



NIST Internal Report
NIST IR 8011v1r1 ipd

Testable Controls and Security Capabilities for Continuous Monitoring

Volume 1 — Overview and Methodology

Initial Public Draft



Eduardo Takamura
Jeremy Licata

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8011v1r1.ipd>

**NIST Internal Report
NIST IR 8011v1r1 ipd**

Testable Controls and Security Capabilities for Continuous Monitoring

Volume 1 — Overview and Methodology

Initial Public Draft

Eduardo Takamura
Jeremy Licata
*Security Engineering and Risk Management Group
Computer Security Division
Information Technology Laboratory*

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8011v1r1.ipd>

February 2025



U.S. Department of Commerce
Jeremy Pelter, Acting Secretary of Commerce

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

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

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

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

NIST Technical Series Policies

[Copyright, Use, and Licensing Statements](#)

[NIST Technical Series Publication Identifier Syntax](#)

Publication History

Approved by the NIST Editorial Review Board on YYYY-MM-DD [Will be added to final publication.]

How to Cite this NIST Technical Series Publication

Takamura E, Licata J (2025) Testable Controls and Security Capabilities for Continuous Monitoring: Volume 1 — Overview and Methodology. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Internal Report (IR) NIST IR 8011v1r1 ipd. <https://doi.org/10.6028/NIST.IR.8011v1r1.ipd>

Author ORCID iDs

Eduardo Takamura: 0000-0002-9978-9050

Jeremy Licata: 0000-0001-8793-5471

Public Comment Period

February 20, 2025 – April 4, 2025

Submit Comments

8011comments@list.nist.gov

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

Additional Information

Additional information about this publication is available at <https://nist.gov/rmf>, including related content, potential updates, and document history.

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

Abstract

According to the NIST Risk Management Framework (RMF) methodology, SP 800-53 security and privacy controls are selected, implemented, assessed, and monitored to help achieve security and privacy objectives. Due to the sheer size, complexity, and scope of information technology footprints, the automation of control assessment and monitoring tasks is desired but not easily achieved. IR 8011 provides a method for identifying SP 800-53 controls that can be tested via automated means based on the assessment objectives and potential assessment methods in SP 800-53A. The IR 8011 methodology also includes a process for developing the actual tests for each testable control. This first volume in the IR 8011 multi-volume series introduces foundational concepts — including the concept of security capability for continuous monitoring — and describes each step of the IR 8011 methodology. This revision includes a new section on the envisioned operationalization of IR 8011 for the development and adoption of potential solutions. Subsequent volumes in the IR 8011 series provide a sample set of testable controls and automatable tests for specific security capabilities for continuous monitoring.

Keywords

actual state; assessment; automated control testing; conceptual implementation; control; desired state specification; information security continuous monitoring; security capability; sub-capability test; testable control; assessment method; authorization boundary; automation; boundary; capability; completeness; continuous monitoring; continuous monitoring dashboard; control assessment; control baseline; control item; control testing; dashboard; data quality test; desired state; foundational test; local test; mitigation; ongoing assessment; ongoing authorization, privacy control; root cause analysis; security automation; security control; sensitivity; specificity; test; test boundary; test object; test plan; testable; timeliness.

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.

Document Conventions

The following conventions apply to this publication:

- The terminology used in the IR 8011 series is consistent with NIST Risk Management Framework (RMF) terminology in [SP800-37] and related publications.
- SP 800-53 generally uses the term *information security* or *security* [SP800-53], but both can be used synonymously for the purposes of the IR 8011 series publications.
- The term *continuous monitoring* refers to *information security continuous monitoring* (ISCM). Although both forms are used interchangeably across RMF-supporting publications, the shorter form is used throughout this publication.
- The terms *information security continuous monitoring system (ISCM-Sys)* and *IR 8011 solution* are used interchangeably. Information security continuous monitoring system or simply ISCM-Sys is used when referring to a general IR 8011 solution that automatically tests controls supporting a security capability.
- In the IR 8011 nomenclature, there is no distinction between *security control* or *privacy control*.
- Early editions of the IR 8011 volumes did not implement leading zeros¹ for the SP 800-53 and SP 800-53A control identifiers. Starting with this revision of Volume 1, control references have been updated to include the leading zero. IR 8011 volumes that have been published but are yet to be revised² will be updated with the leading zeros in the identifiers. The control identifiers used in this revised volume are presented as they are identified in the NIST Cybersecurity and Privacy Reference Tool (CPRT)³ [CPRT] reference datasets for SP 800-53 and SP 800-53A.
- The term *testing* is a shortened form of *testing via automated means*, referring to the use of automation as opposed to manual, visual, procedural or other non-automated testing.
- The term *user* may refer to an individual, a process, or a device acting on behalf of an individual, whichever is most appropriate for the context. In most contexts, the user is an individual authorized to access a resource.
- Building on the concept of security and privacy capabilities⁴ identified in [SP800-53A], the IR 8011 methodology applies the term *capability* specifically to the continuous monitoring of groups of controls that work together in support of a capable defense. While the preferred term is *security capability for continuous monitoring* to preserve the

¹ Leading zeros were first added to control and control item representations in SP 800-53Ar5 (Revision 5) and SP 800-53r5 Update 5.1.1 to address an issue with sorting.

² The IR 8011 volumes include [IR8011v2], [IR8011v3], and [IR8011v4].

³ The CPRT highlights the reference data from NIST publications without the constraints of PDF files, enabling users to interactively browse, search, and export the data in a structured format that is human- and machine-consumable.

⁴ Per [SP800-53A], a security capability or privacy capability is “a combination of mutually reinforcing security controls and privacy controls (i.e., safeguards and countermeasures) implemented by technical means (i.e., functionality in hardware, software, and firmware), physical means (i.e., physical devices and protective measures), and procedural means (i.e., procedures performed by individuals).”

notion that security capabilities are tied to continuous monitoring, a shorter form enables efficient reading.

- A security capability for continuous monitoring is not analogous to the concept of *security automation domain*, which is defined as “information security areas that include a grouping of tools, technologies, and data” [SP800-137]. A security capability focuses on grouping testable controls with a common defense purpose to support both the assessment and monitoring of controls rather than focusing on the tools, technologies, or data for monitoring controls.
- The term *conformance* is used to express adherence to, compliance with, or alignment with a requirement, specification, or expectation. It indicates that a condition is accepted within one or more predefined parameters. If the condition does not exist or does not meet acceptable levels, then the term *non-conformance* is used. *Non-conformance* in IR 8011 testing is similar to the meaning of the term *other than satisfied* in the context of NIST RMF control assessments as used in [SP800-53A]. It indicates that the variance or deviation from a specified parameter is not acceptable. The determination of *conformance* can be achieved via full automation or through a hybrid, semi-automated process in which the non-conformance is automatically detected by a system and procedurally verified by system or authorized personnel.
- Control assessment/monitoring can refer to both the assessment/monitoring of controls or the assessment/monitoring of control implementation.

Assumptions

This report assumes that the reader has a working knowledge of the NIST RMF as described in [SP800-37] and the concepts covered by the following technical publications: [SP800-30], [SP800-39], [SP800-53], [SP800-53A], [SP800-53B], and [SP800-137]. Appendix C provides a listing of RMF-supporting publications that are used as a basis for the IR 8011 methodology.

The IR 8011 methodology assumes that systems are authorized to operate⁵ within the organization and that system artifacts⁶ exist. The control testing process described in this volume is designed to be executed after the initial assessment and authorization⁷ of a system are completed. The focus is on the control tests that support the continuous monitoring process for systems in the *Monitor* step of the RMF or in the operations and maintenance phases of their system development life cycle.

⁵ This is consistent with guidance in [SP800-37].

⁶ Including system plans, control assessment reports, and supporting system documents (e.g., risk assessment plans, system contingency plans, system incident response plans).

⁷ See [SP800-37] for more information on assessments and authorizations.

Note to Reviewers

This major revision to Volume 1 of the IR 8011 series is intended to facilitate understanding of the IR 8011 methodology and expand discussions about its potential implementation. While most of this publication was rewritten to achieve this purpose and to focus on continuous monitoring, the original methodology and foundational concepts have been preserved. No changes to the original methodology and foundational concepts were made other than updates to terminology. The testable controls identified by the capability-specific volumes in the series are only a sample set of controls. The testable controls for a given security capability are not limited to the ones identified by NIST. Finally, the information provided here and throughout the series represents a blueprint for implementation: it does not prescribe *how* to implement the IR 8011 methodology, only *what* to consider when implementing it. A new section on the conceptual implementation of the IR 8011 methodology and considerations (Sec. 4) was added to better support implementers. It describes a conceptual IR 8011 solution⁸ and its basic integrated components to illustrate the automated testing process.

NIST welcomes feedback on this draft revision of IR 8011, Volume 1. Readers may find it beneficial to review NIST Cybersecurity Whitepaper (CSWP) 30, *Automation Support for Control Assessments: Project Update and Vision* [CSWP30], to understand the focus of this revision and the direction of the IR 8011 project. NIST is specifically seeking feedback on the technical accuracy, the ease of navigating and understanding the material, and the added, modified, and removed content.

This public comment period is intended to provide interested parties with an opportunity to contribute to the quality of the material before it is finalized. As always, NIST welcomes comments on this and other IR 8011 publications at any time. Comments can be prepared using the template posted at <https://csrc.nist.gov/pubs/ir/8011/v1/r1/ipd>; send them to 8011comments@list.nist.gov.

⁸ The solution refers to a collection system to support automated data collection and analysis.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

- a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or
- b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:
 - i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
 - ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to 8011comments@list.nist.gov with the subject "IR 8011 call for patent claims."

151	Table of Contents	
152	Executive Summary	11
153	1. IR 8011 Overview	12
154	1.1. Background	12
155	1.2. Purpose and Scope.....	13
156	1.3. Target Audience	14
157	1.4. IR 8011 Series Organization	16
158	2. IR 8011 Fundamentals	18
159	2.1. How to Use IR 8011.....	18
160	2.2. When to Use IR 8011	18
161	2.3. Foundational Concepts	18
162	2.3.1. Boundaries.....	18
163	2.3.2. Security Capabilities for Continuous Monitoring	20
164	2.3.3. Sub-Capabilities	27
165	2.3.4. Adversarial Attack Step Model	29
166	2.3.4.1. Attack Step.....	30
167	2.3.4.2. Defend Step	30
168	2.3.5. Test Automation in IR 8011.....	37
169	2.3.5.1. Actual State.....	37
170	2.3.5.2. Desired State Specification	37
171	3. IR 8011 Methodology	42
172	3.1. Objective #1: Sub-Capability Test Development.....	42
173	3.1.1. Identify Attack Steps	42
174	3.1.2. Identify Defend Steps	44
175	3.1.3. Determine Sub-Capabilities.....	45
176	3.1.4. Identify Control Items.....	45
177	3.1.5. Identify Determination Statements	46
178	3.1.6. Create Sub-Capability Tests.....	49
179	3.1.6.1. Sub-Capability Test Types	49
180	3.1.6.2. Data Quality Measures	50
181	3.1.6.3. Sub-Capability Test Creation.....	52
182	3.1.6.4. Sub-Capability Test Non-Conformance.....	54
183	3.1.6.5. Root Cause Analysis	56
184	3.2. Objective #2: Capability Control Identification.....	65
185	3.2.1. Identify Testable Controls	65

186	3.2.2. Group Testable Controls.....	67
187	3.3. Methodology Summary	67
188	4. Conceptual IR 8011 Implementation and Considerations	69
189	4.1. IR 8011 Solution Developer’s Perspective	76
190	4.1.1. Build a Custom IR 8011 Solution	76
191	4.1.1.1. Design for Automated Control Testing.....	76
192	4.1.1.2. Determine Necessary Data Sources.....	79
193	4.1.1.3. Define Data Relationships.....	80
194	4.1.1.4. Define Solution Functionalities.....	81
195	4.1.1.5. Analysis and Reporting	84
196	4.1.1.6. Develop, Use, and Maintain an IR 8011 Database.....	85
197	4.1.1.7. Control Search (via Keywords).....	86
198	4.1.2. Integrate IR 8011 Sub-Capability Tests into Existing Solutions	88
199	4.1.3. Derive Sub-Capability Tests Outside of IR 8011 Scope.....	89
200	4.1.4. Control Testing as a Service.....	89
201	4.2. IR 8011 Solution Adopter’s Perspective.....	90
202	4.2.1. Roles and Responsibilities	91
203	4.2.2. Buy or Build Considerations	93
204	4.2.3. Support for Internal Automated Control Testing.....	94
205	4.2.4. Support for External Independent Automated Control Testing.....	95
206	4.2.5. Integration Into Existing Continuous Monitoring Programs	96
207	4.3. Understanding Limitations to IR 8011 Operationalization	96
208	4.4. Implementation Validation	96
209	References.....	98
210	Appendix A. Glossary	100
211	Appendix B. List of Abbreviations and Acronyms	105
212	Appendix C. NIST RMF-Related Publications and Their Relationships to IR 8011.....	108
213	Appendix D. Benefits of Breaking Down Security Capabilities Into Elements.....	109
214	D.1. Supports the Strong Systems Engineering of Security Capabilities.....	109
215	D.2. Supports Guidance for Control Selection	109
216	D.3. Simplifies Understanding of the Overall Protection Process.....	110
217	D.4. Enables Testing of Control Outcomes at a Higher Level Than Individual Controls	110
218	D.5. Improves Risk Management by Measuring Control Outcomes.....	111
219	D.6. Helps Organizations Address Organizational, Mission, and Business Risks	111
220	Appendix E. Considerations for IR 8011 Implementation Validation	112

221	Appendix F. Change Log	114
222	List of Tables	
223	Table 1. Potential assessment methods [SP800-53A].....	12
224	Table 2. Ring 1: Manage/Assess Risk.....	22
225	Table 3. Ring 2: Perform Resilient Systems Engineering.....	22
226	Table 4. Ring 3: Manage Operational State	23
227	Table 5. Ring 4: Manage Events	26
228	Table 6. Example of a single control item that supports multiple security capabilities	27
229	Table 7. Select examples of sub-capabilities (HWAM)	28
230	Table 8. Six steps in the IR 8011 attack step model.....	30
231	Table 9. Attack and defend actions for each attack step	31
232	Table 10. Security capabilities work together to defend against attack steps	36
233	Table 11. Types of desired state specifications	38
234	Table 12. Example of equivalence of prohibited and desired state specification.....	39
235	Table 13. Attack step and attack step actions addressed by the HWAM security capability	43
236	Table 14. HWAM attack step and defend step actions.....	44
237	Table 15. Illustrative keyword rules to trace control items to security capabilities	46
238	Table 16. Illustrative control items traced to the HWAM capability	46
239	Table 17. Examples of determination statements highlighting IR 8011 HWAM focus.....	47
240	Table 18. Illustrative control items traced to an associated assessment objective	48
241	Table 19. Example control and determination statements for AC-19, Access Control for Mobile Devices	
242	48
243	Table 20. Example data quality measures	51
244	Table 21. Sub-capability test creation process activities	52
245	Table 22. Sub-capability statement and purpose example.....	53
246	Table 23. Example sub-capability test (from HWAM).....	53
247	Table 24. Relationships between sub-capability tests and determination statements	54
248	Table 25. Example of potential actions for non-conformance response and responsibility assignments	
249	(HWAM)	55
250	Table 26. Sensitivity and specificity notes	56
251	Table 27. Mapping of sub-capability tests to specific control items.....	60
252	Table 28. Example impact scenarios and analyses	61
253	Table 29. HWAM example of documented roles assigned to respond to non-conformances.....	62
254	Table 30. Example of a documented sub-capability test rationale.....	64

255	Table 31. Sub-capability test development workflow output summary.....	65
256	Table 32. Tracing control items to controls/control enhancements.....	66
257	Table 33. Tracing controls to control baselines.....	66
258	Table 34. Example of tracing control items to sub-capabilities (HWAM).....	66
259	Table 35. Data Relationship Between IR 8011 Elements	81
260	Table 36. System owner and security or privacy officer responsibilities	91
261	Table 37. Example of continuous monitoring operational roles for the HWAM security capability	92
262	Table 38. Implementation example of HWAM-F01-Test	94
263	Table 39. NIST RMF-related publications and their relationships to IR 8011	108
264	Table 40. Sample considerations for validating the operationalization of IR 8011	112
265	List of Figures	
266	Fig. 1. IR 8011 methodology elements	13
267	Fig. 2. IR 8011 methodology objectives: Sub-capability test development and capability control	
268	identification	14
269	Fig. 3. Security capabilities for continuous monitoring wheel	21
270	Fig. 4. Overview of an automated control testing process in support of continuous monitoring	40
271	Fig. 5. Sub-capability test development workflow elements	42
272	Fig. 6. Flow of cause and effect from control items to sub-capability test results.....	57
273	Fig. 7. Possible implementation paths.....	70
274	Fig. 8. Conceptual collection system	72
275	Fig. 9. Simplified view of automated testing	74
276	Fig. 10. Solution development elements	77
277	Fig. 11. Control testing process overview	78
278	Fig. 12. Sample data relationships	80
279	Fig. 13. Sample capability narrative in data.....	82
280	Fig. 14. Sub-capability and sub-capability test description	83
281	Fig. 15. Sample analysis logic.....	84
282	Fig. 16. Solution adoption.....	91
283		

284 **Acknowledgments**

285 The authors acknowledge and appreciate the significant contributions of the original authors of
286 this publication, namely Kelley Dempsey, Paul Eavy, and George Moore, who helped establish
287 the foundation for the IR 8011 methodology. Special thanks to Victoria Pillitteri for the
288 comprehensive review and guidance that shaped the direction of this publication and resulted
289 in improved readability. The authors also wish to thank all past collaborators for their historical
290 contributions to this volume, as well as Jim Foti for layout, formatting, styling guidance, and
291 editorial support; Isabel Van Wyk for copy editing this publication; and all of the individuals and
292 organizations who have provided comments and feedback on the IR 8011 series. Their
293 combined input offered insights on some of the improvements that are being applied across
294 the entire IR 8011 series.

