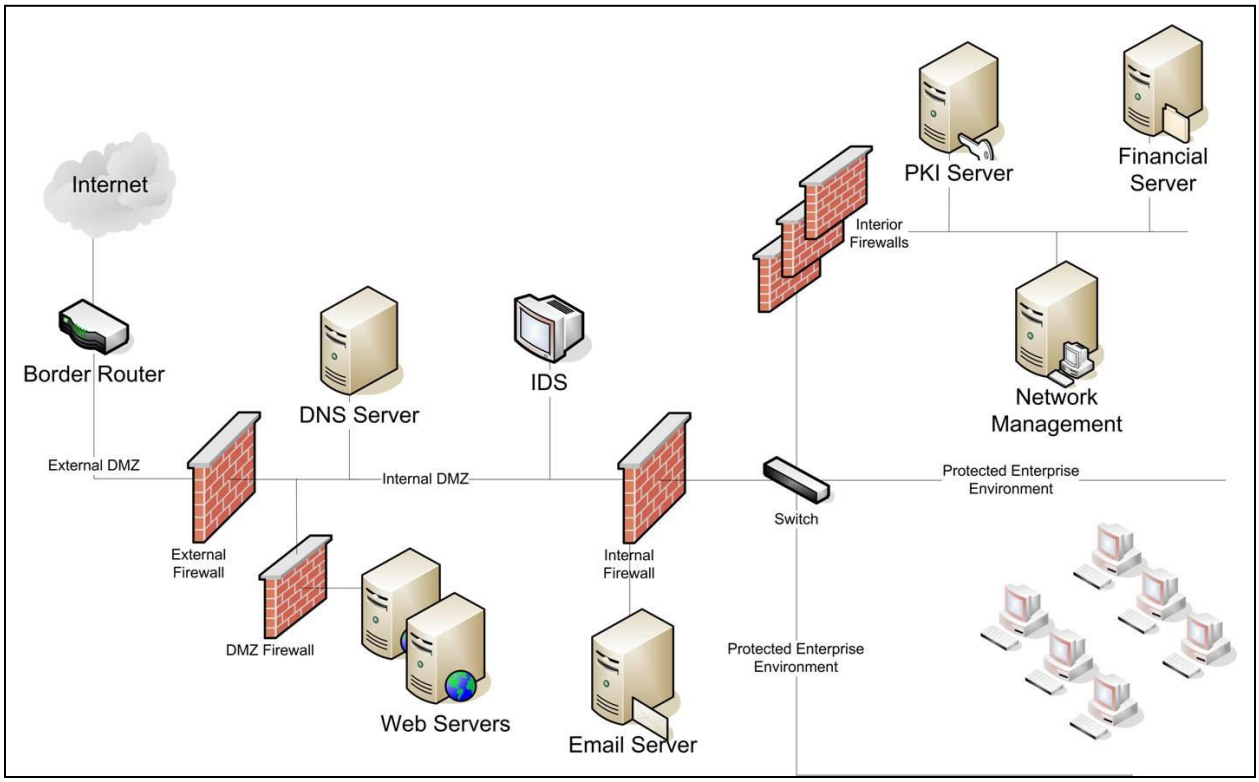

Figure 3-3: Specialized Security-Limited Functionality Environment Example

3.5 Legacy Environments

A Legacy environment is another example of a Custom environment. A Legacy environment contains older systems or applications that may need to be secured to meet today's threats, but they often use older, less secure communication mechanisms and 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. Legacy environments are often subsets of other environments.

An example of a Legacy environment is shown in Figure 3-4. Warehouse workers use wireless personal digital assistant (PDA) devices to collect inventory for shipping and receiving. The PDAs cannot be inexpensively upgraded to support wireless protocols with strong encryption capabilities. However, the location and structure of the warehouse prevents easy intercepts of the wireless traffic. Due to cost considerations, a risk determination was made and a Legacy environment checklist was created for the server/wireless access point. In such cases, compensating controls, such as configuring each application used by the PDAs to encrypt its communications, are typically needed to provide the protection that the legacy protocols cannot.

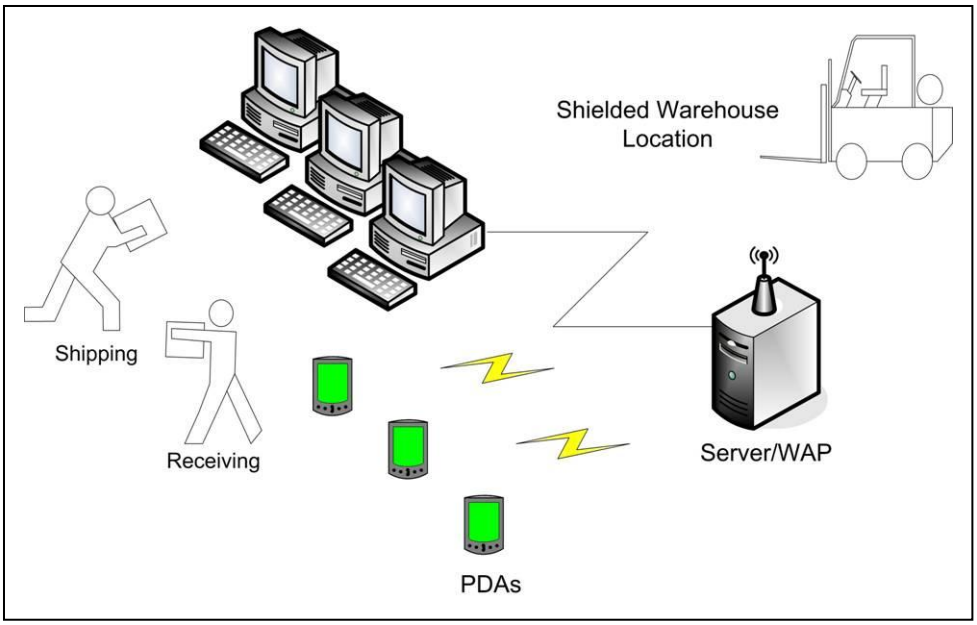

Figure 3-4: Legacy Environment Example

Figure 3-5 shows another simple example of a Legacy environment in which older workstations must be part of a network that uses more recent server technology. The older workstations cannot support newer, more robust aspects of the newer technology, such as a more secure file-sharing protocol, file system, or authentication protocol. Consequently, modifications must be made to support the legacy workstations. In this case, the server would require a Legacy environment checklist.

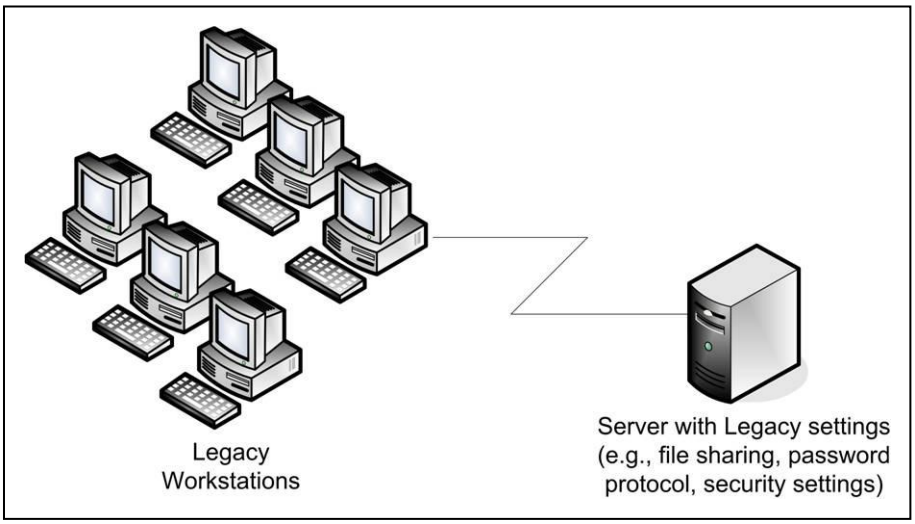

Figure 3-5: Legacy Workstation Environment

3.6 United States Government Environments

A United States Government environment is another example of a Custom environment. This environment contains federal government systems. These systems need to be secured according to prescribed configurations as mandated by policy. For example, the Federal Desktop Core Configuration (FDCC) is a security configuration policy mandated by the Office of Management and Budget (OMB). The original checklists developed in support of the FDCC policy exist for multiple versions of Microsoft Windows, Windows Firewall, and Internet Explorer. These checklists are broader than previous checklists, incorporating settings for Web browsers, personal firewalls, and other software. The configuration settings also include non security-related settings aimed at improving performance, energy efficiency, compatibility, and interoperability. The settings are largely based on the configuration settings recommended by Microsoft in its security guides, but they have been customized to take into account federal government security requirements. Many federal systems have been required to use these checklists by the OMB mandate.

Recently, the US government has focused on developing a new set of security configuration checklists to augment the existing checklists in support of the FDCC policy. These new checklists are known as the United States Government Configuration Baseline (USGCB).[9] Like the original checklists, the USGCB checklists also support the FDCC policy, and the USGCB checklists address a wide variety of security and non-security settings that are largely based on settings recommended by product vendors but customized to meet federal requirements. The USGCB initiative was created in 2010 by the Technology Infrastructure Subcommittee (TIS) of the CIO Council Architecture and Infrastructure Committee (AIC) as an evolution of the FDCC policy. The USGCB checklists are referred to as "baselines" because they define minimum sets of configurations that must be implemented. In 2011, new USGCB baselines will be release to replace the original checklists (Windows XP, Windows Vista, and Internet Explorer 7), and the original checklists will be deprecated.

The original checklists in support of the FDCC policy and USGCB baselines are intended to be applied to systems primarily through automated tools. Organizations should thoroughly test all checklists and baselines before deploying them in operational environments because a number of their settings, such as cryptographic algorithm options and wireless services, may impact system functionality. Settings may also be checked through automated means for compliance with checklists and baselines.

The USGCB configuration settings are intended to be deployed primarily to managed systems. The basic characteristics of Managed environments, such as primary threats against the systems and fundamental technical security practices for the systems, are also basic characteristics of United States Government environments. Section 3.3 contains additional information on Managed environments.

[9] More information on USGCB is available at http://usgcb.nist.gov/.

4. Checklist Usage

This section describes a high-level process for checklist users to follow when retrieving and using checklists. Although all checklist users, ranging from home users to system administrators at large organizations, 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 then describes a process for selecting and retrieving checklists from the NIST checklist repository, and recommends steps for analyzing, tailoring, and applying the checklist.

