The attached DRAFT document (provided here for historical purposes) has been superseded by the following publication:

Publication Number: **NIST Interagency Report 8011 Volume 1**

Title: *Automation Support for Security Control Assessments. Volume 1: Overview*

Publication Date: **June 2017**

- Final Publication: https://doi.org/10.6028/NIST.IR.8011-1 (which links to http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8011-1.pdf).
- Information on other NIST cybersecurity publications and programs can be found at: https://csrc.nist.gov/publications

The following information was posted with the attached DRAFT document:

Feb. 2, 2016

***NIST IR 8011***

***DRAFT Automation Support for Security Control Assessments***
***Volume 1: Overview***
***Volume 2: Hardware Asset Management***

The National Institute of Standards and Technology (NIST) is pleased to announce the initial public draft release of NIST Internal Report (NISTIR) 8011, *Automation Support for Security Control Assessments*, Volumes 1 and 2. This NISTIR represents a joint effort between NIST and the Department of Homeland Security to provide an operational approach for automating security control assessments in order to facilitate information security continuous monitoring (ISCM), ongoing assessment, and ongoing security authorizations in a way that is consistent with the NIST Risk Management Framework overall and the guidance in NIST SPs 800-53 and 800-53A in particular.

NISTIR 8011 will ultimately consist of 13 volumes. Volume 1 introduces the general approach to automating security control assessments, 12 ISCM security capabilities, and terms and concepts common to all 12 capabilities. Volume 2 provides details specific to the hardware asset management security capability. The remaining 11 ISCM security capability volumes will provide details specific to each capability but will be organized in a very similar way to Volume 2.

Public comment period is open through ***March 18, 2016***. Please submit public comments to sec-cert@nist.gov. Comments are accepted in any desired format.

# DRAFT NISTIR 8011
# Volume 1

# Automation Support for Security Control Assessments

***Volume 1: Overview***

Kelley Dempsey
Paul Eavy
George Moore

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.XXXX

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

**DRAFT NISTIR 8011**
**Volume 1**

# Automation Support for Security Control Assessments

***Volume 1: Overview***

Kelley Dempsey
*Computer Security Division*
*Information Technology Laboratory*

Paul Eavy
*Federal Network Resilience Division*
*Department of Homeland Security*

George Moore
*Johns Hopkins University*
*Applied Physics Laboratory*

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.XXXX

February 2016

U.S. Department of Commerce
*Penny Pritzker, Secretary*

National Institute of Standards and Technology
*Willie May, Under Secretary of Commerce for Standards and Technology and Director*

National Institute of Standards and Technology Interagency Report 8011, Volume 1
93 pages (February 2016)

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.XXXX



**Public comment period: February 2, 2016 through March 18, 2016**

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

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

## Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof-of-concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

## Abstract

This volume introduces concepts to support automated assessment of most of the security controls in NIST Special Publication (SP) 800-53. Referencing SP 800-53A, the controls are divided into more granular parts (determination statements) to be assessed. The parts of the control assessed by each determination statement are called control items. These control items are then grouped into the appropriate security capabilities. As suggested by SP 800-53 Revision 4, *security capabilities* are groups of controls that support a common purpose. For effective automated assessment, testable *defect checks* are defined that bridge the determination statements to the broader security capabilities to be achieved and to the SP 800-53 security control items themselves. The defect checks correspond to security sub-capabilities—called sub-capabilities because each is part of a larger capability. Capabilities and sub-capabilities are both designed with the purpose of addressing a series of attack steps. Automated assessments (in the form of defect checks) are performed using the test assessment method defined in SP 800-53A by comparing a desired and actual state (or behavior).

## Keywords

actual state; assessment; assessment boundary; assessment method; authorization boundary; automated security control assessment; automation; capability; continuous diagnostics and mitigation; information security continuous monitoring; dashboard; defect; defect check; desired state specification; ISCM dashboard; mitigation; ongoing assessment; root cause analysis; security automation; security capability; security control; security control assessment; security control item.

# Acknowledgments

The authors, Kelley Dempsey of the National Institute of Standards and Technology (NIST), Dr. George Moore of the Applied Physics Laboratory at Johns Hopkins University, and Paul Eavy of the Department of Homeland Security, wish to thank their colleagues who reviewed drafts of this document, including Nadya Bartol, Craig Chase, Ann Dixon, Jim Foti, Susan Hansche, Amy Heydman, Alicia Jones, Betsy Kulick, Elizabeth Lennon, Susan Pagan, Daniel Portwood, Ron Ross, Martin Stanley, Kevin Stine, Robin Walker, Kimberly Watson, and Jim Wiggins. The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

# Table of Contents

# List of Figures

# List of Tables

## Executive Summary

Evolving threats create a challenge for organizations that design, implement, and operate complex information systems containing many moving parts. The ability to assess all implemented information security controls as frequently as needed using manual procedural methods has become impractical and unrealistic for most organizations due to the sheer size, complexity, and scope of their information technology footprint. Additionally, the rapid deployment of new technologies such as mobile, cloud, and social media brings with it new risks that make ongoing manual procedural assessments of all controls impossible for the vast majority of organizations. Today there is broad agreement in the information security community that once an information system is in production, automation of security control assessments[1] is needed to support and facilitate near real-time information security continuous monitoring (ISCM).

Early in FY 2011, as part of OMB M-11-33, the Office of Management and Budget approved the transition from a static every-three-year security authorization process to an ongoing authorization process via ISCM. In September 2011, NIST published SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, which provided management-level guidance on developing an ISCM strategy and implementing an ISCM program. However, many federal organizations were finding the technical implementation to be challenging.

Recognizing this challenge, the United States Congress funded the Continuous Diagnostics and Mitigation (CDM) program in 2012 at the Department of Homeland Security. The DHS CDM program is designed to facilitate automated security control assessment and continuous monitoring that is consistent with NIST guidance by providing a robust, comprehensive set of monitoring tools, an ISCM dashboard, and implementation assistance.

To help formalize the progress made as a result of OMB M-11-33, NIST SP 800-137, and the Department of Homeland Security CDM program, in November of 2013 OMB issued memorandum M-14-03, which provided instructions and deadlines to federal organizations for development of an ISCM strategy and program. The memorandum stated that each organization may follow one of three approaches for ISCM: 1) develop its own ISCM program; 2) leverage the CDM program from DHS; or 3) establish a hybrid program between its own ISCM program and the DHS CDM program.

This NIST Interagency Report (NISTIR) represents a joint effort between NIST and DHS to provides an operational approach for automating assessments of the selected and implemented security controls from SP 800-53 that is also consistent with the guidance in SP 800-53A.

Organizations implementing ISCM and automating security control assessments by any of these methods are encouraged to share the results with both NIST and DHS so that lessons learned can be shared broadly. If needed, this document will be revised and/or supplemented to document such best practices.

[1] See glossary for definition of *ongoing assessment*.

# 1. Introduction

## 1.1 *Purpose and Scope*

The purpose of this NISTIR is to provide an approach for automating the assessment of security controls in federal information systems and organizations to facilitate information security continuous monitoring, ongoing assessment, and ongoing security authorizations.

Automating security control assessments is important because security threats are materializing at an accelerated pace. Automated assessments have the potential to provide more timely data about security control defects (i.e., the absence or failure of a control), better enabling organizations to respond before vulnerabilities are exploited. Additionally, automated security control assessment has the potential to be less expensive and less human resource-intensive than manual procedural testing. Any realized savings could free up resources to be used on other activities, for example, investing in additional safeguards or countermeasures or responding to security defects and incidents in a more timely manner.

There are potentially many ways to automate the assessment of security controls to determine their effectiveness. This document provides an approach to automated security control assessment that is consistent with both NIST guidance and with the Department of Homeland Security (DHS) Continuous Diagnostics and Mitigation (CDM) program. The CDM approach, while not required, provides an acceptable method for automated security control assessments.

Organizations have the flexibility to innovate and find improved automated security control assessment approaches. When new assessment approaches are found, organizations are encouraged to share such approaches with other organizations by documenting and sending the new approaches to sec-cert@nist.gov.

The transition from manual to automated security control assessment requires time and preparation to implement a data collection system that supports automated security control assessments, as well as an information security continuous monitoring (ISCM) dashboard to visualize assessment results. It also requires resources to modify and update the assessment process. The DHS CDM program is designed to help federal organizations implement a robust data collection system and ISCM dashboard at the agency level.[2] This NISTIR supports the transition to automated security control assessments by providing a customizable security assessment plan that is consistent with both NIST guidance and the DHS CDM program.

This document, Volume 1 of NISTIR 8011, provides an overview of the automation of security controls assessments. Future volumes are planned for each of the security capabilities identified.

The ISCM security capabilities defined in this NISTIR represent sets of security controls logically grouped to fulfill a specific security purpose and to facilitate automated security control assessments. They are not a definitive set of security capabilities and are in no way intended to

[2] See glossary for definition of *agency dashboard*.

limit the flexibility of an organization to define different or additional capabilities.[3] The following are the ISCM security capabilities for which additional volumes will be published:[4]

Volume 2 Hardware Asset Management (HWAM)

Volume 3 Software Asset Management (SWAM)

Volume 4 Configuration Settings Management

Volume 5 Vulnerability Management

Volume 6 Boundary Management (Physical, Filters, and Other Boundaries)

Volume 7 Trust Management

Volume 8 Security-Related Behavior Management

Volume 9 Credentials and Authentication Management

Volume 10 Privilege and Account Management

Volume 11 Event (Incident and Contingency) Preparation Management

Volume 12 Anomalous Event Detection Management

Volume 13 Anomalous Event Response and Recovery Management

This overview volume provides a definition of the terms and overall processes that are common to automated security control assessment for ISCM security capabilities. Specific details regarding automated assessments of the capability and associated security controls are provided in the volumes covering the ISCM security capabilities.

## 1.2 *Target Audience*

This interagency report serves individuals associated with the design, development, implementation, operation, maintenance, and auditing of organizational information security continuous monitoring programs and security control assessment and authorization programs, including individuals with the following responsibilities:

- Information system development and integration (e.g., program managers, information technology product developers, information system developers, information system integrators, enterprise architects, information security architects);
- Information system and/or security management/oversight (e.g., senior leaders, risk executives, authorizing officials, chief information officers, senior information security officers);
- Information system and security control assessment and monitoring (e.g., system evaluators, assessors/assessment teams, independent verification and validation assessors, auditors, information system owners); and

[3]While consistent with the DHS CDM program, the security capabilities in this NISTIR are more granularly defined; however, both the CDM and NISTIR capabilities are designed to address SP 800-53 baseline security controls.
[4] For a description of all ISCM security capabilities, see Section 3.3.5.

- Information security implementation and operations (e.g., information system owners, common control providers, information owners/stewards, mission/business owners, information security architects, information system security engineers/officers, system/network/database administrators).

Note that this interagency report assumes that the reader has a working knowledge of the NIST Risk Management Framework (RMF) in general and specifically NIST Special Publications (SPs) 800-30, 800-39, 800-37, 800-53, 800-53A, and 800-137.

This publication assumes that the target audience has a working knowledge of information technology and information security terms and best practices. For definitions of unfamiliar terms, please see Appendix B of this volume or NISTIR 7298, *Glossary of Key Information Security Terms*.

## 1.3 *Organization of Volume 1*

The remainder of this publication is organized as follows:

Section 2, Overview of an Automated Security Control Assessment Process, describes how existing manual security control assessments can be adapted to an automated assessment approach and addresses concerns about the automation of security control assessment methods.

Section 3, Focusing Security Control Assessments on Security Results, describes the grouping of security controls by purpose (ISCM security capability) that facilitates automated security control assessments.

Section 4, Using Actual State and Desired State Specification to Detect Defects, defines the requisite preparation for automated security control assessment and describes how the process is able to determine the actual state and desired state specification so that it can compare those states.

Section 5, Defect Checks, describes the concept of a defect check.

Section 6, Assessment Plan Documentation, introduces the documentation produced for each security capability.

Section 7, Root Cause Analysis, describes root cause analysis of a security control issue, a defect check failure, or a failure of a security capability to produce the desired overall security result.

Section 8, Roles and Responsibilities, describes operational roles and responsibilities and contrasts them with information system security managerial roles and responsibilities in NIST Special Publications.

Section 9, Relationship of Automated Security Control Assessment to the NIST Risk Management Framework, describes the tasks and function of automated ISCM within the Assessment phase of the RMF.

# 2. Overview of an Automated Security Control Assessment Process

Attacks on information systems are being perpetrated at an accelerating pace. A security defect (i.e., control failure or absence) that is useful to an attacker is likely to be exploited very quickly. Organizations using manual defect detection methods will likely never be able to detect and fix security defects faster than attackers—who are using automated attacks—can detect and exploit those defects. Also, manual assessment is often more expensive than automated assessment (e.g., consider what it would cost to detect unpatched devices manually, compared to the cost of using a vulnerability scanner). Manual security control assessment may not identify defects in a timely manner and is generally more resource-intensive over the long term than automated assessment.

This section discusses how existing manual security control assessments can be adapted to an automated security control assessment process. It also offers solutions to concerns about the automation of security control assessment methods.

## 2.1 *Prerequisites to Automated Security Control Assessment*

The security control assessment process presented in this NISTIR is designed to be used after the initial assessment and authorization (A&A)[5] is completed. While some results from automated security control assessments might be applicable for an information system's initial assessment, this document focuses on the subsequent security control assessments that are embedded in the information security continuous monitoring (ISCM) process for information systems in the operations and maintenance phase of the system development life cycle.

As a corollary to the assumption that an initial A&A was conducted consistent with SP 800-37 and related guidance, there is an assumption that the system(s) being assessed have the normal complement of security documentation, including the system security plan, the initial (or most current) security assessment report, and supporting documents such as the information system contingency plan.

> This NISTIR focuses on automation of the assessment of security controls selected for each impact level baseline, as defined in SP 800-53. More information on the automated assessment of specific security controls is found in the security capability volumes. If a system's *tailored* baseline includes additional security controls not selected in SP 800-53 baselines (i.e., security control supplementation), they may not be covered in this NISTIR. Manual/procedural methods are applied to assess such controls and the manually generated, security-related information is considered when making risk-based decisions.

[5] See SP 800-37 for more information on the security risk management framework and A&A.

## 2.2 *Automating the Test Assessment Method*

Following the initial system security authorization, security control assessments are conducted on an ongoing basis to ensure that implemented security controls are effective and continue to be effective in the operational environment. The assessment method is based on the continuous monitoring strategy developed by the organization, information system owner, and/or common control provider and is approved by the authorizing official. Information on how to plan security control assessments is detailed in SP 800-53A, *Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans*.

Assessment methods define the nature of the assessor actions and include Examine, Interview, and Test. Table 1: SP 800-53A Assessment Methods, provides the definition of each assessment method. The organization uses the results of each assessment method to support the determination of security control existence, functionality, correctness, completeness,[6] and potential for improvement over time.

**Table 1: SP 800-53A Assessment Methods**

| Method | Definition[a] |
|---|---|
| Examine | The process of checking, inspecting, reviewing, observing, studying, or analyzing one or more assessment objects to facilitate understanding, achieve clarification, or obtain evidence. |
| Interview | The process of conducting discussions with individuals or groups within an organization to facilitate understanding, achieve clarification, or lead to the location of evidence. |
| Test | The process of exercising one or more assessment objects under specified conditions to compare actual with expected behavior. |

[a] SP 800-53A, Appendix D

> The TEST assessment method is usually the easiest and most effective to automate and, when automated, provides more accurate results.

A technical implementation of an ISCM program, like Continuous Diagnostics and Mitigation (CDM), uses the Test assessment method wherever it is applicable. Assessment via Test is generally the easiest and most effective assessment method to automate. Moreover, use of the automated Test method may provide more accurate and repeatable results when constructed and implemented correctly. Thus, it may be appropriate to employ the Test assessment method as the sole assessment method for many controls. It is more difficult to automate the Examine and Interview assessment methods as they require people. However, organizations might employ the Examine or Interview methods for root cause analysis of control failures (discussed in Section 7.2, Root Cause Analysis) or if greater assurance, depth, or coverage is needed.

[6] See glossary for definition of *assessment completeness*.

### 2.2.1 Terms for Referring to Assessment Objects

This document generally uses the term *object*, (short for *assessment object* in this publication). The meaning of (assessment) object as used herein is equivalent to the glossary definition but is focused on what could potentially have a security defect. Thus, when used in this NISTIR, the term *object* refers to the following:

> Anything that can have a material security defect (i.e., failed or absent control). Examples include devices, software, people, credentials, accounts, privileges, and things to which privileges are granted (including data and physical facilities).

Although the NIST definition of *assessment object* and this NISTIR definition of *object* use different words, they are synonymous in meaning. This is because anything that may have a defect (as defined in this NISTIR) is also something that needs to be assessed to determine whether the defect is present (NIST definition).

*Object* is a general concept and used where generality is implied. However, in the context of a specific capability (or group of capabilities), it may be clearer to use a more specific term. Many capabilities focus on objects with defects. Hardware Asset Management (HWAM) and Software Asset Management (SWAM) are examples of capabilities with such a focus. In referring to these kinds of objects, the term *asset* may be used (e.g., *assets* with defects).

Most specific capabilities focus on specific object types. Hardware asset management focuses exclusively on defects in and around devices, for example. Because this volume often uses examples from hardware asset management, it often uses the term *devices* when referring to defects in that context.

For the purposes of this NISTIR, all *hardware* assets (objects) are devices, but not all devices are objects. For example, a chip on a circuit board is technically a device and an asset, but in the hardware asset management context, it is not at an abstraction level of focus. Likewise, automated security control assessment does not focus on a device's keyboard, mouse, and monitor, per se, as they are just part of the larger device (or object) being assessed. However, property systems might count them as separate assets.

## 2.3 *Factors for Determining When to Trust Automated Ongoing Assessments*

Automating the appropriate assessment method should be used for assessing security controls at the point that automated security control assessment functionality has an equal or higher probability of detecting defects compared to traditional methods in use. The two factors that contribute most to defect detection are:

- The completeness of automated security control assessment; and
- The timeliness of automated security control assessment.

Completeness means that the automated security control assessment is conducted for *all* defect checks[7] and on all objects that could have the defect. Although 100 percent completeness might not be attained, as automated security control assessment approaches 100 percent completeness, the probability of missing defects approaches zero.

Timeliness means that each cycle of tests on the defect-object combinations assessed occurs at least as often as the frequency specified in the ISCM strategy. Initially, the specified frequency may merely be faster or more frequent than in the past. However, as the automated security control assessment functionality matures, the frequency should be often enough that the automated security control assessment system finds (and allows time for a response to) a high percentage of defects *before* an adversary can exploit them.

Consequently, as part of the risk management process and ISCM strategy, the organization determines the degree of completeness and timeliness required *before* it replaces manual/procedural assessments with an automated security control assessment system. The ISCM dashboard (discussed in the following section) provides maturity metrics to help assess this readiness.

> Automated security control assessment is adequate to replace manual/procedural security control assessment as soon as it is:
>
> - At least as timely; and
> - At least as complete
>
> as the manual/procedural assessments for the capabilities being covered (and their related security controls).

## 2.4 *An Automated Security Control Assessment Program: ISCM*

Figure 1: Overview of an Automated Security Control Assessment Process, provides a functional diagram of an automated security assessment process. This diagram represents the major steps for implementing an ISCM automated security control assessment process. As described in Section 1.3, Organization of Volume 1, the sections of this document are organized to explain each part of the diagram.

[7] "All defects" is limited to those defect checks (see Section 5.3) that are necessary to test the selected controls.