295 **Executive Summary**

296 This NIST Internal Report (IR) provides a methodology for using automation to test the
297 implementation of NIST Special Publication (SP) 800-53 security and privacy controls in support
298 of security capabilities for continuous monitoring. Security capabilities represent foundational
299 defense capabilities against potential cyber attacks to systems and organizations. The IR 8011
300 methodology explores these security capabilities using an attack step model for the purpose of
301 identifying more detailed and specific capabilities, called *sub-capabilities*, that can be tested.
302 Subsequent volumes in this multi-volume series, published separately, present a sample
303 collection of sub-capabilities, sub-capability tests, and associated controls that support specific
304 security capabilities, one capability per volume. All volumes in the IR 8011 series provide a
305 blueprint for the development and adoption of a potential IR 8011 solution.

306 The IR 8011 methodology was designed to be used with the NIST Risk Management Framework
307 (RMF), specifically in support of the *Assess* and *Monitor* steps. Each control in the SP 800-53
308 control catalog has an associated assessment procedure in SP 800-53A, which is leveraged for
309 the development of sub-capability tests. It is possible to apply the IR 8011 methodology using
310 controls other than those from the SP 800-53 catalog following a different framework or
311 methodology as long as controls have associated assessment procedures. The assessment
312 procedures provide the necessary determination statements to support the development of
313 sub-capabilities tests.

314 This major revision to IR 8011, Volume 1, preserves the original methodology first introduced in
315 2017 and focuses on improving the way in which the methodology is presented to facilitate
316 understanding of the foundational concepts and the purpose of the methodology. Key terms
317 and visual aids, such as diagrams and other graphics, were updated to better describe IR 8011
318 processes and their elements. A dedicated section on an envisioned IR 8011 operationalization
319 has been added to provide conceptual implementation examples and considerations for IR
320 8011 solution developers and adopters. These examples are intended to illustrate how IR 8011
321 concepts work to strengthen the understanding of the methodology and facilitate
322 implementation.

1. IR 8011 Overview

1.1. Background

The IR 8011 methodology was designed to work with the NIST Risk Management Framework (RMF) and supporting technical publications,⁹ including the [SP800-53] control catalog, [SP800-53A] control assessment guidance and procedures, [SP800-53B] control baselines, and [SP800-137] continuous monitoring concepts.

An essential aspect of the NIST RMF is the use of security and privacy controls to safeguard information handled by an organizational system and ensure that security and privacy objectives are met. These controls are assessed and monitored periodically to verify that they are in place, operating as expected, and meeting security and privacy objectives.

Monitoring all selected controls as frequently¹⁰ as needed using manual methods is impractical and unrealistic for most organizations due to the sheer size, complexity, and scope of their information technology footprint. The rapid deployment of new technologies may spawn new risks that make the manual or procedural monitoring of controls unattainable for many organizations.

Control assessment objectives for items in the base control and control enhancements, referred to as *control items*, are provided in [SP800-53A]. The potential assessment methods *examine*, *interview*, and/or *test* (see Table 1), are used to compare the actual state (i.e., what is in place; see Sec. 2.3.5.1) against the desired state specification (i.e., what is expected to be implemented; see Sec. 2.3.5.2). The organization uses the results of the assessments — regardless of the method used — to support the determination of control existence, functionality, correctness, and completeness, as well as the potential for improvement over time.

Table 1. Potential assessment methods [SP800-53A]

Method	Definition
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.
Test¹¹	The process of exercising one or more assessment objects under specified conditions to compare actual with expected behavior.

⁹ See Appendix C for a listing of NIST RMF-related publications and their relationships to IR 8011.

¹⁰ The frequency is enough to maintain ongoing awareness of control effectiveness.

¹¹ The implementation of a continuous monitoring program considers the *test* assessment method whenever it is applicable. Use of the automated test method may provide more accurate and repeatable results when constructed and implemented correctly. It is more difficult to automate the examine and interview assessment methods, which require more complex systems to enable capture and accurate interpretation of the input (from examination of artifacts and interviews). Organizations might employ the examine and/or interview methods for root cause analysis (see Sec. 3.1.6) of *other than satisfied* controls or if greater assurance, depth, or coverage is needed.

1.2. Purpose and Scope

The IR 8011 series offers an approach for automating control testing to support continuous monitoring that focuses on the *test* assessment method [SP800-53A] as the method with most potential for automation in support of the RMF *Assess* and *Monitor* steps. IR 8011 supports the *testing* of controls using automation, which is beyond the scope of SP 800-53A.¹² IR 8011 is not about automating the *implementation* of security and privacy controls (RMF *Implement* step).

This volume introduces fundamental concepts and proposes a methodology for creating automatable tests for monitoring SP 800-53 controls by leveraging the determination statements in SP 800-53A¹³ assessment procedures as the basis for the tests. These tests are traced to specific continuous monitoring security capabilities¹⁴, which are groups of controls with a common defense purpose.

The key elements of the IR 8011 methodology are illustrated in Fig. 1, from the development of *sub-capability*¹⁵ tests to the identification of testable controls with a shared common purpose for a specific security capability.

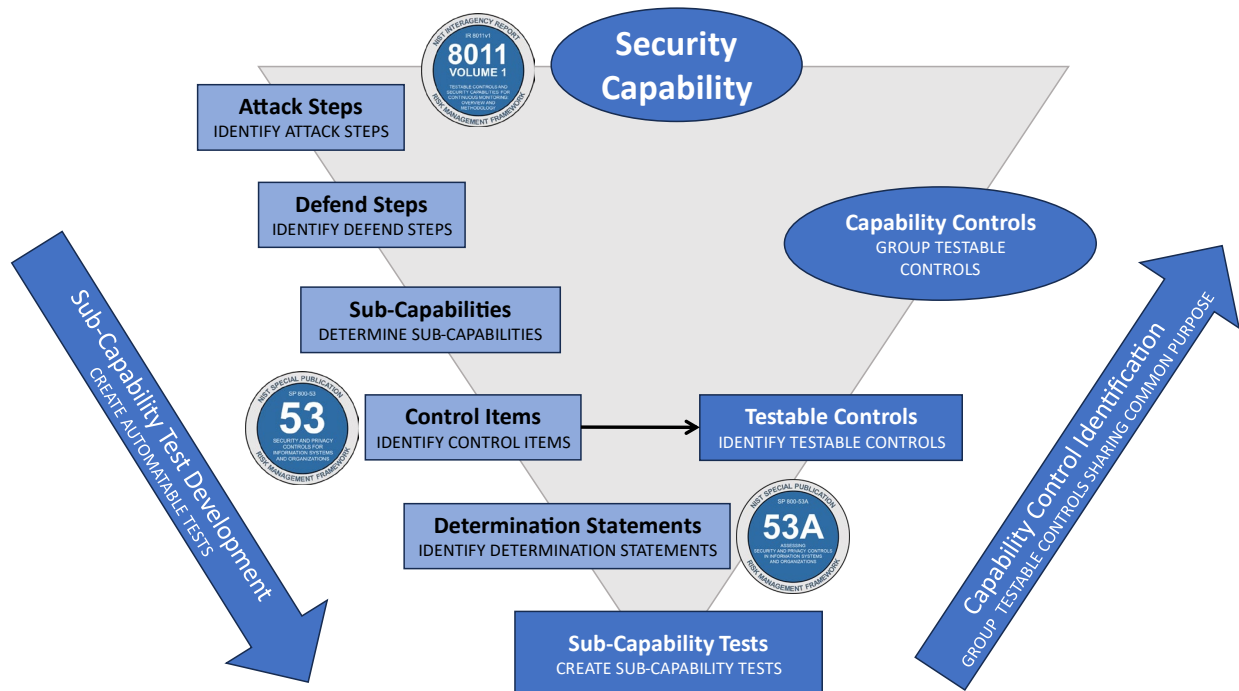


Fig. 1. IR 8011 methodology elements

¹² [SP800-53A] states that “detailed scripts may need to be developed for the specific operating system, network component, middleware, or application employed within the system to adequately assess certain characteristics of a particular security or privacy control. Such test scripts are at a lower level of detail than provided by the assessment procedures contained in SP 800-53A and are beyond the scope of SP 800-53A.”

¹³ Although these tests derive from SP 800-53A assessment procedures that have been designed to assess SP 800-53 controls, IR 8011 is primarily a continuous monitoring initiative.

¹⁴ Security capabilities are discussed in more detail in Sec. 2.3.

¹⁵ Sub-capabilities are discussed in more detail in Sec. 2.3.

Section 3 describes each element in Fig. 1 and their contributions toward meeting the methodology's two main objectives:

1. Sub-capability test development
2. Capability control identification

These objectives are highlighted by the arrows in Fig. 2 and described in detail in Sec. 3.1 and Sec. 3.2.

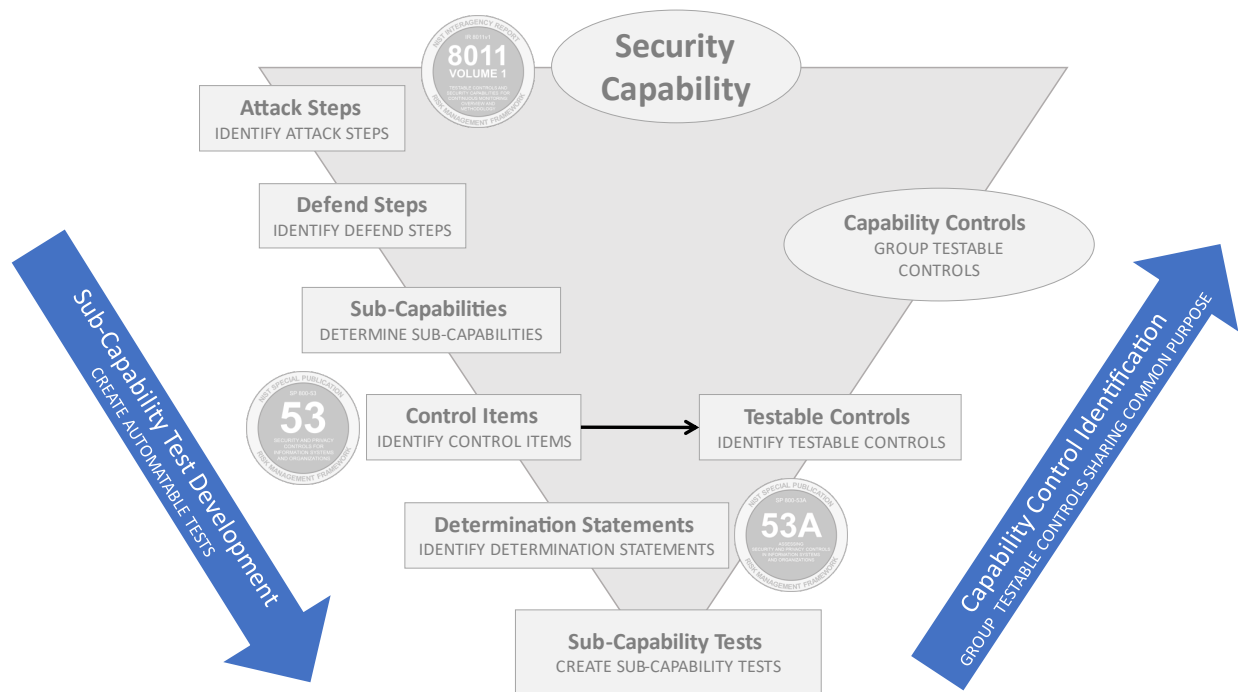


Fig. 2. IR 8011 methodology objectives: Sub-capability test development and capability control identification

1.3. Target Audience

The primary target audience for this publication are the two groups involved with the potential implementation of an automated control testing solution in support of an organization's continuous monitoring program: solution developers/providers and adopters.

Solution developers/providers

Solution developers or providers are the actual implementers of the IR 8011 methodology. They take the IR 8011 blueprints and package test functionalities into a product or solution. Solution providers are system integrators and/or service providers that offer solutions based on the methodology provided in this overview volume to solution adopters, whether as a stand-alone solution or integrated into another product. The automated control testing strategy used by the organization may leverage commercial off-the-shelf, community-built, in-house developed products, or any combination of the three. An IR 8011 solution could be, for instance, an add-on feature to an existing security management application, such as a Governance, Risk, and Compliance (GRC) application.

Solution adopters

Solution adopters are the users or consumers of the product or service solution. Adopters may also be the entity that commissions custom IR 8011 solutions, whether developed in-house or externally contracted. Within the solution-adopting organization, there are additional roles that are identified for a specific security capability, such as operational and managerial roles for a capability (e.g., Device Manager [DM]; Desired State Managers and Authorizers [DSM]; Risk Executive, System Owner, and/or Authorizing Official [RiskExec]). These roles complement existing SP 800-37-defined risk management responsibilities and continuous monitoring operational responsibilities. The capability-specific roles are identified in each capability-specific volume.

A potential third group may or may not be involved with the actual operationalization (implementation) of the IR 8011 methodology: cybersecurity researchers.

Cybersecurity researchers

*Cybersecurity researchers*¹⁶ constitute a potential third audience group due to the existence of opportunities for further research related to the IR 8011 methodology. Cybersecurity research can contribute to the growth and expansion of the IR 8011 methodology with increased speed and accuracy in mind. For example, research into the use of machine learning and natural language processing to identify control items¹⁷ based on context or description could improve the control search process by not relying on the developer to elaborate keywords and the logic behind the keyword search rules. Reliance on an individual's knowledge of a control or a control catalog may result in inaccurate or incomplete identification of controls and limit the full potential of the IR 8011 methodology. This third audience group is not necessarily defined to support NIST's research on improving the IR 8011 methodology, although comments on the methodology are always welcomed. Cybersecurity researchers can be embedded within solution development or adoption teams and contribute problem-solving and innovations that pertain to the development or implementation of an IR 8011 solution.

All target audience groups are encouraged to collaborate and communicate requirements, requirement specifications, maintenance and support strategies, and other development and acquisition concerns to ensure adherence to any applicable organizational requirement for managing security and privacy risks. This may include determining how the solution developers or solution providers keep up with NIST updates to SP 800-53 controls and SP 800-53A assessment procedures and how the products or solutions are kept up to date.

Individuals who are responsible for the design, development, and implementation of continuous monitoring and control assessment processes may also find interest in the IR 8011 series, including those in the following roles:

- Solution development and integration (e.g., software developers, service providers)

¹⁶ This refers to cybersecurity researchers outside of NIST.

¹⁷ The control item identification process is discussed in Sec. 3.1.4.

- System development and integration (e.g., program managers, system developers, system integrators, enterprise architects, security and privacy architects)
- System management (e.g., senior leaders, risk executives, authorizing officials, chief information officers, chief information security officers, chief privacy officers, system owners, security and privacy officers, data managers)
- Control assessment and monitoring (e.g., system evaluators, control assessors, control assessment teams, independent verification and validation assessors, auditors, testers, security operations center personnel)
- Security and privacy control implementation and operations (e.g., system owners; common control providers; information owners or stewards; mission and business process owners; security and privacy architects; security and privacy engineers; security and privacy officers; system, network, database, or application administrators)
- Information technology modernization (e.g., chief modernization officers, chief transformation officers, continuous process improvement managers or specialists)

1.4. IR 8011 Series Organization

The IR 8011 series is organized into multiple volumes. This first volume provides foundational concepts and a methodology for identifying testable controls for security capabilities and developing control tests. Volume 1 is organized into three major sections:

Section 2: *Fundamentals*

Section 2 describes IR 8011, including what it is and is not, the purpose and scope of IR 8011, who is likely to use IR 8011, and how and when IR 8011 is expected to be used. This section provides both a general overview of IR 8011 and an introduction to the IR 8011 methodology to help readers understand how to apply it.