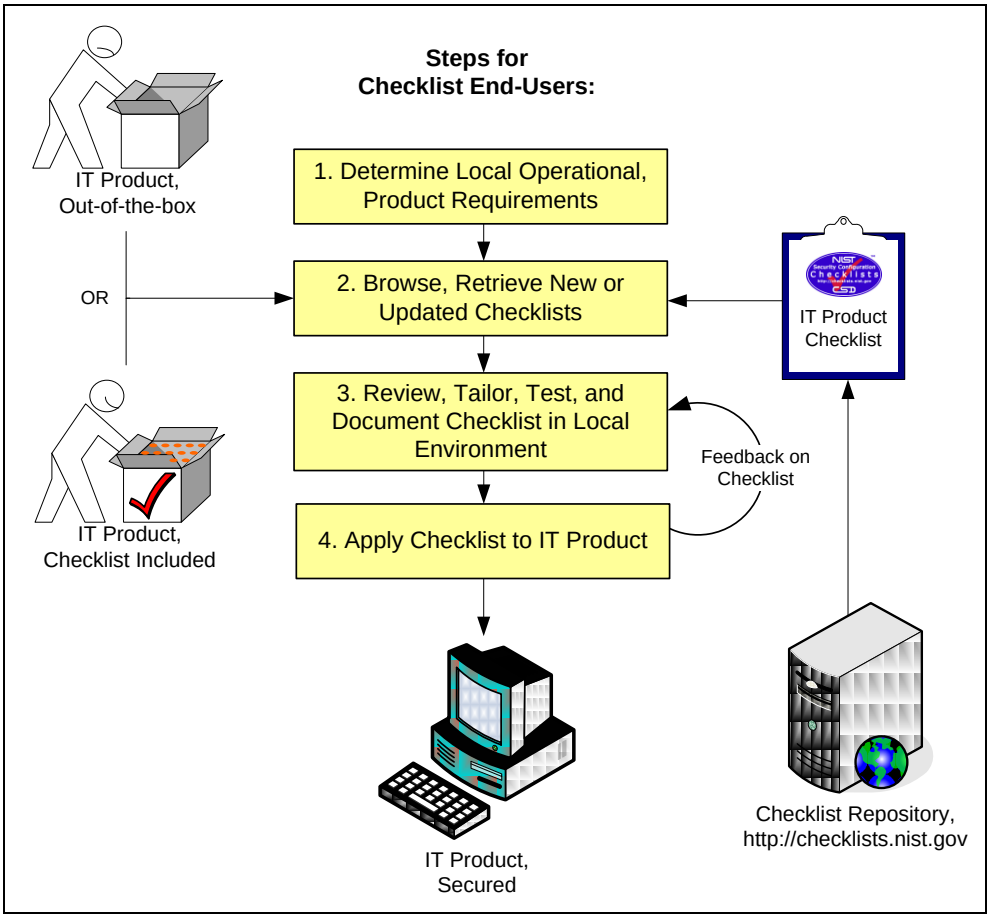

Figure 4-1: Checklist User Process Overview

Figure 4-1 shows the general process for using checklists. In Step 1, a prospective checklist user analyzes local requirements and security needs or policy and identifies the appropriate operational environment model. The user then selects the IT product that best matches those needs. In Step 2, the user browses the NIST repository for checklists that match the IT product and the selected operational environment (and possibly other criteria, such as whether the checklist can be rolled back or whether it is supported by the product vendor). The user downloads the checklists along with any supporting documents and tools. In Step 3, the user reviews the downloaded checklists and then tests and customizes them to reflect local policy and functionality as needed. Feedback on the checklists can be sent to NIST and the developer via the repository. In Step 4, the user prepares to apply the checklists in production by backing up

information that might be affected if the application of the checklists is not successful or if it causes unanticipated problems. Finally, the checklists are applied to production systems. The following sections describe the details of the activities included in each of these steps.

4.1 Determining Local Requirements

Organizations usually conduct a requirements analysis before actually selecting and purchasing a particular IT product. Such an analysis would include identifying the needs of the organization (what the product must do) and the security requirements for the product (e.g., relevant security policies). Individual end users can conduct the same process, although it could be quite informal. Because it is difficult to add security later, it is best to assess requirements upfront when incorporating security into IT operations, big or small.

NIST SP 800-37 Revision 1, *Guide for Applying the Risk Management Framework to Federal Information Systems* [6], contains useful guidance for federal agencies on selecting and implementing security controls, then conducting risk assessments. Organizations use risk assessments to determine the extent of the potential threat and the risk associated with an IT system or product throughout its life cycle. The output of this process helps to identify appropriate controls for reducing or eliminating risk. (Risk is a function of the likelihood of a given threat-source taking advantage of a particular potential vulnerability and the resulting impact of that adverse event on the organization.) Organizations other than federal agencies can also benefit from following the methodology presented in SP 800-37 Revision 1.

The methodology includes steps that are straightforward and simple, even for an individual home user who may not be especially savvy with regard to IT security. Important steps include the following:

- **Identify Functional Needs.** What must the product do? Identifying upfront the end user's requirements, such as remote access for telecommuters or a web server to make internal information available to employees, is necessary to ensure that the security controls selected are appropriate; that is, that they implement an appropriate security solution and still allow the system to meet its requirements for functionality.
- **Identify Threats and Vulnerabilities.** A threat is the potential for a particular threat-source to successfully exercise a particular vulnerability. A vulnerability is a weakness that can be accidentally triggered or intentionally exploited. The goal of this step is to identify potential threat-sources that are applicable to the IT product or system being considered, as well as the 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 exercising 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.

Federal agencies conduct formal requirements analysis and risk assessments as outlined in SP 800-37. For any organization or individual, the threat discussions and general security practices associated with each operational environment described in Section 3 can help identify threats and vulnerabilities and recommended security policies. For example, a home user could study the discussion in Section 3 on the Standalone operational environment before purchasing a product (assuming that the home user's environment matches the description of a Standalone environment). Given that the home user understands the requirements and the type of product that should be acquired, the home user can use the Standalone environment's security model and general recommendations to make an informed choice about which product best meets the needs.

NIST has also written several documents and guides to help federal agencies when selecting information security products and when acquiring and using tested/evaluated products. Another key resource available at NIST for identifying vulnerability-related information about IT products is the National Vulnerability Database (NVD).[10] This website provides a search engine for identified system vulnerabilities and information on patches that are available to correct the vulnerabilities.

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. Figure 4-2 shows an example of the repository home page. To help users obtain checklists that can be processed by SCAP-validated products, the checklists are sorted by default according to tier (described later in this section), from tier IV to tier I. Within each tier, the checklists are also sorted by default based on checklist authority (see the end of Section 4.2 for details). Users can browse the checklists based on the checklist tier, IT product, IT product category, or authority, and also through a keyword search that searches the checklist name and summary for user-specified terms. As shown in Figure 4-2, the search results show the detailed checklist metadata and a link to any SCAP content for the checklist, as well as links to any supporting resources associated with the checklist.

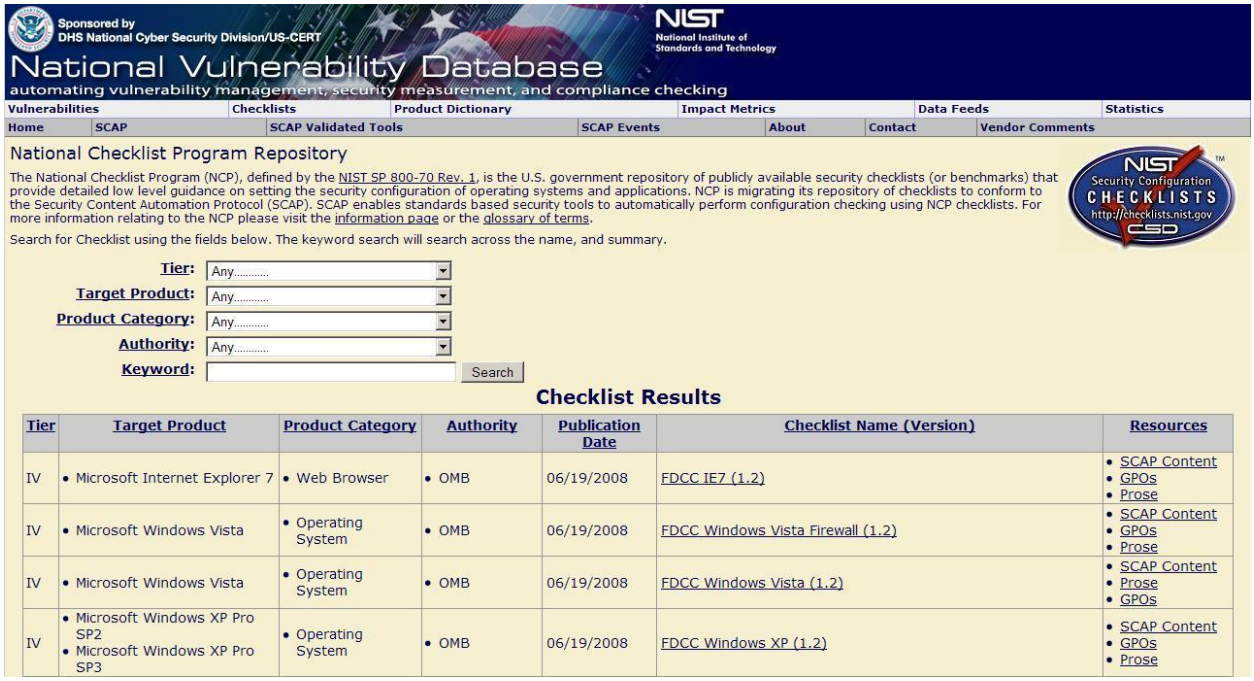

Figure 4-2: NIST Checklist Repository Home Page

Selecting a particular checklist will show a description template, shown in Figure 4-3, that includes extensive information to help users decide whether the checklist will suit their specific purposes. (The list and definition of all the fields used to describe each checklist is presented in Appendix B.)

[10] http://nvd.nist.gov/

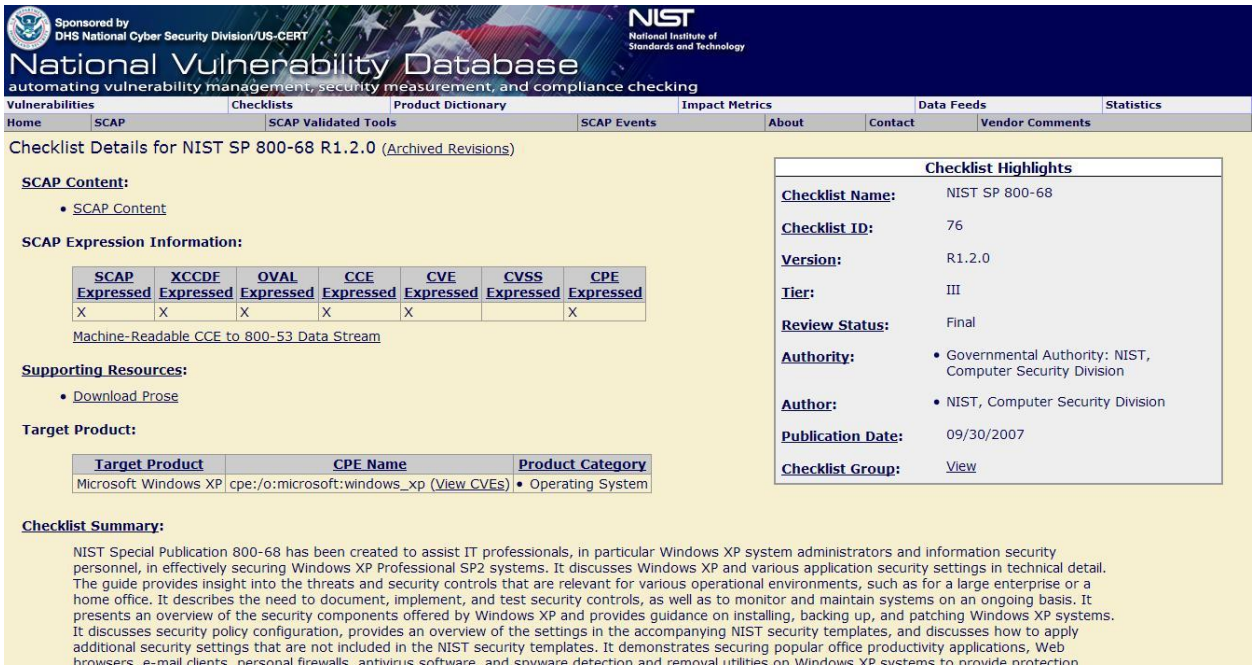

Figure 4-3: NIST Checklist Detail Page

Depending on a user's needs, role, and skills (e.g., home user versus enterprise administrator), some fields in the description will be more important than others. Table 4-1 lists fields that should be helpful to all users in determining whether the checklist meets their specific needs.