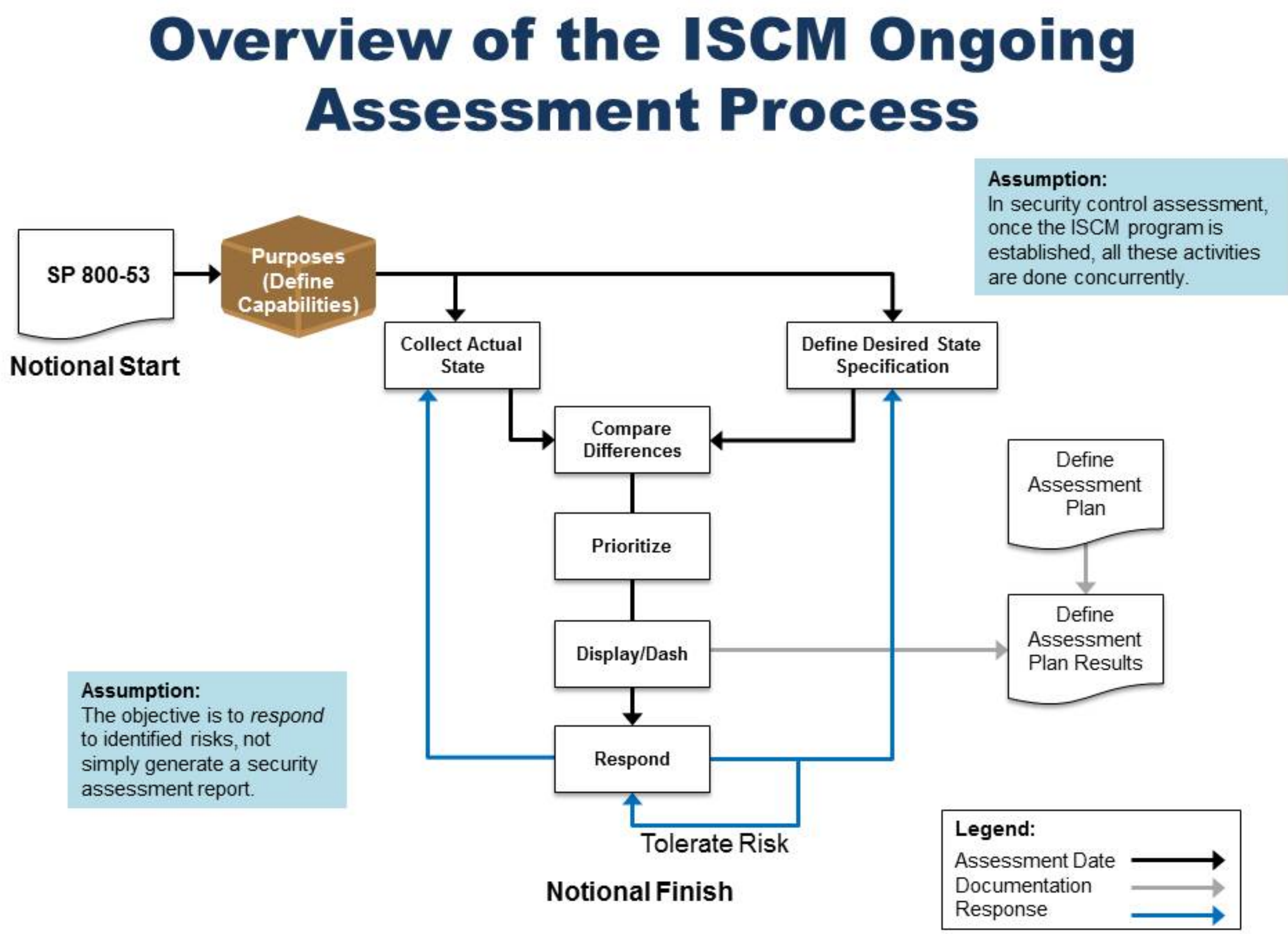


**Figure 1: Overview of an Automated Security Control Assessment Process**

## 2.5 *Preparing for Automated Security Control Assessments*

To effectively automate security control assessments, the following prerequisites must be met:

- Machine-readable actual state or behavior specifications are defined in data;
- Data-based (also machine-readable) desired/expected state/behavior specifications (readily comparable to the actual state) are defined;
- A method to compute/identify defects (differences between desired and actual state/behavior) is defined; and
- A method for producing a human-readable security assessment report to facilitate analysis and risk-based decision making.

When these prerequisites are met, the automated security control assessment system (as part of the ISCM dashboard) can automatically compute the following:

- Where differences between the desired state specification and the actual state (defects) occur;
- The priority of each defect; and
- Assignment of the defects to the appropriate operational team for response.

While specific guidance on risk scoring and risk response is out of scope for this NISTIR, it is still important to define the following in order to most effectively leverage the results/output of the automated security assessment:

- A method to assign a risk value/score (i.e., a priority) to each identified defect; and
- A method to determine operational responsibility to respond to identified defects.

# 3. Focusing Security Control Assessments on Security Results

This section introduces three abstraction layers that focus on achieving security results (as security capabilities) above the level of individual security controls/control items (see Section 3.5).

The following security capability abstraction layers are introduced and are traceable[8] to security requirements and the individual security controls/control items that support them:

(1) Attack Step Layer – Desired Result: Block or delay attacks (see Section 3.2).

(2) Functional Capability Layer – Each capability is a grouping of controls and control items from SP 800-53. Desired Result: Make a broad area of the system more secure (see Section 3.3).

(3) Sub-Capability Layer – Each capability is decomposed into sub-capabilities necessary and sufficient to support the purpose of the larger capability. Each sub-capability is tested with one corresponding defect check. Desired Results: a) To support blocking of attack steps and provide functional security capability; and b) provide clearly testable outcomes (see Section 3.4).

The control items themselves provide a fourth abstraction layer:

(4) Control Items Layer – From SP 800-53 (see Section 3.5).

The four abstraction layers serve the following purposes:

- Support strong systems engineering of security capabilities;
- Support guidance for control selection;
- Simplify understanding of the overall protection process;
- Enable assessment of security results at a higher level than individual controls; and
- Improve risk management by measuring security results that are more closely aligned with desired business results.

To address these purposes, NIST introduced the concept of information security capabilities in SP 800-53 Revision 4, p. 21. *Information security capabilities* (discussed in more detail below) are groups of controls that work together to achieve an information security purpose and enable/protect the organization's ability to perform its mission.

The abstraction layers have been induced from the NIST controls and deduced from what is needed to reduce successful attacks. The intent of documenting these abstraction layers is to show how SP 800-53 controls can work together to achieve important information security outcomes or results, and in turn support security-focused systems engineering.

[8] Traceability of requirements is discussed extensively in NIST SP 800-160.

## 3.1 *Applying Security Capabilities to Automated Assessments*

Presenting security capabilities as abstraction layers above the security control level provides several benefits.

### 3.1.1 Supports Strong Systems Engineering of Security Capabilities

In normal systems engineering, the engineering process begins with general business requirements at a fairly high level of abstraction. More detailed technical requirements are then derived from the business requirements. Information security engineering has generally not been done in this manner. Rather, predefined control sets have been applied to provide detailed technical requirements without documenting traceability of control items to more general requirements.[9]

An unintended and undesirable consequence of this has been that many security programs have focused on the individual controls as a compliance checklist, with little consideration given to how the controls work together to protect the confidentiality, integrity, and availability of information and systems.

The four abstraction layers support integrated systems engineering by making the desired results of a security program clear and measurable at a concrete level. This, in turn, makes those results more understandable to non-security experts and thereby easier to link to desired business/mission results.

Awareness of the results to be produced facilitates better security engineering, by allowing security control designers to look at controls as parts of a system designed to achieve an overall purpose, allowing them to better control design and planning decisions.

### 3.1.2 Supports Guidance for Control Selection

Informed and judicious decision making in security control selection requires an understanding of how security controls work together to achieve broader security protections. Recognizing and documenting how groups of controls work together to block attack steps and support broad security functions facilitates selection of a set of complementary controls that work together to achieve the desired results (i.e., security protections commensurate with risk). As noted in SP 800-53, p. 21:

> *The concept of security capability is a construct that recognizes that the protection of information being processed, stored, or transmitted by information systems, seldom derives from a single safeguard or countermeasure (i.e., security control). In most cases, such protection results from the selection and implementation of a set of mutually reinforcing security controls.*

[9] NIST has published guidance on systems engineering of information security for mission assurance (NIST SP 800-160).

### 3.1.3 Simplifies Understanding of the Overall Protection Process

It is a difficult task to achieve detailed understanding of hundreds of individual security control items. As noted in SP 800-53 (*op. cit.*):

> *[Defining security capabilities] simplifies how the protection problem is viewed conceptually. In essence, using the construct of security capability provides a shorthand method of grouping security controls that are employed for a common purpose or to achieve a common objective.*

Grouping the controls into those that support attack steps, capabilities, and sub-capabilities facilitates better comprehension of information security requirements and implementations. The grouping of security controls into capabilities increases awareness of the results that security controls are expected to produce.

### 3.1.4 Enables Assessment of Security Results at a Higher Level than Individual Controls

Selecting the most appropriate level of abstraction to assess the effectiveness of security control implementations involves trade-offs. If the assessment is at a too-detailed level, one might find that all the parts work, while missing the fact that the sum-of-the-parts does not work. On the other hand, if results are assessed at a higher level of abstraction, and a control failure is detected at that level, then root cause analysis is needed to determine which part is not working. Also, as noted in SP 800-53A (p. 27):

> *This becomes an important consideration, for example, when assessing security controls for effectiveness. Traditionally, assessments have been conducted on a control-by-control basis producing results that are characterized as pass (i.e., control satisfied) or fail (i.e., control not satisfied). However, the failure of a single control or in some cases, the failure of multiple controls, may not affect the overall security capability needed by an organization. This is not to say that such controls are not contributing to the security or privacy of the system and/or organization (as defined by the security requirements and privacy requirements during the initiation phase of the system development life cycle), but rather that such controls may not be supporting the particular security capability or privacy capability. Furthermore, every implemented security control or privacy control may not necessarily support or need to support an organization-defined capability.*

The sub-capability layer of abstraction is the most appropriate level on which to focus automated assessments. The sub-capability layer is closer to results and is easier to automate. That is why defect checks are designed to test effectiveness at the sub-capability layer. When failures are found, root cause analysis can be used to find the specific security control(s)/control items causing the failure. (See Section 7.2, Root Cause Analysis.)

### 3.1.5 Improves Risk Management by Measuring Security Results More Closely Aligned with Desired Business Results

NIST guidance on information security risk stresses the need to focus not just on system-level risk, but also on mission-level risks. (SP 800-30, Figure 4, p. 17 and Section 2.4.1, p. 18f and SP 800-39, Figure 2, p. 10ff).

In SPs 800-37, 800-53, and 800-115, there is an increased focus on assessing results in addition to control effectiveness. Further, SP 800-39, Section 2.2 recommends "a three-tiered approach to risk management that addresses risk-related concerns at: (i) the organization level; (ii) the mission/business process level; and (iii) the information system level." Security controls largely exist at the system level, and business and security results (outcomes, consequences, etc.) are most visible at the organization and mission/business process level. As noted in SP 800-53 (*op. cit.*):

> *Ultimately, authorization decisions (i.e., risk acceptance decisions) are made based on the degree to which the desired security capabilities have been effectively achieved and are meeting the security requirements defined by an organization. These risk-based decisions are directly related to organizational risk tolerance that is defined as part of an organization's risk management strategy.*

The value of these abstraction layers is that because they are more closely aligned to the business mission of any organization, they make it easier for analysts in a specific organization to trace requirements to mission. However, the abstraction layers in this document cannot extend all the way to a specific organization's mission (because this document is written to be adaptable to any organization). Mission-specific layers would need to be added by each organization, based on the contributions of the information systems being managed to support the organization's specific mission. The attack step and security capability abstraction layers are provided to make it easier to trace security controls to the organization's mission.

The following sections describe how the SP 800-53 concept of a security capability can be used to group security controls by the security results they are designed to achieve. With appropriate metrics, this allows risk managers to make better risk management decisions by assessing the extent to which the higher-level objectives are being met.

## 3.2 *Attack Steps*

Ultimately, information security is about blocking or reducing damage to confidentiality, integrity, and availability of information and information systems.

Such damage is caused by one or more of the following threat sources (SP 800-30):

- Hostile cyber or physical attacks;
- Human errors of omission or commission;
- Structural failures of organization-controlled resources (e.g., hardware, software, environmental controls); and

- Natural and man-made disasters, accidents, and failures beyond the control of the organization.

### 3.2.1 Adversarial Attack Step Model

Various attack models have been developed to describe how adversarial (hostile) attacks occur. Attack step models are articulated from the adversarial viewpoint of a malicious attacker with intent to do damage. While non-adversarial events (i.e., events that occur without malicious intent such as natural disasters, hardware failures, etc.) are not addressed by attack step models, organizations remain responsible for considering such events and implementing mitigating security capabilities/controls in order to achieve holistic risk management and as part of a comprehensive information security program.

**Attack Steps and Security Capabilities:** Because specific security controls needed to block or delay attack steps can be mapped, the attack steps correspond to security capabilities designed to block or delay the attacker at each step. The attack step model depicted in Figure 2: Attack Step Model, consists of six steps which are each addressed by specific security capabilities and sub-capabilities identified in this NISTIR. Note also that the attack steps described here are not a definitive set of such steps. They in no way limit the flexibility of organizations to define different or additional attack steps and associated security capabilities for their own situations.

**Defense in Depth:** From the perspective of attack steps, the concept of defense in depth means (in part) that controls are in place at all steps so that if one stage is breached, there are controls at the next step to further protect the system. Examples and/or descriptions of the six attack steps and potential mitigating security controls are provided in Table 2: Descriptions of the Attack Steps.

| Attack Steps |
| --- |
| 1) Gain Internal Entry |
| 2) Initiate Attack Internally |
| 3) Gain Foothold |
| 4) Gain Persistence |
| 5) Expand Control - Escalate or Propagate |
| 6) Achieve Attack Objective |

**Figure 2: Attack Step Model**

**Table 2: Descriptions of the Attack Steps**

| Attack Step | Attacker Action | Defender Action |
|---|---|---|
| 1) Gain Internal Entry | The attacker is outside the target boundaries and seeks entry.<br>Examples include: spear phishing email sent; DDoS attack against .gov initiated; unauthorized person attempts to gain physical access to restricted facility. | 1) Limit attacks or negative events from even initiating in, or having the ability to impact, the local environment.<br>Examples include: multifactor authentication; SPAM filters; access control lists for routers/firewalls; physical protections like locks or guards; link encryption and VPNs; authoritative DNS to prevent poisoning; gateway-level anti-malware applications.<br><br>2) Detect entry; respond and recover.<br>Examples include: network-based intrusion detection systems; surveillance equipment for physical site that identifies attempts at unauthorized physical access to facility. |
| 2) Initiate Attack Internally | The attacker is inside the boundary and initiates attack on some object internally.<br>Examples include: User opens spear phishing email or clicks on attachment; laptop lost or stolen; user installs unauthorized software or hardware; unauthorized personnel gains physical access to restricted facility. | 1) Limit initiating condition from occurring in local environment.<br>Examples include: educating users not to click on attachments; maintaining positive control of assets; restricting privileges for software installation or removable media.<br><br>2) Limit precipitating event from resulting in attack.<br>Examples include: preventing automatic execution of code on removable media; educating users not to share passwords; educating users not to send unencrypted PII outside of the enterprise; host-level anti-malware applications that blocks before execution.<br><br>3) Detect Entry; respond and recover.<br>Examples include: host-based intrusion detection systems; surveillance equipment for physical site that identifies unauthorized physical access to facility. |
| 3) Gain Foothold | The attacker has gained entry to the object and achieves enough actual compromise to gain a foothold, but without persistence.<br>Examples include: Unauthorized user successfully logs in with authorized credentials; browser exploit code successfully executed in memory and call back initiated; person gains unauthorized access to server room. | 1) Limit vulnerable conditions that attack/threat exploits.<br>Examples include: patching; implementation of common secure configurations.<br><br>2) Limit successful completion of exploitation attempt.<br>Examples include: DEP (data execution prevention); recompiling techniques; removing default passwords and accounts; multifactor authentication; disabling accounts; redundant communication paths; restricting physical access to critical resources.<br><br>3) Limit successful foothold on object.<br><br>Examples include: Detect attempts; Blocking access attempts to known bad DNS domains; reviewing audit and event logs.<br><br>4) Detect Foothold; respond and recover.<br>Examples include: Host-based intrusion detection systems; behavioral analysis; surveillance equipment for physical site that identifies unauthorized physical access attempts to internal locations or assets. |

| Attack Step | Attacker Action | Defender Action |
|---|---|---|
| 4) Gain Persistence | The attack has gained a foothold on the object and now achieves persistence. Examples include: Malware installed on host that survives reboot or log off; BIOS or kernel modified; new/privileged account created for unauthorized user; unauthorized person issued credentials/allowed access; unauthorized personnel added to Access Control List (ACL) for server room. | 1) Limit persistent compromise of asset. Examples include: Application whitelisting; malware/intrusion prevention tools; virtualization and sandboxing; one-time password systems; requiring hardware tokens for authentication; restricting physical access with card readers.<br><br>2) Detect persistence; respond and recover. Examples include: File reputation services; file integrity checking; blocking known malicious command and control channels; reviewing audit and event logs; advanced behavioral analysis techniques; surveillance equipment for physical site that identifies successful unauthorized physical access to internal locations or assets. |
| 5) Expand Control - Escalate or Propagate | The attacker has persistence on the object and seeks to expand control by escalation of privileges on the object or propagation to another object. Examples include: Administrator privileges hijacked or stolen; administrator's password used by unauthorized party; secure configuration is changed and/or audit function is disabled; authorized users access resources they do not need to perform job; process or program that runs as root is compromised or hijacked. | 1) Limit escalation of privileges or access propagation to other assets. Examples include: Restricting privileges for accounts, programs, and processes; implementing and following configuration change control processes; using hardware tokens or two-factor authentication for privileged actions; restricting physical access to server rooms.<br><br>2) Detect escalation or propagation activity; respond and recover. Examples include: Use of Intrusion Detection System (IDS) tools; reviewing audit and event logs. |
| 6) Achieve Attack Objective | The attacker achieves an objective. Loss of confidentiality, integrity, or availability of data or system capability. Examples include: Exfiltration of files; modification of database entries; deletion of file or application; denial of service; disclosure of PII. | 1) Minimize impact from successful attack Examples include: Use of data loss prevention tools; laptop and media encryption; outbound boundary filtering; educating users to protect critical information; restricting access to critical information and resources; file and transaction (e.g., email) encryption; link encryption/VPNs.<br><br>2) Detect impact from successful attack; respond and recover. Example methods include: Use of auditing and insider threat tools; network event and analysis tools. |

Note on Table 2: The defender actions (i.e., security controls) are largely covered by the SP 800-53 low baseline, and should generally be in place. This table simply helps make a connection between security controls and the example attack steps. The intent is not to suggest new controls.

## 3.3 *Security Capabilities*

Noting that controls work together to achieve results, NIST defined a security capability as:

> A security capability is a set of mutually reinforcing security controls implemented by technical, physical, and procedural means. Such capabilities are typically defined to achieve a common information security-related purpose.

### 3.3.1 SP 800-53 Control Families and Security Capabilities

SP 800-53 notes in an example that the controls necessary to support a given capability might come from more than one family. It is frequently the case that a single control supports multiple security capabilities. Security control families are not intended to be security capabilities, but rather are general categories used to logically group individual security controls within the control catalogue.

> Security control families were developed with each control only one family. A single control, however, can support multiple capabilities. This makes security control families unsuitable as security capabilities.

### 3.3.2 SP 800-137 Security Automation Domains and Security Capabilities

Appendix D of SP 800-137 defines a set of security automation domains[10] as "information security area[s] that includes a grouping of tools, technologies, and data."[11] These domains are *not* analogous to security capabilities because they are not a collection of controls with a common purpose.

### 3.3.3 Using Security Capabilities in Security Control Assessment

While the term *security capability* is defined in SP 800-53, no specific capabilities are identified, allowing organizations to define security capabilities according to security goals. The next section defines the security capabilities used here as ISCM capabilities. The ISCM capabilities describe the purposes of all SP 800-53 security controls that are selected in the low- through high-impact baseline.[12]

### 3.3.4 Security Capabilities

SP 800-53 also states that:

> *As the number of security controls in Appendix F grows over time in response to an increasingly sophisticated threat space, it is important for organizations to have the*

[10] SP 800-137, D-4, Figure D-1.
[11] *Ibid*, B-12.
[12] SP 800-53, Appendix D, Table D-2.