Section 3: *Methodology*

Section 3 explains the process for developing sub-capability tests – the core of the IR 8011 methodology – and the process for identifying testable controls for security capabilities. This section is intended to help the reader understand how the tests are derived from a single security capability and how each element in the methodology supports the IR 8011 objectives.

Section 4: *Operationalization Vision and Considerations*

Section 4 shares a vision for the potential operationalization of the IR 8011 methodology and identifies considerations for implementation. Conceptual implementation examples are presented from two perspectives: one from an IR 8011 solution developer's perspective and another from an IR 8011 solution adopter's perspective. Making such a distinction is necessary to provide readers with guidance that is specific to their role, including guidance to facilitate collaboration and coordination among implementers.

Each subsequent volume in the series addresses a specific security capability for continuous monitoring. Developed in alignment with the concepts, methodology, and guidance provided in this Volume 1, the capability-specific volumes capture a set of sample sub-capabilities and sub-capability tests for a particular security capability. Testable controls for each security capability are published separately from the capability-specific volumes in the form of machine- and human-readable datasets.

IR 8011 volumes on the following security capabilities for continuous monitoring have been published or are planned¹⁸ to be published as capability-specific volumes in the series:

- Hardware Asset Management (HWAM) [IR8011v2]
- Software Asset Management (SWAM) [IR8011v3]
- Software Vulnerability Management (VUL) [IR8011v4]
- Trust Management (TRUST)
- Security-Related Behavior Management (BEHAVE)
- Credentials and Authentication Management (CRED)
- Privilege and Account Management (PRIV)
- Configuration Settings Management (CSM)
- Boundary Management (Physical, Filters, and Other Boundaries) (BOUND-P, BOUND-N, BOUND-O)
- Event (Incident and Contingency) Preparation Management (PREP)
- Anomalous Event Detection Management (EVENT-DETECT)
- Anomalous Event Response and Recovery Management (EVENT-RESPOND)

These security capabilities are a representative set and do not limit the flexibility of an organization to define additional security capabilities.

¹⁸ The titles of planned volumes and the order in which new volumes are released may change.

2. IR 8011 Fundamentals

2.1. How to Use IR 8011

When reviewing the concepts in this publication, readers are encouraged to understand the *purpose* of each element that comprises the methodology: the *purpose* of the security capability, the *purpose* of a specific attack or attack vector (i.e., ways that a threat can attack), the *purpose* of defend steps associated with specific attacks, and the shared common *purpose* of the controls that comprise a security capability. Each purpose influences the elements and their roles and objectives within the IR 8011 methodology.

Both solution developers and solution adopters are encouraged to become familiar with the purpose, scope, desired outcomes, and limitations of the IR 8011 methodology. Solution developers are likely to become proficient in the IR 8011 methodology, whereas solution adopters are encouraged to understand the operational and management roles and responsibilities for each capability addressed by an IR 8011-based solution. Section 4 provides conceptual implementation guidance for both developers and adopters.

2.2. When to Use IR 8011

Solution developers/providers can start using IR 8011 at any time. The complete IR 8011 methodology is presented in this volume. The sample sets of sub-capabilities and sub-capability tests published in capability-specific volumes support the development of tests supporting security capabilities. These sets are not a prescribed, authoritative, or exhaustive list of sub-capabilities and tests. Solution developers can apply the methodology to develop their own sub-capability tests and group testable controls by security capabilities.

Solution adopters can benefit from an IR 8011 solution after a system has been authorized to operate. Use of an IR 8011 solution may support the *Monitor* step of the RMF, specifically an organization's continuous monitoring program. Solution adopters choose to make or build their own IR 8011 solution if one is not available.

2.3. Foundational Concepts

This section introduces and explains the foundational concepts for understanding the methodology described in Sec. 3: boundaries, security capabilities, sub-capabilities, the adversarial attack step model, and test automation.

2.3.1. Boundaries

In the context of the RMF, control assessment and control monitoring activities are limited to the authorization boundary of the system implementing¹⁹ and/or inheriting the controls. The assessment boundary is the same as the authorization boundary. In the context of ISCM,

¹⁹ Including common control providers.

control monitoring²⁰ can span multiple authorization boundaries if organizations implement a centralized monitoring system²¹ with visibility into the authorization boundary of one or more organizational system. This multi-system boundary, intended for continuous monitoring activities by an organization, is what is referred by the 8011 internal reports as *test boundary* (ISCM-TB).²² The use of the term *test boundary* is intended to differentiate from a traditional RMF assessment boundary or from an authorization boundary.²³ The control testing in support of ISCM activities is conducted within a test boundary featuring implemented controls and test objects.

Once organizations begin to automate the testing of objects, it may not be cost-effective to implement a separate automated collection process within each test boundary. Rather, it may be more economical to implement and manage one central automated test system than to use multiple separate automated test systems within the organization. The most cost-effective test boundary may consist of all devices up to and including the physical, logical, or virtual boundary protection devices (e.g., firewalls, routers, and managed switches) that separate the internal network from separately managed external networks and services, as well as a perimeter network or demilitarized zone (DMZ), extranet, intranet, and internal enclaves.

Because the test boundary is comprehensive, it can be used to test the components of multiple systems across several authorization boundaries offering the following advantages:

- The cost of setting up the collectors, collection system, and continuous monitoring dashboard hierarchy is paid once.
- The cost to maintain and continually use the ISCM system within the test boundary may be reduced if the cost to maintain and use can be amortized over time among users.
- The security-related information that is generated can be used to analyze risk across systems:
 - a. A system may inherit controls, such as boundary protection controls, from other systems.
 - b. A system that provides common controls may have all inheritable controls implemented correctly, but it may have other vulnerabilities that could be exploited to compromise the strength of the common control implementation. If that is the case, the common control provider may provide security-related information about the vulnerability to common control implementers through, for instance, an organization risk dashboard.
 - c. Organizational components that are within authorization boundaries but outside of the test boundary may be vulnerable and become vectors to attack components in the test boundary. This may not be evident if the testing only

²⁰ Which may include control assessment/testing activities.

²¹ Such as an ISCM system.

²² In addition to possibility of encompassing multiple authorization boundaries, the test boundary may take into consideration specific functionalities across the organization or other factors such as physical boundaries, network boundaries, and other logical boundaries to determine the extent to which automation can support boundary management controls (e.g., AC-04, SA-09, SC-07, SC-32) and capabilities (i.e., BOUND-P, BOUND-N, BOUND-O).

²³ In the IR 8011 narratives, however, all three terms are used when referring to a specific type of boundary.

looks for risks within the test boundary since systems can inherit risks from a component outside of the test boundary without inheriting controls from that component.

The extra inherited risk information described in the preceding cases b. and c. can provide valuable information about aggregated risks from the mission/business process level and organization level.

2.3.2. Security Capabilities for Continuous Monitoring

IR 8011 focuses on identifying testable controls in support of *security capabilities for continuous monitoring*. Per [SP800-53A], a security capability is:

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

[SP800-53] allows organizations to define security capabilities according to their security goals. The security capabilities for continuous monitoring in IR 8011 address all the [SP800-53B] security baselines by identifying and grouping testable controls that are allocated to baselines.

Continuous monitoring programs²⁴ define specific security capabilities to focus on during implementation. Each capability has a clearly defined outcome that allows monitoring activities — through automated testing — to better inform risk analysis and response. A continuous monitoring security capability consists of a set of SP 800-53 controls needed to obtain the desired outcomes of that security capability. A security capability for continuous monitoring has the following additional characteristics:

- The desired outcome of each security capability is to address specific attack scenarios or exploits.
- Each security capability focuses on attacks on specific targets within the test range.
- There is a viable way to automate many of the control tests that comprise the security capability.

The interaction among the various security capabilities is illustrated in Fig. 3 below.

²⁴ For NIST guidance on the development of a continuous monitoring strategy and the implementation of a continuous monitoring program, see [SP800-137].

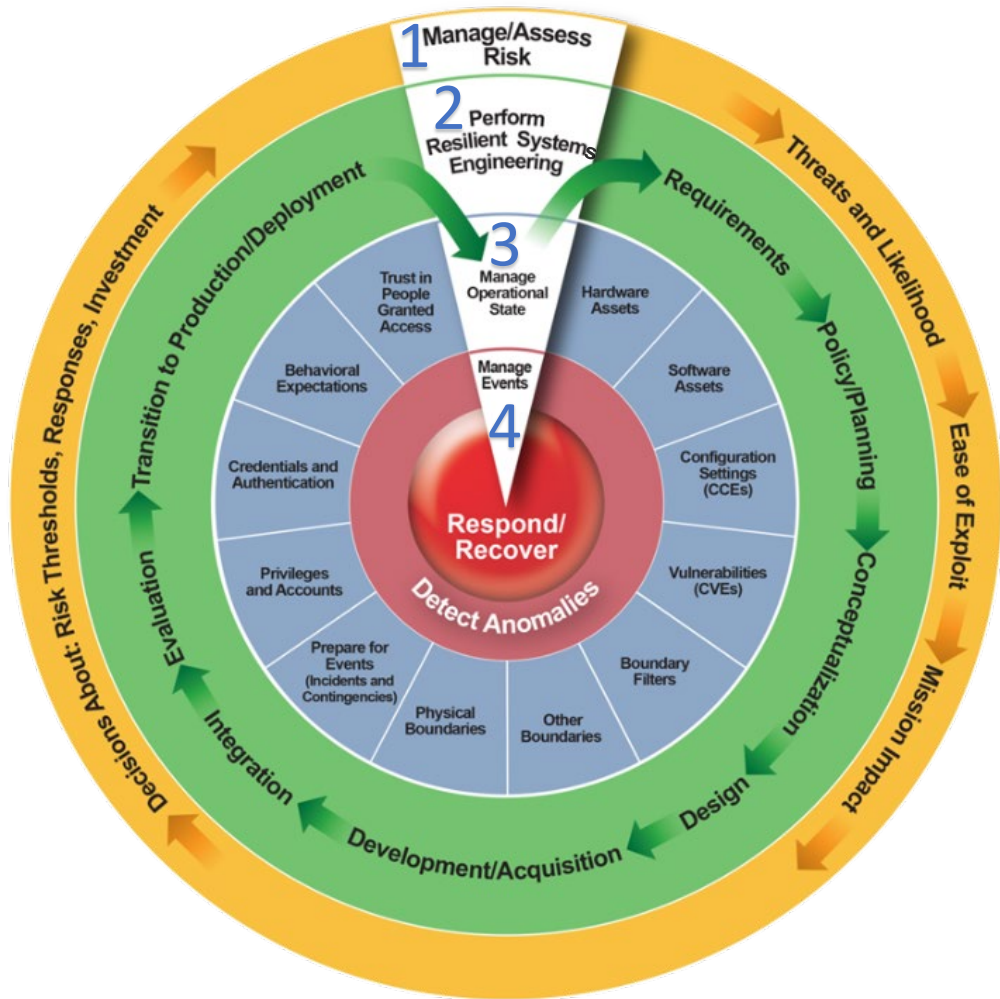


Fig. 3. Security capabilities for continuous monitoring wheel

The security capabilities identified in Fig. 3 are not definitive. Organizations are encouraged to initially automate their control testing approach using the security capabilities listed and described in this volume and later customize capabilities as needed. As different attack scenarios and exploits emerge, existing security capabilities and sub-capabilities can be adapted to address the changing risks, and new capabilities and sub-capabilities can be defined if needed.

Ring 1, Manage/Assess Risk, is the outermost ring. It refers to the overall purpose of continuous monitoring and is informed by and applied to all of the inner rings that cover other continuous monitoring capabilities. Table 2 summarizes the capability, desired outcome, and considerations for Ring 1.

588

Table 2. Ring 1: Manage/Assess Risk

Security Capability Name	Desired Outcome	Considerations
Manage and Assess Risk (RISK)	Reduce impactful exploits that occur in other capabilities because the risk management process failed to correctly identify and prioritize the actions and investments needed to lower the risk profile.	Continuous monitoring dashboards ideally provide scoring and maturity metrics for each capability to prioritize risk response at operational (e.g., system administrator), tactical (e.g., security or privacy officer), and strategic levels (e.g., chief executive officer, chief information officer, chief information security officer, chief privacy officer).

589 Ring 2, Perform Resilient Systems Engineering, focuses on the application of the overall systems
590 engineering process to design resilience into systems. Systems engineering is applied to all of
591 the inner rings of the wheel. It is informed by risk management, risk assessment, and the
592 lessons learned from continuous monitoring of the inner rings.

593 Systems engineering steps may be tailored in a number of ways and may be done in an agile or
594 spiral manner. The terms used in Fig. 3 are illustrative, not prescriptive. For more guidance on
595 resilient systems engineering and effective steps, refer to [SP800-160v1] and [SP800-160v2].

596 The systems engineering outputs are initially assessed outside of continuous monitoring before
597 they go into operations. **Volumes in the IR 8011 series do not provide guidance for the**
598 **automated assessment of the systems engineering phases apart from what might be adapted**
599 **from the operational tests of other capabilities.** Table 3 summarizes the capability, desired
600 outcome, and considerations for Ring 2.

601

Table 3. Ring 2: Perform Resilient Systems Engineering

Security Capability Name	Desired Outcome	Considerations
Perform Resilient Systems Engineering (SE)	Reduce successful exploits that occur in the inner ring capabilities due to inadequate definitions of policy, requirements, planning, and/or other management issues in designing, implementing, and/or monitoring the controls within a given capability.	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 the exploits through more comprehensive requirements, policy, and planning. Monitoring the controls that comprise the inner ring capabilities reveals when exploits are successful. Root cause analysis may determine that exploits resulted from non-conformance in the pre-operational design stages of the life cycle.

602

Ring 3, Manage Operational State, enumerates security capabilities that provide the primary security protections for information and systems during the operations and maintenance phases of the system life cycle and can be largely assessed by automated means. The security capabilities in this inner ring of the security capabilities wheel also identify systemic problems in operations that might be fixed with improved engineering. Table 4 summarizes the capability, desired outcome, and considerations for Ring 3.

Table 4. Ring 3: Manage Operational State

Security Capability Name	Desired Outcome	Considerations
Hardware Asset Management (HWAM)	Ensure that unauthorized and unmanaged devices are identified to enable the organization to prevent attackers from using those devices to compromise systems.	<p>Maintain an inventory of authorized hardware and who manages it. Treat other hardware discovered within the test boundary as a non-conformance.</p> <p>The HWAM inventory may be in any desired format that is machine-readable. The inventory may be maintained via manual or automated means based on organizational needs.</p>
Software Asset Management (SWAM)	Ensure that unauthorized software is identified to prevent attackers from using it to compromise systems.	<p>Maintain an inventory of authorized software at both the product and executable levels. Treat other software discovered within the test boundary as a non-conformance.</p> <p>The SWAM inventory may be in any desired format that is machine-readable. The inventory may be maintained via manual or automated means based on organizational needs.</p> <p>The definition of a software product includes its version, release date, patch level, and other differentiators.</p>
Configuration Settings Management (CSM)	Ensure that common secure configurations (i.e., CCEs, Common Configuration Enumerations) are established and applied to prevent attackers from compromising a system or device that may be used to compromise other systems or devices.	<p>Maintain a record of authorized settings. Treat deviations discovered within the test boundary as non-conformance.</p>

Security Capability Name	Desired Outcome	Considerations
Software Vulnerability Management (VUL)	Ensure that software and firmware vulnerabilities (i.e., CVEs, Common Vulnerabilities and Exposures) are identified and patched to prevent attackers from compromising a system or device that may be used to compromise other systems or devices.	The National Vulnerability Database (NVD) [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 as vulnerabilities that are discovered and assigned a CVE.
Trust Management (TRUST)	Ensure that unauthorized/uncleared persons are not entrusted with system access.	Track the completion of personnel screening processes (e.g., clearances, background checks, suitability reviews) that are designed to identify evidence of untrustworthiness.
Security-Related Behavior Management (BEHAVE)	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 (e.g., training, rules of behavior, access and use agreements, courseware, certifications) that are designed to specify and enable secure behavior.
Credentials and Authentication Management (CRED)	Ensure that authorized users have the credentials and authentication methods necessary to perform their duties, and limit access to only that which is necessary.	Derive the needed credentials and authentication methods from assigned user roles and verify that no extra credentials/methods are provided.
Privilege and Account Management (PRIV)	Ensure that authorized users have the privileges necessary to perform their duties, and limit access to only that which is necessary.	Establish the needed privileges for assigned user roles and verify that no extra privileges are provided.
Physical Boundary Management (BOUND-P)	Ensure that movement (e.g., of people, media, equipment) into and out of the physical facility does not compromise information.	Restrict and monitor physical access using automated tools and collectors to help track and control movements.
Network Boundary Management (BOUND-N)	Ensure that data traffic into and out of the network and, out of the physical facility protection does not compromise information. 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.