Table 4-1: Checklist Description Fields

Field Name	Description
Checklist Name	The name of the checklist.
Version	The version or release number of the checklist.
Review Status	The status of the checklist within the internal NCP review process. A status of "Final" signifies that NCP has reviewed the checklist and has accepted it for publication within the program. Possible status options are: Candidate, Final, Archived, or Under Review.
Entry Date	States the date when the checklist record was first listed in the NCP repository, in the format MM/DD/YYYY.
Publication Date	States the date when the actual checklist document was published, in the format MM/DD/YYYY.
Last Modified Date	States the date when the checklist record was last revised within the NCP repository, 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 Product(s)	The set of specific IT systems or applications that the checklist provides guidance for.
CPE Name	The CPE representation of a specific Target Product.
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).

Field Name	Description
Tier	The checklist tier (Tier I, II, III, or IV). ▪ Tier I checklists are prose-based, such as narrative descriptions of how a person can manually alter a product's configuration. ▪ Tier II checklists document their recommended security settings in a machine-readable but non-standard format, such as a proprietary format or a product-specific configuration script. These checklists may include some elements of SCAP (for example, they may contain CCE identifiers), but do not meet the Tier III requirements. ▪ Tier III checklists use SCAP to document their recommended security settings in machine-readable standardized SCAP formats that meet the definition of "SCAP Expressed" specified in NIST SP 800-126 [21]. Tier III checklists can be processed by SCAP-validated tools, which are products that have been validated by an accredited independent testing laboratory as conforming to applicable SCAP specifications and requirements. ▪ Tier IV checklists include all properties of Tier III checklists. Additionally, Tier IV checklists are considered production-ready and have been validated by NIST or a NIST-recognized authoritative entity to ensure, to the maximum extent possible, interoperability with SCAP-validated products. Tier IV checklists also demonstrate the ability to map low-level security settings (for example, standardized identifiers for individual security configuration issues) to high-level security requirements as represented in various security frameworks (e.g., SP 800-53 controls for FISMA), and the mappings have been vetted with the appropriate authority.
SCAP Expressed	Checklists that are designed to be processed by SCAP-validated products. For more details regarding the definition of SCAP Expressed, see NIST SP 800-126 [21].
XCCDF Expressed	Whether the checklist is expressed in XCCDF (yes or no). If yes, the checklist is expressed in XCCDF and validates against the published version of the XCCDF schema. The checklist also validates against the NIST-provided XCCDF reference implementation.
CCE Expressed	Whether the checklist has valid CCEs (yes or no). If yes, each configuration setting has an associated CCE.
CPE Expressed	Whether the checklist has valid CPEs (yes or no). If yes, the checklist expresses its applicability to systems using CPE.
CVE Expressed	Whether the checklist has valid CVEs (yes or no). If yes, each software flaw and patch has an associated CVE or CVEs.
CVSS Expressed	Whether the checklist has valid CVSSs (yes or no). If yes, each CVE identifier has an associated CVSS base score.
OVAL Expressed	Whether the checklist is expressed in OVAL (yes or no). If yes, each OVAL definition must validate according to the OVAL reference implementation.[11]
Checklist Summary	Summarizes the purpose of the checklist and its settings.
Known Issues	Summarizes issues that may arise after application of the checklist to help users pinpoint any functional and operational problems caused by the checklist.
Target 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.
Target Operational Environment	The IT product's operational environment, such as Standalone, Managed, or Custom (with description, such as Specialized Security-Limited Functionality, Legacy, or United States Government).
Checklist Installation Tools	Describes the functional tools required to use the checklist to configure the system, if they are not included with the checklist.

[11] More information on the OVAL reference implementation is available at http://ovaldi.wiki.sourceforge.net/.

Field Name	Description
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 any additional testing-related information such as summary of testing procedures used. Should specify any operational testing performed in production or mirrored production environments.
FIPS 140-2 Compliance	Whether the product can operate in a Federal Information Processing Standards (FIPS) 140-2 validated mode (yes or no).
Regulatory Compliance	Whether the checklist is consistent with various regulations (e.g., Health Information Portability and Accountability Act [HIPAA], Gramm-Leach-Bliley Act [GLBA], FISMA [such as mappings to NIST SP 800-53 controls], ISO 27001, Sarbanes-Oxley, Department of Defense [DoD] 8500).
Comments, Warnings, Miscellaneous	Any additional information that the checklist developer wishes to convey to users.
Disclaimer	Legal notice pertaining to the checklist.
Product Support	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 for usage of 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.
Authority	The organization responsible for producing the original security configuration guidance represented by the checklist. Authorities are ranked according to their "Authority Type." Within the NCP website, authorities are grouped with their authority types through the syntax of *Authority Type*: *Authority*. If it is not clear which checklists(s) should be analyzed, users from Federal civilian agencies should first search for checklists produced by authorities of type "Governmental Authority." If "Governmental Authority" produced checklists exist, the user 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) or the National Security Agency (NSA) should be used. If no "Governmental Authority" checklists exist, the user should search for checklists produced by authorities of type "Software Vendor." If none of these checklists exist, the user should search for checklists produced by authorities of type "Third Party."
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 NIST SP 800-68, the organization that created the SCAP content will be listed as the Author, but NIST will remain the Authority.
Authority Type	Type of organization that lends its authority to the checklist. The three types are Governmental Authority, Software Vendor, and Third Party (e.g., security organizations).
Point of Contact	An email address where questions, comments, suggestions, and problem reports can be sent in reference to the checklist. The point of contact should be an email address that the checklist developer monitors for checklist problem reports.
SCAP Content	A link to the machine-readable content representing the configuration guidance. This guidance is expressed using SCAP.
Supporting Resource	A link to any supporting information, or content, relating 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 product.
Change History	Running log detailing any changes made to the checklist since its inclusion in the repository. This field is updated with each version of checklist.
Dependency/ Requirement	Indicate 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.

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, a Checklist Group link is available; it represents the grouping of checklists based on a common source material. For example, the DISA Desktop Checklist contains configuration settings for multiple products including browsers and antivirus products. The NCP decomposes the checklist metadata 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 and the target audience (e.g., providing general recommendations versus complying with Federal agency-specific requirements). To assist checklist users in being able to readily identify the major differences among checklists, NIST has defined four tiers of checklists. The minimum requirements for each tier are listed below.

- Tier I checklists are prose-based, such as narrative descriptions of how a person can manually alter a product's configuration.
- Tier II checklists document their recommended security settings in a machine-readable but non-standard format, such as a proprietary format or a product-specific configuration script. These checklists may include some elements of SCAP (for example, they may contain CCE identifiers), but do not meet the Tier III requirements.
- Tier III checklists use SCAP to document their recommended security settings in machine-readable standardized SCAP formats that meet the definition of "SCAP Expressed" specified in NIST SP 800-126 [21]. Tier III checklists can be processed by SCAP-validated tools, which are products that have been validated by an accredited independent testing laboratory as conforming to applicable SCAP specifications and requirements. When evaluated using the NIST SCAP Content Validation Tool[12], a Tier III checklist provides a clean compile/run result.
- Tier IV checklists include all properties of Tier III checklists. Additionally, Tier IV checklists are used in the NIST validation program to ensure, to the maximum extent possible, interoperability with SCAP-validated products. Tier IV checklists also provide a complete mapping of low-level security settings (for example, standardized identifiers for individual security configuration issues) to high-level security requirements as represented in various security frameworks (e.g., SP 800-53 controls for FISMA), and the mappings have been vetted with the appropriate authority.

Table 4-2 summarizes the main differences in the requirements for the four tiers.

[12] The NIST SCAP Content Validation Tool is available for download on the SCAP specification website at http://scap.nist.gov/revision/index.html. This tool validates the correctness of the SCAP data stream according to the SCAP version specified in SP 800-126.

Table 4-2: Checklist Tier Requirement Summary

Tier	Machine Readable?	Automated Format?	References to Security Compliance Framework?
Tier I	No	N/A	Optional
Tier II	Yes	Non-standard (proprietary, product-specific, etc.)	Optional
Tier III	Yes	Complete SCAP-expressed checklist that can be processed by SCAP-validated tools and runs cleanly using the SCAP content validation tool.	Optional
Tier IV	Yes	Complete SCAP-expressed checklist that can be executed by SCAP-validated tools; has been validated by NIST or a NIST-accredited laboratory; and maps low-level security settings to high-level security requirements.	Required; must be vetted with at least one governance organization authoritative for the security compliance framework. Must demonstrate mapping capability from low level enumerations (CCE) to high level categorization (e.g., SP 800-53 controls).

Each checklist, regardless of tier, should provide checklist metadata, security configuration recommendations, and a description of the threat model on which the recommendations are based.

When multiple checklists are available for a particular product, organizations should take into consideration the tier of each checklist. Generally, checklists from higher tiers can be used more consistently and efficiently than checklists at lower tiers. There may be other significant differences among checklists that are not indicated by the tier; 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., environment, threat model). A checklist user should identify such differences and determine which checklist(s) seem appropriate and merit further analysis. If it is not clear which checklist(s) should be analyzed, users from Federal civilian agencies should first search for government-authorized or mandated checklists. In general, 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) or the National Security Agency (NSA) should be used if available. If formal government-authored checklists do not exist, then organizations are encouraged to use vendor-produced checklists. If vendor-produced checklists are not available, then checklists from other trusted third parties 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 the concept of a "Checklist Revision." A Checklist Revision indicates 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 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 archive the current checklist 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 and 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 modified[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 (e.g., having JavaScript disabled in a browser might make some web pages unusable). 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.[14] This is particularly true for checklists 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 FIPS 140 encryption requirements. 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 reason 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 as well as to 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-critical systems, preferably in a controlled non-operational environment. (Such testing may be difficult for home or small business users who do not have extra systems and networks for testing purposes.) 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, and some of these differences may affect checklist deployment. The testing configuration of the IT product should match the deployment configuration. In some cases, a security control modification can have a negative impact on a product's functionality and usability, or on other products or security controls. For example, installing a patch could inadvertently break another patch, or enabling a firewall could inadvertently block antivirus software from updating its signatures or disrupt patch management software. Consequently, 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 contains recommendations for performing backups and other suggestions to prevent or recover from potential damage or unwanted effects that could occur if applying an untested checklist.

Before using a checklist to verify product settings without altering them, users should test it. 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

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

[14] This may not be applicable to checklists that are mandatory for an organization to adopt.

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. The following provides recommendations for both ways of applying checklists:

- Setting Modification
  - Each checklist will include specific installation instructions to help with deployment. Even after review and testing, users should handle deployment carefully to minimize any issues that might arise from applying a security 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 completely and to determine if an initial backup is required. The *Rollback Capability* field in the checklist description (see Table 4-1) 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 the system can be restored to its pre-checklist state if necessary. (Making a full backup is recommended before making any major system change; it does not apply only to implementing a checklist.) Large organizations should also follow this procedure and, if possible, first select several operational systems as pilots to provide "real-world" testing for the checklist before enterprise-wide deployment.
- Setting Verification
  - Each checklist will include specific installation instructions to assist with using it to verify settings. Even after review and testing, users should handle verification carefully to ensure that product settings are not inadvertently altered.

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. For selected checklists, NIST may maintain a mailing address list of users, and users who subscribe to the list will receive announcements of updates or other issues connected with the checklist. Instructions for subscribing to the mailing address list will be included in the selected checklist's description on the checklist repository. 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.

NIST welcomes all feedback, "bug" reports, comments, and suggestions from checklist users in regard to individual checklists or the repository itself. Where applicable, NIST will encourage feedback from checklist users so that the developers are better able to gauge the effectiveness and appropriateness of their checklists.