*ability to describe key security capabilities needed to protect core organizational missions/business functions, and to subsequently define a set of security controls that if properly designed, developed, and implemented, produce such capabilities.*[13]

To facilitate the implementation of automated security control assessments, an ISCM program defines specific security capabilities to guide and focus implementation. Each capability has a clearly defined result, which allows assessment activities to better inform risk analysis and response.

#### *3.3.4.1 ISCM Security Capabilities*

An ISCM security capability consists of the SP 800-53 security controls needed to achieve the purpose of each capability. A capability has the following additional traits:

- The *purpose* (desired result) of each capability is to address specific kinds of attack scenarios or exploits;
- Each capability focuses on attacks toward specific *objects*; and
- There is a viable way to automate many of the assessments of the security controls that comprise the security capability.

The complete set of security capabilities provides protection against current and relevant attack scenarios/exploits and thus includes all SP 800-53 baseline controls (i.e., all the controls selected in the SP 800-53 high-impact baseline) in at least one capability.

Note that when organizations implement controls not selected in the NIST SP 800-53 high-impact baseline (i.e., tailoring– supplementation), it is important that those additional controls are also assessed at the appropriate frequency (as determined by the organization's ISCM strategy). Supplemental controls may be added to an existing capability if appropriate, or new capabilities may be created as needed.

As significantly different attack scenarios/exploits emerge, it may be necessary to augment the set of security capabilities. Further, if an objective of a sound security program is found not to be included already, the intention is to add it, once discovered.

### 3.3.5 Example Security Capabilities Listed and Defined

This NISTIR identifies a set of security capabilities designed to achieve complete coverage of SP 800-53 controls and to effectively display interaction among the various security capabilities. Figure 3: ISCM Security Capabilities Used in this NISTIR, shows the view of security capabilities used in this document. The narratives in Table 3: ISCM Security Capabilities, describe each capability in Figure 3. Since the DHS CDM program also defines security capabilities (shown on their website), differences between the two capability sets are noted in footnotes to Table 3.

[13] SP 800-53 Rev. 4, p. 21.

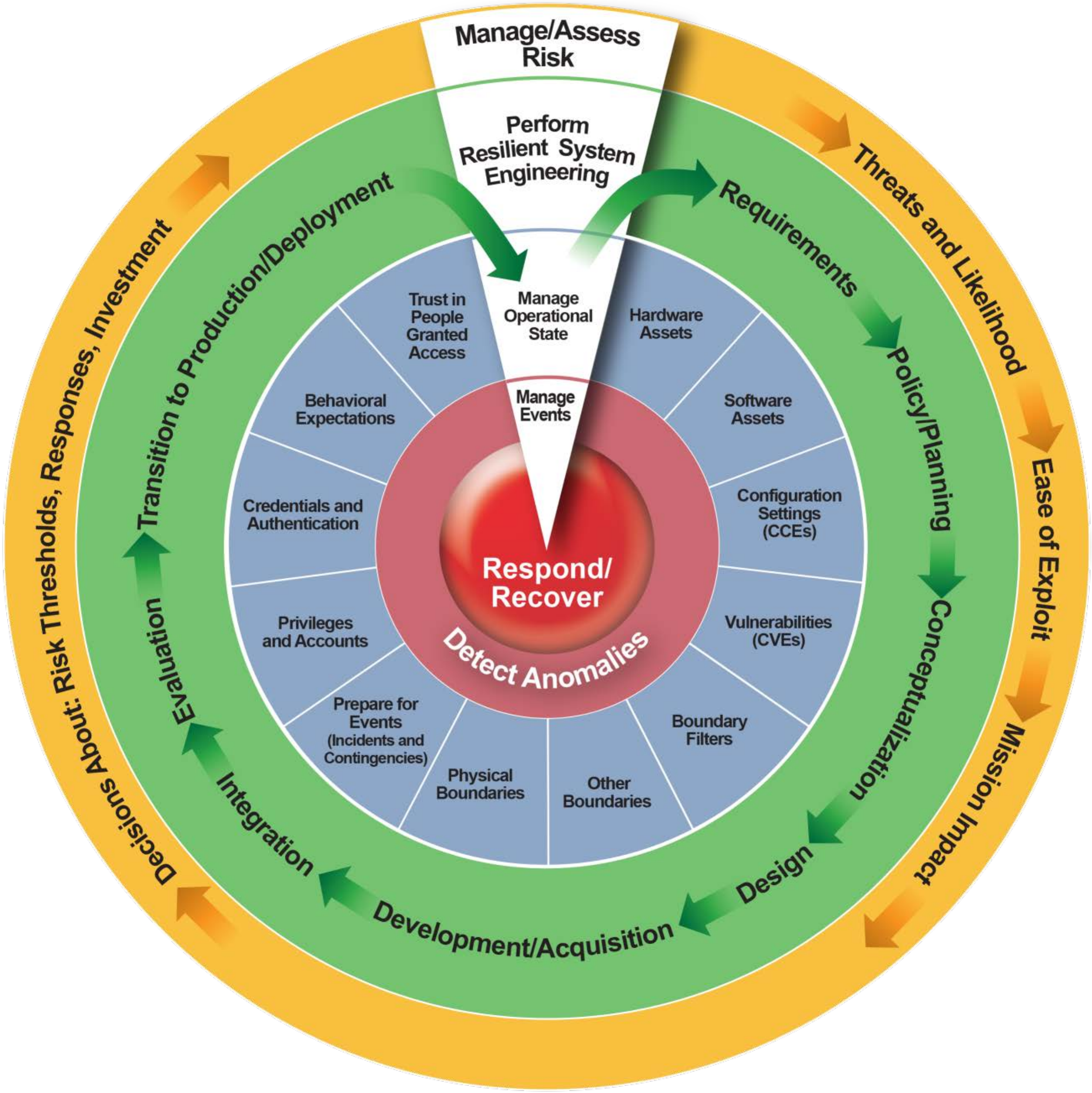


**Figure 3: ISCM Security Capabilities Used in this NISTIR**

## Table 3: ISCM Security Capabilities

| Ring 1: Manage and Assess Risk (Orange ring plus wedge touching all other rings in Figure 3) | | |
|---|---|---|
| Risk management (and assessment) is the overall purpose of ISCM and is informed by and applied to all inner rings, i.e., all other ISCM capabilities. | | |
| **Security Capability Name** | **Purpose (Desired Result)** | **Considerations** |
| Manage and Assess Risk | To reduce successful exploits that occur in other capabilities because the risk management process failed to correctly identify and prioritize actions and investments needed to lower the risk profile. | ISCM dashboards ideally provide scoring and maturity metrics for each capability to prioritize risk response not only at the operational (system administration) and tactical (ISSO) levels, but also at the strategic (CISO, CIO, CEO) level. |

| Ring 2: Perform Resilient Systems Engineering[a] (Green ring in Figure 3) | | |
|---|---|---|
| Resilient Systems Engineering is focused on applying the overall systems engineering process to design resilience into information systems.<br>System Engineering is applied to all the inner rings of the wheel. It is informed by risk management and assessment and by lessons learned from ISCM of the inner rings on the wheel.<br>Systems engineering steps may be tailored in a number of ways and may be done in an agile or spiral manner. The words in Figure 2 are illustrative, not normative. For more guidance on resilient systems engineering and effective steps, see SP 800-160.<br>The systems engineering outputs should be initially assessed outside of ISCM before they go into operations. Therefore, this NISTIR does not provide guidance for the automated assessment of the systems engineering phases (per se), apart from what might be adapted from the operational tests of other capabilities. | | |
| **Security Capability Name** | **Purpose (Desired Result)** | **Considerations** |
| Perform Resilient Systems Engineering | To reduce successful exploits that occur in the blue and red ring capabilities because there was inadequate definition of requirements, policy, planning, and/or other management issues in designing, implementing, and/or monitoring the controls in those capabilities. | Requirements and policy are documented in the desired state specification for each of the other capabilities. If exploits are repeatedly successful, additional controls may be introduced to block those exploits through more comprehensive requirements, policy, and planning.<br><br>Monitoring the controls that comprise the blue and red ring capabilities reveals when exploits are successful. Root cause analysis may determine that the exploit(s) resulted from defects in the pre-operational design stages of the lifecycle. |

| Ring 3: Manage the Operational State (Blue ring in Figure 3) | | |
|---|---|---|
| These capabilities can be assessed by automated means and provide the primary security protections to information and information systems during the operations and maintenance phase of the SDLC. They also serve the role of identifying systemic problems in operations that might be fixed with improved engineering. | | |
| **Security Capability Name** | **Purpose (Desired Result)** | **Considerations** |
| Hardware Asset Management | Ensure that unauthorized and unmanaged devices are identified to prevent use by attackers as a platform from which to extend compromise of information systems. | Maintain a list of authorized hardware and who manages it. Treat other hardware discovered within the assessment boundary as a defect. |
| Software Asset Management | Ensure that unauthorized software is identified to prevent use by attackers as a platform from which to extend compromise of information systems. | Maintain a list of authorized software at both the product and executable level. Treat other software discovered within the assessment boundary as a defect. |
| Configuration Settings Management | Ensure that common secure configurations (Common Configuration Enumerations: CCEs) are established and applied to prevent attackers from compromising an information system or device which in turn may be used as a platform to compromise other information systems or devices. | Maintain a list of authorized settings. Treat deviations discovered within the assessment boundary as defects. |
| Vulnerability (Patch) Management (CVEs) | Ensure that software and firmware vulnerabilities (Common Vulnerabilities and Exposures: CVEs) are identified and patched to prevent attackers from compromising an information system or device which in turn may be used to compromise other information systems or devices. | The National Vulnerability Database (NVD) provides a library of vulnerabilities mapped to vulnerable software. Responses may include applying patches, installing more secure versions, or accepting the risk. Common Weakness Enumeration (CWE) scanning tools may identify poor coding practices that are directly associated with conditions that often manifest themselves as vulnerabilities that are discovered and assigned a CVE. |
| Manage Trust for Persons Granted Access | Ensure that unauthorized/uncleared persons are not entrusted with information system access. | Track completion of personnel screening processes (such as clearances, background checks, suitability reviews, etc.) designed to identify evidence of untrustworthiness. |
| Manage Behavioral Expectations | Ensure that authorized users are aware of expected security-related behavior and understand how to avoid and/or prevent purposeful and inadvertent behavior that may compromise information in the course of performing their duties. | Track evidence (such as Training, Rules of Behavior/Access and Use Agreements, Courseware and Skill Certifications, etc.) designed to specify and enable secure behavior. |
| Manage Credentials and Authentication | Ensure that authorized users have the credentials and authentication methods necessary to perform their duties, while limiting access to only that which is necessary. | Establish the needed credentials and authentication methods from assigned user roles and verify that no extra credentials/methods are provided. |
| Manage Privileges and Accounts | Ensure that authorized users have the privileges necessary to perform their duties/limit access to only that which is necessary. | Establish the needed privileges from assigned user roles and verify that no extra privileges are provided. |
| Manage Physical Boundaries[b] | Ensure that movement (of people, media, equipment, etc.) into and out of the physical facility does not compromise security. | Restrict and monitor physical access using automated tools and collectors to help track and control movements. |

| | | |
|---|---|---|
| Manage Network Boundaries[b] | Ensure that traffic into and out of the network (and thus out of the physical facility protection) does not compromise security. Do the same for enclaves that subdivide the network. | Configure secure information flow and other traffic-related boundary protections to monitor and control internal and external network boundaries. |
| Manage Other Boundaries[b] | Ensure that the confidentiality and integrity of information is protected in transit and at rest. This is especially important when information is exposed (as in an Internet or wireless link) or residing on equipment that will be outside a secure space (as in a laptop or mobile device). Encryption is the most commonly used technique. | Ensure that boundary controls not related to physical and network boundaries (e.g., encryption of VPN traffic, RF Spectrum management) are secure to protect data in motion and at rest. |
| Manage Preparation for Events (Incidents and Contingencies) | Ensure that procedures and resources are in place to respond to both routine and unexpected events that can compromise security.<br>• Potential responses include a wide range of possible actions, including, but not limited to, continuity of operations, recovery, and forensics.<br>• Unexpected events include actual attacks and natural disasters like floods, earthquakes, tsunamis, etc. | Identify the desired preparations (e.g., extra capacity, backups, uninterruptible power supplies, generators, hot site, redundant site, etc.) and verify that they are present (and ideally performing). |

| Ring 4: Manage Anomalous Events (Red ring in Figure 3) | | |
|---|---|---|
| Notwithstanding best efforts in implementing the surrounding rings for risk management and assessment, resilient systems engineering, and operational state management, it is still likely that some successful attacks and some damaging contingencies will adversely affect the information system. These capabilities are designed to detect and inform a response to such events.<br>The detection and response activities need to relate to each of the sections of the blue ring. That is, anomalous events could appear in any of the blue ring capabilities. In fact, most attacks or contingencies will touch multiple capabilities related to operational state and/or behavior of the objects covered by those capabilities. | | |
| **Security Capability Name** | **Purpose (Desired Result)** | **Considerations** |
| Manage Anomalous Event Detection | Ensure that routine and unexpected events that compromise security can be identified within a specified time frame such that impact is minimized to the greatest extent possible. | Use various methods to correlate audit records, system events, IDPS logs, etc., and track patterns to identify unexpected patterns or indicators of harmful activity. Set desired thresholds for impact (e.g., servers are never down more than 24 hours) and detect when thresholds are not met. |
| Manage Anomalous Event Response and Recovery | Ensure that routine and unexpected events that require a response to maintain functionality and security are responded to (once identified) within a specified time frame such that impact is minimized to the greatest extent possible. | Implement desired response procedures and verify that they are performing. |

[a] The DHS CDM program identifies some capabilities slightly differently than this NISTIR as follows: a) design and build in requirements, policy, and planning; b) design and build in quality; c) manage audit information; and d) manage operation security. This NISTIR includes a) and b) in systems engineering (green ring), c) in manage events – detect anomalies (red ring), and d) in manage the operational state (as part of the overall blue ring).
[b] The three boundary capabilities (Physical, Filters, Other) listed here are considered a single capability in the CDM program. They have been separated based on more detailed assessment of the corresponding controls.
.

### 3.3.6 Tracing Requirements: Mapping Capability to Attack Steps

Each capability-specific volume of this NISTIR includes a more detailed description of how the capability maps to attack steps described in Section 3.2, Attack Steps. For example, the HWAM volume includes Table 4: Tracing the HWAM Capability to Blocking Attack Steps, that shows how the purpose of the capability blocks or delays the attack.

**Table 4: Tracing the HWAM Capability to Blocking Attack Steps**

| Attack Step Name | Attack Step Purpose | Examples |
|---|---|---|
| 2) Initiate Attack Internally | The attacker is inside the boundary and initiates attack on some object internally.<br>Examples include: User opens spear phishing email or clicks on attachment; hurricane hits site; laptop lost or stolen; user installs unauthorized software or hardware; unauthorized personnel gains physical access to restricted facility. | Block Internal Access: Prevent or minimize unauthorized/compromised devices from being installed and/or staying deployed on the network.<br>Reduce amount of time unauthorized devices are present before detection. |

| Attack Step Name | Attack Step Purpose | Examples |
|---|---|---|
| 3) Gain Foothold | The attacker has gained entry to the object and achieves enough actual compromise to gain a foothold, but without persistence.<br>Examples include: Unauthorized user successfully logs in with authorized credentials; browser exploit code successfully executed in memory and initiates call back; person gains unauthorized access to server room; backup server fails due to overload during storm after primary server fails. | Block Foothold: Reduce number of unauthorized and/or easy-to-compromise devices that aren't being actively administered. |
| 6) Achieve Attack Objective | The attacker achieves an objective. Loss of confidentiality, integrity, or availability of data or system capability.<br>Examples include: Exfiltration of files; modification of database entries; deletion of file or application; denial of service; disclosure of PII. | Block Physical Exfiltration: Prevent or minimize copying information to unauthorized devices. |

### 3.3.7 Organization-Defined Security Capabilities

The security capabilities identified herein are not a definitive set of security capabilities. The defined capabilities in no way limit the flexibility of organizations to define different or additional security capabilities.

Organizations may define new security capabilities, additional capabilities, or revise the functional security capabilities, and then execute the general automated security control assessment paradigm defined in this NISTIR at the organizational level. Note, though, that this would require development of a comprehensive organization-specific automated security control assessment approach and a plan to address the organization-specific capabilities. Organizations are encouraged to automate their security control assessment approach using the functional security capabilities initially, in order to gain experience, and then decide at a later point whether customization is necessary.

## 3.4 *Sub-Capabilities*

Capabilities are composed of sub-capabilities.

A key feature of the sub-capabilities defined here is that they were designed to be testable by automated means. For each sub-capability, this NISTIR defines one *defect check*, which is used to assess whether the purpose of that sub-capability is being met, which in turn contributes to an overall determination of security program effectiveness (control items, controls, sub-capabilities, and capabilities).[14]

For example, an HWAM capability related to removing high-risk hardware could have sub-capabilities related to:

[14] Finding defective control items may require root cause analysis as described in Section 8.2, Root Cause Analysis.

- Removing unauthorized hardware;
- Ensuring all hardware is managed; and
- Validating that the hardware supply chain is secure.

These sub-capabilities, for HWAM, support the broader purpose of removing high-risk hardware. The assumption is that unauthorized devices, unmanaged devices, and devices with unapproved supply chains are all higher risk.

In the capability-specific volumes of this NISTIR, sub-capabilities within each broader capability have been identified to illustrate the way control items in the capability work together to achieve the overall capability goal.

The security sub-capabilities identified herein are not a definitive set of security sub-capabilities. The defined sub-capabilities in no way limit the flexibility of organizations to define different or additional security sub-capabilities.

Because sub-capabilities are defined under each capability, each sub-capability belongs to exactly one (one and only one) capability. Note, though, that there are often *similar* sub-capabilities identified for different capabilities.

### 3.4.1 Examples of Sub-Capabilities (from HWAM)

As described in Table 4: Tracing the HWAM Capability to Blocking Attack Steps, HWAM provides a high-level ability to block or delay attack steps related to the exploitation of hardware devices. After mapping relevant security controls to this capability (see Tracing Security Control Items to Capabilities), sub-capabilities were derived to more fully demonstrate how the HWAM controls work together to achieve the purposes of HWAM (see Tracing Security Control Items to Sub-Capabilities). Similar analyses will be presented in each capability-specific volume of the NISTIR. Table 5: Selected Examples of Sub-Capabilities (HWAM), taken from the HWAM capability volume, lists example definitions of HWAM sub-capabilities.

**Table 5: Selected Examples of Sub-Capabilities (HWAM)**

| Sub-Capability Name | Defect Check ID | Sub-Capability Purpose |
|---|---|---|
| Prevent unauthorized devices. | HWAM-F01 | Prevent or reduce the presence of unauthorized devices thus reducing the number of potentially malicious or high risk devices. |
| Reduce number of devices without assigned device manager. | HWAM-F02 | Prevent or reduce the number of devices without an assigned device manager within the assessment boundary, thus reducing delay in mitigating device defects (when found). |
| Reduce exploitation of devices before removal, during use elsewhere, and after return. | HWAM-L01 | Prevent exploitation of devices before removal, during use elsewhere, and after return (or other mobile use) by a) appropriately hardening the device prior to removal; b) checking for organizational data before removal; and c) sanitizing the device before introduction or reintroduction into the assessment boundary. |

| Sub-Capability Name | Defect Check ID | Sub-Capability Purpose |
|---|---|---|
| Reduce insider threat of unauthorized device. | HWAM-L02 | Use separation of duties (i.e., requiring multiple persons to authorize adding a device to the authorization boundary) to limit the ability of a single careless or malicious insider to authorize high-risk devices.<br><br>Note 1: The organization might choose to use access restrictions to enforce the separation of duties. If so, that would be assessed under the PRIV capability. What is assessed here is that the separation of duties occurs.<br>Note 2: See HWAM-L11 for authorization boundary. |
| Reduce denial of service attacks from missing required devices. | HWAM-L03 | Prevent or reduce denial of service attacks and/or attacks on resilience by ensuring that all required devices are present in the assessment boundary. |
| Reduce unauthorized components. | HWAM-L06 | Detect and remove unauthorized sub-components and/or subcomponent types to implement least functionality in order to prevent or reduce the introduction of sub-component and sub-component types that could enable attacks. |
| Verify ongoing business need for device. | HWAM-L07 | Prevent Require periodic and/or event driven consideration of whether a device is still needed for information system functionality to fulfill mission requirements in support of least functionality).<br><br>Note: A good practice might be to require DMs to review what they manage and System Owners to review what is needed in their authorization boundaries. |
| Ensure required device data is collected. | HWAM-L08 | Ensure that data required to assess risk are collected. These data may relate to other than a HWAM defect but may need to be generated by the HWAM collector. For example, devices with inadequate memory to support basic OS and defensive security components may need to be detected as defects. |

### 3.4.2 Tracing Sub-Capabilities to Attack Steps

By tracing the sub-capabilities of a security capability to attack steps, it is clearer how the security capability addresses the attack step. For example, in each capability-specific volume, there will be a table with similar columns as Table 5 above, showing only the sub-capabilities that support blocking that attack step.