Security Capability Name	Desired Outcome	Considerations
Other Boundary Management (BOUND-O)	Ensure that the confidentiality and integrity of information are protected in transit and at rest. This protection is especially important when information is exposed (e.g., in an internet or wireless link) or resides on equipment that could be outside of a secure space (e.g., a laptop or mobile device). Encryption is the most commonly used technique to protect the confidentiality and integrity of information.	Ensure that boundary controls not related to physical and network boundaries are secure to protect data in transit and at rest. Examples of protection include the encryption of network traffic, the encryption of data at rest, and radio frequency spectrum management.
Event (Incident and Contingency) Preparation Management (PREP)	<p>Ensure that procedures and resources are in place to respond to both routine and unexpected events that can compromise information.</p> <p>Potential responses include a wide range of possible actions, including continuity of operations, recovery, and forensics.</p> <p>Unexpected events include actual attacks and natural disasters (e.g., floods, earthquakes, tsunamis).</p>	Identify the desired preparations and verify that they are present and performing as intended. Examples of protection include extra capacity, backups, uninterruptible power supplies, generators, hot sites, and redundant sites.

Ring 4, Manage Events, the innermost ring in the security capability wheel, features security capabilities designed to detect and inform a response to events such as successful attacks and natural disasters that could adversely affect the system despite best efforts to implement the surrounding rings for risk management, risk assessment, resilient systems engineering, and operational state management. The detection and response activities relate to each of the sections of the inner ring. For example, anomalous events could appear in any of the inner ring (“Manage Operational State”) capabilities. In fact, contingency planning and incident response may involve multiple capabilities related to the operational state and/or behavior of the assessment objects covered by the inner ring capabilities. Table 5 summarizes the capability, desired outcome, and considerations for Ring 4.

620

Table 5. Ring 4: Manage Events

Security Capability Name	Desired Outcome	Considerations
Anomalous Event Detection Management (EVENT-DETECT)	Ensure that routine and unexpected events that compromise information can be identified within a specified time frame such that impacts are minimized to the greatest extent possible.	Use various methods to correlate audit records, system events, intrusion detection and prevention system logs, and track patterns to identify unexpected patterns or indicators of harmful activity. Set desired thresholds for impact (e.g., "Servers are never down for more than 24 hours") and detect when thresholds are not met.
Anomalous Event Response and Recovery Management (EVENT-RESPOND)	Ensure that routine and unexpected events that compromise security can be identified within a specified time frame such that impacts are minimized to the greatest extent possible.	Implement desired response procedures and verify that the procedures are performing as intended.

621 A comprehensive and organization-specific automated control testing approach is developed to
 622 address organization-specific capabilities. If an organization creates new controls or control
 623 items, that organization can define the test objectives, identify the testable objects, and
 624 determine the test methods necessary to fully test the new controls or control items.
 625 Automatic testing of the new controls or control items can then be integrated into the IR 8011
 626 solution.

627 **Relationship Between Controls, Control Items, and Security Capabilities**

628 Controls often complement one another to achieve specific security and privacy objectives.
 629 By defining a capability as a group of related controls with a common purpose, testing those
 630 controls as part of a security capability enables organizations to understand system and
 631 organization risk management weaknesses beyond the *other than satisfied* status of
 632 individual controls. IR 8011 focuses on security capability testing in support of continuous
 633 monitoring and not on testing individual controls.

634 **Control Items Support Multiple Capabilities**

635 Most control items support more than one capability because:

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

639 Table 6 provides an example of a single control item that supports multiple capabilities (CM-
 640 03, Configuration Change Control, item b. "*Review proposed configuration-controlled
 641 changes to the system and approve or disapprove such changes with explicit consideration
 642 for security and privacy analyses*").

643

Table 6. Example of a single control item that supports multiple security capabilities

SP 800-53 Control Item	Security Capability Supported	Application
CM-03b.	Network boundary [BOUND-N]: Firewall and routing rules; content filtering rules	Review changes for firewall rules
CM-03b.	Configuration setting management [CSM]	Review changes for configuration settings
CM-03b.	Generic auditing, logging, and monitoring [EVENT-DETECT] to detect incidents and contingencies	Review changes to auditing, logging, and monitoring rules
CM-03b.	Hardware asset management [HWAM]	Review changes to hardware configurations
CM-03b.	Plan and prepare [PREP] for incidents and contingencies	Review changes to required preparations
CM-03b.	Respond [EVENT-RESPOND] to incidents and contingencies	Review changes to planned responses
CM-03b.	Manage risk [RISK] and budget at the management level	Review changes to funding for operational and event-driven risk management actions
CM-03b.	Software asset management [SWAM]	Review changes to authorized software products and executables
CM-03b.	Systems engineering [SE]	Review changes to requirements, designs, etc.

644

Differences Between Controls That Support a Security Capability and Control Families

645

Control families are intended to be general categories used to logically group individual controls within the control catalog. Control families were developed with each control allocated to only one family. The controls necessary to support a specific security capability might come from more than one SP 800-53 control family and a single control may support multiple security capabilities, making control families unsuitable as security capabilities.

646

647

648

649

650

2.3.3. Sub-Capabilities

651

Sub-capabilities refer to the specific protections that provide the ability of the organization or system to be defended. They derive from and support the purpose, objectives, and requirements of a security capability. When they work together, sub-capabilities help achieve the purposes of a security capability.²⁵

652

653

654

655

For example, the HWAM security capability provides a high-level ability to defend against attack steps related to the exploitation of hardware devices. Sub-capabilities are derived to better demonstrate how the HWAM-supporting controls work together to achieve the desired

656

657

²⁵ A security capability can be viewed as a set of sub-capabilities that fulfill the security capability's purpose.

outcome of the HWAM capability. Similar analyses are presented in each capability-specific volume of the IR 8011 series. Table 7 is from the HWAM capability volume [IR8011v2] and lists example definitions of HWAM sub-capabilities. A more in-depth discussion on sub-capabilities, including sub-capability test types that explain the letters F and L in the Sub-Capability ID field, can be found in Sec. 3.1.6.1.

Table 7. Select examples of sub-capabilities (HWAM)

Sub-Capability ID	Sub-Capability Statement	Sub-Capability Purpose
HWAM-F01	Only authorized devices allowed in the boundary	Prevent or reduce the presence of unauthorized devices, reducing the number of potentially malicious or high-risk devices.
HWAM-F02	All devices are assigned to a device manager	Prevent or reduce the number of authorized devices without an assigned device manager within the authorization boundary, reducing delay in responding to non-conformance findings.
HWAM-L01	Devices resistant to exploitation before removal from, during use elsewhere besides, and after return to boundary	Prevent the exploitation of devices before removal from the boundary, during use within the boundary, and after return into the boundary by appropriately hardening the device prior to removal from protected spaces, checking for residual organizational data on the devices before removal from protected spaces, and by sanitizing the devices before introduction or reintroduction into the boundary.
HWAM-L02	Multiple personnel required for authorizing device admission to boundary	Require multiple personnel to authorize a device to be added to the authorization boundary to limit ²⁶ the ability of a single careless or malicious insider to authorize devices.
HWAM-L06	Only authorized sub-components in a device	Detect and remove unauthorized device sub-components to prevent or reduce the introduction of device sub-components that could enable attacks.
HWAM-L07	Business need for use of devices is reviewed regularly	Require periodic and/or event-driven consideration of whether a device is still needed for system functionality to fulfill mission requirements in support of least functionality. ²⁷
HWAM-L08	Required device data is collected	Ensure that data required to assess risk are collected. Some data may relate to non-conformance unrelated to HWAM but may need to be generated by the HWAM collector.

²⁶ 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. See HWAM-L11 for authorization boundary.

²⁷ A good practice requires device managers to review managed devices while system owners review device functionality required within the authorization boundary and identify non-supportable/end-of-life devices in a timely manner.

Sub-capabilities are designed to be tested by automated means. Only one test is defined for each sub-capability to verify whether the objective of that sub-capability is being met. Sub-capability tests contribute to an overall determination of security program effectiveness, especially due to their ability to assess controls/control items and their association with security capabilities.²⁸ This association is the main differentiator between control assessments for authorization purposes and testing a control for monitoring purposes. For this reason, IR 8011 emphasizes *monitoring over assessments*.

Sub-capabilities are defined for each capability, and each sub-capability belongs to exactly one capability. There may be similar sub-capabilities identified for different capabilities. However, a single control or control item can support multiple sub-capabilities and sub-capability tests. A single sub-capability test may involve testing multiple control items that comprise a sub-capability. The sub-capability testing validates whether the sub-capability is effective in supporting the defend actions expressed as part of a sub-capability.

The ability to automate the testing of control items as a collective entity provides a meaningful context for the testing results and supports the identification of root causes of non-conformance. For example, an HWAM sub-capability related to removing high-risk hardware could have sub-capabilities related to removing unauthorized hardware, ensuring that all hardware is managed, or validating that the hardware supply chain is secure. For HWAM, such sub-capabilities support the broader objective of removing high-risk hardware vulnerabilities since unauthorized devices, unmanaged devices, and devices with unapproved supply chains increase risk to organizations.

In the capability-specific volumes, samples of sub-capabilities within each security capability are identified to illustrate how control items in each sub-capability work together to achieve the overall protection goal of the security capability. The sub-capabilities are provided as examples, not requirements, specifications, or conventions, allowing organizations to identify different or additional sub-capabilities.

The IR 8011 methodology organizes sub-capability tests into specific security capabilities for continuous monitoring, each representing the capability to defend against threats. The threats that sub-capabilities protect from are modeled after an adversarial attack step model.

2.3.4. Adversarial Attack Step Model

The IR 8011 series uses a six-step *adversarial attack step model*²⁹ to understand potential attacks and attack vectors, which is one of the first steps in determining sub-capabilities. The security capability is tightly coupled with the protection needs for risk reduction.³⁰ Identifying the necessary protection depends on understanding the different vectors or forms of attack against the security of information. Various attack models have been developed to describe how adversarial attacks occur. Attack step models are articulated from the adversarial

²⁸ Finding non-conforming control items may require root cause analysis (see Sec. 3.1.6).

²⁹ The steps provided in this model are generalized steps.

³⁰ See [SP800-39] for risk concepts, including *risk reduction*.

viewpoint of a malicious attacker. While non-adversarial events³¹ are not directly addressed by attack step models, the attack step model and associated attack and defend actions may be applied to non-adversarial events since many similarities are easily inferred. Organizations consider both adversarial and non-adversarial events as they implement security capabilities as part of a comprehensive risk management and information security continuous monitoring program.

2.3.4.1. Attack Step

The specific controls needed to defend against attack steps can be mapped to specific attack actions.³² The *attack steps* correspond to the security capabilities designed to defend against the attacker at each step. The attack step model depicted in Table 8 consists of a sample of attack steps that are addressed by specific security capabilities identified in this volume and the sub-capabilities identified in each capability-specific volume.

Table 8. Six steps in the IR 8011 attack step model

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

The attack steps described here are simplified, and organizations have the flexibility to define different or additional attack steps and associated security capabilities for their own environments of operation.

2.3.4.2. Defend Step³³

Defense is the target's³⁴ response to attacks (potential or actual) and is the primary protection mechanism against threats. Understanding the intent of an attack, attack objectives, and the attack vectors³⁵ help determine an adequate defense. For each *defend step*, actions are taken to detect and to respond to attacks on the organization, on the system, or on the system component. These actions are referred to as *defend actions*.

³¹ Non-adversarial events occur without malicious intent (e.g., natural disasters, hardware failures, human error).

³² An attack step can be viewed as the purpose or objective of an attack. An attack action can be viewed as a way to achieve the attack purpose or objective.

³³ The defend step was formerly referred to as the *block step*.

³⁴ The target is the system or organization under attack.

³⁵ Attack vectors refer to the ways that a threat can attack.

Defense in Depth

The concept of defense in depth³⁶ means that controls are in place at all defend steps so that if one defensive measure is breached, there are additional layers of protection that will protect the system.

Descriptions and examples of the six attack steps and potential mitigating controls are provided in Table 9.

Table 9. Attack and defend actions for each attack step

Attack Step	Attacker Action	Defender Action
ATTACK STEP 1: Gain internal entry.	<p>ATTACKER ACTION: The attacker is outside of the test boundary and seeks entry, such as spear phishing email sent, distributed denial-of-service (DDoS) attack against .gov initiated, or unauthorized person attempts to gain physical access to restricted facility.</p> <p>In a DDoS attack, the attack traffic only gets into the firewall or another boundary device. Still, this traffic disrupts the connection to the internet, which is inside of the boundary.</p>	<p>DEFENDER ACTION: Limit attacks or negative events from even initiating in or having the ability to impact the local environment.</p> <p>Examples include multi-factor authentication, spam filters, access control lists for routers/firewalls, physical protections (e.g., locks, guards), link encryption and virtual private networks (VPNs), authoritative domain name system (DNS) to prevent poisoning, or gateway-level anti-malware applications.</p> <p>DEFENDER ACTION: Detect entry, respond, and recover.</p> <p>Examples include network intrusion detection systems or surveillance equipment for physical sites that identifies attempts to physical access a facility without authorization.</p>

³⁶ [SP800-160v1] identifies three pillars to defense-in-depth: 1) multiple lines of defenses or barriers are placed along loss scenario sequences; 2) loss control does not rely on a single defensive element; and 3) the successive barriers are diverse in nature and include technical, operational, and organizational barriers.

Attack Step	Attacker Action	Defender Action
ATTACK STEP 2: Initiate attack internally.	<p>ATTACKER ACTION: The attacker is inside of the boundary and initiates an attack internally.</p> <p>Examples include a user opening a spear phishing email with links to malicious content or opens a malicious attachment, a laptop that is lost or stolen, a user installing unauthorized software or hardware, or unauthorized personnel gaining physical access to restricted facility.</p>	<p>DEFENDER ACTION: Limit an initiating condition from occurring in the local environment.</p> <p>Examples include educating users to not open suspicious attachments, maintaining a positive control of assets, or restricting privileges for software installation or removable media.</p> <p>DEFENDER ACTION: Limit a precipitating event from resulting in an attack.</p> <p>Examples include preventing the automatic execution of code on removable media, identifying authorized software for execution, educating users not to share credentials, educating users not to send/receive unencrypted personally identifiable information (PII) and other controlled unclassified information (CUI) without the required protection, or host-level anti-malware applications that block before execution.</p> <p>DEFENDER ACTION: Detect entry, respond, and recover.</p> <p>Examples include host-based intrusion detection systems or surveillance equipment for physical sites that identifies unauthorized physical access to a facility.</p>

Attack Step	Attacker Action	Defender Action
ATTACK STEP 3: Gain foothold.	<p>ATTACKER ACTION: The attacker has gained entry and achieved enough actual compromise to gain a foothold without persistence.</p> <p>Examples include an unauthorized user successfully logging in with authorized credentials, a browser exploit code successfully executing in memory and initiating a call back, or a person gaining unauthorized access to server room.</p>	<p>DEFENDER ACTION: Limit vulnerable conditions that can be exploited by an attack or threat.</p> <p>Examples include patching software or implementing common secure configurations.</p> <p>DEFENDER ACTION: Limit the successful completion of an exploitation attempt.</p> <p>Examples include data execution prevention, recompiling techniques, removing default passwords and accounts, multi-factor authentication, disabling accounts, redundant communication paths, or restricting physical access to critical resources.</p> <p>DEFENDER ACTION: Limit a successful foothold on a test object.</p> <p>Examples include detecting attempts, blocking access attempts to known malicious DNS domains, or reviewing audit and event logs.</p> <p>DEFENDER ACTION: Detect foothold, respond, and recover.</p> <p>Examples include host-based intrusion detection system, behavioral analysis, or surveillance equipment that identifies unauthorized physical access attempts to locations or assets.</p>

Attack Step	Attacker Action	Defender Action
ATTACK STEP 4: Gain persistence.	<p>ATTACKER ACTION: The attack has gained a foothold and now achieves persistence.</p> <p>Examples include malware installed on a host that survives reboot or log off, modified firmware or kernel, a new/privileged account created for an unauthorized user, an unauthorized person issued credentials/allowed access, or unauthorized personnel added to an access control list for a server room.</p>	<p>DEFENDER ACTION: Limit the persistent compromise of an asset.</p> <p>Examples include application allow lists, malware/intrusion prevention tools, virtualization, sandboxing, one-time password systems, requiring hardware tokens for authentication, or restricting physical access with card readers.</p> <p>DEFENDER ACTION: Detect persistence, respond, and recover.</p> <p>Examples include file reputation services, file integrity checking, blocking known malicious command and controls channels, reviewing audit and event logs, advanced behavioral analysis techniques, or surveillance equipment that identifies successful unauthorized physical access to locations or assets.</p>
ATTACK STEP 5: Expand control — escalate or propagate.	<p>ATTACKER ACTION: The attacker has persistence and seeks to expand control through the escalation of privileges or propagation.</p> <p>Examples include hijacked or stolen elevated user privileges, elevated user credentials used by an unauthorized party, changes to secure configuration, disabled audit functions, authorized users accessing resources that they do not need to perform their job, or a compromised or hijacked process or program that runs as an elevated privileged user.</p>	<p>DEFENDER ACTION: Limit the escalation of privileges or access propagation to other assets.</p> <p>Examples include restricting privileges for accounts, programs, and processes; implementing and following configuration change control processes; using hardware tokens or multi-factor authentication for privileged actions; or restricting physical access to server rooms.</p> <p>DEFENDER ACTION: Detect escalation or propagation activity; respond and recover.</p> <p>Examples include intrusion detection system tools or reviews of audit and event logs.</p>

Attack Step	Attacker Action	Defender Action
ATTACK STEP 6: Achieve attack objective.	<p>ATTACKER ACTION: The attacker achieves an objective, resulting in a loss of confidentiality, integrity, or the availability of data or a system capability.</p> <p>Examples include the exfiltration of data, the modification of database entries, a successful DDoS attack, the deletion of data or software, or the disclosure of PII.</p>	<p>DEFENDER ACTION: Minimize the impacts of a successful attack.</p> <p>Examples include data loss prevention tools, device and media encryption, outbound boundary filtering, educating users to protect critical information, restricting access to critical information and resources, file and email encryption, link encryption, or VPNs.</p> <p>DEFENDER ACTION: Detect the impacts of a successful attack; respond and recover.</p> <p>Examples include tools for auditing, insider threat detection, or network event and analysis.</p>

729

730 Multiple Security Capabilities Address Each Attack Step

731 Multiple capabilities can mutually address each attack step and combine to protect a system
732 or organization at all steps in the attack step model even though the defend actions may not
733 be immediately evident.