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 NIST will follow to screen the checklist submissions and publish them in its repository, and the process NIST and developers will follow to update the checklist or to archive the checklist. Individual developers and organizations that want to submit checklists to NIST should review the appendices of this document, which contain the administrative requirements for participation in the NCP. Before submitting a checklist to NIST, developers should ensure they have the most recent version of this document. The most recent version is available as a separate file at http://checklists.nist.gov/.

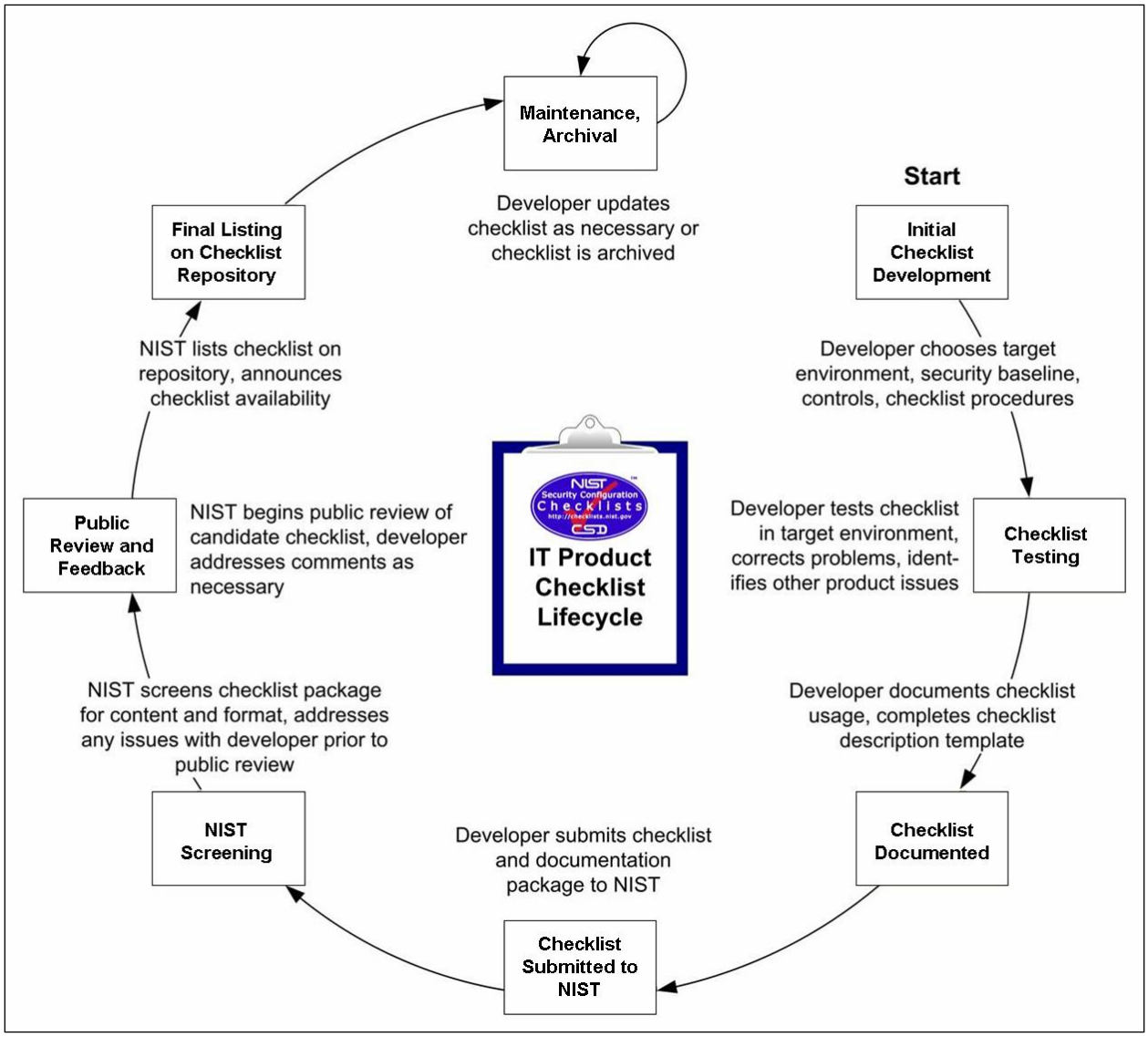

Figure 5-1: NCP Checklist Development Steps

The life-cycle steps shown in Figure 5-1 are straightforward. Each step should be carried out to ensure the checklist is accurate, tested, and documented during its development and subsequent publication, update, or archival. The following sections describe considerations for each step. USGCB checklists for the US Government environment follow the steps in this section, but they must also meet additional requirements as detailed in Appendix E.

5.1 Background on Security-Related Criteria for Checklists

This section discusses the security-related criteria that NIST recommends developers follow to enhance consistency of the technical security policy practices among the checklists. NIST recognizes that detailed checklist development cannot be covered extensively in this document. Therefore, NIST based the security-related criteria on commonly accepted technical security principles and practices, as catalogued in NIST SP 800-53 [13], other NIST publications, and other literature [24]. Additional considerations are contained in NIST SP 800-27, *Engineering Principles for Information Technology Security (A Baseline for Achieving Security)* [4]. To aid in designing secure information systems, NIST compiled a set of engineering principles for system security that are discussed in this document. These principles provide a foundation upon which a consistent and structured approach to the design, development, and implementation of IT security capabilities can be constructed. SP 800-27's guidance is based in part on the *Information Assurance Technical Framework (IATF)* [24].

The checklist must be consistent with one of the general operational environments described in Section 3 (excepting the Custom environment). This will require consulting the guidance in Section 3, the checklist format and content guidelines in the remainder of this section and in Appendix C, and other generally recommended practices and procedures. If no recommended practices guidance is available for a product or class of products, general security recommended practices should be used (e.g., defense in depth and layered security; least privilege, confidentiality, integrity, and availability controls).

In terms of vulnerability coverage, the security objectives should take into account the most up-to-date vulnerabilities and generally be consistent with recognized sources of vulnerability-related information, including the Department of Homeland Security's (DHS) United States Computer Emergency Readiness Team (US-CERT), the Computer Emergency Response Team/Coordination Center (CERT/CC), and NIST's NVD.[15]

Developers of checklists for products that are used by the federal government should consult the FISMA-associated security control requirements. NIST SP 800-53, *Recommended Security Controls for Federal Information Systems and Organizations* [13], provides a catalog of security controls, using groups of the controls to create three minimum security control sets for federal information systems—low, moderate, and high impact as specified in FIPS 199 [22]. Developers of IT products that will be used in federal information systems are encouraged to help federal agencies meet the mandatory requirements in FISMA by creating checklists that provide recommended configuration settings in a variety of operational environments or for information systems of differing impact levels, as described in FIPS 199 and SP 800-53. Developers are also encouraged to consider requirements imposed by HIPAA and other sources.

5.2 Developer Steps for Creating, Testing, and Submitting Checklists

The first four steps in the development methodology shown in Figure 5-1 begin with the developer becoming familiar with the procedures and requirements of the checklist program, and then performing the initial development of the checklist. Following initial development, the developer tests the checklist and refines it as needed. The third step involves documenting the checklist according to the guidelines of

[15] US-CERT website is http://www.us-cert.gov/. CERT/CC website is http://www.cert.org/. NVD is at http://nvd.nist.gov/.

the program. In the fourth step, the developer prepares and submits a checklist submission package to NIST for screening and public review. Sections 5.2.1 through 5.2.4 describe considerations in each of these steps.

5.2.1 Initial Checklist Development

During initial checklist development, a developer becomes familiar with the requirements of the checklist program and all procedures involved during the checklist life cycle (as described throughout this section). At this point, a developer would presumably agree to the requirements for participation in the NCP before continuing to develop the checklist. The participation requirements are described in this document, but are presented in administrative and programmatic terms in Appendix C, which is intended less for technical developers and more for those in developer organizations who must formally agree to NCP requirements. The participation agreement is contained in Appendix D.[16]

After agreeing to NCP requirements, the developer decides in which operational environment (see Section 3) the checklist should be implemented, and builds the checklist accordingly, using the security-related criteria presented in Sections 3 and 5.1. The output of this step is an initial checklist for the product.

Appendix B describes the complete set of fields for a checklist description on the repository; users can browse and view these fields when using the repository.

Table 5-1 shows the fields of the checklist description that would be completed at this step:

Table 5-1: Fields Completed at Initial Checklist Development

Field Name	Description
Checklist Name	The name of the checklist.
Product Category	The main product category of the IT product (e.g., firewall, IDS, operating system, web server).
Target Product(s)	The set of specific IT systems or applications that the checklist provides guidance for.
CPE Name	The CPE representation of a specific Target Product.
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	Summarizes the purpose of the checklist and its settings.
Known Issues	Summarizes issues that may arise after application of the checklist to help users pinpoint any functional and operational problems caused by the checklist.
Target 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.
Target Operational Environment	The IT product's operational environment, such as Standalone, Managed, or Custom (with description, such as Specialized Security-Limited Functionality, Legacy, or United States Government).

[16] The latest updates to these sections and to this document are available at http://checklists.nist.gov/. This updated material should be consulted before formally agreeing to participate in the program.

Field Name	Description
Checklist Installation Tools	Describes the functional tools required to use the checklist to configure the system, if they are not included with the checklist.
FIPS 140-2 Compliance	Whether the product can operate in a FIPS 140-2 validated mode (yes or no).
Regulatory Compliance	Whether the checklist is consistent with various regulations (e.g., Health information Portability and Accountability Act [HIPAA], Gramm-Leach-Bliley Act [GLBA], FISMA [such as mappings to NIST SP 800-53 controls], ISO 27001, Sarbanes-Oxley, Department of Defense [DoD] 8500).
Authority	The organization responsible for producing the original security configuration guidance represented by the checklist. Authorities are ranked according to their "Authority Type." Within the NCP website, authorities are grouped with their authority types through the syntax of *Authority Type*: *Authority*. If it is not clear which checklists(s) should be analyzed, users from Federal civilian agencies should first search for checklists produced by authorities of type "Governmental Authority." If "Governmental Authority" produced checklists exist, the user 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) or the National Security Agency (NSA) should be used. If no "Governmental Authority" checklists exist, the user should search for checklists produced by authorities of type "Software Vendor." If none of these checklists exist, the user should search for checklists produced by authorities of type "Third Party."
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 NIST SP 800-68, the organization that created the SCAP content will be listed as the Author, but NIST will remain the Authority.

5.2.2 Checklist Testing

Before a checklist is submitted to NIST, it should be fully tested in a configuration that meets the target environment and platform. The checklist should be tested with a variety of applications and hardware platforms, if applicable. 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; however, the developer should retain the data for review as appropriate.

Table 5-2 shows fields in the checklist description that would be completed at this step.

Table 5-2: Fields Completed During Checklist Testing

Field Name	Description
SCAP Expressed	Checklists that are designed to be processed by SCAP-validated products. For more details regarding the definition of SCAP Expressed, see NIST SP 800-126 [21].
XCCDF Expressed	Whether the checklist is expressed in XCCDF (yes or no). If yes, the checklist is expressed in XCCDF and validates against the published version of the XCCDF schema. The checklist also validates against the NIST-provided XCCDF reference implementation.

Field Name	Description
CCE Expressed	Whether the checklist has valid CCEs (yes or no). If yes, each configuration setting has an associated CCE.
CPE Expressed	Whether the checklist has valid CPEs (yes or no). If yes, the checklist expresses its applicability to systems using CPE.
CVE Expressed	Whether the checklist has valid CVEs (yes or no). If yes, each software flaw and patch has an associated CVE or CVEs.
CVSS Expressed	Whether the checklist has valid CVSSs (yes or no). If yes, each CVE identifier has an associated CVSS base score.
OVAL Expressed	Whether the checklist is expressed in OVAL (yes or no). If yes, each OVAL definition must validate according to the OVAL reference implementation.
Known Issues	Summarizes issues that may arise after application of the checklist to help users pinpoint any functional or operational problems caused by the checklist.
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 any additional testing-related information such as summary of testing procedures used. Should specify any operational testing performed in production or mirrored production environments.