## 3.5 *Security Control Items*

In many cases, SP 800-53 security controls include multiple requirements—in the base controls and also in control enhancements. Some control requirements may support one ISCM or organization-defined capability, while other requirements contained in the same control may support a different capability or multiple capabilities.

Therefore, to isolate the requirements for automated security control assessment planning purposes, the concept of a *security control item* is used.

Control items are identified as follows:

(1) **Each base control** is a separate control item (apart from its enhancements). If the base control has sub-requirements designated in SP 800-53 by (a), (b), (c), etc., each sub-requirement is also a separate control item.

(2) **Each enhancement** is a separate control item (apart from other enhancements and base controls). As with the base control, if it has sub-requirements designated by (a), (b), (c), etc., then each sub-requirement is also a separate control item.

SP 800-53 security controls are divided into control items:

- So that each control requirement is individually testable; and
- To simplify defining security capabilities.

This aligns the control items more closely to the individual determination statements in SP 800-53A, the difference being that control items identified here are sometimes further subdivided in SP 800-53A.

### 3.5.1 Tracing Security Control Items to Attack Steps

Sub-capabilities are mapped to attack steps and control items. This makes it possible to produce a list of the control items that are mapped to attack steps (i.e., control items that support blocking or delaying an attack step). See the example in Table 6: Example of Tracing HWAM Security Control Items to Attack Steps, which covers just one attack step and HWAM control items associated with it. See Appendix B of each capability-specific volume of this NISTIR for a complete listing of security control items for that capability mapped to attack steps.

**Table 6: Example of Tracing HWAM Security Control Items to Attack Steps**

| Example Attack Stage | Sortable Control Item Code | NIST Control Item Code |
|---|---|---|
| 2) Initiate Attack Internally | AC-19-a | AC-19(a) |
| 2) Initiate Attack Internally | AC-19-b | AC-19(b) |
| 2) Initiate Attack Internally | AC-19-z-05-z | AC-19(5) |
| 2) Initiate Attack Internally | AC-20-z-02-z | AC-20(2) |
| 2) Initiate Attack Internally | CM-02-z-07-a | CM-2(7)(a) |
| 2) Initiate Attack Internally | CM-02-z-07-b | CM-2(7)(b) |
| 2) Initiate Attack Internally | CM-03-b | CM-3(b) |
| 2) Initiate Attack Internally | CM-03-c | CM-3(c) |
| 2) Initiate Attack Internally | CM-03-d | CM-3(d) |
| 2) Initiate Attack Internally | CM-03-f | CM-3(f) |
| 2) Initiate Attack Internally | CM-03-g | CM-3(g) |
| 2) Initiate Attack Internally | CM-03-z-01-a | CM-3(1)(a) |
| 2) Initiate Attack Internally | CM-03-z-01-b | CM-3(1)(b) |
| 2) Initiate Attack Internally | CM-03-z-01-c | CM-3(1)(c) |
| 2) Initiate Attack Internally | CM-03-z-01-d | CM-3(1)(d) |

| Example Attack Stage | Sortable Control Item Code | NIST Control Item Code |
|---|---|---|
| 2) Initiate Attack Internally | CM-03-z-01-f | CM-3(1)(f) |
| 2) Initiate Attack Internally | CM-08-a | CM-8(a) |
| 2) Initiate Attack Internally | CM-08-b | CM-8(b) |
| 2) Initiate Attack Internally | CM-08-z-01-z | CM-8(1) |
| 2) Initiate Attack Internally | CM-08-z-03-b | CM-8(3)(b) |
| 2) Initiate Attack Internally | MA-03-z-01-z | MA-3(1) |
| 2) Initiate Attack Internally | MA-03-z-03-a | MA-3(3)(a) |
| 2) Initiate Attack Internally | MA-03-z-03-b | MA-3(3)(b) |
| 2) Initiate Attack Internally | MP-07-z-01-z | MP-7(1) |
| 2) Initiate Attack Internally | PS-04-d | PS-4(d) |
| 2) Initiate Attack Internally | SC-15-a | SC-15(a) |

### 3.5.2 Tracing Security Control Items to Capabilities

In defining individual security control items from SP 800-53 Revision 4, keyword search rules were developed and used to map control items to capabilities in an automated manner. A systematic process was followed to validate the keyword rules mappings—testing for missed control items and evaluating false positives and false negatives.

Table 7: Illustrative Keyword Rules to Map to Capabilities, provides two examples of keyword rules used for mapping control items to capabilities.

**Table 7: Illustrative Keyword Rules to Map to Capabilities**

| A control item maps to the Hardware Asset Management (HWAM) capability if one or more of the following are true: |
|---|
| It contains “inventory” |
| It contains “supply chain,” and NOT “monitoring” |
| ….And about 12 other conditions…. |

Each capability-specific volume of this NISTIR addresses a defined capability. Each volume documents both (1) the keyword search rules used (by reference) to search the control text and identify the controls/control items that support the capability; and (2) the list of controls/control items. As a result, there is no need for organizations to repeat the mapping work if the capabilities are used as defined.

Table 8: Tracing Control Items to the HWAM Capability (EXAMPLE), provides a sampling of the control items that are traceable to the HWAM capability.

**Table 8: Tracing Control Items to the HWAM Capability (EXAMPLE)**

| Capability | Security Control Baseline | Sortable Control Item Code | NIST Control Item Code |
|---|---|---|---|
| HWAM | Low | AC-19-a | AC-19(a) |
| HWAM | Low | CM-08-b | CM-8(b) |
| HWAM | Low | PS-04-d | PS-4(d) |
| HWAM | Low | SC-15-b | SC-15(b) |
| HWAM | Moderate | AC-19-z-05-z | AC-19(5) |
| HWAM | Moderate | CM-02-z-07-a | CM-2(7)(a) |
| HWAM | Moderate | CM-03-a | CM-3(a) |
| HWAM | Moderate | CM-03-d | CM-3(d) |
| HWAM | Moderate | CM-08-z-03-b | CM-8(3)(b) |
| HWAM | Moderate | MA-03-z-01-z | MA-3(1) |
| HWAM | Moderate | MP-07-z-01-z | MP-7(1) |
| HWAM | High | CM-03-z-01-a | CM-3(1)(a) |
| HWAM | High | CM-03-z-01-e | CM-3(1)(e) |
| HWAM | High | CM-03-z-01-f | CM-3(1)(f) |
| HWAM | High | CM-08-z-02-z | CM-8(2) |
| HWAM | High | MA-03-z-03-a | MA-3(3)(a) |
| HWAM | High | SA-12 | SA-12 |

### 3.5.3 Tracing Security Control Items to Sub-Capabilities

The control items supporting each sub-capability are listed in Section 3.2 of each capability-specific volume of this NISTIR. For each sub-capability, this is documented in a table similar to Table 9, which includes a sample of control items that trace to the sub-capability of preventing or reducing the number of authorized devices without an assigned device manager within the assessment boundary.

**Table 9: Tracing Control Items to the Sub-Capabilities: Selected Examples for the *Prevent Authorized Devices without a Device Manager* Sub-Capability**

| Defect Check ID | Baseline | Sortable Control Item Code | NIST Control Item Code |
|---|---|---|---|
| HWAM-F02 | Low | AC-19-b | AC-19(b) |
| HWAM-F02 | Low | CM-08-z-04-z | CM-8(4) |
| HWAM-F02 | Moderate | CM-03-b | CM-3(b) |
| HWAM-F02 | Moderate | MA-03-z-01-z | MA-3(1) |
| HWAM-F02 | High | CM-03-z-01-a | CM-3(1)(a) |

## 3.6 *Synergies Across Each Abstraction Level*

Capabilities can be mutually supportive, but because this NISTIR documents the types of traceability within a defined security capability, the synergies that operate across capabilities might not be immediately evident. There are many synergies that cut across security capabilities that can be identified and are useful for security program planning and overall risk management. Two examples are shown below.

### 3.6.1 Multiple Capabilities Support Addressing Each Attack Step

There is a many-to-many relationship between security capabilities and attack steps. Attack steps focus on the attacker's view of the system, i.e., ways to find and exploit vulnerabilities. Security capabilities focus on the defender's view of the system, i.e., ways to prevent attacks or reduce the harm from attacks. Figure 4: Capabilities Work Together to Block Attack Steps, shows which security capabilities support each of the attack steps.

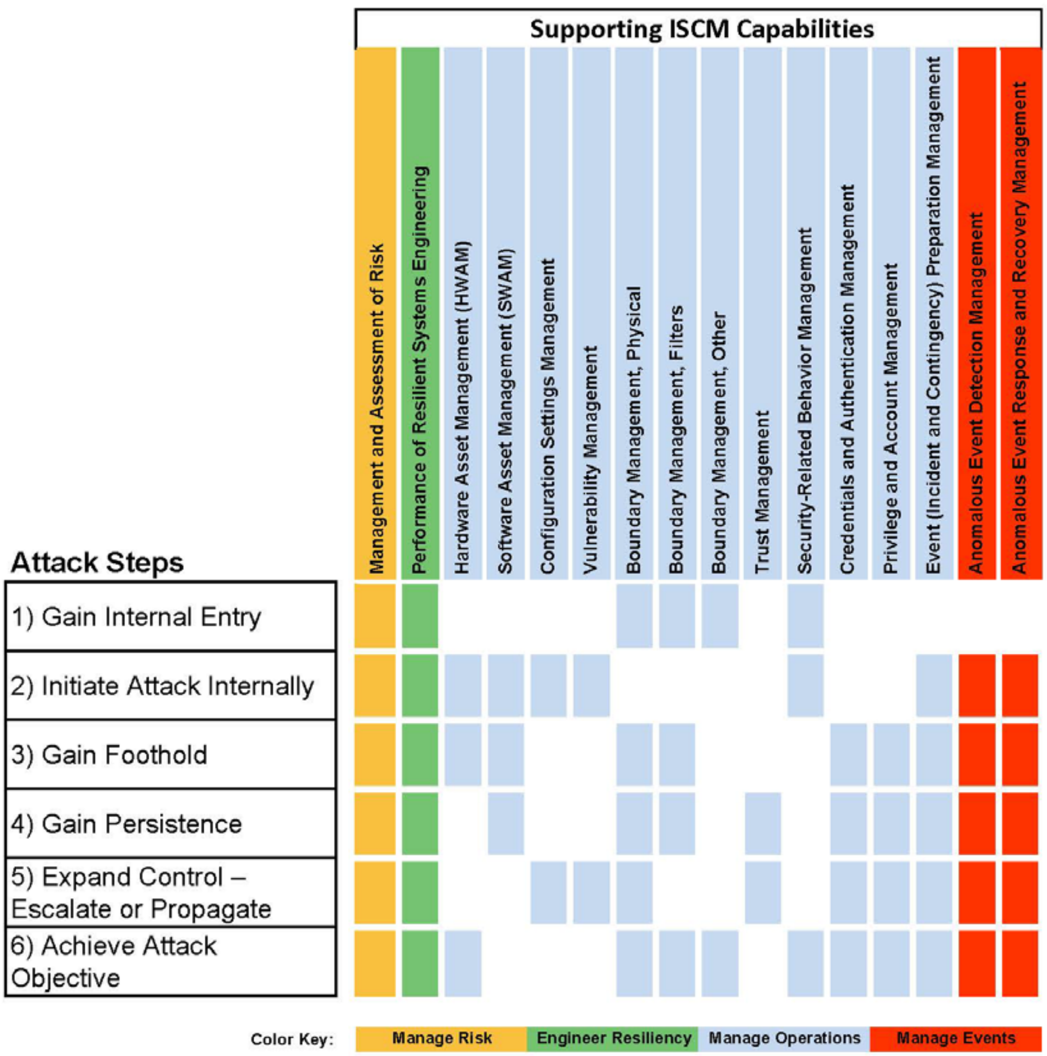


**Figure 4: Capabilities Work Together to Block Attack Steps**

Consider three capabilities that support blocking or delaying an attack from initiating internally:

- HWAM can prevent the entry of malware by detecting unauthorized/unmanaged devices;
- SWAM can do the same through both blacklisting and whitelisting of software; and
- Security-related behavior management can block entry by helping the user avoid phishing attacks and by preventing users from installing unauthorized hardware and software, etc.

Working together, these capabilities might be said to provide defense in depth (or more accurately, defense in breadth), to block attacks at each attack step.

### 3.6.2 Many Controls Support Multiple Capabilities

Most control items support more than one capability. This is because:

- Control items do not consider capabilities; and
- Some control items reflect generic processes (e.g., configuration management) that support multiple capabilities.

Table 10 illustrates an example of a control item that supports multiple capabilities.

**Table 10: Example of a Control Item Supporting Multiple Capabilities**

| NIST Control Item Code | Security Capability Supported |
|---|---|
| CM-3(b) | Network Boundary 1: Firewall and Routing Rules; Content Filtering Rules |
| CM-3(b) | Configuration Setting Management |
| CM-3(b) | Generic Auditing, Logging, and Monitoring to Detect Incidents and Contingencies |
| CM-3(b) | Hardware Asset Management |
| CM-3(b) | Plan and Prepare for Incidents and Contingencies |
| CM-3(b) | Respond to Incidents and Contingencies |
| CM-3(b) | Manage Risk and Budget at Management Level |
| CM-3(b) | Software Asset Management |
| CM-3(b) | Systems Engineering |

Synergies will be summarized in the final volume, to be published after security capability-specific volumes are completed.

# 4. Using Actual State and Desired State Specification to Detect Defects

This section explains the requisite preparation for automated ISCM assessment, to describe how the assessment process recognizes the actual state and desired state specification so that it can compare them. Because it is often inefficient to set up an automated security control assessment regime for each information system separately, this section introduces the concept of an assessment boundary, which may be different from (typically much larger than) authorization boundaries as defined in SP 800-37. The final part of this section discusses a key requirement for automation of a security control assessment—to have the desired state specification expressed in computable data (rather than in free-form text) that can be compared to the actual state digitally or mechanically.

## 4.1 *Actual State and Desired State Specification*

SP 800-53A defines the test method as the process of exercising one or more assessment objects (i.e., activities or mechanisms) under specified conditions to compare actual with expected behavior. In the rest of this document, the terms *actual state* and *desired state specification* are used instead of actual behavior and expected behavior. See Section 4.4, The Desired State Specification, for an explanation of why *state* is used instead of *behavior*. In the current climate of security automation, the actual state is the security-related information most likely to be available. The automated security control assessment model assumes that data about the actual state of the objects being assessed can be collected by tools called collectors.

## 4.2 *Collectors and the Collection System*

### 4.2.1 Actual State Collectors

Collectors[15] are the part of the collection system that interfaces with the objects being assessed and with those who set policy for those objects. The collectors might be scanners, agents, appliances, data entry processes, etc. How the collectors work is unimportant as long as they provide reliable and valid (accurate) data that are timely and complete.

### 4.2.2 Collection of Desired State Specifications

The system must be able to manage (collect, process, store, present, etc.) desired state specification data for each automated security control assessment implementation.

Some desired state specifications are federally defined (e.g., CVEs, or federal configuration settings such as the U.S. Government Configuration Baseline [USGCB]). The organization's agency dashboard can receive federally defined desired state specification data[16] from the federal

[15] Collectors may also be referred to as sensors.
[16] The desired state specification data are received in the form of defect checks. See Section 5, Defect Checks.

dashboard. Other desired state specifications are organization-specific (e.g., lists of authorized devices or frequency of training requirements).

The collection system itself and the agency dashboard work together to represent organization-defined desired state specifications. For example:

- Inventories of authorized devices/software are provided by the collection system (which provides the functionality to automatically import or enter these data).
- Values for organization-specific configuration settings are managed (collected, processed, stored, presented, etc.) by the defect check list in the agency dashboard.

### 4.2.3 The Collection System

A *collection system*, depicted in Figure 5: *ISCM Collection System*, manages the collectors, generates actual state data, collects desired state data, and compares the collector data (actual state) to the desired state specification to find defects.

> The ISCM collection system is an instance of the Continuous Asset Evaluation, Situational Awareness, and Risk Scoring Reference Architecture (CAESARS). This creates some confusion, because CAESARS contains a *collection subsystem*. The CAESARS collection subsystem functionally approximates the *collectors* as described above. Thus, the ISCM collection system and the CAESARS collection subsystem are not the same. The CAESARS Framework is defined in IR 7756.

In addition to the collector functions of the CAESARS collection subsystem, the ISCM collection system includes: (1) a repository to hold data; (2) an orchestration engine to coordinate collectors to collect time and event-driven data and to coordinate time- and event-driven communications with an agency dashboard; and (3) an analysis engine to find defects and identify the event-driven data collection needed. Typically, the collection system's graphical user interface (GUI) and reporting function is minimal because data are sent directly to the agency dashboard to provide these functions.

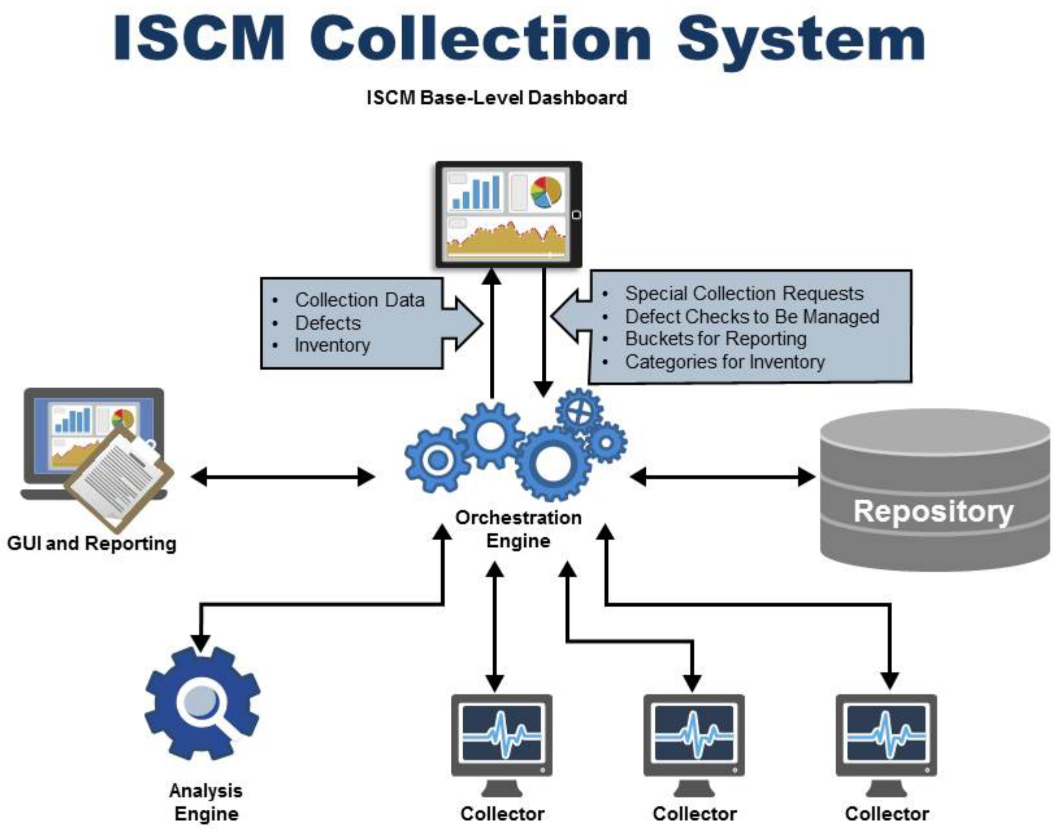


**Figure 5: ISCM Collection System**

## 4.3 *Authorization Boundary and Assessment Boundary*

For security-related information generated by the collectors and processed by the collection system to be of maximal usefulness, all defects on a system that impose a risk to that system must be mapped, including the following:

- Defects in the controls implemented at the system level;
- Defects in common controls that the system inherits; and
- Defects in otherwise unrelated objects that allow an attack path to be established that can damage the system.