734 There is a many-to-many relationship between security capabilities and attack steps. Attack
735 steps focus on the attacker's view of the system, including ways to find and exploit
736 vulnerabilities. Security capabilities focus on the defender's view of the system, including
737 ways to prevent attacks or reduce the harm from attacks. Table 10 shows how the security
738 capabilities work together to defend against the six attack steps by identifying which attack
739 steps are addressed by each security capability and how, combined, the security capabilities
740 cover all steps in the attack step model.

741

742

Table 10. Security capabilities work together to defend against attack steps

Security Capabilities	Attack Step 1 (Gain Internal Entry)	Attack Step 2 (Initiate Attack Internally)	Attack Step 3 (Gain Foothold)	Attack Step 4 (Gain Persistence)	Attack Step 5 (Expand Control)	Attack Step 6 (Achieve Attack Objective)	Attack Steps Covered
Hardware Asset Management (HWAM)		2	3			6	2, 3, 6
Software Asset Management (SWAM)		2	3	4			2, 3, 4
Configuration Settings Management (CSM)		2			5		2, 5
Vulnerability Management (VUL)	1	2			5		1, 2, 5
Trust Management (TRUST)	1			4	5		1, 4, 5
Security-Related Behavior Management (BEHAVE)			3	4	5	6	3, 4, 5, 6
Credentials and Authentication Management (CRED)			3	4	5	6	3, 4, 5, 6
Privileges and Account Management (PRIV)		2	3			6	2, 3, 6
Boundary Management, Physical (BOUND-P)	1		3	4	5	6	1, 3, 4, 5, 6
Boundary Management, Filters (BOUND-N)	1		3	4		6	1, 3, 4, 6
Boundary Management, Other (BOUND-O)						6	6
Event Preparation Management (Contingency and Incident) (PREP)		2		4	5	6	2, 4, 5, 6

743

744 Consider five capabilities in Table 10 that support defending an organization from a malware
745 attack initiated from within:

- 746 1. Hardware asset management (HWAM) can prevent the entry of malware by detecting
747 unauthorized/unmanaged devices.
- 748 2. Software asset management (SWAM) can prevent the entry of malware prohibiting or
749 permitting specific software.
- 750 3. Configuration settings management (CSM) ensures that the configurations are defined,
751 established, and implemented to prevent unauthorized changes to devices.
- 752 4. Vulnerability management (VUL) can identify and address exploitable weaknesses in the
753 components before an attack can occur.

5. Security-related behavior management (BEHAVE) can block entry by helping users avoid phishing attacks and preventing users from installing unauthorized hardware and software.

When combined effectively, security capabilities provide defense-in-depth and defense-in-breadth to defend against attacks at each attack step.

2.3.5. Test Automation in IR 8011

The *test method*³⁷ is the process of evaluating one or more test objects under specified conditions to compare an actual state value with a desired state specification. The use of automation in the IR 8011 methodology involves comparing *actual state* and *desired state specification* values in machine-readable format, analyzing them against criteria (e.g., acceptable values, ranges, and thresholds), and reporting the results from the analyses. Automating the testing process enables results to be returned in a timely manner to notify management of potential events or weaknesses in the implementation so that action can be taken before an attack is successful.

2.3.5.1. Actual State

In the test method, the actual state is the security-related information of interest that will be compared and analyzed to verify a control implementation. The automated control testing model illustrated in Sec. 4 assumes that data about the actual state of the items being monitored can be collected by tools called *collectors*.³⁸

2.3.5.2. Desired State Specification

The desired state specification is a defined value against which the actual state value can be compared and explains the implementation of the sub-capabilities. The desired state defines the ability to defend by identifying specifications and acceptable parameters or thresholds for each defense action. Differences³⁹ between the two values may indicate non-conformance in the effectiveness of one or more controls. These specifications are the requirements to be satisfied and can be traced to a control item. For example, an organizational policy states that user accounts are locked after three unsuccessful logon attempts. The desired state specification is: “Applicable devices are configured to lock accounts after three unsuccessful logon attempts.” If, during an automated control testing, the security-related information collected indicates that a specific device is configured such that accounts are locked after five unsuccessful logon attempts, a mismatch between the desired state specification (i.e., three attempts allowed before lockout) and the actual state (i.e., five attempts allowed before

³⁷ The automated *test method* may provide more accurate and repeatable results when constructed and implemented correctly. Automating the *examine* and *interview* methods can be challenging since they require more complex systems to enable capture and accurate interpretation of the input. However, using the automated *test method* does not discard the use of the *examine* and *interview* methods as they can help organizations perform root cause analyses of *other than satisfied* controls or when greater assurance, depth, or coverage is needed.

³⁸ For more on *collectors*, see *collectors and collection system* in Sec. 4.

³⁹ Including values outside of an acceptable range or threshold.

lockout) is identified. A mismatch, whether the actual state exceeds or falls short of the desired state, may reflect a problem with the implementation of SP 800-53 controls AC-07, Unsuccessful Logon Attempts; AC-02, Account Management; CM-02, Baseline Configuration; or others.

The strategy for test automation depends on expressing the desired state specifications for each testable control item in a machine-readable data format that matches, or is comparable to, the data specification format for actual state values. Having a machine-readable dataset of desired state specifications is fundamental to the automation of control testing. For example, the desired state specification may be applied categorically to groups of components or applied on a per component basis.

Examples of desired state specification information include:

- Authorized devices
- Authorized device roles
- Permitted and prohibited software for each device role
- Required frequency of security and privacy awareness training
- Authorized configuration settings for each device role
- Vulnerable software versions (provided by the NIST National Vulnerability Database [NVD])
- Authorized users and privileges

Types of desired state specifications

The desired state specification is expressed in different ways depending on specification type, as shown in Table 11.

Table 11. Types of desired state specifications

Type of Desired State Specification	Simplified Examples (Actual cases may be more complex)
Desired state	If software product X is present, setting Z is expected to have value Y.
Prohibited state	If software product X is present, specified patch levels have CVEs that produce risk and are prohibited.
Expected state	If software product X is present, the device is expected to 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 who receive email validate the origin of the message before clicking on links or opening attachments in the message.

Type of Desired State Specification	Simplified Examples (Actual cases may be more complex)
Prohibited behavior	Users with privileged accounts that are allowed to install software are not permitted to browse the internet or use email from the privileged accounts.
Expected behavior	User B normally logs in from devices in the [geographical location] area during the period from 8 a.m. to 6 p.m. This activity would constitute expected behavior. Other patterns of login activity might indicate account compromise.

Desired and prohibited states and behaviors express normative policy. In contrast, expected states and behaviors are not normative policy but descriptions of patterns. The analysis of expected states and behaviors can detect unusual states and behaviors, including anomalous or suspected malicious activity, that might require manual intervention, response, or recovery. Expected states and behaviors are not typically used outside of the Anomalous Event Detection Management and Anomalous Event Response and Recovery Management security capabilities.

It is possible to restate a prohibited state as desired behavior, as shown in Table 12.

Table 12. Example of equivalence of prohibited and desired state specification

Prohibited Behavior	Equivalent Desired Behavior
Users with accounts that are allowed to install software are not permitted to browse the internet or use email from such accounts.	Users with accounts that are allowed to install software do not browse the internet or use email from such accounts.

While expected behavior can be restated as desired behavior, expected behavior indicates a symptom of a *possible* problem rather than a *definite* problem.

Prerequisites for Automation

The following are prerequisites for effectively automating the testing of control items:

- The actual state and actual behavior parameters are stored as machine-readable data.
- Desired or expected state or behavior expectations are defined and recorded as machine-readable data and are readily comparable to the actual state.
- A method to compute or identify non-conformance based on differences between desired and actual state and behavior is defined.
- A method for producing a human-readable control test report to facilitate analysis and risk-based decision-making is defined.
- A threshold (e.g., upper limit, lower limit) for the actual state is defined as part of the desired state specification.

When the prerequisites are met, the automated control testing system can automatically compute the following:

- Where differences occur between the desired state specification and actual state

- The priority of each finding⁴⁰
 - Assignment of the findings to the appropriate operational team for response⁴¹
- Not all controls can be fully automated for testing using the methods described above. Testing some controls may be partially automated or not automated at all.
- A functional, overarching view of the major steps in the automated control testing process is illustrated in Fig. 4.

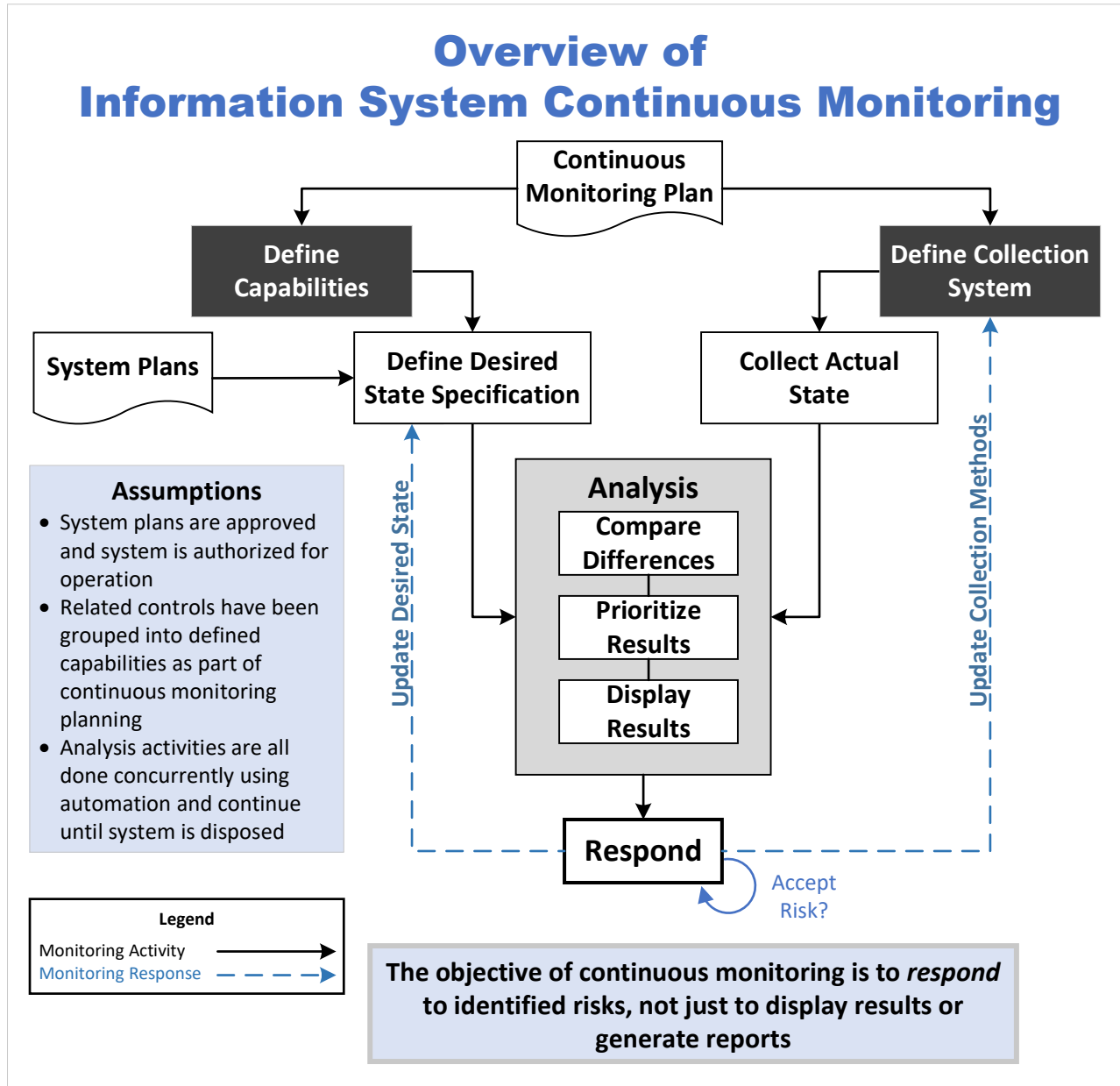


Fig. 4. Overview of an automated control testing process in support of continuous monitoring

⁴⁰ This refers to risk scoring to automatically prioritize responses to each finding, which is out of scope for the IR 8011 series.

⁴¹ Responsibilities are security capability-specific and defined and described in each capability-specific volume.

842 Section 4 provides a conceptual example of the potential inner workings of collectors and the
843 collection system, including the collection of actual state data, the automated comparison of
844 actual state values and desired state specifications, and subsequent reporting of this
845 comparison.

846 The success of using automation to support a control test is predicated on:

- 847 • The effective execution of manual processes to define the controls that comprise the
848 sub-capability,
- 849 • The actions that result from the control items included in each security capability,
- 850 • Understanding the implementation of each control item, and
- 851 • Defining the thresholds and variances associated with each implementation.

3. IR 8011 Methodology

This section provides an overview of the processes and relationships between the different elements in the IR 8011 methodology organized by methodology objectives:

3.1. Objective #1: Sub-Capability Test Development

The development of sub-capability tests is one of the two major objectives of the IR 8011 methodology. Testing capabilities and sub-capabilities enables an understanding of weaknesses beyond individual controls being determined to be *other than satisfied*.

The sub-capability test development process helps identify testable controls that can be used to support the continuous and automated testing of sub-capabilities within a larger security capability. The tests are elaborated exclusively for control items that can be tested via automated means. The elements in the test development process are illustrated in Fig. 5.



Fig. 5. Sub-capability test development workflow elements

Section 2.3 identifies and describes security capabilities for continuous monitoring, including the security capabilities addressed by IR 8011.

The following sections describe each of the elements in the test development process, their expected outcomes, and their relationship in supporting the sub-capability test development process.

3.1.1. Identify Attack Steps



The purpose and desired outcome of each security capability listed in Fig. 3 and in the *adversarial attack step model* can assist in the identification of a sequential attack step that can be evaluated when determining specific defense steps.⁴² Identifying the attack steps relies on understanding the desired outcome of the security capability and the potential attack actions to be defended against. This understanding contributes to the identification of the defend steps that solidify the protection requirements and form the basis for identifying testable controls later in the test development workflow.

⁴² The defense steps are described in Sec. 3.1.2.