Selecting the most appropriate set of security controls can be a daunting task because many security controls have limited system functionality and usability. In some cases, a security control can have a negative impact on other security controls. For example, installing a patch could inadvertently break another patch, or enabling a personal firewall could inadvertently block antivirus software from updating its signatures or disrupt patch management software. Therefore, it is important to perform testing for all security controls to determine what impact they have on system security, functionality, and usability, and to take appropriate steps to address any significant issues.

NIST has produced SP 800-115, *Technical Guide to Information Security Testing and Assessment* [19], to help administrators in testing systems for vulnerabilities and configuration problems. Although this publication is focused more on testing systems than testing on individual IT products, it may be useful to checklist developers.

5.2.3 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. In some cases, it may not be possible to roll back checklist settings, in which case the checklist documentation should recommend procedures such as backups and 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 cause the product to operate incorrectly. Ideally, assistance is available for (registered) users of the product if there are problems.

Table 5-3 shows additional fields in the checklist description that would be completed in this step.

Table 5-3: Additional Documentation Fields

Field Name	Description
Comments, Warnings, Miscellaneous	Any additional information that the checklist developer wishes to convey to users.
Disclaimer	Legal notice pertaining to the checklist.
Product Support	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 for usage of 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 where questions, comments, suggestions, and problem reports can be sent in reference to the checklist. The point of contact should be an email address that the checklist developer monitors for checklist problem reports.
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.
Licensing	States the license agreement (e.g., the checklist is copyrighted, open source, General Public License [GPL], free software, shareware).
SCAP Content	A link to the machine-readable content representing the configuration guidance. This guidance is expressed using SCAP.
Supporting Resource	A link to any supporting information, or content, relating 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 product.
Dependency/ Requirement	Indicate 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 describe the checklist accurately and minimize user confusion as to what the checklist accomplishes.

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

- Complete and accurate checklist description
- Statement of the security objectives, including the targeted environment and the expected behavior of the product after applying the checklist
- The target 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 operation of the product and any functionality 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
- Procedures for uninstalling the checklist (if applicable)

- Troubleshooting instructions or other information and references.

5.2.4 Checklist Submitted to NIST

At this point, the checklist developer has completed, tested, and documented the checklist. The developer now submits the package of materials to NIST. The package includes the following:

- Checklist and configuration files, templates, scripts, etc.
- 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 C, which also includes the appropriate NIST contact information.

Checklist packages may be submitted to NIST through the NCP Submission website. The website walks the checklist developer through a series of screens that collect all of the information and materials needed for checklist submission. In addition to that, the website allows checklist developers to view the checklists they have submitted, see tasks that have been assigned to them (such as fixing errors on a previously submitted checklist), update an existing checklist, and perform other actions. NIST also provides web services for submitting, fetching, and maintaining checklists. Additional information on the NCP Submission website and web services, including detailed instructions on their use, will be posted to http://web.nvd.nist.gov/view/ncp/information.

5.3 NIST Steps for Reviewing and Finalizing Checklists for Publication

The NIST process for screening and publishing a checklist is described in the following sections. Figure 5-1 shows the general steps; steps 4 and 5 may loop depending on the amount of feedback to the developer. Steps 6 and 8 may repeat depending on the magnitude of updates to an already-published checklist (most changes should not require additional public reviews).

5.3.1 NIST Screening of the Checklist Package

This step involves determining if the checklist is sufficiently accurate and complete to be publicly reviewed. NIST screens the checklist materials for completeness and accuracy, and examines the testing procedures used to evaluate the checklist. NIST may contact the developer with questions about the submitted materials during the screening period. NIST completes the screening and, if all issues are addressed, posts the checklist and its description as a candidate for public review for a period of 30 to 60 days.

The criteria used to screen the checklist are the same criteria that were used for checklist description development, which are described in Section 5.1. Essentially, the security objectives of the checklist should be consistent with recommended guidance from NIST and other recognized security organizations. The checklist must be documented according to the guidelines in this section and in Appendix C. Some of the questions typically posed by NIST when screening checklist submissions include the following:

- Documentation

  - Does it specify the checklist's automation, SCAP, OVAL, and XCCDF compliance?
  - Does it specify the target audience?
  - Does it identify the targeted environment?
  - Does it specify if the checklist is designed specifically for federal agencies?
  - Does it explain the security objectives?
  - Does it contain a complete, clear, and concise description of the checklist settings?
- Best Practices
  - Are the checklist settings consistent with recommended practices?
  - Do the checklist settings take into account 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 settings causes the product to meet the security objectives of the checklist?
  - Do any of the checklist settings cause the product to become inoperable or unstable?
  - 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 procedures to verify that the installation is successful included?
  - 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?
  - Does the documentation contain information for troubleshooting if errors occur or if the checklist settings cause the product to operate incorrectly?
  - Is there 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?

5.3.2 Public Review and Feedback for the Candidate Checklist

After the checklist has been screened and the developer has addressed any issues, NIST will post it for public review for a period of typically 30 to 60 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, it is included in the checklist repository.

A checklist reviewer will complete a form to provide comments as well as other information about 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 the following:

- 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.

At the end of the public review period, NIST will announce that the comment period is closed. Depending on the number of comments received and the ramifications of those comments to the checklist, NIST will specify a timeframe (typically 15 to 30 days from the end of the review period) in which the developer must respond to comments.

5.3.3 Final Listing on Checklist Repository, Maintenance, and Archival

After any outstanding issues are addressed, NIST lists the final checklist and announces that the checklist is now listed on the repository. 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 the developer provides assistance for the checklist. Requirements for use of the logo are described in Appendix D.

NIST will also announce procedures for accepting further comments or questions about the checklist throughout its life cycle. Depending on the product and how frequently updates occur, NIST may maintain a mailing address for the associated checklists. Users who subscribe to the mailing 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 mailing address list. Throughout the checklist life cycle, NIST will continue to collect feedback and pass this information to the checklist developer.

When the final checklist is listed, NIST will establish a periodic review schedule with the developer. Typically, the timeframe for the review will be 1 year; however, it could be sooner depending on factors such as the discovery of new vulnerabilities. If the developer decides to update the checklist, NIST will announce that the checklist is in the process of being updated. If the revised checklist contains major changes, it will be accepted as if it were a new submission, and will be required to undergo the same review process as a new submission.

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

Appendix A. References

Appendix A contains a list of the documents used to develop this publication.

[1] Cyber Security Research and Development Act of 2002, http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ305.107.pdf

[2] Federal Information Security Management Act (FISMA) of 2002, http://csrc.nist.gov/drivers/documents/FISMA-final.pdf

[3] OMB Circular A-130, http://www.whitehouse.gov/omb/circulars/a130/a130trans4.pdf

[4] NIST SP 800-27 Revision A, *Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A,* http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf

[5] NIST SP 800-28 Version 2, *Guidelines on Active Content and Mobile Code*, http://csrc.nist.gov/publications/nistpubs/800-28-ver2/SP800-28v2.pdf

[6] NIST SP 800-37, *Guide for Applying the Risk Management Framework to Federal Information Systems*, http://csrc.nist.gov/publications/nistpubs/800-37-rev1/sp800-37-rev1-final.pdf

[7] NIST SP 800-40 Version 2, *Creating a Patch and Vulnerability Management Program*, http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

[8] NIST SP 800-41 Revision 1, *Guidelines on Firewalls and Firewall Policy*, http://csrc.nist.gov/publications/PubsSPs.html

[9] NIST SP 800-44 Version 2, *Guidelines on Securing Public Web Servers*, http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf

[10] NIST SP 800-45 Version 2, *Guidelines on Electronic Mail Security*, http://csrc.nist.gov/publications/nistpubs/800-45-version2/SP800-45v2.pdf

[11] NIST SP 800-46 Revision 1, *Guide to Enterprise Telework and Remote Access Security*, http://csrc.nist.gov/publications/nistpubs/800-46-rev1/sp800-46r1.pdf

[12] NIST SP 800-48 Revision 1, *Guide to Securing Legacy IEEE 802.11 Wireless Networks*, http://csrc.nist.gov/publications/nistpubs/800-48-rev1/SP800-48r1.pdf

[13] NIST SP 800-53 Revision 3, *Recommended Security Controls for Federal Information Systems and Organizations,* http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final-errata.pdf

[14] NIST SP 800-61 Revision 1, *Computer Security Incident Handling Guide*, http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf

[15] NIST SP 800-68 Revision 1, *Guide to Securing Microsoft Windows XP Systems for IT Professionals: A NIST Security Configuration Checklist*, http://csrc.nist.gov/itsec/guidance_WinXP.html

[16] NIST SP 800-69, *Guidance for Securing Microsoft Windows XP Home Editions: A NIST Security Configuration Checklist*, http://csrc.nist.gov/itsec/guidance_WinXP_Home.html

[17] NIST SP 800-83, *Guide to Malware Incident Prevention and Handling*, http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf

[18] NIST SP 800-97, *Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i*, http://csrc.nist.gov/publications/nistpubs/800-97/SP800-97.pdf

[19] NIST SP 800-115, *Technical Guide to Information Security Testing and Assessment*, http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

[20] NIST SP 800-123, *Guide to General Server Security*, http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf

[21] NIST SP 800-126, *The Technical Specification for the Security Content Automation Protocol (SCAP)*, http://csrc.nist.gov/publications/PubsSPs.html

[22] FIPS PUB 199, Standards for Security Categorization of Federal Information and Information Systems, http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

[23] Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIG), available for .mil and .gov domains at https://iase.disa.mil/techguid/stigs.html and, for other domains, at http://iase.disa.mil/stigs/index.html

[24] *Information Assurance Technical Framework* (IATF), Release 3.0, 2000, http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA393328

[25] National Information Assurance (IA) Glossary, CNSS Instruction no. 4009, revised April 2010, http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

[26] National Security Agency (NSA) - Security Configuration Guides, available at http://www.nsa.gov/ia/

Appendix B. Checklist Description Template

Appendix B describes the fields of the checklist description that is maintained for each checklist on the checklist repository. The completed fields provide information about the checklist to users. Checklist developers must complete a checklist description form for each checklist. The latest version of the checklist description form can be downloaded from the checklist repository at http://checklists.nist.gov/.

Table B-1 lists all fields of the checklist description, with sample data from the NIST Microsoft Windows XP Professional checklist [15].