In order for the collection system collectors to detect and process all three types of defects, objects being assessed are grouped into the following categories:

- Objects and defects within the information system authorization boundary; and
- Objects and defects from common controls which the system inherits.

This allows the agency dashboard to compute risk from both groups.

### 4.3.1 Information System Authorization Boundary

The concept of an information system authorization boundary is well described in multiple NIST publications. The following is the formal definition of authorization boundary from IR 7298:

> All components of an information system to be authorized for operation by an authorizing official and excludes separately authorized systems, to which the information system is connected.

In other words, authorization boundaries are used to ensure that information systems are distinct in order to facilitate information security management and responsibility. SP 800-53 includes the following control and a control enhancement requiring components to be assigned to an information system and that those system components are not duplicated in other system component inventories:

> **CM-8** INFORMATION SYSTEM COMPONENT INVENTORY
>
> Control: The organization:
>
> a. Develops and documents an inventory of information system components that:
>    1. Accurately reflects the current information system;
>    2. Includes all components within the authorization boundary of the information system;
>    3. Is at the level of granularity deemed necessary for tracking and reporting; and
>    4. Includes [Assignment: organization-defined information deemed necessary to achieve effective information system component accountability]; and
>
> b. Reviews and updates the information system component inventory [*Assignment: organization-defined frequency*].
>
> **CM-8(5)** INFORMATION SYSTEM COMPONENT INVENTORY | NO DUPLICATE ACCOUNTING OF COMPONENTS
>
> The organization verifies that all components within the authorization boundary of the information system are not duplicated in other information system inventories**.**

### 4.3.2 ISCM Assessment Boundary

Once organizations begin to automate security control assessment of information system components, it is not cost-effective to implement a separate automated collection process within each authorization boundary. Thus, the concept of an ***assessment*** boundary (generally larger and inclusive of more information systems and system components than an authorization boundary) is introduced as part of an ISCM program.

The most cost-effective assessment boundary consists of all devices connected to a network that is bounded by traffic filters (firewalls) and other boundary protections (e.g., routers, switches), out to Internet devices managed separately from the network itself. Typically, this boundary would include a perimeter network or demilitarized zone (DMZ), extranet, intranet, and perhaps internal enclaves. Within the federal government, the boundary to the outside network (typically the Internet) is mediated through a trusted Internet connection (TIC), which is the external boundary of the network.

Because the assessment boundary is comprehensive, it can be used to assess the components of multiple systems within the assessment boundary. This has the following advantages:

(1) The fixed cost of setting up the collectors, collection system, and ISCM dashboard hierarchy is paid only once.

(2) The security-related information that is generated can be used to look at risk across systems, in several forms:

   a. A system may inherit controls from other systems. For example, most systems within an organization are likely to inherit boundary controls from a network system. This is typically covered by the concept of inheritance of common controls.

   b. The system from which the common controls are inherited may have all the inherited controls implemented correctly, but it may have other defects that could be attacked to compromise the strength of the common control implementation. Though the security assessment reports and Plan of Action and Milestones (POA&M) information for systems providing common (inheritable) controls are to be made available to inheriting system staff, such information is not always included in traditional system assessment analysis; however, the security-related information about the common control-providing system can be seen automatically along with the system-specific information through a properly constructed agency dashboard.

   c. Component(s) (objects) on a given network that are within specific authorization boundaries may be vulnerable and become attack vectors through which other objects on the network may be compromised. This risk cannot be seen by looking inside a single authorization boundary; information systems can inherit risk from another object without inheritance of controls from that object.

   The extra inherited risk information described in the preceding cases b and c is not only useful at the system-level tier; it also provides valuable information about aggregated risk from the missions/business tier and organizational tier perspectives regarding how risk from one particular system can affect the entire organization.

(3) In large networks, there are typically components that fail to be assigned to any authorization boundary. Such components may regularly appear, disappear, and reappear on large networks creating an ongoing problem. These components may be invisible to those looking only within authorization boundaries, because by definition they are outside such boundaries. By looking at the component inventory across the assessment boundary and identifying unassigned components, it becomes more feasible to structure a process to assign these components to a system for appropriate device management. In other words, this helps ensure that all components are already assigned to an authorization boundary, flagged to be assigned to one, or removed from the network.

Throughout this NISTIR, the ISCM assessment boundary is referred to as the ISCM Target Network (ISCM-TN).

### 4.3.3 Tracing System Risk to its Sources

For an automated security control assessment system to accurately track the risks associated with each authorization boundary (information system), it must be able to identify the following:

(1) Components (objects) and controls implemented at the system level;

(2) Components of other systems that provide common controls and controls implemented on those components;

(3) Components within the assessment boundary that are unmanaged/unassigned; and

(4) Components on potential attack paths to the system.

For Item 1, identifying the components inside the system's authorization boundary may be a manual process. However, it is often possible to identify markers (registry entries, specific executables, etc.) that allow the asset management actual state collection system to identify devices that are in the boundary of a system. Identifying markers are preferable whenever possible, as they are more likely to be current and complete.

For Item 2, identifying the objects from which a system inherits controls may be as simple as identifying the information system(s) or business processes providing the common controls, and then including all of the objects when assessing the effectiveness of common controls. In other cases, the scope of common control components included in a system's automated security control assessment is narrowed when the system is supported by only one or some components of a given common control information system.

For Item 3, unmanaged and/or unassigned devices within the assessment boundary impose risk on all connected components. Item 4 may help clarify how much unmanaged/unassigned components affect the system being considered.

Finally, for Item 4, potential attack paths can only be considered when data and tools are adequately structured to compute likely and exploitable attack paths within the assessment boundary to see which components are on attack paths that may impose risk to the system. Components on the potential attack paths may include unmanaged or unassigned devices.

Once the components to be assessed are identified for an information system, an agency dashboard should be able to process the assessment results and derived known risks for the system from the three sources listed in Section 4.3, Authorization Boundary and Assessment Boundary. The agency dashboard should then be able to provide a view of the system's risk and promptly alert designated roles when any of the following are identified:

- Defects in system components;
- Defects in components providing common controls; and/or
- Defects in other components within the assessment boundary.

## 4.4 *The Desired State Specification*

The strategy to increase the number of security controls for which monitoring for effectiveness can be automated depends on defining a desired state specification and expressing the desired state specification in a machine-readable data format that can be compared with the actual state. The desired state specification is a defined value (specification) to which the actual state value can be compared. Mismatches of the two values indicate a defect is present in the effectiveness of one or more security controls. For example, an organizational policy states that user accounts will be locked after three unsuccessful logon attempts. The desired state specification would thus be that applicable devices are configured to lock accounts after three unsuccessful logon attempts. If, during automated security control assessment, the security-related information collected indicates a specific device is configured such that accounts are locked after five unsuccessful logon attempts, a mismatch between the desired state specification (three attempts allowed before lockout) and the actual state (five attempts allowed before lockout) is identified, which may reflect a problem with the effectiveness of SP 800-53 controls AC-7, Unsuccessful Logon Attempts, AC-2, Account Management, CM-2, Baseline Configuration, etc.

> Having a machine-readable data-based desired state specification is fundamental to automation of security control assessments.

The automated security control assessment system model assumes that data about the desired state specification is communicated to the collection system by the organization managing the information system.

Examples of desired state specification information include the following:

- Authorized devices;
- Authorized device roles;
- White-listed software for each device role;
- Required frequency of security awareness training;
- Authorized configuration settings for each device role;
- Vulnerable software versions (provided by NVD);
- Authorized users and privileges; and
- Many others.

### 4.4.1 Types of Desired State Specifications

The desired state specification may be as expressed in any of the following examples. For simplicity, the shorter phrase *desired state specification* is used, instead of the more complete and explicit but cumbersome phrase, “desired/allowed/prohibited state/behavior specification.”

**Table 11: Types of Desired State Specifications**

| Type of Desired State Specification | Simplified Examples (Actual cases might be more complex.) |
|---|---|
| Desired state | If software product X is present, setting Z should have value Y to increase security. |
| Prohibited state | If software product X is present, specified patch levels have CVEs that produce risk and are prohibited. |
| Expected state[a] | If software product X is present, the device should have [a list of executables with hashes to identify them]. The expected state of a software product may be that it is fully installed with the correct hashes, but the actual state may be that some files have altered hashes. |
| Desired behavior | Users receiving email will validate the origin of the e-mail before using links or attachments in the email. |
| Prohibited behavior | Users using accounts allowed to install software, i.e., privileged accounts, are not permitted to browse the Internet or use email from those accounts. |
| Expected behavior | User B normally logs in from devices in the [City] area during the period from 8 a.m. to 6 p.m. This would constitute expected behavior. Other patterns of login activity might indicate account compromise. |

[a] Desired and prohibited states and behaviors express normative policy. In contrast, expected states and behaviors are not normative policy but descriptions of patterns. Expected states and behaviors are used to detect unusual (thus anomalous and suspected as malicious) states and behaviors that might require responses and recovery. Expected states and behaviors do not tend to be used outside the capabilities of Anomalous Event Detection Management and Anomalous Event Response and Recovery Management.

Note that the prohibited state/behavior can always be restated as a desired behavior. Table 12: Equivalence of Prohibited and Desired State Specification – An Example, provides such a restatement.

**Table 12: Equivalence of Prohibited and Desired State Specification – An Example**

| Prohibited Behavior | Equivalent Desired Behavior |
|---|---|
| Users using accounts allowed to install software are not permitted to browse the Internet or use email from those accounts. | Users using accounts allowed to install software do not browse the Internet or use email from those accounts. |

Expected behavior can sometimes be restated as desired behavior, except that it indicates a symptom of a *possible* problem rather than of a definite problem.

### 4.4.2 Desired State Specification Reflects Policy

As noted above, the desired state specification is an expression of policy in a machine-readable form (database) that can be easily compared to actual state data collected by automated means.

Organizations develop policies to support security control implementations and information security in general. If these policies are expressed in the form of data that can be used to automate testing and display both the results of such testing and the policy in human-readable form via an ISCM dashboard, then:

- The organization does not need to manually produce the same policies in a traditional text form (Word or PDF document, for example) because this can be generated from the authoritative automated specification (as noted above); and
- The use of data to express the policies in machine-readable form supports automated testing.

These conclusions assume that the interface that displays results of the automated testing clearly communicates in human-readable form both the policy and the defects tested/found.

### 4.4.3 Desired State Specification Demonstrates the Existence of Policy

It is often assumed that only technical controls can be assessed for effectiveness via automation and that management and operational controls cannot be assessed via automation; however, it is actually often possible to assess the effectiveness of management and operational controls via automation by placing the desired state specification in data.

Consider that the desired state specification itself is often policy. Thus, the existence of a desired state specification is evidence that the organization has policy within a given security capability. To the extent that the organization can automate collection of corresponding actual state data to identify where desired and actual state do and do not match, the organization is clearly using automation to assess whether or not the policy is applied.

When an organization demonstrates that it is assessing whether policies are followed, it also demonstrates that the policy exists and is documented in the desired state specification database. To automate this process, the automated security control assessment system must be able to automatically compare the policy with the actual state.

An example is the control for periodic awareness training (AT-2). The organization must decide how frequently this training is to be provided. If the specified time-frame parameter is 360 days, that information is stored in data as the “policy definition,” i.e., the desired state specification. Then the parameter can be compared to the actual time elapsed since the last recorded awareness training completion as it was recorded in the organizational learning management system. If the training has not occurred within the specified period, a defect would be recorded.

The example demonstrates how nontechnical controls can be automatically tested more often than might be expected. The operational key is developing an adequate desired state specification that expresses the policy.

**Note**

Even as organizations seek to automate security control assessments to the greatest extent possible using methods as described in the example, the fact remains that while the assessment of many controls can be fully automated, the assessment of some controls might be only partially automated or might not be automated at all. Organizations must carefully consider the assessment approach and specific assessment methods to be used as part of the ISCM strategy.

## 4.5 *Using Automation to Compare Actual State and Desired State Specification*

When conducting manual/procedural security control assessments, the security assessment plan, actual state, desired state specification, and defects found are largely managed in text documents. This requires that they be written and edited by humans, which is a slow and often expensive process. A security assessment report could be out of date by the time it is finished, simply because the system changes so fast (machines added, patched, etc.) that the manual assessments cannot keep up.

The assumption of the automated security control assessment approach presented here is that the actual state results collected and the desired state specification are both expressed in data, such that a computer can effectively analyze the results. This means that the collection system's analysis engine must be able to do the following:

- Match objects being assessed with their respective desired state specifications;
- Match object state data with the defect check that is relevant to assess the object;
- Match the objects in real time without significant human intervention; and
- Send the resulting defects to the agency dashboard for prioritization and response.

For the automated security control assessment system to be able to produce useful results, it must be able to match an object identifier in the actual state with an object identifier in the desired state specification for objects like devices, software products, etc.

For more on this topic, see the material on assessment criteria in Section 5.4, Defect Check Documentation.

# 5. Defect Checks

This section describes the concept of a defect check. Defect checks provide a way to assess control items in an automated fashion based on the determination statements. Defect checks verify the determination statements for control items that support the purpose (capability or sub-capability) being assessed. Defect checks are key to the automated security control assessment process.

Another way to look at a defect check is as a statement defining the desired state specification in data by finding what is NOT in the desired state specification.

## 5.1 *Defect Checks and Determination Statements*

In SP 800-53A, which provides guidance for assessing SP 800-53 security controls, an assessment objective, in the form of one or more **determination statements**, is specified for each control item. The determination statements begin with "Determine if." They then deconstruct the control items further into assessable parts. The assessment objective is to determine if the control is effective. See the example in Table 13: Example Control and Determination Statements.

**Table 13: Example Control and Determination Statements**

| AC-2(2) – ACCOUNT MANAGEMENT | |
|---|---|
| The Control Statement (800-53 Revision 4) | **The Determination Statement (800-53A Revision 4)** |
| The information system automatically removes or disables temporary and emergency accounts after [*Assignment: organization-defined time period for each type of account*]. | ASSESSMENT OBJECTIVE: *Determine if:*<br>[1] *The organization defines a time period after which the information system automatically removes or disables temporary and emergency accounts; and*<br>[2] *The information system automatically removes or disables temporary and emergency accounts after the organization-defined time period for each type of account.* |

In this example, the control item is deconstructed into two assessment objectives. Assessment objective [1] asks whether the organization specified the relevant desired state specification. Assessment objective [2] asks whether the desired state specification is being implemented.

A **defect check** is a way to verify determination statements. It has the following additional properties. A defect check:

- Is stated as a test (wherever appropriate);
- Can be automated;[17]
- Explicitly defines a particular desired state specification that will then be compared to the corresponding actual state to determine the test result; and
- Is typically at a higher level of abstraction than a single determination statement (see the next section).

## 5.2 *Interpreting Defect Checks as Tests of Control Items*

The defect check is designed to focus on the purpose a set of controls are designed to achieve. Because a defect check is designed intentionally to determine whether a collection of controls are achieving their purpose, the defect check is at a higher level of abstraction than the determination statement(s) for a single control item.

> For example, in hardware asset management there is a supply chain defect check to verify whether the hardware supplier and/or manufacturer are on the approved list. This defect check:
>
> - Is directly supported by one control, SA-12, which calls for consideration of supply chain issues in approving devices; and
> - Is indirectly supported by other controls such as the parts of CM-3, which require a configuration management process to consider security impacts explicitly in the change control process (implicitly including supply chain, where appropriate).

This relationship of defect checks to control items is illustrated in Figure 6: Focus of Defect Checks and Determination Statements.

## 5.3 *Interpreting Defect Checks as Tests of Sub-Capabilities*

As discussed in the last section, the collection of control items assessed by a defect check work together to achieve a purpose. In the example, the purpose is to reduce the potential consequences of supply chain attacks—one part of the overall hardware asset management capability and, in effect, a *sub-capability* of HWAM (see Sub-Capabilities).

While the defect check assesses the individual *controls* or *control items* that work together to achieve a purpose, at the same time the defect check also tests the overall effectiveness of the

[17] When assessing a control item cannot be automated efficiently, manual/procedural assessment approaches are used.

controls *working together as a sub-capability*. In NISTIR 8011, defect checks are designed so that there is one defect check for each defined sub-capability.

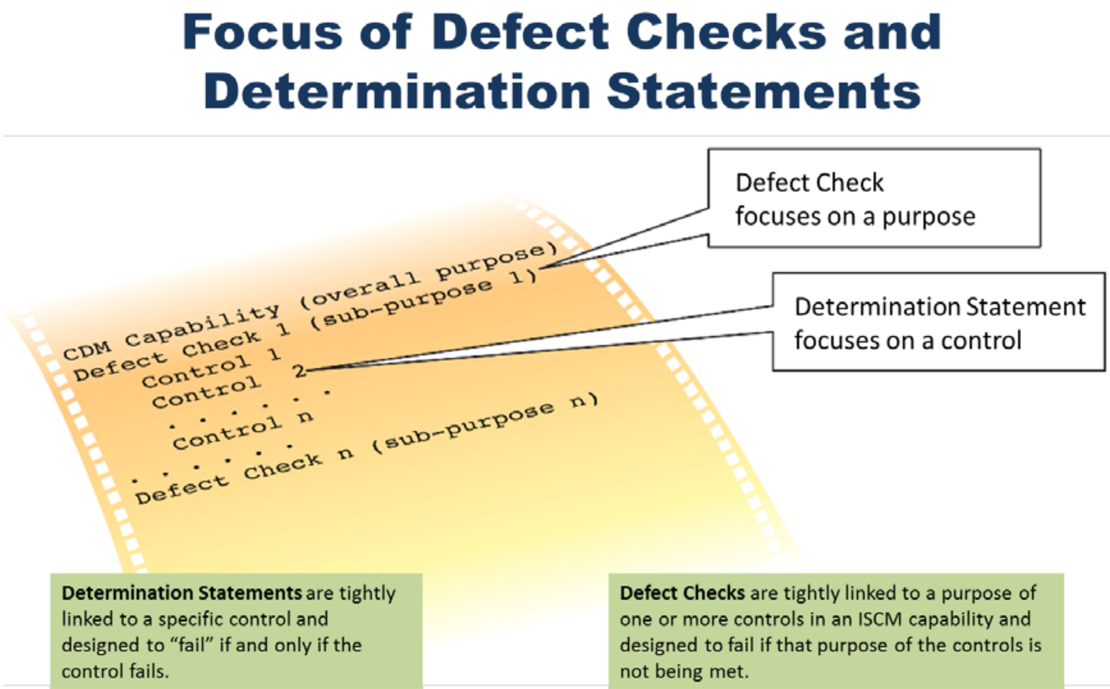


**Figure 6: Focus of Defect Checks and Determination Statements**

The difference in the level of focus has a significant impact on how a found defect is interpreted. The difference relates to the sensitivity and specificity of the result.

**Sensitivity:**

> A sensitive test is one which finds all of the cases where a defect occurs; that is, it has a low false negative rate.

The focus of defect checks on the purpose of a set of controls can be highly sensitive because it directly measures a result of interest (achievement of purpose).