879 **Table 13. Attack step and attack step actions addressed by the HWAM security capability**

Attack Step	Attack Step Actions
2) Initiate attack internally	<p>The attacker is inside of the boundary and initiates an attack internally.</p> <p>Examples include a user opening a spear phishing email or clicking on an attachment; a laptop being lost or stolen; a user installing unauthorized software or hardware; or unauthorized personnel gaining physical access to restricted facility.</p>
3) Gain foothold	<p>The attacker has gained entry to the boundary and achieves enough actual compromise to gain a foothold without persistence.</p> <p>Examples include an unauthorized user successfully logging in with valid credentials; a browser exploit code successfully executing in memory and initiating a call back; or a person gaining unauthorized access to a server room.</p>
6) Achieve attack objective	<p>The attacker achieves an objective, resulting in the loss of confidentiality, integrity, or availability of data or system capability.</p> <p>Examples include the exfiltration of files; the modification of database entries; the deletion of a file or application; denial of service; or the disclosure of PII.</p>

880 For example, the desired outcome of the HWAM security capability in Fig. 3 and Table 4 is to
881 “ensure that unauthorized and unmanaged devices are identified to enable the organization to
882 prevent attackers from using those devices to compromise systems”:

- 883 • Unauthorized and unmanaged devices that are already in the boundary can be traced to
884 step 2 of the adversarial attack step model as the attacker is already inside.
- 885 • The fact that the device is already in the boundary indicates that the attacker has gained
886 a foothold, as indicated in step 3 of the attack model.
- 887 • The unauthorized and unmanaged devices in the boundary could potentially enable an
888 attacker to make lateral or upward movements, which would constitute step 6 in the
889 attack model.

890 All three steps are identified regardless of whether the attack or attacker is successful or not.
891 Organizations understand this scenario to ensure that appropriate defenses are in place.

892 In each capability-specific volume in the IR 8011 series, the attack steps associated with each
893 security capability are identified.

894

3.1.2. Identify Defend Steps



After determining the potential attacker actions derived from the identified attack steps, a set of defender actions is elaborated as *defend steps*. The identification of the defend steps promotes an understanding of the expected ability to respond to an attack and protect the organization, system, or component that is targeted or attacked. Table 14 illustrates the defend steps for each of the identified attack steps in the HWAM example.

Table 14. HWAM attack step and defend step actions

Attack Step Action	Defend Step Action
ATTACK STEP 2: Initiate attack internally. The attacker is inside of the boundary and initiates an attack internally. Examples include a user opening a spear phishing email or opening an attachment; a laptop that is lost or stolen; a user installing unauthorized software or hardware; or unauthorized personnel gaining physical access to a restricted facility.	DEFEND STEP: Block or limit internal access. Prevent or minimize access to trusted network resources by unauthorized or compromised devices; reduce the amount of time that unauthorized devices are present before detection.
ATTACK STEP 3: Gain foothold. The attacker has gained entry and achieves enough actual compromise to gain a foothold without persistence. Examples include an unauthorized user successfully logging in with valid credentials; a browser exploit code successfully executing in memory and initiating a call back; or a person gaining unauthorized access to server room.	DEFEND STEP: Block foothold. Reduce the number of unauthorized and/or easy-to-compromise devices that are not being actively administered.
ATTACK STEP 6: Achieve attack objective. The attacker achieves an objective, resulting in a loss of confidentiality, integrity, or availability of data or system capability. Examples include the exfiltration of files; the modification of database entries; the deletion of a file or application; denial of service; or the disclosure of PII.	DEFEND STEP: Block physical exfiltration. Prevent or minimize the copying of information to unauthorized devices.

The defend steps and actions associated with a security capability are identified in each capability-specific volume in the IR 8011 series.

3.1.3. Determine Sub-Capabilities



Sub-capabilities help clarify how security capabilities address attack steps. After the defend steps are identified, specific sub-capabilities are elaborated based on the defender actions of each defend step. The intent is to support the defense of attack steps and provide the necessary testable protection. As such, each sub-capability is designed so that it can be tested individually via automated means, with one test for each sub-capability.

The sub-capability element is the most appropriate level in the methodology⁴³ on which to focus automated testing for continuous monitoring. The sub-capability layer has enough details on a defense objective to identify potential protective measures to achieve that objective. These details can also reduce the number of false positives during control searches, discussed in the next section. When non-conformances are found, root cause analyses can be used to find the specific control items causing the non-conformance. For further discussion on why the sub-capability element offers the optimum level for automating control testing among the different elements in the methodology, see Appendix D.4.

3.1.4. Identify Control Items



The *control items* element is where specific parts of a control are identified as potential defenses that can achieve the desired outcomes of the sub-capability. Control items can be matched to a sub-capability via a control item search against a control catalog, producing a many-to-one relationship. The IR 8011 approach to finding control items is through the use of *keywords*⁴⁴ against a control catalog. The IR 8011 methodology uses the SP 800-53 security and privacy control catalog with control names, control statements, and discussion text as the search scope. For each capability-specific volume,⁴⁵ keyword search rules⁴⁶ help automate the identification of control items that are relevant to the specific security capability. A systematic process validates the keyword rules, including testing for missed control items and evaluating the results for false positives and false negatives. The internal process and procedures for control item search by keyword are not covered in this volume, but each capability-specific volume documents both the keyword search rules used and the identified control items that support the security capability.

⁴³ This refers to the different elements in the sub-capability test development process in Fig. 5: attack steps, defend steps, sub-capabilities, control items, determination statements, and sub-capability tests.

⁴⁴ IR 8011 implementers are encouraged to use any approach, including an automated approach, as long as a match can be found between a control item objective and a sub-capability purpose.

⁴⁵ The capability-specific volumes are published separately.

⁴⁶ Control search by keyword is discussed in Sec. 4.1.1.7.

The sample keywords in Table 15 were identified from the sub-capabilities defined in the previous step, “Determine Sub-Capabilities,” with text that describes the purpose of the sub-capability.

Table 15. Illustrative keyword rules to trace control items to security capabilities

A control item traces to the HWAM security capability if one or more of the following are true:
It contains “inventory”
It contains “supply chain” and NOT “monitoring”
...and multiple other conditions...

“Inventory” is a recurring term that appears in multiple sub-capabilities in [IR8011v2]. The second rule example shows the use of the logical operator “NOT” in the keyword search to narrow the search results and reduce the number of false positives.

Table 16. Illustrative control items traced to the HWAM capability

IR 8011 Security Capability	SP 800-53B Control Baseline	SP 800-53 Control Item
HWAM	Low	AC-19a.
HWAM	Low	AC-19b.
HWAM	Low	CM-08a.
HWAM	Low	CM-08b.
HWAM	Moderate	AC-19(05)
HWAM	Moderate	CM-02(07)(a)
HWAM	Moderate	CM-02(07)(b)
HWAM	High	CM-03(01)(a)
HWAM	High	CM-03(01)(e)
HWAM	High	CM-03(01)(f)
HWAM	High	CM-08(02)

Section 4.1 expands the discussion geared toward IR 8011 solution developers with additional insights and considerations regarding control item search by keyword, which is an important element in the sub-capability test development process.

3.1.5. Identify Determination Statements



The identification of control items only returns results if there is a match between a control item objective and the desired results of a sub-capability. The next step is to verify that the identified control items can be assessed or monitored through automation using the test assessment method.

[SP800-53A] breaks down SP 800-53 controls into singular, assessable parts that are either *satisfied* or *other than satisfied*. This granularization facilitates the individual assessment of

each part of the control regardless of the assessment method. These smaller, assessable parts take shape in the form of *determination statements* for control assessments, which provide a foundation for the development of automatable tests. Each determination statement of an assessment objective in [SP800-53A] is associated with an individual control item⁴⁷ in [SP800-53]. Many control items have more than one associated determination statement. The example below displays two determination statements for a single control item (CM-03f.) following the control text:

Control Item: CM-03f.: Configuration Change Control
f. Monitor and review activities associated with configuration-controlled changes to the {devices and device components of the} system.

The insertion of “{devices and device components of the}” into the example above is intended to clarify the scope for the HWAM capability and determination statement to provide context, facilitate development, and promote understanding of the application of sub-capability tests. Similar capability-specific insertions are present in each capability volume.

Table 17. Examples of determination statements highlighting IR 8011 HWAM focus

SP 800-53A Determination Statement ID	SP 800-53A Determination Statement Text with IR 8011 HWAM Focus
CM-03f.[01]	Determine if: f. Activities associated with configuration-controlled changes to the {devices and device sub-components of the} system are monitored .
CM-03f.[02]	Determine if: f. Activities associated with configuration-controlled changes to the {devices and device sub-components of the} system are reviewed .

The determination statements in [SP800-53A] control assessment objectives correspond directly to the control items from [SP800-53]. While there is often a one-to-one match between a control item and an associated determination statement, the granularization⁴⁸ of the control item can result in multiple assessment objectives, as shown in Table 18.

⁴⁷ A single control item may support multiple capabilities. Within a capability, only the way in which the control item supports that capability is considered.

⁴⁸ [SP800-53A], Sec. 2.4.3, discusses the application of granularization to facilitate the assessment of the distinct parts of an [SP800-53] control item. The granularized determination statements use square brackets in the [SP800-53A] assessment objectives identifiers, as shown in the third column of Table 18.

974

Table 18. Illustrative control items traced to an associated assessment objective

SP 800-53 Control Item	SP 800-53 Control Item (JSON identifiers from the CPRT dataset) ⁴⁹	SP 800-53A Assessment Objective Identifier
AC-19a.	AC-19-a	AC-19a.[01] AC-19a.[02] AC-19a.[03]
AC-19(05)	AC-19(05)	AC-19(05)
CM-08b.	CM-08-b	CM-08b.
PE-06b.	PE-06-b	PE-06b.[01] PE-06b.[02]
PS-04d.	PS-04-d	PS-04d.
SC-15b.	SC-15-b	SC-15b.

975 The granularized determination statements for the same control item may or may not support
 976 testing using automated methods. Additional effort may be required by the individual(s)
 977 performing the identification of determination statements to validate whether testing for each
 978 determination statement for a control item can be automated or whether non-automated
 979 methods will be required to fully test the control item. Table 19 provides an example in which
 980 the control item is granularized into four determination statements that each determine
 981 whether the desired state specification is implemented.

982

Table 19. Example control and determination statements for AC-19, Access Control for Mobile Devices

Control Statement (SP 800-53)	Determination Statement (SP 800-53A Assessment Objective)
a. Establish configuration requirements, connection requirements, and implementation guidance for organization-controlled mobile devices, to include when such devices are outside of controlled areas.	Determine if: AC-19a.[01] configuration requirements are established for organization-controlled mobile devices, including when such devices are outside of the controlled area. AC-19a.[02] connection requirements are established for organization-controlled mobile devices, including when such devices are outside of the controlled area.
b. Authorize the connection of mobile devices to organizational systems.	AC-19a.[03] implementation guidance is established for organization-controlled mobile devices, including when such devices are outside of the controlled area. AC-19b. the connection of mobile devices to organizational systems is authorized.

⁴⁹ This is for illustrative purposes only and is intended to provide implementers using datasets from the CPRT with an alternative view of the control item identifiers.

3.1.6. Create Sub-Capability Tests



Sub-capability tests provide a context for testing activities, including what is being tested, how it is being tested, the expected outcome of the test, the actual outcome of the test, and the implication of the testing activity result relative to the risk posture of the component, system element, system, authorization boundary, or organization. The tests are designed to provide a valid measure of whether, and to what extent, the desired outcomes of the sub-capability are being achieved.

The intent of the sub-capability test development process is to identify testable controls for security capabilities and to elaborate the actual sub-capability tests. Once the desired state specifications are set, a comparison between the actual state and the desired state is performed to verify that control item requirements are being satisfied. This comparison is executed in an automated fashion based on the determination statements within the assessment procedures of the control or control item by verifying the determination statements for the control items that support the purpose of the sub-capability.

Sub-capability tests can also determine the control implementation variance. If the difference between desired and actual state values is within an acceptable range, then the control item is *satisfied*. If the test results are not within acceptable parameters, then the control item's actual state does not conform with the requirement and is considered *other than satisfied*.

Sub-capability tests:

- Are **stated as a test** (wherever appropriate), whether in support of a control assessment or of control monitoring.
- Identify **automated** approaches to the test method for assessing and monitoring controls. Thus, manual or procedural test methods are not sub-capability tests.
- Define an explicit **desired state specification** that is then compared to the corresponding actual state value to determine the test result.
- Are typically **at a higher level of abstraction** than a single control determination statement.

3.1.6.1. Sub-Capability Test Types

Sub-capability tests are designated as one of three types: *foundational*, *local*, or *data quality* tests:

1. **Foundational Tests** — Tests that are fundamental to help achieve the desired outcomes of the capability.
2. **Local Tests** — Tests that an organization determines whether or not to implement. Local tests provide greater test depth and may be selected by the organization based on their

1018 risk tolerance and need for greater assurance when corresponding controls are
1019 implemented. Regarding local sub-capability tests, the organization:

- 1020 • Might not implement a test because the test checks a control item that is in a
1021 control baseline not found within the organization or within a specific organizational
1022 system;
- 1023 • Might not implement a test because the test checks a control item that is not
1024 implemented within the organization or within a specific organizational system;
- 1025 • Might only implement a test for specific test targets for which an associated control
1026 is implemented;
- 1027 • Might implement an alternative version of the local sub-capability test; or
- 1028 • Might use manual or procedural testing for certain control items.

1029 Organizations may also add or edit local tests as appropriate to manage their own risk.
1030 For example, sub-capability tests may be added for controls that are implemented as
1031 supplemented controls.

1032 3. **Data Quality Tests** — Tests to determine whether the data collected is both *complete*
1033 and *timely*. If metrics for completeness and timeliness are not adequate, the test is not
1034 reliable because it may yield inaccurate results.

1035 In order to automate control testing to the greatest extent possible and to support ongoing
1036 authorization, implementation of the applicable foundational and local tests defined in this
1037 volume and enumerated in the separate capability-specific volumes is needed for *all*
1038 implemented control items.

1039 3.1.6.2. Data Quality Measures

1040 An automated test method can be used when the automated control testing functionality has
1041 an equal or higher probability of detecting non-conformance compared to interview and
1042 examination methods. The two factors that contribute most to the detection of non-
1043 conformance are:

- 1044 1. The *completeness* of automated control testing
- 1045 2. The *timeliness* of automated control testing

1046 **Completeness**

1047 This refers to the extent to which the security-related information includes testing all
1048 relevant non-conformance on all test targets within a defined scope, such as a capability.
1049 *Relevant non-conformance* can pose risks. An incomplete measure tends to bias the results
1050 by underestimating total risk. The *completeness measure* type is to ensure that the
1051 automated control testing is conducted for *all* sub-capability tests and on all test targets that
1052 could be non-compliant. Although total completeness might not be attained, the probability

1053 of missing non-conformance approaches zero as automated control testing approaches
1054 100% completion.

1055 **Timeliness**

1056 This refers to the extent to which the security-related information has been refreshed within
1057 the last hours or days as required by the organization. The objective of automated testing is
1058 to collect data to identify and respond to non-conformances faster than an attacker's ability
1059 to compromise a system.⁵⁰ The timeliness measurement ensures that each cycle of tests on
1060 the *non-conformance test target* combinations occurs at least as often as the frequency
1061 specified in the continuous monitoring strategy. Initially, the specified frequency may merely
1062 be faster or more frequent than reported in previous readings. As the automated control
1063 testing functionality matures, the frequency is often enough that the automated control
1064 testing data collection system identifies and allows time for a response to vulnerabilities due
1065 to non-conformances *before* an adversary can exploit them.

1066 As part of the risk management process and continuous monitoring strategy, the organization
1067 determines the degree of completeness and timeliness required *before* it replaces manual or
1068 procedural testing with an automated control test system. The continuous monitoring
1069 dashboard provides metrics to help evaluate this readiness. Table 20 shows an example of data
1070 quality measures.

1071 **Table 20. Example data quality measures**

Measure Type	Description	When to Use This Measure
Completeness	Percent of devices for which complete data is being collected	Setting an organization-defined threshold on completeness metrics triggers an alarm when the overall level of completeness is too low to provide reliable results on non-conformance.
Timeliness	Percent of devices for which data is being collected within organizationally defined time periods	Setting an organization-defined threshold on timeliness metrics triggers an alarm when measures have not been captured within a defined time period, which may indicate a failure of the process.

1072 Automated control testing is adequate to replace manual or procedural control testing as soon
1073 as it is at least as *timely* and *complete* as the manual or procedural test methods for the
1074 capabilities being covered and their related controls. The data quality measures included in the
1075 capability-specific volumes use the letter prefix Q in their identifiers.

⁵⁰ While not always feasible to implement, event-driven testing that can detect non-conformance when introduced provides the best timeliness.

3.1.6.3. Sub-Capability Test Creation

Sub-capability tests are created for each individual sub-capability to verify whether the controls that support the sub-capability are operating effectively and whether the desired outcomes of the sub-capability can continue to be fulfilled. The activities in Table 21 describe the test creation process.