Table B-1: Fields in the Checklist Description Template

Field Name	Description	Example Data
Checklist Name	The name of the checklist.	https://www.nist.gov/csrc/checklists/repository/glossary.html - name*NIST SP 800-68*
Checklist ID	Uniquely identifies the checklist in the NCP repository. This will be generated during the NCP submission process and assigned to the checklist.	*76*
Version	The version or release number of the checklist.	*R1.2.1*
Review Status	The status of the checklist within the internal NCP review process. A status of "Final" signifies that NCP has reviewed the checklist and has accepted it for publication within the program. Possible status options are: Candidate, Final, Archived, or Under Review.	https://www.nist.gov/csrc/checklists/repository/glossary.html - status*Final*
Entry Date	States the date when the checklist record was first listed in the NCP repository, in the format MM/DD/YYYY.	https://www.nist.gov/csrc/checklists/repository/glossary.html - creation_date*10/30/2007*
Publication Date	States the date when the actual checklist document was published, in the format MM/DD/YYYY.	*04/01/2009*
Last Modified Date	States the date when the checklist record was last revised within the NCP repository, in the format MM/DD/YYYY.	*06/15/2009*
Product Category	The main product category of the IT product (e.g., firewall, IDS, operating system, web server).	*Operating System*
Target Product(s)	The set of specific IT systems or applications that the checklist provides guidance for.	*Microsoft Windows XP*
CPE Name	The CPE representation of a specific Target Product.	*cpe:/o:microsoft:windows_xp*
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).	*Operating System*

Field Name	Description	Example Data
Tier	The checklist tier (Tier I, II, III, or IV). ▪ Tier I checklists are prose-based, such as narrative descriptions of how a person can manually alter a product's configuration. ▪ Tier II checklists document their recommended security settings in a machine-readable but non-standard format, such as a proprietary format or a product-specific configuration script. These checklists may include some elements of SCAP (for example, they may contain CCE identifiers), but do not meet the Tier III requirements. ▪ Tier III checklists use SCAP to document their recommended security settings in machine-readable standardized SCAP formats that meet the definition of "SCAP Expressed" specified in NIST SP 800-126 [21]. Tier III checklists can be processed by SCAP-validated tools, which are products that have been validated by an accredited independent testing laboratory as conforming to applicable SCAP specifications and requirements. ▪ Tier IV checklists include all properties of Tier III checklists. Additionally, Tier IV checklists are considered production-ready and have been validated by NIST or a NIST-recognized authoritative entity to ensure, to the maximum extent possible, interoperability with SCAP-validated products. Tier IV checklists also demonstrate the ability to map low-level security settings (for example, standardized identifiers for individual security configuration issues) to high-level security requirements as represented in various security frameworks (e.g., SP 800-53 controls for FISMA), and the mappings have been vetted with the appropriate authority.	*Tier III*
SCAP Expressed	Checklists that are designed to be processed by SCAP-validated products. For more details regarding the definition of SCAP Expressed, see NIST SP 800-126 [21].	*Yes*

Field Name	Description	Example Data
XCCDF Expressed	Whether the checklist is expressed in XCCDF (yes or no). If yes, the checklist is expressed in XCCDF and validates against the published version of the XCCDF schema. The checklist also validates against the NIST-provided XCCDF reference implementation.	*Yes*
CCE Expressed	Whether the checklist has valid CCEs (yes or no). If yes, each configuration setting has an associated CCE.	*Yes*
CPE Expressed	Whether the checklist has valid CPEs (yes or no). If yes, the checklist expresses its applicability to systems using CPE.	*Yes*
CVE Expressed	Whether the checklist has valid CVEs (yes or no). If yes, each software flaw and patch has an associated CVE or CVEs.	*Yes*
CVSS Expressed	Whether the checklist has valid CVSSs (yes or no). If yes, each CVE identifier has an associated CVSS base score.	*No*
OVAL Expressed	Whether the checklist is expressed in OVAL (yes or no). If yes, each OVAL definition must validate according to the OVAL reference implementation.[17]	*Yes*
Checklist Summary	Summarizes the purpose of the checklist and its settings.	https://www.nist.gov/csrc/checklists/repository/glossary.html - checklist_summary*NIST SP 800-68 Revision 1.2.1 has been created to assist IT professionals, in particular Windows XP system administrators and information security personnel, in effectively securing Windows XP Professional SP2 and SP3 systems. It provides detailed information on Windows XP security, including security configuration guidelines for popular applications and Windows XP. The guide provides insight into the threats and security controls that are relevant for various operational environments, such as for a large enterprise or a home office. It describes the need to document, implement, and test security controls, as well as to monitor and maintain systems on an ongoing basis. It presents an overview of the security components offered by Windows XP and provides guidance on installing, backing up, and patching Windows XP systems. It discusses security policy configuration, provides an overview of the settings in the accompanying NIST security templates, and discusses how to apply additional security settings that are not included in the NIST security templates.*

[17] More information on the OVAL reference implementation is available at http://ovaldi.wiki.sourceforge.net/.

Field Name	Description	Example Data
Known Issues	Summarizes issues that may arise after application of the checklist to help users pinpoint any functional and operational problems caused by the checklist.	https://www.nist.gov/csrc/checklists/repository/glossary.html - known_issues*Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. These recommendations should be applied only to the Windows XP Professional SP2 and SP3 systems and will not work on Windows 9X/ME, Windows NT, Windows 2000 or Windows Server 2003. The security templates have been tested on WinXP Professional SP2 and SP3 systems and will not work on Windows 9X/ME, Windows NT, Windows 2000 or Windows Server 2003. The Specialized Security-Limited Functionality template should not be used by home users and should be used with caution since it will restrict the functionality and reduce the usability of the system.*
Target 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.	*This checklist has been created for IT professionals, particularly Windows XP system administrators and information security personnel. The document assumes that the reader has experience installing and administering Windows-based systems in domain or standalone configurations.*
Target Operational Environment	The IT product's operational environment, such as Standalone, Managed, or Custom (with description, such as Specialized Security-Limited Functionality, Legacy, or United States Government).	** Small Office/Home Office (SOHO)* ** Enterprise* ** Specialized Security-Limited Functionality (SSLF)* ** Legacy*
Checklist Installation Tools	Describes the functional tools required to use the checklist to configure the system, if they are not included with the checklist.	https://www.nist.gov/csrc/checklists/repository/glossary.html - checklist_intsllation_tools*The Microsoft Windows tools (e.g., Security Templates MMC snap-in, Security Configuration Analysis MMC snap-in, Group Policy MMC snap-in, Group Policy Management Console MMC snap-in) can be used to customize and apply the NIST security templates to Windows XP systems.*
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.	*There is no automated way of rolling back the settings unless a full system backup was performed before a security template was applied to the system.*
Testing Information	Platforms on which the checklist was tested. Can include any additional testing-related information such as summary of testing procedures used. Should specify any operational testing performed in production or mirrored production environments.	https://www.nist.gov/csrc/checklists/repository/glossary.html - testing_information*The security templates have been tested on Windows XP Professional SP2 and SP3 systems and will not work on Windows 9X/ME, Windows NT, Windows 2000 or Windows Server 2003.*
FIPS 140-2 Compliance	Whether the product can operate in a FIPS 140-2 validated mode (yes or no).	*Yes*
FIPS 140-2 Compliance Verification	Whether the checklist enumerates the required settings which must be configured on a product for the product to be FIPS 140-2 compliant.	*Yes*

Field Name	Description	Example Data
Regulatory Compliance	Whether the checklist is consistent with various regulations (e.g., Health information Portability and Accountability Act [HIPAA], Gramm-Leach-Bliley Act [GLBA], FISMA [such as mappings to NIST SP 800-53 controls], ISO 27001, Sarbanes-Oxley, Department of Defense [DoD] 8500).	*The recommendations are consistent with the security controls advocated in SP 800-53 (NIST FISMA implementation project publication).*
Comments, Warnings, Miscellaneous	Any additional information that the checklist developer wishes to convey to users.	*Refer to Known Issues.*
Disclaimer	Legal notice pertaining to the checklist.	https://www.nist.gov/csrc/checklists/repository/glossary.html - disclaimer*Do not attempt to implement any of the settings in this guide without first testing them in a non-operational environment. NIST assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. NIST would appreciate acknowledgement if the document and template are used.*
Product Support	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 for usage of 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.	*Microsoft will provide best efforts support, in line with the customer's support contract, to assist in removing the worst results of such file and registry permissions, but Microsoft can only guarantee returning to the recommended out-of-the-box settings by reformatting and reinstalling the operating system.*

Field Name	Description	Example Data
Authority	The organization responsible for producing the original security configuration guidance represented by the checklist. Authorities are ranked according to their "Authority Type." Within the NCP website, authorities are grouped with their authority types through the syntax of *Authority Type*: *Authority*. If it is not clear which checklists(s) should be analyzed, users from Federal civilian agencies should first search for checklists produced by authorities of type "Governmental Authority." If "Governmental Authority" produced checklists exist, the user 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) or the National Security Agency (NSA) should be used. If no "Governmental Authority" checklists exist, the user should search for checklists produced by authorities of type "Software Vendor." If none of these checklists exist, the user should search for checklists produced by authorities of type "Third Party."	*Governmental Authority: NIST, Computer Security Division*
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 NIST SP 800-68, the organization that created the SCAP content will be listed as the Author, but NIST will remain the Authority.	*NIST, Computer Security Division*
Authority Type	Type of organization that lends its authority to the checklist. The three types are Governmental Authority, Software Vendor, and Third Party (e.g., security organizations).	*Governmental Authority*
Point of Contact	An email address where questions, comments, suggestions, and problem reports can be sent in reference to the checklist. The point of contact should be an email address that the checklist developer monitors for checklist problem reports.	*itsec@nist.gov*
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.	*Vlad Pigin and Chase Carpenter, Microsoft Corporation*

Field Name	Description	Example Data
Licensing	States the license agreement (e.g., the checklist is copyrighted, open source, General Public License [GPL], free software, shareware).	https://www.nist.gov/csrc/checklists/repository/glossary.html - licensing*This document was developed at the National Institute of Standards and Technology, which collaborated with NSA, DISA, CIS, and Microsoft to produce the Windows XP security templates. Pursuant to title 17 Section 105 of the United States Code this document and template are not subject to copyright protection and are in the public domain.*
SCAP Content	A link to the machine-readable content representing the configuration guidance. This guidance is expressed using SCAP.	*http://nvd.nist.gov/scap/content/SCAP-WinXPPro.zip* sha1 = 07F6F12B9644AF79C63469F059EE4CA0B000C76E sha256 = 8FDAA4AF17890E1277DB381705175E0E4C45908E5A580679008DE9ACC66A093B
Supporting Resource	A link to any supporting information, or content, relating 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 product.	*Prose - http://csrc.nist.gov/itsec/guidance_WinXP.html*
Resource Description	A prose description of the resource.	NIST prose guide for Windows XP.
Resource Type	The format of the resource. Examples include SCAP content, prose, GPOs, security templates, etc.	Prose
SHA-1	The SHA-1 hash for the resource.	0C4020ADF1B066858F910FB3A627EF8D29F3D989
SHA-256	The SHA-256 hash for the resource.	1F618FD4C63A784C849B467D6402AD2E8D67BF61EC438489EC9A1A336FC427BC
Resources	Provides a logical grouping of the two content types within the National Checklist Program. Content found under this column includes SCAP Content and Supporting Resources.	SCAP Content (http://nvd.nist.gov/fdcc/fdcc-files-1.2.1.0/fdcc-winxp.zip) Prose (http://csrc.nist.gov/itsec/guidance_WinXP.html)

Field Name	Description	Example Data
Change History	Running log detailing any changes made to the checklist since its inclusion in the repository. This field is updated with each version of the checklist.	*Security Templates (.inf files)* *2007-05-08 - Release R1.2.1* *2005-11-02 - Release R1.2.0* *2004-08-24 - Draft Update R1.0.2* *2004-07-04 - Draft Update R1.0.1* *2004-06-24 - Draft Release R1.0* *SP 800-68 document* *2008-10-10 - Final Release of Revision 1* *2008-07-25 - Draft Release of Revision 1* *2005-11-02 - Final Release of Original Version* *2004-08-24 - Draft Update of Original Version* *2004-07-04 - Draft Update of Original Version* *2004-06-24 - Draft Release of Original Version* *SP 800-68 SCAP Content* *2009-07-29 - Final Release of Revision 1*
Dependency/ Requirement	Indicate 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.	https://www.nist.gov/csrc/checklists/repository/glossary.html - references*DISA, NSA, CIS, Microsoft and other security guides.*

Appendix C. Checklist Program Operational Procedures

NIST
National Institute of Standards and Technology



Operational Procedures
for
The NIST National Checklist Program
for Information Technology Products

Version 1.2

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 developer organizations who would 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 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—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* is a *Technical Configuration Checklist*, which is a checklist that refers to 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 public review or listing of the checklist. They work independently of other reviewers and are considered expert 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 PUB 199, Standards for Security Categorization of Federal Information and Information Systems, http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf
- NIST SP 800-27 Revision A, *Engineering Principles for Information Technology Security (A Baseline for Achieving Security*), *Revision A,* http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf
- NIST SP 800-53 Revision 3, *Recommended Security Controls for Federal Information Systems and Organizations*, http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final-errata.pdf
- Draft NIST SP 800-70 Revision 2, *National Checklist Program for IT Products*, http://csrc.nist.gov/publications/PubsSPs.html

1. Overview and General Considerations

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

(a) **Checklist Lifecycle Overview:** Checklists typically have the following lifecycle:

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. General information about checklists is discussed in Section 1. Checklist submission requirements and procedures are discussed in Section 2.
2. NIST verifies that all information is complete and performs a screening on the checklist. Checklists meeting the requirements for listing receive further consideration and are referred to as "candidate checklists." Section 2 discusses screening criteria and procedures. Section 1d discusses issue resolution processes.