> In the example of supply chain controls, the defect check for hardware supply chain would fail if either:
>
> - A list of approved suppliers and manufacturers was not set up per SA-12; OR
> - A device from a supplier not on that list was approved by the change control process per CM-3.

**Specificity:**

> A specific test is one which does not report a defect when it is not present; that is, it has a low false positive rate.

Because defect checks measure the result to be achieved by a set of controls, they can be fairly specific about whether that result was achieved. However, failure to achieve the result does not imply that ALL of the controls supporting that sub-capability failed. Thus, the defect check is not specific at the control or control item level of abstraction.

> In the example of supply chain controls, the failure of the defect check does not help determine whether the control failed because:
>
> - A list of approved suppliers and manufacturers was not set up per SA-12; OR
> - A device from a supplier not on that list was approved by the change control process per CM-3.
>
> The defect could have occurred because either or both failed. Thus, a defect check failure should NOT be interpreted to mean that ALL the supporting controls failed.

**Root Cause Analysis adds specificity at the control level:**

In epidemiology, it is commonly understood that it is hard to make a single test both sensitive and specific. As criteria are changed to improve one measure, the other measure deteriorates. Thus, a common testing strategy is to use two tests in phases:

(1) A very sensitive test is used to find as many “positive” results as possible even though it may include some false positives.

(2) A very specific test is given to the cases that failed the first test, to eliminate the false positives.

This combination of two tests is often the most cost-effective way to identify all true positives in a population.

In the case of security control testing, the defect check is like a health screening test. It provides warning that one or more controls that support its purpose are likely failing, but because it is possible that only one control failed, it cannot be assumed that all the supporting controls failed.

To resolve this issue—the issue of not necessarily knowing which control failure led to the defect check failure—root cause analysis is used to determine which control(s) supporting the sub-capability failed. See Section 7.2, Root Cause Analysis.

> In the example of supply chain controls, imagine a scenario in which root cause analysis showed that an approved list of device manufacturers was maintained, but a device purchased from an unapproved manufacturer was installed. Root cause analysis might show that the failure was a problem within the change control process (CM-3).
>
> A trend analysis could further indicate whether the weakness in the change control process was a recurring problem.

Conclusions when a defect check falls outside of an acceptable threshold:

- One or more of the supporting control(s) failed;
- Root cause analysis is used to determine *which* control(s) failed; and
- It is NOT necessarily the case that all supporting controls failed.

## 5.4 *Defect Check Documentation*

Defect checks are documented with a table in the following form:

**Table 14: Sample Rows from a Hypothetical Sub-Capability and Defect Check Description[a]**

***Prevent Unauthorized Devices Sub-Capability and Defect Check HWAM-F01***

The purpose of this sub-capability is defined as follows:

| Sub-Capability Name | Sub-Capability Purpose |
|---|---|
| Prevent Unauthorized Devices | Prevent or reduce the presence of unauthorized devices thus reducing the number of potentially malicious or high risk devices. |

The defect check to assess whether this sub-capability is operating effectively is defined as follows:

| Defect Check ID | Defect Check Name | Assessment Criteria Summary | Assessment Criteria Notes | Selected |
|---|---|---|---|---|
| HWAM-F01 | Unauthorized devices | Device is In Actual State but not in Desired State [See supplemental criteria in L02] | Assessment Criteria Notes:<br>1) The actual state is the list (inventory) of all devices (within an organizationally defined tolerance) in the assessment boundary as determined by the ISCM system.<br>2) The desired state specification is a list of all devices authorized to be in the assessment boundary.<br>3) A defect is a device in the actual state but not in the desired state, and is thus unauthorized. This is computed by simple set differencing. | Yes |

[a] Responsibility is illustrative operational responsibility. The illustrative assignments do not change the overall management responsibilities defined in other NIST standards and guidelines. Moreover, the responsibilities can be customized by each organization to adapt to local circumstances.

- **Sub-Capability Name** column provides a short name to address the purpose of the sub-capability.
- **Sub-Capability Purpose** column contains a full description of the purpose of the sub-capability.
- **The Defect Check ID** column includes:
  - The ISCM security capability abbreviation (HWAM in the example);
  - A letter F, L or Q, to indicate whether the provisional level of the defect check is:
    - Foundational (F);
    - Local (L) security-related defect check (see Section 5.9, Foundational and Local Defect Checks); or
    - Data quality (Q) defect check (see Section 5.5 Data Quality Measures); and
  - A number to uniquely identify the check.
- **The Defect Check Name** column includes a short name to identify the defect check.
- **The Assessment Criteria Summary** includes a short description of how to decide (compute) whether a defect is present.
- **The Assessment Criteria Notes** expand on the assessment criteria. At a minimum, these notes define the following:
  - What data are used
    - to define the actual state; and
    - to define the desired state specification; and
  - How these two data sets are used to identify a defect.
- The selected column contains a yes if the organization has decided to select the defect check for implementation.

The potential most likely actions needed to resolve a defect, and the responsible roles, are listed in an additional table. For example:

**Example Mitigation/Responses:** The following responses and/or mitigations (with example assignments) are common ones appropriate when a defect is discovered in the *prevent unauthorized devices* sub-capability. The example assignments shown do not change the overall management responsibilities defined in other NIST documents. Moreover, they can be customized by each organization to best adapt to local circumstances.

| Defect Check ID | Mitigation/Response Description | Primary Responsibility |
|---|---|---|
| HWAM-F01 | Remove Device | DM |
| HWAM-F01 | Authorize Device | DSM |
| HWAM-F01 | Accept Risk | RskEx |
| HWAM-F01 | Primary Responsibility | DSM |

A *primary responsibility* is also suggested in this table. The role with primary responsibility determines the most appropriate response and ensures that the response action is allocated to the appropriate role. Responsibility is defined in terms of both NIST managerial roles and/or operational roles. See Section 8, Roles and Responsibilities.

The assessment criteria notes are intentionally somewhat general to allow organizations flexibility in implementation. However, the notes are specific enough to allow the organization to design a reliable (repeatable) test.

The individual security capability volumes explain the specific purposes to be achieved by each sub-capability and the supporting controls as they relate to the capability covered in that volume. The defect checks are designed to provide a valid measure of whether (and to what extent) the purpose of the sub-capability is being achieved.

## 5.5 *Data Quality Measures*

The measures described previously are of little value unless the data collected are both complete and timely. These defect checks use letter prefix "Q" in their ID code.

**Completeness** means the extent to which the security-related information includes assessment of all relevant defects on all objects (within some scope like a capability). Relevant defects are those that produce significant risk, e.g., the top two orders of magnitude. Incomplete metrics tend to bias the results by underestimating total risk.

**Timeliness** means the extent to which the security-related information has been refreshed within the last X hours or days. Data must be collected (and defects mitigated) faster than the attacker(s) can act, in order to be able to stay ahead of their ability to compromise a system.

If metrics for completeness and timeliness are not adequate, the assessment is not useful because the results will underestimate the risk.

**Table 15: Data Quality Measures**

| Measure Type | Description | When to Use this Measure |
| --- | --- | --- |
| Completeness and/or Timeliness Measures | Percent of devices for which complete and timely data (respectively) are being collected. | Setting an organization-defined threshold on completeness and timeliness metrics triggers an alarm when the overall level of completeness and timeliness (respectively) is too low to provide reliable results on defects. |

## 5.6 *Assessment Criteria Device Groupings to Consider*

In order to manage risk for information systems as defined in SP 800-37, devices are grouped by information system (i.e., the authorization boundary) to look at system-level risk.

However, the security-related information produced by automated security control assessment across the larger assessment boundary means that the risk executive has the ability to look at risk for other groupings of devices to better identify risk concentrations and aggregate risk. Groupings that might be useful include devices that are:

- Identified as mission critical;
- Necessary for an integrated business function;
- Managed by a separate business partner;
- Supporting a specific mission across the entire organization; or
- Supporting a particular customer.

Looking at risk (with organization-defined thresholds) across such large groupings of devices helps the organization address organizational and mission/business risk as described in SP 800-39.

## 5.7 *Why Not Call Defects Vulnerabilities or Weaknesses?*

Assessment methods are designed to detect a control failure or control absence. In a quality engineering concept, these failures are typically called *defects*.

> For example, in Six Sigma terms, a defect is a product (assessment object) that has some property (actual state) that is outside the specification limit (desired state).

To avoid confusion, this NISTIR generally uses the term *defect*, meaning security defect, rather than the terms *vulnerability* or *weakness*, to describe control failure or control absence. Using *vulnerability* or *weakness* could create ambiguity between the broadly applied concept of control failure/control absence and the much more specific concepts of Common *Vulnerabilities* and Exposures (CVEs) and Common *Weakness* Enumerations (CWEs). However, it is important to note that while using the terms *vulnerability* and *weakness* is avoided here, it is recognized that from a risk management perspective, a security defect *does* represent a vulnerability or weakness in the information system or its environment of operation.

## 5.8 *Security Controls Selected/Not Selected and Defect Checks*

The controls to be assessed as part of the ISCM program are limited to SP 800-53 controls selected in the low, moderate, and high baselines.

The defect checks are organized so that it is easily determined which defect checks apply to the relevant baseline.

SP 800-53 includes controls and enhancements that are not selected in any baselines. If a system has been tailored to implement one or more of these controls, the organization may create an automated defect check or conduct a manual/procedural assessment to assess that control. Each capability-specific volume of this NISTIR links to a list of the not selected controls related to that capability.

## 5.9 *Foundational and Local Defect Checks*

SP 800-53A states that:

> *Organizations are not expected to employ all of the assessment methods and assessment objects contained within the assessment procedures identified in this publication for the associated security controls deployed within or inherited by organizational information systems. Rather, organizations have the inherent flexibility to determine the level of effort needed for a particular assessment (e.g., which assessment methods and assessment objects are deemed to be the most useful in obtaining the desired results). This determination is made on the basis of what will accomplish the assessment objectives in the most cost-effective manner and with sufficient confidence to support the subsequent determination of the resulting mission or business risk.*[18]

Likewise, organizations are not expected to employ all the defect checks (which are themselves assessment methods) described in this NISTIR.

Defect checks are designated in this NISTIR as one of three types: foundational, local, or data quality defect checks. Note that data quality defect checks are described in Section 5.4.

- **Foundational defect checks**–Defect checks that are fundamental to the purposes of the capability (e.g., HWAM, SWAM, or Configuration Setting Management) in which the defect check appears.
- **Local defect checks** – Defect checks that a given organization determines whether or not to implement. With regard to local defect checks, the organization:

[18] SP 800-53A, pp. 3-4.

- Might not implement a check because the check assesses a security control item that is in a baseline not found within the organization (e.g., the control item is in the high-impact baseline, but the organization has only low- and moderate-impact systems) or within a specific organizational system;
- Might not implement a check because the check assesses a control item that is not implemented at all within the organization or within a specific system (i.e., has been tailored out with appropriate rationale);
- Might implement a check only for specific system objects on which an associated security control is implemented;
- Might implement an alternative version of the local defect check; or
- Might use manual/procedural assessment methods for certain control items.

The organization may customize the defect check tables by adjusting the description of defect checks (adding checks, editing checks, clarifying roles, deselecting checks). Table 14: Sample Rows from a Hypothetical Sub-Capability and Defect Check, provides an example of part of a defect check table.

> In order to automate security control assessments to the greatest extent possible and to support ongoing authorization, implementation of the applicable foundational and local defect checks defined in this NISTIR is needed for *all* implemented security control items.

## 5.10 *Documenting Tailoring Decisions*

Organizations may indicate the rationale for defect check selection decisions in the defect check table's *Select* column.

Organizations may also add or edit local defect checks as appropriate to manage their own risk, e.g., defect checks may be added for security controls implemented as supplemental controls.

Role names and/or assessment boundary names may also be changed to more concrete values applicable to the organization.

# 6. Assessment Plan Documentation

Building on the definitions of actual state, desired state specification, and defect checks in the preceding sections, this section describes documentation that can be produced for each ISCM security capability.

Consistent with the Office of Management and Budget's Memorandum OMB M-14-03, an organization may:

- Use the federal-wide ISCM assessment plan, without change;[19]
- Develop its own assessment plan independently; or
- Create a hybrid that combines elements of both.

## 6.1 *Introduction to Security Assessment Plan Narratives*

The NISTIR volumes for each security capability include security assessment plan narratives that serve as the security assessment plan as defined in SPs 800-37 and 800-53A.

*Note*: The narratives are designed to be consistent with NIST guidance and can be adopted with minimal change as the organization's security assessment plan documentation to address for security controls/control items assessed via defect checks. Section 6.8, Documenting Selected Controls and Tailoring Decisions, describes how an organization might choose to customize these narratives. An example of a possible security assessment plan narrative template follows in Figure 7: Example of a Security Assessment Plan Narrative.

[19] A federal-wide ISCM assessment plan has not been developed to date.

### *Control Item CM-8(a): INFORMATION SYSTEM COMPONENT INVENTORY*

**Control Item Text:**

Control: The organization:

a. Develops and documents an inventory of information system components that:

1. Accurately reflects the current information system;
2. Includes all components within the authorization boundary of the information system;
3. Is at the level of granularity deemed necessary for tracking and reporting; and
4. Includes [Assignment: organization-defined information deemed necessary to achieve effective information system component accountability].

**Determination Statement 1:** [See Section 6.3]

| Determination Statement ID | Determination Statement Text |
|---|---|
| CM-8(a)(1) | Determine if the organization:<br>a. Develops and documents an inventory of information system components {for devices and device components} that:<br>1. Accurately reflects the current information system;<br>2. Includes all components within the authorization boundary of the information system; |

**Roles and Assessment Methods:** [See Section 6.4]

| Determination Statement ID | Implemented By | Assessment Boundary | Assessment Responsibility | Assessment Methods | Selected | Rationale for Risk Acceptance | Frequency of Assessment | Impact of not implementing |
|---|---|---|---|---|---|---|---|---|
| CM-8(a)(1) | DSM | ISCM-TN | ISCM-Sys | Test | | | | |

**Defect Check Rationale Table:** [See Section 6.5]
**A failure in control item effectiveness will create a defect in one or more of these defect checks:**

| Determination Statement ID | Defect Check ID | Defect Check Name | **Rationale**<br>If an [organization-defined measure] for this defect check is above [the organization-defined threshold], ***then defects in an inventory of the {devices and device subcomponents of the} information system that includes all components within the authorization boundary being developed/documented or being accurate related to this control item*** might be the cause of ... |
|---|---|---|---|
| CM-8(a)(1) | HWAM-F01 | Unauthorized devices | the presence of unauthorized devices. |
| CM-8(a)(1) | HWAM-L03 | Required device not installed | a required device not being found in the assessment boundary. |

Note that this example is not complete or authoritative. See the appropriate volume of this NISTIR for the complete and authoritative version. The list of Defect Checks here is exemplary only.

**Figure 7: Example of a Security Assessment Plan Narrative**

## 6.2 *Assessment Scope*

Note that a single control item may support multiple capabilities. Within a capability, only how the control item supports that capability is considered. The insertion of “{devices and device sub-components of the}" into the example in Figure 7: Example of a Security Assessment Plan Narrative, is included to clarify its scope for the HWAM capability and determination statement. Such insertions are included in each capability volume.

## 6.3 *Determination Statements within the Narratives*

Many control items have more than one associated determination statement. The Security Assessment Plan Narrative example in Figure 7: Example of a Security Assessment Plan Narrative, addresses a single determination statement, CM-3f-1; however, CM-3f includes two determination statements, CM-3f-1 and CM-3f-2. Table 16: Example of a Control Item and Its Determination Statements, shows the control item text and the two determination statements. Note that each determination statement has its own assessment narrative.

**Table 16: Example of a Control Item and Its Determination Statements**

| | |
|---|---|
| Control Item Text | **CM-3f** : The organization audits and reviews activities associated with configuration-controlled changes to the {devices and device sub-components of the} information system; and |
| Determination Statement 1 | **CM-3f(1): Determine if the organization:**<br>f. Audits activities associated with configuration-controlled changes to the {devices and device sub-components of the} information system. |
| Determination Statement 2 | **CM-3f(2): Determine if the organization:**<br>f. Reviews activities associated with configuration-controlled changes to the {devices and device sub-components of the} information system. |

The notation for a determination statement includes the control item identifier from SP 800-53—in this case CM-3f—followed by a dash and the determination statement number. Note that the determination statements include the same qualifying language that applied to the control item (per Section 6.2, Assessment Scope).

## 6.4 *Roles and Assessment Methods in the Narratives*

In addition to the control item determination statement, the security assessment plan narrative identifies the following:

- Role responsible[20] for control item implementation (to clarify responsibility for defects);
- Assessment boundary (to clarify scope of assessment, see Section 4.3, Authorization Boundary and Assessment Boundary);
- Role responsible[21] for the security control assessment; and
- Assessment method(s) to be used (see Section 2.2, Automating the Test Assessment Method).

[20] Roles specified are management roles defined in NIST standards and guidelines (Section 8.1) or operational roles (Section 8.2).
[21] See preceding footnote.

## 6.5 *Defect Check Rationale Table*

Within the security assessment plan narrative, a defect check rationale table maps the assessment criteria for each applicable defect check to the determination statement. The table indicates which defect checks fail if the given determination statement is not satisfied, and the table explains (in the rationale column) how the defect check applies (see example in Figure 7: Example of a Security Assessment Plan Narrative). The defect check rationale table indicates how the defect check is, in fact, assessing the control item determination statement in question and includes all the applicable defect checks for each determination statement. The *Defect Check* and *Rationale* columns in the assessment criteria table provide the following:

- The *Defect Check* column identifies the defect checks from the defect check tables that assess the security control/control item. Refer to the defect check tables within each capability volume for a description of how the defect check applies to a given object.
- The *Rationale* column describes the conditions under which a failure of the defect check might be caused by a failure of the control. Moreover, if the control fails too often (per an organization-defined threshold), then it will cause a failure of the security test criteria for a defect check.

Note that the defect check might also fail because another control associated with it fails (see Section 5.2, Interpreting Defect Checks as Tests of Control Items). The mere failure of a defect check does not prove that a given control failed, since the defect check is not specific at the control or control item level. Rather, the assessment criteria are designed so that if the control item fails, the defect check control item-determination statement (CI-DS) assessment criteria will *show* that it failed. See Section 7.2 on root cause analysis for information on how to determine *which* control item(s) caused the defect check to fail. If the CI is determined to have failed, then its control has at least partially failed.

## 6.6 *Tailoring of Security Assessment Plan Narratives*

As noted previously, only the defect checks that assess *implemented* security controls need be applied. The local defect checks provide greater assessment depth and may be selected by the organization based on their risk tolerance and need for greater assurance when corresponding controls are implemented. In addition, each organization has the flexibility to use the narratives as written or to modify them for consistency with organizational risk management requirements, policies, and procedures. Modifications may include (but are not limited to) the following:

- Removing or adding local defect checks;
- Providing an organization-specific definition for such terms as *ISCM Assessed Systems*, *ISCM Target Network*, etc.;
- Adding, modifying, or removing potential response options;
- Clarifying the organization-specific processes that go with each potential response option;
- Using organization-specific terms for the response actions, roles, and responsibilities; and

- Noting which checks are selected.

Tailoring decisions may be documented in the control allocation tables described in Section 6.7, per methods described in Section 6.8.

## 6.7 *Control Allocation Tables*

Control Allocation Tables (CATs) were developed to document security assessment plans for high-, moderate-, and low-impact security control baselines within each security capability.

| |
|---|
| CAT tables are designed to provide a summary of the security assessment plan narratives and are used to indicate which controls are selected. This helps to define which defect checks are required. |

CATs are provided in each capability-specific volume of this NISTIR. The CATs provide a summary of the security assessment plan narratives discussed above. Table 17: Control Allocation Table Column Explanations, provides definitions of the columns in the CAT. Table 18: Notional Control Allocation Table – Example, provides an example of a control allocation table. The example illustrates how the table summarizes the narratives: The narrative in Figure 7: Example of a Security Assessment Plan Narrative, can be compared with the corresponding row in Table 18: Notional Control Allocation Table – Example, to see how the narrative is summarized. If organizations tailor the security assessment plan narratives, the Control Allocation Tables should be revised for consistency.