Table 21. Sub-capability test creation process activities

Activity	Identifier	Process Description
Identify the sub-capability with a statement	Sub-Capability Statement	Identify the sub-capability by a short name to address its purpose.
Elaborate the desired outcomes of the sub-capability	Sub-Capability Purpose	Provide a full description of the desired outcomes of the sub-capability. ⁵¹
Assign a unique identifier for each test	Test ID	Include the security capability abbreviation (e.g., HWAM) followed by a dash and a designator for the test type (i.e., Foundational (F) test, a Local (L) test, or a Data Quality (Q) test), a unique sequential number, and the words “-Test” to differentiate the test from the sub-capability (e.g., HWAM-F01-Test).
Label the test	Test	Provide a short descriptor to distinctly identify the test within the sub-capability.
Define and summarize the test criteria	Test Criteria Summary	Provide a short description of how to decide (i.e., compute) whether non-conformance is present.
Document any notes about the test criteria	Test Criteria Notes	<p>Expand on the test criteria summary. At a minimum, the test criteria notes define the following:</p> <ul style="list-style-type: none"> What data is used <ul style="list-style-type: none"> To define the actual state To define the desired state specification How the actual state and desired state specification data sets are used to identify non-conformance <p>The language in the test criteria notes is intentionally generic to provide greater implementation flexibility. However, the notes are also specific enough to facilitate the design of reliable and repeatable tests.</p>

The following example shows how sub-capability tests are featured in capability-specific volumes using an HWAM sub-capability test.

⁵¹ This is generated by the *Determine Sub-Capabilities* element (see Sec. 3.1.3) in the sub-capability test development process.

Only Authorized Devices Allowed in the Boundary Sub-Capability and Sub-Capability Test

Table 22. Sub-capability statement and purpose example

Sub-Capability Statement	Sub-Capability Purpose
Only authorized devices allowed in the boundary	Prevent or reduce the presence of unauthorized devices, reducing the number of potentially malicious or high-risk devices.

Table 23 defines an example sub-capability test to verify whether the sub-capability is operating effectively.

Table 23. Example sub-capability test (from HWAM)

Test ID	Test	Test Criteria Summary	Test Criteria Notes
HWAM-F01-Test	Unauthorized devices in the boundary	The device is present in the authorization boundary (i.e., <i>is</i> in actual state) but has not been authorized to be there (i.e., <i>is not</i> in the desired state specification). See supplemental criteria in L02.	<p>1) The actual state is the list (i.e., inventory) of all devices within an organization-defined tolerance in the authorization boundary, as determined by the continuous monitoring system.</p> <p>2) The desired state specification is a list of all devices that are authorized to be in the authorization boundary.</p> <p>3) Non-conformance refers to a device that is in the actual state but not in the desired state and is, thus, unauthorized. The non-conformance verification is computed by simple set differencing.</p>

Since the sub-capability tests are designed to focus on the *purpose* that a set of controls is intended to achieve, they are at a higher level of abstraction than the determination statements for a single control item. The HWAM security capability defines a sub-capability test to verify whether the hardware supplier and/or manufacturer are on an approved list. This sub-capability test is:

- Directly supported⁵² by one control, SR-03(02), Supply Chain Controls and Processes,⁵³ which calls for the consideration of supply chain issues in approving devices

⁵² The direct and indirect support of controls are related to the focus of the control objectives with regard to the purpose of the sub-capability test.

⁵³ SR-03(02) is not selected for any baseline; it is only mentioned here to illustrate a point.

- Indirectly supported by other controls (e.g., parts of CM-03, Configuration Change Control) that require a configuration management process to consider security and privacy impacts explicitly in the change control process (implicitly, including supply chain, where appropriate)

The control items that can be tested by a sub-capability test via automated means work together to achieve the desired outcomes of a specific sub-capability. In the Table 22 example, the purpose is to reduce the potential consequences of supply chain attacks, which is one part of the overall hardware asset management capability and, in effect, a *sub-capability* of HWAM. The sub-capability test checks the individual control items and the overall effectiveness of the controls working together as a sub-capability.

Table 24 summarizes the relationship between sub-capability tests and determination statements.

Table 24. Relationships between sub-capability tests and determination statements

Sub-Capability Tests (IR 8011)	Determination Statements (SP 800-53A)
Focus on a <i>purpose</i>	Focus on a <i>control</i>
Tightly linked to a <i>purpose</i> of one or more controls in a security capability	Tightly linked to a <i>specific control</i>
Reported as <i>other than satisfied</i> or <i>non-conforming</i> if the purpose of those controls is not being met	Reported as <i>other than satisfied</i> if the control has not been demonstrated to be fully implemented, is not operating as expected, or is not providing the expected level of protection

3.1.6.4. Sub-Capability Test Non-Conformance

System personnel can compare the automated test results to the variances to determine whether 1) the implementation is operating within the defined specifications, or 2) an investigation is necessary because the test result was outside of the defined specifications.

Responses to Non-Conformance

When non-conformance is identified through the test process, specific roles may be predefined for notification and response activities.

Table 25 provides a sample of non-conformance responses and the roles that may have the responsibilities to address identified issues.

1119 **Table 25. Example of potential actions for non-conformance response and responsibility assignments (HWAM)**

Test ID	Response Description	Primary Responsibility
HWAM-F01-Test	Remove Device	Device Manager (DeviceMgr)
HWAM-F01-Test	Authorize Device	Desired State Manager (DesiredStateMgr)
HWAM-F01-Test	Accept Risk	Risk Executive, System Owner, and/or Authorizing Official (RiskExec)
HWAM-F01-Test	Ensure Correct Response	Desired State Manager (DesiredStateMgr)

1120 The example responsibility assignments do not change the overall risk management
1121 responsibilities defined in other NIST publications, and risk management responsibilities can
1122 be customized by each organization to best adapt to local circumstances.

1123 Table 25 suggests a *Primary Responsibility* to determines the most appropriate response and
1124 ensure that the response action is allocated to the appropriate role.

1125 The difference in the level of focus between sub-capability tests and determination
1126 statements has a significant impact on how non-conformance is interpreted once
1127 discovered. The difference relates to the *sensitivity* and *specificity* of the result.

1128 Sensitivity

1129 A *sensitivity test* is one that finds *all* of the cases in which non-conformance occurs, resulting
1130 in low false negative⁵⁴ rates.

1131 A sub-capability test focused on whether the desired outcome of a set of controls is met
1132 reflects a high degree of sensitivity if it correctly reports on all cases in which the non-
1133 conformance occurs.

1134 In the example of supply chain-related controls, the sub-capability test for hardware supply
1135 chain would fail if either:

- 1136 • A list of approved suppliers and manufacturers was not set up per SR-03(02); or
- 1137 • A device from a supplier not on the approved list of suppliers was approved by the
- 1138 change control process per CM-03.

1139 Since this sub-capability test focuses on reducing the potential consequences of supply chain
1140 attacks, and the sub-capability test directly measures all of the cases in which that purpose is
1141 not met, the sub-capability test can be said to be highly *sensitive*.

1142 Specificity

1143 A *specificity test* is one that does not report non-conformance when non-conformance is not
1144 present, resulting in low false positive⁵⁵ rates.

1145 Sub-capability tests measure the specific results to be achieved by a set of controls. Failure
1146 to achieve the result does not imply that *all* of the controls supporting that capability failed.

⁵⁴ A false negative reports that non-conformance does not exist when there is non-conformance.

⁵⁵ A false positive reports that non-conformance exists when there is none.

While the sub-capability test is *specific* to the desired outcome of the sub-capability, it is *not specific* at the control or control item level.

In the example of supply chain-related controls, the failure of the sub-capability test does not help determine whether the control failed to achieve its objectives because:

- A list of approved suppliers and manufacturers was not set up per SR-03(02),
- A device from a supplier not on the list of approved suppliers was approved by the change control process per CM-03, or
- The non-conformance could have occurred because one or both objectives failed, though a failed sub-capability test does not mean that *all* of the supporting controls are *other than satisfied*.

The considerations about *sensitivity* and *specificity* are applied in root cause analysis. Table 26 illustrates how the level of sensitivity and specificity of a test can be summarized.

Table 26. Sensitivity and specificity notes

Level of Testing	Degree of Sensitivity (at the Specified Level of Testing)	Degree of Specificity (at the Specified Level of Testing)
Sub-Capability Desired Result	Sensitive	Specific
Control Item (Determination Statement) Effectiveness	Sensitive	Not Specific (Specific with root cause analysis)

There is no prescribed scale or range for either the degree of *sensitivity* or *specificity*. Organizations determine their own acceptable scales or ranges based on their operating conditions and risk management policies.

3.1.6.5. Root Cause Analysis

Root cause analysis adds *specificity* at the control level. Throughout the normal course of sub-capability testing, issues with control implementation due to weaknesses that result in non-conformances may be detected after sub-capability test failure and with a given security capability failing to achieve its desired outcomes. Root cause analysis is performed to determine the cause of the test failures.

Making a single test both sensitive and specific is challenging because specificity deteriorates as criteria are changed to improve sensitivity. Likewise, sensitivity can degrade as criteria are changed to improve specificity. A potential testing strategy is to use two tests in phases:

1. A *highly sensitive test* is performed first to find as many positive results as possible with the understanding that it may include some false positives.
2. A *highly specific test* is subsequently administered to the cases that failed the *highly sensitive test* to evaluate and to eliminate the false positives.

This combination of tests is intended to identify all true positives in a population. Control testing provides a warning that one or more controls that support a security capability are *other than satisfied*. However, because it is possible that only one control is *other than satisfied*, it cannot be assumed that *all* of the supporting controls are *other than satisfied*. Root cause analysis helps determine which specific controls supporting the sub-capability are *other than satisfied*.

In the example of the supply chain-related 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-03). A trend analysis further indicates whether the weakness in the change control process was a recurring problem. Some valid conclusions to draw when a sub-capability test falls outside of an acceptable threshold include:

- One or more of the supporting controls failed to achieve its objectives.
- Root cause analysis is used to determine which controls are *other than satisfied*.
- It is not necessarily the case that all supporting controls failed to achieve their objectives.

The risk management responses defined in [SP800-39] are used to address non-conformance: accepting, avoiding, mitigating, sharing, or transferring risk. In general, under a continuous monitoring program, the responsibility for risk response belongs to the organization.

Root cause analysis operates on the logical flow of cause and effect from control items to a sub-capability test results that is the objective of a security capability, as shown in Fig. 6.

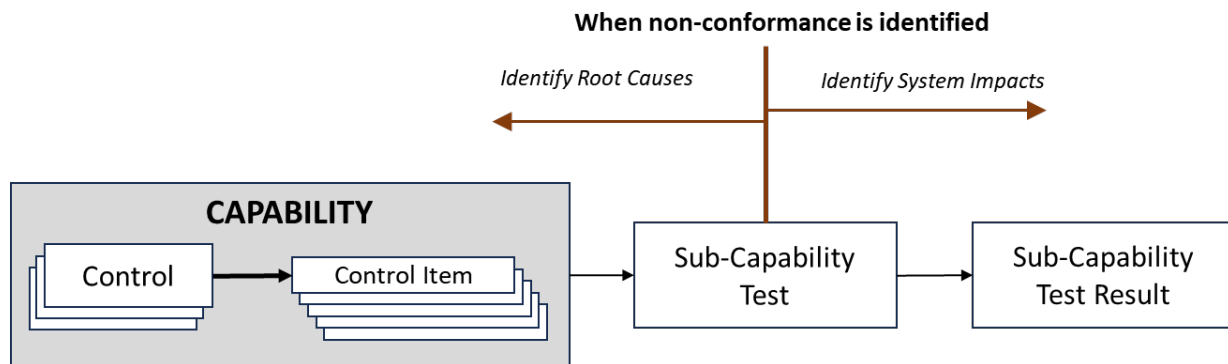


Fig. 6. Flow of cause and effect from control items to sub-capability test results

The desired sub-capability test result is to make attack scenarios and/or exploits more difficult to conduct by reducing the number of vulnerabilities and the likelihood that those vulnerabilities can be exploited. Desired outcomes are identified for each capability in the subsequent volumes in the IR 8011 series.

A non-conformance might be noticed at the control item, the whole control, the sub-capability test, and/or at the security capability level. Root cause analysis includes:

- 1206 • Looking **back** toward the control items to see which failures may have caused the non-
1207 conformance and
- 1208 • Looking **forward** to review the positive or negative impact on the desired security
1209 outcome.

1210 By looking forward, one might find that the failures are *not* compromising the desired security
1211 outcome or that the failure is *not* having a significant negative impact on the security outcome.
1212 The information discovered from root cause analysis is used to prioritize efforts to address non-
1213 conformance or to help determine whether the risk from the non-conformance of a particular
1214 control can be accepted.

1215 The following topics illustrate how root cause analysis is applied.

1216 **Analyzing Controls**

1217 It is important to understand why a particular control or control item is determined to be
1218 *other than satisfied*. The reason may be obvious, so it may be appropriate to fix the control
1219 implementation to satisfy the requirements. When determining why a capability comprised
1220 of multiple controls is not conforming, the root cause may be more subtle to discern and
1221 require greater effort to resolve.

1222 If a needed patch has not been applied or a configuration setting is incorrect, an immediate
1223 risk reduction action could include the application of the software patch or adjusting the
1224 appropriate settings. If test results consistently demonstrate non-conformance with defined
1225 requirements, it is advisable to look deeper. One key factor to look for in this situation is
1226 whether there is a systemic cause preventing or delaying the satisfactory resolution of the
1227 issue.

1228 Consider whether an engineering misstep from an early stage in the system life cycle is
1229 preventing conformance to requirements. Questions that can help with this analysis include:

- 1230 • Was the capability or control functionality supporting the capability added at the end of
1231 the system life cycle so that not enough preparation and planning were done or the
1232 security functionality is not yet optimal?
- 1233 • Has an appropriate policy been established to guide control implementation and
1234 management?
- 1235 • Were requirements appropriately defined?
- 1236 • Is the responsibility for avoiding and fixing non-conformance clearly defined?
- 1237 • Is the non-conformance occurring outside of system boundaries where it has been
1238 overlooked?
- 1239 • What can be done to change user behavior to increase conformance to requirements?
1240 For example, is additional training needed to ensure that users are aware of the
1241 penalties for not following policies and/or procedures?
- 1242 • Can operators easily obtain the necessary information to prevent non-conformance? For
1243 example, it may be difficult to know what privileges are inherited by a user from parent
1244 groups in certain directory services.

- 1245 • Was the control implementation automated? For example, is an automated centralized
1246 patch management system in place? Is the automation working?
- 1247 • For manually implemented and managed controls, do personnel have the necessary
1248 resources, training, and tools?
- 1249 • Were appropriate tools and methods used to implement the control?
- 1250 • Did planning for the implementation ensure that adequate funds, staff, and other
1251 resources were provided for implementation?
- 1252 • Are operational staff members tasked by policy to do so many things for security that
1253 they are overwhelmed?
- 1254 • Was the control implementation adequately tested?

1255 Finding non-conformance in an organization, especially if the non-conformance spans across
1256 multiple systems, can be an important function for either the organization or control
1257 assessors. These non-conformance findings are more important than a list of specific non-
1258 conformances from a penetration test exercise or a control assessment of a single system.
1259 While this analysis is more difficult than just reporting individual control non-conformance,
1260 finding and resolving non-conformance can have a much more profound effect on improving
1261 security and privacy programs.

1262 **Levels of Root Cause Analysis**

1263 Three levels of root cause analysis are needed for sub-capability test failures:

1264 **Level 1: Determine *case-specific causes*.** This determination typically involves affirming
1265 whether the desired specification or the actual state is in error:

1266 a. Was the desired state specification insufficient?

1267 b. Was the actual state not captured correctly?

1268 In coordination with the system owner and security and privacy officer, designated
1269 operational staff review each specific case to decide whether option a. or b. applies to the
1270 non-conformance and consider what caused a. or b. to be the non-conformance. For
1271 example, a system administrator has connected multiple components to the production
1272 network without first adding them to the component inventory, configuring them correctly,
1273 and patching them. Determining that this is the root cause indicates that option b. is the
1274 issue because the actual state is the non-conformance due to unpatched and misconfigured
1275 components in the boundary that are not in the component inventory. The solution is not
1276 just to get the component authorized, configured, and patched, but to make sure the system
1277 administrator understands the importance of following operational procedures. The failure
1278 includes one or more of the controls or control items related to managing the actual state.

1279 In another example, a system administrator has connected multiple components to the
1280 production network after getting them authorized and correctly configuring and patching
1281 them. However, the administrator forgot to put them in the component inventory first.
1282 Determining that this is the root cause indicates that option a. is the issue because the
1283 desired state specification is the non-conformance due to failure to include a correctly

authorized component in the inventory. The solution is to enter the component into the inventory and ensure that the system administrator understands the need to add authorized components to the component inventory before putting them in the boundary. The failure includes one or more of the controls or control items related to managing the desired state specification.