3. NIST lists the candidate checklist on the repository for public review, typically for a period of 30 to 60 days, as discussed in Section 3.
4. NIST forwards comments from public reviewers to the developer. When all issues are addressed, the checklist is listed on the FCL, as discussed in Section 4.
5. The developer contacts NIST on typically an annual basis to determine whether the listing should continue, be updated, or be archived, as discussed in Section 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 the 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 not exclusive of the 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 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 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.

2. Checklist Submission and Screening

This section outlines the procedures and requirements for submitting checklists to NIST and the process by which NIST determines if 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 on the repository a complete set of information for developers. The information outlines the requirements for participation in the program and describes materials and timeframes.

(b) **Materials Required From the Developer:** Developers provide the following information:

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, and contact information for a backup or deputy point of contact. The information must include postal address, direct telephone number, facsimile number, and 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 to checklist developers. 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 checklists meet the program requirements. The following paragraphs summarize the screening criteria, which are described more fully in NIST Special Publication 800-70 Revision 2.

1. The checklist settings reflect consideration of recommended security and engineering practices.
2. The checklist contains a complete, clear, and concise description of the configuration settings.
3. The checklist has been tested and configuration or compatibility issues have been identified.
4. The documentation explains how to install and uninstall the checklist.
5. Checklist-related help is available.

3. Candidate Checklist Public Review

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

(a) **Public Review Period:** NIST typically lists candidate checklists for a 30 to 60 day comment period. NIST reserves the right to extend the review cycle, particularly for long or complicated checklists. NIST uses the following disclaimer (or very similar words) 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 complete a web-based feedback form to capture their comments as well as other information about the reviewer's test environment, procedures, and other relevant information. The contents of the feedback forms are considered public records.

(c) **Maintaining Records:** NIST maintains copies of all correspondence and feedback between the public and developers by creating a unique email address for each checklist. NIST will archive the information.

(d) **Addressing Comments:** At the end of the public review period, NIST announces that the comment period is closed. Depending on the number of comments received and the ramifications of those

comments to the checklist settings, NIST determines a timeframe in which the developer must respond to comments. This timeframe typically ranges from 15 to 30 days from the date the comments were submitted or from the end of the review period. At no time will this period be less than 15 days.

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. NIST uses the following disclaimer (or very similar words) for final 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) **Handling Comments:** NIST continues to accept comments about final checklists by maintaining a central email address on the repository. NIST lists the procedures to be used for contacting the developer, along with the contact information for the developer, such as an email address or URL.

(c) **Scheduling Periodic Reviews:** NIST determines whether a final checklist should be reviewed periodically and typically sets a review timeframe of one year. NIST may request that a checklist be reviewed sooner for reasons such as new vulnerabilities or threats. NIST schedules reviews with the developer's points of contact. If at any time the point of contact changes, NIST must be notified immediately.

5. Final Checklist Update, Archival, and Delisting

NIST follows the subsequent procedures for periodic update, archival, and delisting of final checklists.

(a) **Periodic Reviews:** NIST contacts developers at least annually to identify changes in the status of checklists. NIST also may contact developers, as appropriate, to determine if there are changes in the status of a checklist, in which case 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 periodic 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:** When a developer no longer provides support for the checklist, at the developer and NIST's discretion, the checklist can remain in the repository, but it will be reclassified as an archive. Typical reasons for archiving a checklist are that the product is no longer supported or is obsolete or that the developer no longer wants to provide support for the checklist.

(d) **Delisting:** NIST removes the checklist from the FCL. NIST may send announcements to various email lists maintained by NIST or other organizations.

(e) **Automatic Delisting:** If a final checklist is not reviewed annually, it is automatically removed from the FCL. At the developer and NIST's discretion, it can be reclassified as an archive.

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:** During the period that a checklist has been submitted to NIST, and during the period that a checklist is listed on the FCL as a final or archived checklist, and for three years thereafter, NIST 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. All comments submitted as part of the public review
4. All comments submitted to NIST regarding the checklist.

(b) **Developer Records:** During the period that a checklist has been submitted to NIST, and during the period that 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 D. 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 they have the most recent version of this appendix. The most recent version is available as a separate file at http://checklists.nist.gov/.

National Institute of Standards and Technology

NIST Security Configuration CHECKLISTS http://checklists.nist.gov CSD ™

Participation and Logo Usage Agreement Form for The NIST National Checklist Program for Information Technology Products

Version 1.3
December 10, 2010

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 C of NIST SP 800-70 Revision 2).

2. You will respond to comments and issues raised by a public review of your checklist submission. Any comments from reviewers and your responses may be made publicly available.

3. You agree to maintain the checklist and provide a timely response 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.

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, within one business week, appeal the rejection and provide supporting evidence.

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. 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 quality of the use of the phrase "NIST National Checklist Program for Information Technology Products" and the NIST National Checklist Program logo.

9. Your permission for advertising 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 for advertising participation in the NIST National Checklist Program and use of the logo is conditional on and limited to those checklist developers who provide assistance and help to users of the checklist with regard to 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 2.

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 E. Additional Requirements for USGCB Baselines

As mentioned in the Section 5 introduction, USGCB baselines have additional requirements that supplement those presented in Section 5. This appendix details these additional requirements and presents them based on the NCP Checklist Development Steps from Sections 5.2 and 5.3.

E.1 Developer Steps for Creating, Testing, and Submitting USGCB Baselines

A new USGCB baseline's development is led by any US federal agency, which is referred to in this appendix as the *champion agency*.

This portion of the appendix lists additional requirements related to creating, testing, and submitting USGCB baselines that the champion agency must follow. See Section 5.2 for the base requirements.

E.1.1 Initial Baseline Development

Each baseline originates from one or more agencies' recommendations regarding the appropriate configuration settings to achieve a given security state. For example, an agency may develop a comprehensive list of configuration settings for a popular operating system. If the settings may be applicable to a broad range of federal systems, the agency should consider submitting the settings as a USGCB baseline. USGCB baselines should be consistent with the guidance from NIST SP 800-53 Revision 3, which states that a baseline is "chosen based on the security category and associated impact level of the information system determined in accordance with FIPS 199 and FIPS 200, respectively."

USGCB settings are compiled by platform; a single platform may include one or more versions (e.g., Windows 7 32-bit and Windows 7 64-bit). The champion agency must ensure that a discrete setting is defined for each baseline configuration. Providing general guidance does not meet the settings requirement for a USGCB candidate. NIST recognizes that some configurations may be site specific and defining discrete settings that could be mandated for all Federal agencies is not a trivial task. During the creation of the candidate settings, the champion agency should remember that these settings are intended to be used by all Federal agencies; therefore, the USGCB settings may be considered a common subset applicable to all. USGCB candidates should reflect the minimum or core set of configurations that are applicable for all Federal agencies. Agencies using a USGCB baseline may customize it, making the settings more restrictive or appending additional settings. In the case of configurations applicable to a broad number of environments but not appropriate for all, USGCB introduces the notion of "Conditional" status. For example, the use of wireless technologies may be allowed at some sites, but not at others. The baseline would provide discrete wireless configurations applicable only to sites where wireless technology is allowed.

Developing a viable USGCB baseline requires expertise with the IT product and the ability to balance security and operational needs. During baseline development, discrete settings are defined, reviewed, and tested with the goal of arriving at a baseline that provides protection while allowing operational functionality. The champion agency should draw on field experience and available security configuration resources, such as government security guidelines, product security guidelines, and industry recommendations when developing baseline settings. Each baseline should be referenced to a security guide, such as an NSA Systems and Network Analysis Center security recommendation guide or a vendor security guide, if available. Champion agencies should also engage the product vendor during the baseline creation phase to ensure supportability and applicability. After settings are selected, the champion agency considers how each setting functions (e.g., registry value or file version) and identifies available methods for assessing compliance or determining a setting's value. As the baseline is created, the developers will test the system's behavior when settings are changed (e.g., examine the registry value or service status).

Each USGCB candidate must be a Tier III checklist, so it must be expressed as SCAP content. Each instance of SCAP content should contain a single USGCB baseline and should have only a single profile in its XCCDF component. The SCAP content must comply with the requirements in any revision of NIST SP 800-126, which defines the versions of SCAP, and the SCAP content must pass validation using the current version of the NIST SCAP Content Validation Tool (see Section 4.2). Using the latest version of SCAP is generally advantageous because the baseline can take advantage of newer specifications for more accurate checking, but it is not mandatory to use the latest SCAP version. The champion agency should identify all baseline settings that do not have OVAL checks, and then work with the MITRE Corporation and preferably with the product vendor to ensure that future versions of OVAL support these checks. Similarly, the champion agency should identify all configurations that do not have CCE identifiers, and then either the champion agency or the vendor should submit properly formatted CCE entries for these missing identifiers to the MITRE Corporation. All missing OVAL checks and CCE identifiers should be noted by the champion agency in the known issues document that is included with the USGCB candidate submission.

In addition to configuration checks, the champion agency should include up-to-date patch content, and the champion agency should continue to update the patch content before, during, and after baseline submission.

E.1.2 Baseline Testing

There are two major aspects to USGCB candidate testing: verifying that the SCAP content is compliant with SCAP technical requirements, and evaluating the baseline settings in an operational environment.

The champion agency should validate and test all SCAP content using the NIST SCAP Content Validation Tool (see Section 4.2) and the XCCDF Reference Implementation Tool[18], which was developed by NIST and the MITRE Corporation. The binary distribution is packaged with OVAL and OCIL open source checking engines: ovaldi and ocilqi. These tools are revised periodically as the SCAP specifications are updated.

Testing with the XCCDF Reference Implementation Tool should include assessing a system in three configurations:[19]

- Exact compliance: The configuration settings are equal to the discrete settings defined in the baseline.
- Reduced compliance: The configuration settings are less restrictive that those defined in the baseline.
- Enhanced compliance: The configuration settings are more restrictive than those defined in the baseline.

SCAP content testing must also include at least one validated SCAP validated product; the product chosen is at the discretion of the champion agency. If possible, validated product testing should simulate the environment that USGCB consumers will experience. A list of current SCAP Validation products can be found at http://scap.nist.gov/validation/index.html.

In addition to verifying baseline compliance with SCAP requirements, the champion agency should also test the baseline in an operational enterprise environment of considerable size and representative of a typical Federal agency. This testing ensures the viability of the baseline in an operational environment. NIST recommends testing the baseline for a minimum of three months. Evidence of field testing should be documented and include information about the location, duration, number of systems, issues identified,

[18] http://scap.nist.gov/specifications/xccdf/

[19] These terms come from NIST IR 7511, *Security Content Automation Protocol (SCAP) Version 1.0 Validation Program Test Requirements*.

and successful resolution to known issues. The Field Testing Report template is provided in Appendix E.3.