Note that the table does not include the explanation of how each defect check helps to assess the control; see assessment criteria tables within the security assessment plan narratives for such explanations.

**Table 17: Control Allocation Table Column Explanations**

| Column | Explanation |
|---|---|
| Determination Statement ID | Maps back to the SP 800-53 control item being tested. |
| Implemented by | The role or system that is primarily responsible for implementing the SP 800-53 control and control items being assessed. |
| Assessment Boundary | The ISCM assessment boundary where the control item is found. |
| Assessment Responsibility | The entity that performs the assessment. |
| Assessment Method | Generally "Test" for automated assessment and "TBD" for Manual assessment. |
| Selected? | Documents whether or not the given organization or information system selects and uses the test. |
| Rationale for Risk Acceptance | Documents a rationale for non-selection or for risk acceptance of a selected control when assessment results reflect other than satisfied. |
| Frequency of Monitoring[a] | The minimum frequency with which the test is to be conducted. |
| Impact of not Implementing | The impact to organizational objects, individuals, other organizations, and the Nation that a failure of this control may create. |

[a] Frequencies specified in this column are at least as often as the frequency determinations in the organization's information system continuous monitoring strategy.

## 6.8 *Documenting Selected Controls and Tailoring Decisions*

In addition to summarizing the security assessment plan narratives, several of the CAT columns provide a space to document how and why the security control baseline was tailored by the organization. This allows the table to help document the system security plan in the following ways:

- The *Selected* column can be used to document which controls are selected for implementation; and
- When controls are tailored out of an applicable baseline:
  - The *Impact* column can be used to document the assumed impact of non-selection; and
  - The *Risk Acceptance* column can be used to document the rationale for risk acceptance (i.e., justification is provided for security control tailoring decisions).

**Table 18: Notional Control Allocation Table – Example**

| Determination Statement ID | Implemented By | Assessment Boundary | Assessment Responsibility | Assessment Methods | Selected[a] | Rationale for Risk Acceptance[a] | Frequency of Assessment[a] | Impact of not implementing[a] |
|---|---|---|---|---|---|---|---|---|
| CM-8(a)(1) | DSM | ISCM-TN | ISCM-Sys | Test | | | | |
| CM-8(a)(2) | ISCM-Sys | ISCM-TN | ISCM-Sys | Test | | | | |
| CM-8(a)(3) | ISCM-Sys | ISCM-TN | ISCM-Sys | Test | | | | |
| CM-8(b)(1) | DM | ISCM-TN | ISCM-Sys | Test | | | | |
| CM-8(b)(2) | DSM | ISCM-TN | ISCM-Sys | Test | | | | |
| CM-8(4)(1) | DSM | ISCM-TN | ISCM-Sys | Test | | | | |
| PS-4(d)(1) | DM | ISCM-TN | ISCM-Sys | Test | | | | |
| SC-15(a)(1) | DM | ISCM-TN | ISCM-Sys | Test | | | | |
| SC-15(b)(1) | MAN | ISCM-TN | ISCM-Sys | TBD | | | | |

[a] To be completed by the organization. Note that this table is an example; the authoritative tables for control allocations are in the appropriate volumes.

# 7. Root Cause Analysis

Responding to defect checks is done using the normal risk management responses defined in SP 800-39. In general, under an ISCM program, responsibility for risk response belongs to the owning organization.

## 7.1 *Knowing Who Is Responsible*

For the agency dashboard to generate effective prioritized to-do lists to assign responsibility for defects, the dashboard must have a way to:

- Identify the person to assign each defect (maintained as part of the desired state specification); and
- Compute risk.

Because defect checks could be symptoms of one or more controls failing, the response is likely to include some amount of root cause analysis to find the source of the defect.

## 7.2 *Root Cause Analysis*

As noted above, root cause analysis is often needed to determine which control or control item has failed when a defect is found within a capability.

Root cause analysis operates on the logical flow of cause to effect from control items to the security result that is the objective of a security capability (Figure 8: Flow of Cause and Effect from Control Items to Security Results). The desired security result is to make attack scenarios and/or exploits more difficult to conduct by reducing the number of defects that can be exploited and the likelihood that defects will be exploited. Desired security results will be identified for each capability in the subsequent volumes of this NISTIR.

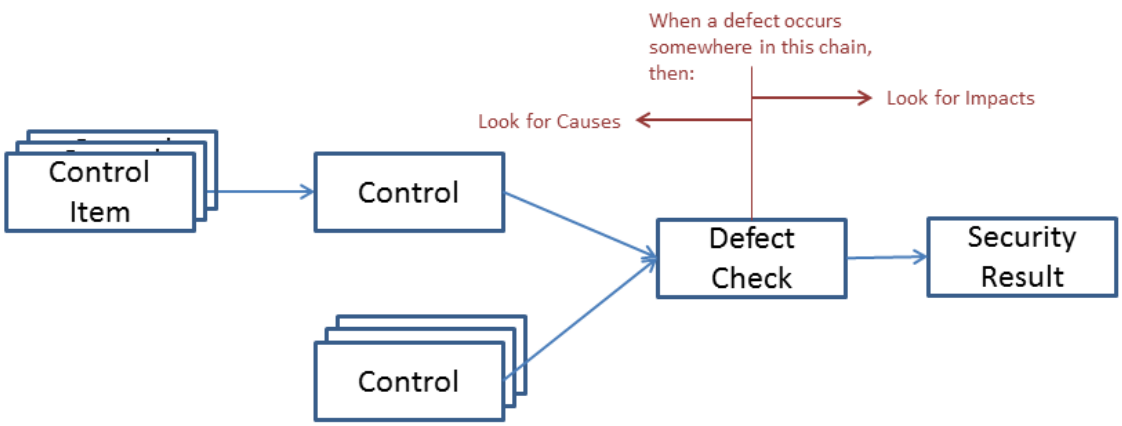


**Figure 8: Flow of Cause and Effect from Control Items to Security Results**

A defect might be noticed at the control item, the whole control, a defect check, and/or at the ultimate result level. Root cause analysis includes:

- Looking back toward the control items to see which failures may have caused the defect; and
- Looking forward to see the impact (positive or negative) on the desired security result.

The second step should not be ignored because, by looking forward, one might find the failure is *not* compromising the desired security result, or that the failure is *not* having a significant negative impact on the security result. The information discovered from root cause analysis is used to prioritize efforts to fix malfunctioning controls or to help determine if the risk from a particular control malfunction can be accepted.

### 7.2.1 Root Cause Analysis How-to: Controls

When a particular control or control item is found to be failing, it is important to consider why. In some cases, the reason may be obvious, and it may be appropriate to simply fix the individual defect. In other cases, the root cause may be more subtle.

Clearly, if a needed patch has not been applied or a configuration setting is incorrect, one can usually reduce the risk immediately by applying the patch or adjusting the setting. However, if such problems consistently recur, it is advisable to look deeper. One key factor to look for in this kind of root cause analysis is whether there is a systemic problem causing the recurring defects.

In this case, it is useful to think about the expected life cycle of control implementation to see whether a defect from early in the life cycle (i.e., an engineering defect) is causing the problem. Questions that can help with this analysis include (but are not limited to) the following:

- Was the capability or control functionality supporting the capability added at the end of the system life cycle, so that too little preparation and planning was done or security functionality is not yet optimal?
- Has sound policy been established to guide control implementation and management?
- Were requirements appropriately defined?
- Is responsibility for avoiding and fixing defects clearly defined?
- Is the defect something that occurs in the space between systems, where it may be overlooked by both systems?
- Are users behaving in ways that inhibit or decrease security (e.g., not following policies and/or procedures), and what can be done to change their behavior?
- Can operators easily get the information they need to avoid problems? (For example, in Active Directory, it is difficult to know what privileges are inherited by a user from parent groups.)
- Was control implementation automated (e.g., automated centralized patch management)? Is the automation working?

- For manually implemented and managed controls, does staff have the necessary resources, training, and tools?
- Were appropriate tools and methods used to implement the control?
- Did planning for implementation ensure that adequate funds, staff, and other resources were provided for implementation?
- Are operational staff members tasked to do so many things for security by policy that they are overwhelmed?
- Was the control implementation adequately tested?
- Other?

Finding issues like these in an organization, especially if the issues span across multiple systems, can be an important function for either the organization or auditors. Such findings are orders of magnitude more important than a list of specific defects from a red team exercise or single system assessment. While this analysis is more difficult than just reporting individual control defects, finding and resolving these systemic problems can have a much more profound effect in improving security programs than fixing miscellaneous controls.

### 7.2.2 Root Cause Analysis How-to: Defect Types

Three levels of root cause analysis are needed for defect check failures:

(1) Determine case-specific causes.

(2) Determine which control failed.

(3) Determine systemic causes.

**LEVEL 1: Determine the case-specific causes**. This typically involves affirming whether the desired specification or the actual state is in error.

a. Was the desired state specification wrong?

b. Was the actual state wrong?

In coordination with the information system owner (ISO) and ISSO, designated operational staff looks at each specific case to decide whether option (a) or (b) applies to the defect. It is equally important to consider what caused (a) or (b) to be the defect.

**Example 1:** Perhaps a system administrator has connected multiple devices to the production network without first adding them to the authorized inventory, configuring them correctly, and patching them. Determining that this is the root cause indicates that option (b), actual state error, is the issue because the actual state (unpatched, misconfigured devices on the network and not in the inventory) is the defect. In this case, the solution is not just to get the devices authorized, configured, and patched, but also to make sure the system administrator understands the importance of following operational procedures.

For Example 1, note that the failure includes one or more of the controls/control items related to managing the actual state.

**Example 2:** Perhaps a system administrator has connected multiple devices to the production network after getting them authorized and correctly configuring and patching them. However, the administrator forgot to put them in the authorized information system component inventory first. Determining that this is the root cause indicates that option (a), desired state specification error, is the issue because the desired state specification (failure to include a correctly authorized device in the inventory) is the defect. In this case, the solution is just to enter the devices into the inventory and make sure that the system administrator understands the need to add authorized devices to the information system component inventory before putting them on the network.

For Example 2, note that the failure includes one or more of the controls/control items related to managing the desired state specification.

In summary, the determination of whether (a) or (b) is the cause also helps clarify which control items failed: those related to desired state specification or to actual state. Within those groups, additional analysis may be needed to determine if specific control items are failing.

**LEVEL 2: Identify which control(s) failed**. Use the Control-to-Defect Check Mapping tables that map specific defect checks to specific control items that might be causing the defect check to fail. The tables may provide more resolution, as the various control items that might cause the defect check failure are more detailed and thus more useful for analysis. A mapping table is included in each capability-specific volume. The mapping tables notionally look like Table 19: Notional Way to Look up Controls Tested by a Defect Check.

**Table 19: Notional Way to Look up Controls Tested by a Defect Check**

**Supporting Control Items:** The sub-capability assessed by this defect check is supported by each of the following control items. Thus, if any of the supporting controls fail, the defect check assessing the sub-capability will fail. Thus, the defect check also, indirectly, tests the control items.

| Defect Check ID | Baseline | Sortable Control Item Code | NIST Control Item Code |
|---|---|---|---|
| HWAM-F01 | Low | AC-19-b | AC-19(b) |
| HWAM-F01 | Low | CM-08-a | CM-8(a) |
| HWAM-F01 | Low | CM-08-b | CM-8(b) |
| HWAM-F01 | Moderate | AC-20-z-02-z | AC-20(2) |
| HWAM-F01 | Moderate | CM-03-b | CM-3(b) |
| HWAM-F01 | Moderate | CM-03-c | CM-3(c) |
| HWAM-F01 | High | CM-03-z-01-a | CM-3(1)(a) |
| HWAM-F01 | High | CM-03-z-01-b | CM-3(1)(b) |
| HWAM-F01 | High | CM-03-z-01-d | CM-3(1)(d) |

This example does not include all controls that might cause this defect check to fail. See the corresponding capability volume for the complete list.

These tables of supporting control items, in their entirety, are found in each capability volume, in a section numbered 3.2, called Sub-Capabilities and Defect Check Tables and Templates. Each defect check there contains a table called Supporting Control Items.

In this case, the root cause analyst determines whether or not all of the implemented security controls related to the defect check are operating as intended. If they are not, repairs/changes may need to be made by control implementers, or a risk acceptance decision can be made by the authorizing official (with appropriate justification).

Note that once failing controls are identified, additional (root cause) analysis is conducted, as described in Section 7.2, Root Cause Analysis, to determine why they are failing.

**LEVEL 3:** Systemic analysis: The systemic analysis looks for causes of repeated failures or engineering defects and seeks to find systemic solutions. In Example 1 for Level 1 above, the defect(s) in question may have occurred repeatedly because the system administrator:

- Has no way to properly configure and patch the devices until they are on the production network,
- Lacks the training to know how to prepare devices before putting them on the production network;
- Has too much to do and is cutting corners to keep up with assigned workload;
- Is unaware of the operational procedures; and/or
- Other possible causes.

As noted above, conducting root cause analysis to determine whether there are underlying systemic defects and finding those causes may be more relevant than focusing on individual defects.

Once the causes are identified, the impacts are also considered. The question is: How important is a specific failure in the context of the overall organization and its risk tolerance? For example, consider the three cases in Table 20: Impact Scenarios/Impact Analysis, of a failure to assign a manager to a device on the network.

**Table 20: Impact Scenarios/Impact Analysis**

| Case | Example Scenario: | Example Impact Analysis |
|---|---|---|
| A | No device manager is specifically designated, and, though someone is carefully managing the devices, the person forgot to record the device in the information system component inventory. | Relatively low risk short-term because the device is actually being managed, but the lack of a designated device manager should be addressed so that the responsible person receives and responds to relevant defect lists going forward. |

| Case | Example Scenario: | Example Impact Analysis |
|---|---|---|
| B | A device was put on the production network for test purposes, so it was not added to the information system component inventory. The device has become vulnerable over time due to lack of patching and configuration management, and downstream objects can be attacked through it. | Relatively high risk that will likely increase. In addition to removing the device from the network, attention needs to be given to device manager training to prevent such behavior in the future. |
| C | There was a need to rapidly expand the network for disaster response purposes, and management accepted the risk for (for example) 10 weeks of putting unauthorized and higher-risk devices in a segment of the network without prior authorization to address this need. Authorization and other cleanup are to occur before the 10 weeks have elapsed. | Moderate to high risk. The fact that risk was accepted by the appropriate management official indicates that no systemic problem occurred. Perhaps, however, the organization could find a way to better prepare for such incidents to avoid needing to accept such risk in the future. |

Because the automated security control assessment system typically identifies defects at the defect check level, the ability to identify both root causes and the impacts from defect check failures, as described above, is an essential activity. When significant systemic conclusions are reached, it may imply the need for new desired state specifications in supporting areas (e.g., training of system administrators in a specific skill). Policy changes and related defect checks for the new desired state specifications should then be established.

# 8. Roles and Responsibilities

The purpose of this NISTIR is to provide an operational approach for implementing automated security control assessments. Here, operational roles and responsibilities are defined, in addition to managerial responsibilities.

## 8.1 *NIST-Defined Management Responsibilities*

NIST-defined information security management roles and responsibilities indicate those who have the ultimate responsibility and authority to oversee the security of an information system and ensure that security requirements as documented in the system security plan are met. Responsibility for the operational task of actually finding and responding to defects on the system is not specified, but typically those performing operational roles report to the management-level roles specified in NIST guidance.

NIST guidance assigns the management responsibility to discover and respond to security defects for a system to the ISO and the ISSO. This is illustrated by the language quoted in Table 21: ISO and ISSO Responsibilities, from SP 800-37, Appendix D.

**Table 21: ISO and ISSO Responsibilities**

| Role | Responsibilities |
|---|---|
| Information System Owner (ISO) | The *information system owner* is an organizational official responsible for the procurement, development, integration, modification, operation, maintenance, and disposal of an information system. The ISO is responsible for addressing the operational interests of the user community (i.e., users who require access to the information system to satisfy mission, business, or operational requirements) and for ensuring compliance with information security requirements. |
| Information System Security Officer (ISSO) | The *information system security officer* is an individual responsible for ensuring that the appropriate operational security posture is maintained for an information system and as such, works in close collaboration with the information system owner. The ISSO also serves as a principal advisor on all matters, technical and otherwise, involving the security of an information system. |

It is unlikely that the ISO or the ISSO will connect devices to the network, install software, set configuration values, and patch software as part of their daily duties. Yet these are daily operational tasks by which most endpoint security defects are managed. Thus, while they have overall management responsibility for the system and its security posture, the ISO and ISSO roles can be supplemented by more detailed operational roles as needed in order to execute day-to-day information security tasks.

## 8.2 *ISCM Operational Responsibilities*

ISCM operational roles and responsibilities are illustrative operational roles for completing tasks that those with managerial roles would typically delegate to others (see Table 22: Notional Example of ISCM Operational Roles for HWAM). Depending on the size and complexity of the system, the operational roles may be full-time positions or the tasks may be performed along with other duties. While each organization might define these operational roles in different ways, the goal is to ensure that operational duties are assigned to roles and then to individuals or teams with enough capacity to perform the role. Thus, the roles defined here are examples to help implement ongoing assessment and response and to maintain the desired system security posture. Organizations have great flexibility in how to designate these roles. For example, organizations may want to subdivide these roles, rename and/or combine them to reflect local practice. The appropriate allocation will likely vary significantly between large and small organizations.

**Table 22: Notional Example of ISCM Operational Roles for HWAM**

| Role Code | Role Title | Role Description | Role Type |
|---|---|---|---|
| DM | Device Manager (DM) | Assigned to a specific device or group of devices, device managers are (for HWAM) responsible for adding/removing devices from the network, and for configuring the hardware of each device (adding and removing hardware components). The device managers are specified in the desired state inventory specification. The device manager may be a person or a group. If a group, there is a group manager in charge. | Operational |
| DSM | Desired State Managers and Authorizers (DSM) | Desired State Managers are needed for both the ISCM Target Network and each object. The desired state managers ensure that data specifying the desired state of the relevant capability is entered into the ISCM system's desired state data and is available to guide the actual state collection subsystem and to identify defects. The DSM for the ISCM Target Network also resolves any ambiguity about which information system authorization boundary has defects (if any).<br><br>Authorizers share some of these responsibilities by authorizing specific items (e.g., devices, software products, or settings) and thus defining the desired state. The desired state manager oversees and organizes this activity. | Operational |

Note that for the purpose of this example, not all roles are shown. See the relevant capability volume for the complete list of roles.

A primary output of ISCM is a list of defects for which a response is needed. The defect lists are targeted at predetermined operational roles and/or teams and thus reflect just the defects for which that role and/or team is responsible. If the defect lists are not targeted at specific roles and/or teams, defect response actions may not be appropriately allocated or taken on a day-to-day basis. To address this, the ISCM dashboard hierarchy can be configured to efficiently allocate response actions to the appropriate roles/teams given the correct operational role information.

The operational roles help describe which individual or team is assigned to respond to specific defect types. As such, the defect tables list the role responsible for coordinating response to each defect. Potential response actions are suggested in the defect tables but may require the input or approval of the ISO and/or ISSO. Additionally, if risk is to be accepted, approval of the authorizing official is required.

Finally, some of the operational roles address defects that cannot be assigned to a specific system. For example, if unauthorized devices are detected on the ISCM Target Network, their system assignment will be unknown. A specific role is thus defined at the network level to manage unassigned defects.

The operational roles are supplementary to those defined in SP 800-37. However, the additional detail is provided with each capability to clarify how to operationalize automated security control assessments. Each organization also has the flexibility to decide to which of the NIST-defined management roles those performing the operational roles report.

# 9. Relationship of Automated Security Control Assessment to the NIST Risk Management Framework

Now that the automated security control assessment process has been defined, it is important to show how the process maps to the equivalent Risk Management Framework (RMF) Step 4 (Assessment) tasks from SP 800-37, and to document how the ISCM-specific processes can be leveraged to produce the required RMF documentation.