During the testing period, the baseline will be refined, arriving at a viable USGCB candidate baseline that is secure while accommodating operational requirements. The concept of leveraging a field tested configuration that provides security benefit without negative impact in an operational environment is paramount to the USGCB process. If baseline adjustments are needed to accommodate mission needs, the baseline is updated and redeployed to the same group of operational systems for additional field testing.

The configuration methods and materials are to be used for automating the configuration of test systems. The intended use of the configuration materials is facilitating lab setup for USGCB end users who test the baseline prior to deploying on operational systems. The format of these configuration materials may vary between products. For example, Microsoft provides Virtual Hard Disk (VHD) images and Group Policy Objects (GPOs), whereas Red Hat may provide kickstart scripts.

The champion agency should work with the vendor during baseline development and ensure the configuration automation materials produce a system that is USGCB compliant. NIST recommends the vendor choose the method and materials for configuration support. All configuration methods and materials in the USGCB candidate package should be fully tested, if possible during the field testing activities, and include end user instructions. At a minimum, test cases should ensure the methods and materials function as expected and produce a system that is compliant with the USGCB candidate. It is preferable that these materials be supported by the product vendor.

The USGCB candidate settings should be reviewed in a collaborative environment comprised of subject matter experts from the champion agency and product vendor. During this review, the group determines whether the baseline will have operational impact, addresses known issues discovered during field testing or revealed by the vendor security expert, and determines how to assess each setting with the Open Vulnerability and Assessment Language (OVAL). Because the product vendor will eventually be supporting operational implementations of the baseline, NIST encourages the product vendor to participate in the settings review and provide the following:

- Highlight settings that may have operational impact on systems
- Determine how each configuration setting can most accurately be assessed using an SCAP checking language (e.g., OVAL, OCIL)

Although not required, NIST recommends having a subject matter expert working group that reviews each baseline setting and reaches consensus about its viability for the USGCB candidate.

E.1.3 Baseline Documented

In addition to the baseline documentation already mentioned, such as the SCAP Tier III content and the automated configuration materials, other documentation is required for USGCB baselines.

Each baseline must be documented in a settings spreadsheet (.xls or .xlsx), which lists a discrete setting for every configuration in the baseline. NIST recognizes that inherent differences in products will dictate variations in the settings documentation; however, the following fields are required:

- CCE Identifier - Refer to the MITRE Corporation CCE List at http://mitre.org/lists/cce_list.html for version and platform group.
- Description of the setting – Include information needed to manually configure or assess. This will vary between products. For example, Windows documents define the Policy Path and Policy Setting Name, whereas Red Hat documents define the Technical Mechanism and Configuration Details.

- Setting – List the discrete setting recommended for the baseline
- Rationale – Describe the reason for this setting
- Impact – Describe potential operational impacts of this setting
- Category – Use this column to indicate "Conditional" settings
- Suggested SP 800-53 mapping – Map the setting to a NIST SP 800-53 security control.[20] NIST will verify the mappings and provide feedback.

Additional information may be included in the settings spreadsheet to provide explanation or technical details about the setting. Refer to http://usgcb.nist.gov for complete settings spreadsheets.

E.1.4 Baseline Submitted to NIST

Once the configuration baseline is defined, SCAP content is developed, and field testing is complete, the champion agency will submit the USGCB candidate package to the NIST checklist repository. A complete USGCB candidate submission must include the following:

- Baseline settings spreadsheet
- SCAP content: automated Tier III checklist with validated SCAP data streams
- Known issues spreadsheet, which lists all issues with the settings or SCAP data streams
- Frequently Asked Questions (FAQ) document that addresses the questions that baseline consumers are most likely to have
- Automated configuration materials (discussed below)
- Field testing report

E.2 NIST Steps for Reviewing and Finalizing USGCB Baselines for Publication

This portion of the appendix lists additional requirements related to NIST screening and publishing USGCB baselines. See Section 5.3 for the base requirements.

E.2.1 NIST Screening of the Baseline Package

NIST reviews the USGCB candidate submission and determines whether the submission meets all requirements for candidacy, namely the elements required for all NCP submissions plus the required USGCB elements, as listed in Appendix E.1.4. If the submission meets the requirements, NIST engages the TIS, Federal CIO Council, and OMB, and recommends the baseline as a USGCB candidate. If the submission does not meet all USGCB candidate requirements, NIST provides feedback to the champion agency, reporting on areas that should be addressed before resubmission.

After the TIS accepts the USGCB candidate submission, the NIST Security Automation Team conducts a formal engineering exercise. During this engineering exercise, all components of the candidate package are analyzed and tested in a lab at NIST. Subject matter experts review the baseline settings, ensuring they are consistent with existing USGCB baselines and suitable for broad implementation across Federal agencies. SCAP content is validated with the NIST SCAP Content Validation Tool and tested with the Reference Implementation Tool. Members of the NIST Security Automation Team test the configuration methods and materials, including the user instructions, by building test systems in the NIST lab. The SCAP content is imported into SCAP validated tools and used to assess the test systems. Test systems in exact compliance, reduced compliance, and enhanced or more restrictive compliance configurations are

[20] See http://web.nvd.nist.gov/view/800-53/home for additional information on the SP 800-53 security controls.

assessed during this exercise. NIST encourages the participation of technical representatives from the champion agency and product vendor during the engineering exercise to resolve any discrepancies.

At the conclusion of the NIST engineering exercise and prior to the USGCB draft release, the champion agency, in coordination with NIST, makes updates to the USGCB candidate. The champion agency updates the patch content and addresses findings from the engineering exercise. These revisions may affect one or all components that comprise the USGCB package. All findings from the engineering exercise must be either corrected or documented as known issues before the candidate can be released. NIST provides status of the engineering exercise to the TIS and champion agency. The TIS reserves the right to adjust the release schedule of the USGCB candidate based on the results of the NIST engineering exercise.

E.2.2 Public Review and Feedback for the Candidate Baseline

Once the USGCB candidate is approved by the TIS, NIST becomes the primary custodian of the baseline. The USGCB candidate is published for a 30-day comment period on the http://usgcb.nist.gov website as an alpha release. During this review period, community wide testing is expected and Federal agency representatives assigned to the TIS are required to submit consolidated comments on behalf of their respective agencies. NIST collects public comments sent to the usgcb@nist.gov mailing list and feedback from Federal agency TIS representatives regarding the viability of settings and the completeness and accuracy of the supporting materials. NIST consolidates and reviews this feedback and may propose changes to be included in the beta release. The TIS Change Control Board (CCB), in cooperation with NIST and other subject matter experts, adjudicates the proposed setting changes for the beta release and provides the results to NIST. TIS approved changes are included in the beta release, which is posted to the USGCB website for another 30-day review period. NIST consolidates feedback received during the beta review period, and again may propose changes to consider for the final release. Again, the TIS CCB adjudicates the proposed changes and can either declare the settings final or ask NIST to conduct a second or third beta release depending on the degree and nature of the TIS desired changes.

E.2.3 Final Listing on Checklist Repository, Maintenance, and Archival

After the TIS CCB approves the final configuration, OMB, the TIS, and the CIO Council formally release the USGCB final version and may provide a date for mandated implementation. The final USGCB is posted to http://usgcb.nist.gov. This final package includes the requisite settings documentation, SCAP content, automated configuration scripts or virtual disk images, an FAQ document, and a known issues document.

The TIS, OMB, and NIST maintain the USGCB package until the baseline is deprecated by the TIS. The TIS CCB continues vetting requested changes to the baseline, while NIST oversees maintenance of all USGCB components posted to http://usgcb.nist.gov/. These maintenance activities include updating the settings documentation, SCAP content, FAQ, and patch content as appropriate. Policy changes to the USGCB can only occur with approval from the TIS. NIST continues to update the technical content as appropriate to reflect the most accurate and consistent representation of the policy based on the public feedback and leveraging new advancement of capabilities as the SCAP matures. During maintenance, NIST coordinates with the product vendor, ensuring all automated configuration files are kept current in accordance with the vendor's update cycle for patches and at the request of NIST as per Appendix C, item 5a.

E.3 Field Testing Report Template

The following is the Field Testing Report template required for all USGCB candidate submissions.



National Institute of Standards and Technology

U.S. Department of Commerce

This Field Testing Report verifies successful testing of a USGCB candidate configuration in an operational environment. This report must be included with the USGCB candidate package submitted to the NIST National Checklist Program.

Champion Agency	
Champion Agency Point of Contact Name	
POC Email	
POC Phone	
Field Testing Site Location (Organization and location)	
Field Testing Technical Point of Contact Name	
POC Email	
POC Phone	
Dates of field testing	
Number of systems tested at field site	
Issue identified with the baseline	
Resolution to issue	
Report all known issues, extending this template as needed.	

Appendix F. Acronyms and Abbreviations

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

AIC	Architecture and Infrastructure Committee
CCB	Change Control Board
CCE	Common Configuration Enumeration
CERT®/CC	Computer Emergency Response Team/Coordination Center
CIS	Center for Internet Security
CMVP	Cryptographic Module Validation Program
COTS	Commercial Off-the-Shelf
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
FDCC	Federal Desktop Core Configuration
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GLBA	Gramm-Leach-Bliley Act
GPL	General Public License
GPO	Group Policy Object
HIPAA	Health Information 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
LAN	Local Area Network
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
PDA	Personal Digital Assistant
SCAP	Security Content Automation Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOHO	Small Office/Home Office
SP	Special Publication
SSLF	Specialized Security-Limited Functionality
STIG	Security Technical Implementation Guide
TIS	Technology Infrastructure Subcommittee
US-CERT	United States Computer Emergency Readiness Team
USGCB	United States Government Configuration Baseline
VHD	Virtual Hard Disk
VPN	Virtual Private Network
XCCDF	Extensible Configuration Checklist Description Format
XML	Extensible Markup Language

Appendix G. Glossary

Selected terms used in this guide are defined below. Definitions for some terms have been adapted from [25].

Availability: Timely, reliable access to data and information services for authorized users.

Candidate Checklist: Checklist approved by NIST for public review.

Confidentiality: Assurance that information is not disclosed to unauthorized individuals, processes, or devices.

Consortia: Associations or societies (e.g., Internet Engineering Task Force).

Consumer: Organization or private individual using checklists.

Custom Environment: Specialized operational environment.

Final Checklist: Checklist approved by NIST for placement on the repository.

Independent Qualified Reviewer: Reviewer tasked by NIST to make a recommendation about a checklist.

Integrity: Quality of a system or product reflecting the logical correctness and reliability of the operating system; verification that the original contents of information have not been altered or corrupted.

Inward-Facing: Description of a system that is connected on the interior of a network behind a firewall.

Legacy Environment: Typical Custom environment usually involving older systems or applications.

Logo: NIST National Checklist Program logo.

Managed Environment: Inward-facing environment that is typically very structured and centrally managed.

Operational Environment: Standalone, Managed, or Custom (including Specialized Security-Limited Functionality, Legacy, and United Stated Government).

Outward-Facing: Description of a system that is connected directly to the Internet.

Producer: Developer of a checklist.

Public Reviewer: Member of the general public who reviews a candidate checklist and sends comments to NIST.

Repository: NIST checklist repository; http://checklists.nist.gov/.

Specialized Security-Limited Functionality (SSLF) Environment: Environment encompassing systems with specialized security requirements, in which higher security needs typically result in more limited functionality.

Standalone Environment: Small office/home office environment.

Template: XML-encoded checklist description template that describes aspects of a checklist.

United States Government Environment: A Custom environment that contains federal government systems to be secured according to prescribed configurations as mandated by policy.