> Note that although the term *documentation* is used, there is no requirement that the various documents be printed or that they be narrative documents. In fact, it may be possible to observe many of the required documents directly in ISCM dashboards.
>
> It is valuable to keep trend data at appropriate levels of aggregation. However, there is no requirement to keep detailed (object-level) assessment results from each day, unless the organization finds a good risk management reason to keep historical data. In general, having current detailed assessment results and summary trend data is adequate.

## 9.1 *Linking ISCM to Specific RMF Assessment Tasks*

The following sections relate to RMF Tasks 4-1 through 4-4, as defined in SP 800-37, and they explain how automated ISCM outputs can be used to produce more timely documentation.

**TASK 4-1: Develop, review, and approve a plan to assess the security controls.**

The capability-specific volumes in this NISTIR provide a template for developing and reviewing the required security assessment plan. They do not play a role in the approval of the plan to assess security controls because that is an organizational responsibility.

The security assessment plan template is expressed first in the control narrative for each control, as shown in the example in Figure 7: Example of a Security Assessment Plan Narrative, and then supplemented by the defect check tables as shown in Table 14: Sample Rows from a Hypothetical Sub-Capability and Defect Check.

The volumes on each capability provide a security assessment plan narrative for each applicable control. Organizations may use this narrative as is, customize it, and/or develop their own. Examples of areas where organizations may customize the narratives include (but are not limited to) the following:

- Use of organization-specific names for the roles and responsibilities in the narrative;
- Clarification of the scope of the ISCM Target Network(s); and/or
- Conduct of additional types of assessments.

Together, the defect check tables and the security assessment plan narratives constitute documentation of the security assessment plan for controls and control items within the scope of ISCM automated security control assessment capabilities, and are in accordance with SP 800-37

Task 4-1 guidance. The control narratives are summarized in the control allocation tables for each baseline, described in Section 6.7, Control Allocation Tables. Note that when controls and control items are assessed using manual procedural methods, the security assessment plan is also documented in accordance with SP 800-37 Task 4-1 guidance.

### TASK 4-2: Assess the security controls in accordance with the assessment procedures defined in the security assessment plan.

The control allocation tables include a column for diagnostic responsibility (see Table 18: Notional Control Allocation Table – Example). Where this is assigned to ISCM Check, the ISCM program automates the defect checks specified. Where diagnostic responsibility is not assigned to ISCM Check, it is assigned to organizational staff for manual procedural assessment. Refer to the control allocation tables in each capability-specific volume of this NISTIR for details.

### TASK 4-3: Prepare the security assessment report documenting the issues, findings, and recommendations from the security control assessment.

The agency dashboard provides the required documentation of the assessment findings, if properly configured by the organization. This configuration includes grouping the assessed objects by authorization boundary and also by inherited common controls.

Security assessment report information includes:

- Detailed lists of defects by system, responsible party, device, etc.;
- Detailed lists of which defects contribute the most overall risk;
- Federal- and organization-defined prioritization of which defects to address first;
- Summary levels of risk by capability, mitigation manager, system, etc.; and
- Estimates of the consequences of the given level of risk, to facilitate risk management decisions, investment decisions, etc.

The security assessment report information generated by the agency dashboard is acceptable whether it is printed on paper or presented electronically. As with the security assessment plan from Task 4-1, security assessment reporting for controls and control items assessed using manual/procedural methods is also documented in accordance with Task 4-3 guidance.

### TASK 4-4: Conduct initial remediation actions on security controls based on the findings and recommendations of the security assessment report and reassess remediated control(s), as appropriate.

The agency dashboard presents the defect findings in the form of a prioritized to-do list for each person/team responsible for mitigation (remediation). The response action is the responsibility of each authorizing official (for risk acceptance), ISO, ISSO, and the persons (operational roles) designated in the agency dashboard to mitigate risk (e.g., device managers).

Automated assessment tools are often capable of providing a standard of periodic assessment of control effectiveness on a much more frequent basis than has been generally conducted previously, or than is possible with manual/procedural assessments. While organizations retain the flexibility to determine the frequency of defect checks and associated dashboard-based reports, if defects are checked every four (4) days (or more frequently) at least two purposes are served:

- It lets the responsible party know whether the mitigation action was successful; and
- It raises a flag should the defect appear again in the future.

While actual remediation actions are not conducted, ISCM's prioritized to-do lists and frequency of defect checks strongly supports Task 4-4 activities for controls under ISCM assessment.

## Appendix A. References


| POLICIES, DIRECTIVES, INSTRUCTIONS, REGULATIONS, AND MEMORANDA |
|---|

1. Office of Management and Budget Memorandum M-10-28, *Clarifying Cybersecurity Responsibilities and Activities of the Executive Office of the President and the Department of Homeland Security (DHS)*, July 6, 2010. https://www.whitehouse.gov/sites/default/files/omb/assets/memoranda_2010/m10-28.pdf

2. Office of Management and Budget Memorandum M-11-33, *FY 2011 Reporting Instructions for the Federal Information Security Management Act and Agency Privacy Management*, September 14, 2011. https://www.whitehouse.gov/sites/default/files/omb/memoranda/2011/m11-33.pdf

3. Office of Management and Budget Memorandum M-14-03, *Enhancing the Security of Federal Information and Information Systems*, November 18, 2013. https://www.whitehouse.gov/sites/default/files/omb/memoranda/2014/m-14-03.pdf

| STANDARDS |
|---|

1. NIST National Institute of Standards and Technology Federal Information Processing Standards Publication (FIPS) 199, *Standards for Security Categorization of Federal Information and Information Systems*, February 2004. http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

| GUIDELINES AND INTERAGENCY REPORTS |
|---|

1. National Institute of Standards and Technology Special Publication (SP) 800-30 Revision 1, *Guide for Conducting Risk Assessments*, September 2012. http://dx.doi.org/10.6028/NIST.SP.800-30r1

2. National Institute of Standards and Technology Special Publication (SP) 800-37 Revision 1, *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*, February 2010 (updated June 5, 2014). http://dx.doi.org/10.6028/NIST.SP.800-37r1

3. National Institute of Standards and Technology Special Publication (SP) 800-39, *Managing Information Security Risk: Organization, Mission, and Information System View*, March 2011. http://dx.doi.org/10.6028/NIST.SP.800-39

4. National Institute of Standards and Technology Special Publication (SP) 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations*, April 2013 (updated January 22, 2015). http://dx.doi.org/10.6028/NIST.SP.800-53r4

5. National Institute of Standards and Technology Special Publication (SP) 800-53A Revision 4, *Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans*, December 2014 (updated December 18, 2014). http://dx.doi.org/10.6028/NIST.SP.800-53Ar4

6. National Institute of Standards and Technology Special Publication (SP) 800-115, *Technical Guide to Information Security Testing and Assessment*, September 2008. http://dx.doi.org/10.6028/NIST.SP.800-115

7. National Institute of Standards and Technology Special Publication (SP) 800-137, *Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations*, September 2011. http://dx.doi.org/10.6028/NIST.SP.800-137

8. National Institute of Standards and Technology Special Publication (SP) 800-160 (DRAFT), *Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems*, May 2014. http://csrc.nist.gov/publications/PubsSPs.html#800-160

9. National Institute of Standards and Technology Interagency Report (NISTIR) 7298 Revision 2, *Glossary of Key Information Security Terms*, May 2013. http://dx.doi.org/10.6028/NIST.IR.7298r2

10. National Institute of Standards and Technology Interagency Report (NISTIR) 7756 (DRAFT), *CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Architecture*, January 2012. http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7756

# Appendix B. Glossary

| | |
|---|---|
| Actual State | The observable state or behavior of an object (device, software, person, credential, account, etc.) at the point in time when the collector generates security-related information. In particular, the actual state includes the states or behaviors that might indicate the presence of security defects. |
| Anomalous Event Response and Recovery Management | See *Capability, Anomalous Event Response and Recovery Management*. |
| Agency Dashboard | An organizational-level dashboard that: a) collects data from a collection system; and b) shows detailed object-level data and object-level defects to organizationally authorized personnel. |
| Assessment Boundary | The scope of (assessment objects included in) an organization's ISCM implementation to which assessment of objects is applied. Typically, assessment boundary includes an entire network to its outside perimeter. |
| Assessment Completeness | The degree to which the continuous monitoring-generated, security-related information is collected on all objects for all applicable defect checks within a defined period of time. |
| Assessment Criterion/Criteria | A rule (or rules) of logic to allow the automated or manual detection of defects. Typically, the assessment criterion in ISCM defines a) what in the desired state specification b) will be compared c) to what in the actual state and d) the conditions that indicate a defect. |
| Assessment Object | See *Object, Assessment*. |
| Assessment Timeliness | The degree to which the continuous monitoring-generated, security-related information is collected within the specified period of time (or frequency). |
| Asset | Resources of value that an organization possesses or employs. |
| Behavior Management | See *Capability, Behavior Management*. |
| Capability | See *Capability, Security*. |
| Capability, Anomalous Event Detection Management | An ISCM capability that identifies routine and unexpected events that can compromise security within a time frame that prevents or reduces the impact (i.e., consequences) of the events to the extent possible. |

| | |
|---|---|
| Capability, Anomalous Event Response and Recovery Management | An ISCM capability that ensures that both routine and unexpected events that require a response to maintain functionality and security are responded to (once identified) within a time frame that prevents or reduces the impact (i.e., consequences) of the events to the extent possible. |
| Capability, Behavior Management | An ISCM capability that ensures that people are aware of expected security-related behavior and are able to perform their duties to prevent advertent and inadvertent behavior that compromises information. |
| Capability, Boundary Management | An ISCM capability that addresses the following network and physical boundary areas:<br>**Physical Boundaries –** Ensure that movement (of people, media, equipment, etc.) into and out of the physical facility does not compromise security.<br>**Filters –** Ensure that traffic into and out of the network (and thus out of the physical facility protection) does not compromise security. Do the same for enclaves that subdivide the network.<br>**Other –** Ensure that information is protected (with adequate strength) when needed to protect confidentiality and integrity, whether that information is in transit or at rest. |
| Capability, Configuration Settings Management | An ISCM capability that identifies configuration settings (Common Configuration Enumerations [CCEs]) on devices that are likely to be used by attackers to compromise a device and use it as a platform from which to extend compromise to the network. |
| Capability, Credentials and Authentication Management | An ISCM capability that ensures that people have the credentials and authentication methods necessary (and only those necessary) to perform their duties, while limiting access to that which is necessary. |
| Capability, Event Preparation Management | An ISCM capability that ensures that procedures and resources are in place to respond to both routine and unexpected events that can compromise security. The unexpected events include both actual attacks and contingencies (natural disasters) like fires, floods, earthquakes, etc. |
| Capability, Hardware Asset Management | An ISCM capability that identifies unauthorized and unmanaged devices that are likely to be used by attackers as a platform from which to extend compromise of the network to be mitigated. |
| Capability, ISCM | See *ISCM Capability*. |
| Capability, Manage and Assess Risk | An ISCM capability that focuses on reducing the successful exploits of the other non-meta capabilities that occur because the risk management process fails to correctly identify and prioritize actions |

| | |
|---|---|
| | and investments needed to lower the risk profile. |
| Capability, Perform Resilient Systems Engineering | An ISCM capability that<br>• Focuses on reducing successful exploits of the other non-meta capabilities that occur because there was inadequate design, engineering, implementation, testing, and/or other technical issues in implementing and/or monitoring the controls related to those capabilities.<br>• Reducing the successful exploits of the other non-meta capabilities that occur because there was inadequate definition of requirements, policy, planning, and/or other management issues in implementing and/or monitoring the controls related to those capabilities. |
| Capability, Privilege and Account Management | An ISCM capability that ensures that people have the privileges necessary (and only those necessary) to perform their duties, to limit access to that which is necessary. |
| Capability, Security | A set of mutually reinforcing security controls implemented by technical, physical, and procedural means. Such controls are typically selected to achieve a common information security-related purpose. |
| Capability, Software Asset Management | An ISCM capability that identifies unauthorized software on devices that is likely to be used by attackers as a platform from which to extend compromise of the network to be mitigated. |
| Capability, Trust Management | An ISCM capability that ensures that untrustworthy persons are prevented from being trusted with network access (to prevent insider attacks). |
| Capability, Vulnerability Management | An ISCM capability that identifies vulnerabilities [Common Vulnerabilities and Exposures (CVEs)] on devices that are likely to be used by attackers to compromise a device and use it as a platform from which to extend compromise to the network. |
| CDM | See *Continuous Diagnostics and Mitigation*. |
| CMaaS | See *Continuous Monitoring as a Service* |
| Collection System | A CAESARS instance (system) that manages the collectors, collects actual state data, and compares the collector data (actual state) to the desired state specification to find security defects. |
| Collector | Typically, an automated component that gathers actual state data. Part of the collection system. See Section 4.2, Collectors and the Collection System. |

| | |
|---|---|
| Configuration Settings Management | See *Capability, Configuration Settings Management*. |
| Continuous Diagnostics and Mitigation (CDM) | A Congressionally established program to provide adequate, risk-based, and cost-effective cybersecurity assessments and more efficiently allocate cybersecurity resources targeted at federal civilian organizations. |
| Continuous Monitoring as a Service | a. As a general concept, continuous monitoring services provided by a third party (as a service), typically from the cloud.<br>b. With respect to the DHS CDM program, a GSA-procured Blanket Purchasing Agreement to provide strategic sourcing for the CDM program, all federal agencies, and state, local, and tribal governments. |
| Control Item | See *Security Control Item*. |
| Defect | An occurrence of a defect check that failed on an object. It indicates a weakened state of security that increases risk. |
| Defect Check | A **defect check** is a way to assess determination statements. It has the following additional properties. A defect check:<br>• Is stated as a test (wherever appropriate);<br>• Can be automated;<br>• Explicitly defines a particular desired state specification that will then be compared to the corresponding actual state to determine the test result;<br>• Provides information that may help determine the degree of control effectiveness/level of risk that is acceptable;<br>• Suggests risk response options; and<br>• Assesses a corresponding sub-capability. |
| Defect Instance | The term *instance* emphasizes that this term is used to refer to a single occurrence of a defect on an object, and not a type of defect that could occur on many objects. Also see Defect. |
| Defect Type | A kind of defect that could occur on many objects. Generally, a defect check tests for the presence or absence of a defect type. |
| Desired State | See *Desired State Specification*. |
| Desired State Specification | A defined value, list, or rule (specification) that a) states or b) allows the computation of the state that the organization desires in order to reduce information security risk. Desired state specifications are generally statements of policy. |

| | |
|---|---|
| Device | In automated assessment, a type of assessment object that is either an IP addressable (or equivalent) component of a network or a removable component that is of security significance. |
| Device Role | A device role is a group of devices with the same rules. For example, the list of white-listed software for a server is likely different from that for a workstation. This would cause servers and devices to have separate device roles.<br><br>Roles can be federally and/or organization-defined. Examples of high-level roles include user-endpoint, server, networking device, cellular device, and other devices. Each might be further subdivided. For example, servers might be divided into many sub-categories (e.g., database-server, email-server, file-server, DNS-server, DHCP-server, authentication-server). A device role is needed whenever the organization wants a group of devices to have different rules for authorized software, settings, and/or patching, for example. |
| Federal Dashboard | A CAESARS instance that:<br>• Collects summary data from the base-level dashboards across multiple organizations; and<br>• Does not collect defects at the object-level data or defects. It summarizes federal level defects and object categories, but not local (base) level defects or local (base) categories. |
| Federal Risk Scoring Working Group | A federal government-wide team that defines which defect checks will be required for reporting to the federal dashboard and what risk scoring parameters will be used to score the defects found. |
| Foundational Defect Checks | Defect checks that expose ineffectiveness of controls that are fundamental to the purposes of the capability (e.g., HWAM, or SWAM, or Configuration Setting Management) in which the defect check appears. |
| Hardware Asset Management | See *Capability, Hardware Asset Management*. |
| Identifier | Something (data) that identifies an object or other thing of interest (like a defect check). In database terms, it is a primary or candidate key that can be used to uniquely identify the object so it is not confused with other objects. |
| ISCM Capability | A security capability with the following additional traits:<br>• The purpose (desired result) of each capability is to address specific kind(s) of attack scenarios or exploits.<br>• Each capability focuses on attacks towards specific objects.<br>• There is a viable way to automate ISCM on the security |

capability.

- The capability provides protection against current attack scenarios.

| | |
|---|---|
| ISCM Dashboard | A hierarchy of dashboards to facilitate reporting of appropriate security-related information at multiple organizational levels. |
| Limit, Specification | A condition indicating that risk has exceeded acceptable levels and that immediate action is needed to reduce the risk, or the system/object may need to be removed from production (lose authority to operate). |
| Local Defect Checks | Those defect checks that an organization adds to Foundational defect checks based on an assessment of its own needs and risk tolerance. A local defect check supports or strengthens the Foundational defect checks. Agencies might choose not to apply a given local defect check in cases where the supporting controls have not been selected/implemented. |
| Manage Boundaries | See *Capability, Boundary Management*. |
| Manage Credentials and Authentication | See *Capability, Credentials and Authentication Management*. |
| Manage Privileges | See *Capability, Privilege and Account Management*. |
| Object | As used in this NISTIR, anything that can have a material security defect (failed or absent control). Examples of objects include (but are not limited to) devices, software products, software executables, people, credentials, accounts, account-privileges, things to which privileges are granted (including data and physical facilities), etc. Equivalent to assessment object in this NISTIR. |
| Object, Assessment | Assessment objects identify the specific items being assessed and include *specifications*, *mechanisms*, *activities*, and *individuals*. See NIST SP 800-53A Revision 4, page 10. |
| Ongoing Assessment | The continuous evaluation of the effectiveness of security control implementation; it is not separate from ISCM but in fact is a subset of ISCM activities. |
| Prepare for Events | See *Capability, Event Preparation Management*. |
| Regular Expression | A sequence of characters (or words) that forms a search pattern, mainly for use in pattern matching with strings, or string matching. |

| | |
|---|---|
| Risk | A measure of the extent to which an organization is threatened by a potential circumstance or event, and typically a function of the following:<br>a. The adverse impacts that would arise if the circumstance or event occurs; and<br>b. The likelihood of occurrence. Likelihood is influenced by the ease of exploit(s) required and the frequency with which an exploit or like-objects are being attacked at present. |
| Risk (Acceptance) Limit | Judgment about whether risk is acceptable. |
| Risk (ISCM Capability) | See *Capability, Manage and Assess Risk*. |
| Risk Management | See *Capability, Manage and Assess Risk*. |
| Security Capability | See *Capability, Security*. |
| Security Control Item | All or part of a SP 800-53 security control requirement, expressed as a statement for implementation and assessment. Both controls and control enhancements are treated as control items. These are further subdivided if they have a list of parts: a, b, c, etc. |
| Specification Limit | See *Limit, Specification*. |
| Software Asset Management | See *Capability, Software Asset Management*. |
| Sub-Capability | A capability that supports the achievement of a larger capability. In this NISTIR, each defined capability is decomposed into the set of sub-capabilities that are necessary and sufficient to support the purpose of the larger capability. |
| Trust | See *Capability, Trust Management*. |
| Trust Management | See *Capability, Trust Management*. |
| Vulnerability Management | See *Capability, Vulnerability Management*. |

## Appendix C. Acronyms

| | |
|---|---|
| A&A | Assessment and Authorization |
| CAESARS | Continuous Asset Evaluation, Situational Awareness, and Risk Scoring |
| CCE | Common Configuration Enumeration |
| CDM | Continuous Diagnostics and Mitigation |
| ISCM-TN | Information Security Continuous Monitoring Target Network |
| CSM | Configuration Settings Management |
| CVE | Common Vulnerabilities and Exposures |
| CWE | Common Weakness Enumeration |
| DHS | Department of Homeland Security |
| HWAM | Hardware Asset Management |
| ISCM | Information Security Continuous Monitoring |
| ISO | Information System Owner |
| ISSO | Information System Security Officer |
| MOE | Measure of Effectiveness |
| NVD | National Vulnerability Database |
| RMF | Risk Management Framework |
| SWAM | Software Asset Management |
| US-CERT | U.S. Computer Emergency Response Team |
| USGCB | U.S. Government Configuration Baseline |