The determination of whether a. or b. is the cause also helps clarify which control items failed: control items related to desired state specification or to actual state. Additional analysis may be needed to determine the specific control items that are failing.

Level 2: Determine *which controls failed to achieve their objectives*. Use the tables that map specific sub-capability tests to specific control items that might be causing the sub-capability test to fail. A mapping table similar to Table 27 is included in the discussion of each capability-specific volume.

Table 27. Mapping of sub-capability tests to specific control items

IR 8011 Test ID	SP 800-53B Baseline	SP 800-53 Control Item Code	SP 800-53A Assessment Objective Identifier
HWAM-F01-Test	Low	AC-19(b)	AC-19b.
HWAM-F01-Test	Low	CM-08(a)	CM-08a.
HWAM-F01-Test	Low	CM-08(b)	CM-08b.
HWAM-F01-Test	Moderate	AC-20(2)	AC-20(02)
HWAM-F01-Test	Moderate	CM-03(b)	CM-03b.
HWAM-F01-Test	Moderate	CM-03(c)	CM-03c.
HWAM-F01-Test	High	CM-03(01)(a)	CM-03(01)(a)
HWAM-F01-Test	High	CM-03(01)(b)	CM-03(01)(b)
HWAM-F01-Test	High	CM-03(01)(c)	CM-03(01)(c)

The sub-capability assessed by the HWAM-F01-Test is supported by each of the above control items. If any of the supporting controls fails to achieve its objectives, the sub-capability test also fails. Therefore, the sub-capability test indirectly tests the control items.⁵⁶

The root cause analysis determines whether all of the implemented controls related to the sub-capability test are operating as intended. If some or all of the controls are not operating as intended, repairs or changes may be necessary, or the authorizing official can make a risk acceptance decision with appropriate justification.

Once the controls implicated in the sub-capability test non-conformance are identified, additional root cause analysis efforts can determine why the controls are not operating as intended.

Level 3: Determine *systemic causes*. The root cause analysis looks for systemic causes of repeated failures or engineering missteps and seeks to identify appropriate resolutions. In

⁵⁶ This example does not include all of the control items associated with this sub-capability test to fail. See the corresponding capability volume for the complete list.

the first example for Level 1, the non-conformances in question may have occurred repeatedly because the system administrator:

- Has no way to properly configure and patch the devices until the devices are on the production network;
- Lacks the training to know how to prepare devices before putting them on the production network;
- Is overwhelmed with multiple tasks and is skipping procedural steps to keep up with their assigned workload; and/or
- Is unaware of the operational procedures.

There may be other possible causes, and finding those root causes may be more relevant than focusing on individual non-conformances. Once causes are identified, the impacts are also analyzed. The organization considers how important a specific failure is in the context of the overall organizational risk tolerance. Table 28 shows three example scenarios involving the failure to assign a manager to a device in the boundary.

Table 28. Example impact scenarios and analyses

Case	Example Scenario	Example Impact Analysis ⁵⁷
A	The role of the device manager (DeviceMgr) exists but is not specifically designated. Someone has been managing the devices but forgets to record the device in the component inventory.	There is a relatively low short-term risk because the device is actually being managed, but the lack of a designated device manager is addressed so that the responsible person receives and responds to relevant non-conformance lists going forward.
B	A device was put on the production network for test purposes so it was not added to the component inventory. The device has become vulnerable over time due to the lack of patching and configuration management, and downstream target objects can be attacked through it.	There is a high risk because the device is not being managed with the potential for increased risks as the device becomes vulnerable over time. Potential response includes the removal of the unmanaged device from the boundary and ensure the device manager completes appropriate role-based 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 of putting unauthorized and higher risk devices in a segment of the network without prior authorization for 10 weeks. Authorization and other cleanup are to occur before the 10 weeks have elapsed.	There is a moderate to high risk because while the authorization official accepted that risk, the devices remained in the boundary without authorization for longer than permitted. Potential response includes the development of better approaches to address similar situations in the future and avoid having to accept such risk in the future.

The ability to identify both root causes and the impacts of sub-capability test failures is an essential activity to support the automated control test system that typically identifies non-conformance at the sub-capability test level. Reaching significant systemic conclusions may

⁵⁷ Table 28 uses a *low*-, *moderate*-, and *high-risk* scale for the example impact analysis. Organizations determine their own risk scale with regard to impact based on their risk management policies. For more on quantitative assessment and scales, see Sec. 2.3.2 of [SP800-30].

imply the need for new desired state specifications in supporting areas (e.g., identifying a need for training system administrators in a specific topic or skill). Policy changes and related sub-capability tests for the new desired state specifications can then be established.

Assignment of Responsibility

For an organization dashboard to generate effective lists of actionable items for responding to non-conformances, the dashboard requires the functionality to identify the specific operational role responsible for responding to each non-conformance maintained as part of the desired state specification. Depending on the size and complexity of the system, the operational roles may be performed by a specific individual or a group with an assigned supervisor. Responsibility is clearly assigned to ensure that response tasks to nonconformances are completed and can be documented in a variety of ways. Table 29 shows an example of a partial table from the HWAM capability volume.

Table 29. HWAM example of documented roles assigned to respond to non-conformances

Determination Statement ID	Implemented By	Test boundary	Test Responsibility	Assessment Method(s)	Rationale for Risk Acceptance	Test Frequency	Impact of Not Implementing
CM-08a.[01]	DesiredStateMgr	ISCM-TB	ISCM-Sys	Test			
CM-08a.[02]	ISCM-Sys	ISCM-TB	ISCM-Sys	Test			
CM-08a.[03]	ISCM-Sys	ISCM-TB	ISCM-Sys	Test			
CM-08b.[01]	DeviceMgr	ISCM-TB	ISCM-Sys	Test			
CM-08b.[02]	DesiredStateMgr	ISCM-TB	ISCM-Sys	Test			

The example in Table 29 captures the following for each control item determination statement:

- *Determination Statement ID*: The unique SP 800-53A Determination Statement identifier to trace back to the SP 800-53 control item being tested
- *Implemented By*: The role⁵⁸ or system that is primarily responsible for control item implementation to clarify responsibility for non-conformances, such as:
 - DeviceMgr: Device Manager
 - DesiredStateMgr: Desired State Managers and Authorizers
 - ISCM-Sys: The system that collects, analyzes, and displays ISCM security-related information (e.g., an IR 8011 solution)
- *Test boundary*: The test boundary to clarify the scope of the test. It is the user-defined test boundary, or portion of the test boundary
- *Test Responsibility*: The role responsible for the control testing
- *Assessment Methods*: The assessment methods to be used⁵⁹
- *Rationale for Risk Acceptance*: Rationale for the non-selection or risk acceptance of a selected control when test results reflect *other than satisfied*

⁵⁸ The roles listed under the “Implemented By” column are described in the HWAM capability volume [IR8011v2].

⁵⁹ In most cases, this will likely be the test method.

- 1357 • *Test Frequency*: The minimum frequency with which the test is to be conducted⁶⁰
- 1358 • *Impact of Not Implementing*: The potential impacts to organizational test objects,
- 1359 individuals, other organizations, or the Nation that may occur if this control is *other*
- 1360 *than satisfied* or if a sub-capability test is not implemented.

1361 Documenting Sub-Capability Test Rationale

1362 Within the test plan narrative, a sub-capability test rationale table traces the test criteria for
1363 each applicable sub-capability test to control determination statements. The table indicates
1364 which sub-capability tests fail if the given determination statement returns an *other than*
1365 *satisfied* status as well as an explanation of how the sub-capability test applies. The sub-
1366 capability test rationale table indicates how the sub-capability test is assessing the item
1367 using the determination statement in question and includes all of the applicable sub-
1368 capability tests for each determination statement. The sub-capability test and rationale
1369 columns provide the following:

- 1370 • The *Sub-Capability Test* columns (*Test ID* and *Sub-Capability Test*) identify the sub-
1371 capability tests from the sub-capability test tables that assess or monitor the control
1372 item. Refer to the sub-capability test tables within each capability volume for a
1373 description of how the sub-capability test applies to a given test object.
- 1374 • The *Rationale* column describes the conditions under which a failure of the sub-
1375 capability test might be caused by a failure of the control to achieve its objectives.
1376 Moreover, if the control is deemed *other than satisfied* too often relative to an
1377 organization-defined threshold, it may cause a failure of the test criteria for a sub-
1378 capability test.

1379 The non-conformance of a sub-capability test does not prove that a control was not satisfied
1380 since the sub-capability test is not specific to a control or control item. Refer to the root
1381 cause analysis discussion in Sec. 3.1.6.5 for information on how to determine which control
1382 items caused the sub-capability test to fail. If the control item is determined to have failed,
1383 then its control has at least partially failed, both resulting in an *other than satisfied* condition
1384 by a control assessment.

1385 Documenting the sub-capability test rationale helps support the development and
1386 implementation of sub-capability tests. Table 30 provides an example of documented
1387 rationale for specific data quality tests for a single determination statement.

⁶⁰ The frequencies specified in the “Test Frequency” column are at least as often as the frequency determinations in the organization’s continuous monitoring strategy.

1388

Table 30. Example of a documented sub-capability test rationale

Determination Statement ID	Test ID	Sub-Capability Test	Sub-Capability Test Rationale ⁶¹ If an [organization-defined measure] for this sub-capability test is to improve [the organization-defined threshold], <i>then non-conformance or discrepancies in an inventory of the {devices and device sub-components of the} system that includes all components within the authorization boundary being developed/documentd or being accurate related to this control item</i> might be the cause of...
CM-03f.[01]	HWAM-Q01-Test	Devices in the Boundary Not Reporting to ISCM-Sys	A device failing to report within the specified time frame
CM-03f.[01]	HWAM-Q02-Test	Non-Reporting of Sub-Capability Test Results to ISCM-Sys	Specific sub-capability checks failing to report
CM-03f.[01]	HWAM-Q03-Test	Missing Report(s) from Selected Sub-Capability Test(s)	The completeness of overall ISCM reporting failing to meet the threshold
CM-03f.[01]	HWAM-Q04-Test	Selected Sub-Capability Tests Do Not Report on Time	The poor timeliness of overall ISCM reporting

1389

Final Thoughts on the Sub-Capability Test Development Objective

1390

The ability to automate tests can support continuous monitoring efforts but does not fully replace other evaluation methods for determining the effectiveness of controls. IR 8011 supports automation of the collection, evaluation, and reporting of implementation data. It is up to the organization to review the reported data and take any appropriate actions. It is also the organization's responsibility to ensure that the parts of the control or control enhancement that are not tested through automated means are tested via other methods. Sub-capability tests support the assessment and monitoring of controls through automation and decrease the assessment and monitoring levels of effort by speeding up processes.

1397

1398

The sub-capability tests identified in IR 8011 capability-specific volumes are a sample set of tests in support of a given security capability. **Organizations are not expected to employ all the sub-capability tests described in each volume.**

1399

1400

1401

Table 31 provides a summary of the outputs for each element in the sub-capability test development process.

1402

⁶¹ Note on formatting: for the rationale statement in the Table 29 header, items within square brackets represent universal parameters applicable to any capability. Items within curly brackets are capability-specific. Since we are using HWAM as an example, "{devices and device sub-components of the}" is specific to HWAM.

1403

Table 31. Sub-capability test development workflow output summary

Element	Output	Output Summary
Attack Steps	Attacker actions are understood	An understanding of the threat, exploitable vulnerability, and potential attack vectors that can be used to exploit the vulnerability
Defend Steps	Defender actions are understood	An understanding of the expected ability to detect an attack and protect the organization, system, or component being attacked or targeted
Sub-Capabilities	Defender capability actions are defined	Definitions for specific actions to support risk/threat management efforts for each functional capability
Control Items	Defender actions are identified	The identification of specific control items through a control search by keyword
Determination Statements	Test objectives are identified	The identification of test procedures for each control item identified
Sub-Capability Tests	Implementation variance is determined	An understanding that a <i>non-conformance</i> can occur when the <i>actual state</i> operates outside of the bounds of the <i>desired state</i> of the implementation

1404 3.2. Objective #2: Capability Control Identification

1405 The identification of testable controls for security capabilities is another major objective of the
 1406 IR 8011 methodology. It is a follow-on to the sub-capability test development whose output
 1407 confirms the testability of control items. The identification of testable controls derives from the
 1408 control items that support the sub-capability tests. Together, the testable controls share the
 1409 same common purpose in support of a security capability.

1410 3.2.1. Identify Testable Controls



1411

1412 Potential control items that can be tested via automated means are identified via control
 1413 search by keyword during the sub-capability test development process.⁶² Once the sub-
 1414 capability test is developed, the control item associated with the test can be confirmed as being
 1415 testable. The control item is then traced to a control or control enhancement, as shown in
 1416 Table 32.

⁶² Each capability volume includes sample keywords for identifying potential testable controls in support of the security capability.

1417

Table 32. Tracing control items to controls/control enhancements

SP 800-53 Control Item	SP 800-53 Control/Control Enhancement
AC-19b.	AC-19
CM-08(04)	CM-08(04)
CM-03b.	CM-03
MA-03(01)	MA-03(01)
CM-03(01)a.	CM-03(01)

1418 The identification of testable controls is one step toward identifying the controls that share a
1419 common defense purpose for a security capability. After the testable controls are identified, the
1420 baselines to which they belong can be identified, as shown in Table 33.

1421

Table 33. Tracing controls to control baselines

SP 800-53 Control	SP 800-53B Security Control Baseline ⁶³
AC-19	Low
CM-08(04)	Low
CM-03	Moderate
MA-03(01)	Moderate
CM-03(01)	High

1422 Tracing controls to control baselines can facilitate certain monitoring activities (e.g., planning,
1423 reporting), just as control items are traced to controls to assist with planning and
1424 documentation. For this reason, both control-item-to-control and control-to-control-baseline
1425 mappings can be an explicit or internal function or feature of an IR 8011 solution. Adopters
1426 ensure that any tailoring of the control baselines is addressed.

1427 Tracing control items to sub-capabilities can also be beneficial to implementers when there is a
1428 need to enumerate all control items for a given sub-capability. The capability-specific volumes
1429 in the IR 8011 series list the control items that support each sub-capability. This is documented
1430 in a table similar to Table 34, which includes a sample of control items that trace to a sub-
1431 capability.

1432

Table 34. Example of tracing control items to sub-capabilities (HWAM)

SP 800-53 Control Item	Test ID
AC-19b.	HWAM-F02-Test
CM-08(04)	HWAM-F02-Test
CM-03b.	HWAM-F02-Test
MA-03(01)	HWAM-F02-Test
CM-03(01)a.	HWAM-F02-Test

⁶³ This refers to the lowest baseline to which the control is allocated.

1433 Depending on the implementation, a similar listing may be produced to return all of the sub-
1434 capabilities that a given control item can support.

1435 3.2.2. Group Testable Controls

1436 Identify Testable Controls

GROUP TESTABLE CONTROLS

1437 The last step in the IR 8011 methodology is to group testable controls that share a common
1438 purpose. At this stage, the control set can be further organized by control baselines using
1439 [SP800-53B] as a reference to facilitate the deployment of an operationalized IR 8011
1440 implementation.⁶⁴

1441 This group of controls is intended to be *continuously* monitored so that the security capability
1442 can be monitored independently from any set of controls selected for assessments. Testing
1443 security capabilities once is not enough to keep up with evolving threats.

1444 3.3. Methodology Summary

1445 The IR 8011 methodology:

- 1446 • Identifies controls and control items that can be tested to help automate the test
1447 assessment method in SP 800-53A
- 1448 • Proposes an approach for developing tests that can be automated to offer criteria for
1449 the development of *sub-capability*⁶⁵ tests
- 1450 • Groups identified controls as sets of controls with a shared, common purpose to
1451 organize automatable tests by security capabilities
- 1452 • Provides sample capability and test narratives that can be used when developing control
1453 assessment and continuous assessment plans and to facilitate the operationalization⁶⁶
1454 and implementation of an IR 8011 solution on a per security capability basis
- 1455 • Establishes guidelines for using automation in support of testing and continuous
1456 monitoring
- 1457 • Supports the adaptation of existing manual control assessments to an automated
1458 testing process
- 1459 • May be used in support of control-based frameworks or methodologies other than the
1460 NIST RMF and SP 800-53
- 1461 • Supports the development of control tests for a specific security capability, a specific
1462 control family, or stand-alone controls

⁶⁴ IR 8011 solutions may provide the functionality to list testable controls for a given security capability by baseline to facilitate assessments and monitoring activities.

⁶⁵ See Sec. 2.3.

⁶⁶ Section 4 describes a vision for IR 8011 operationalization.

- 1463 • **Does not** assist in automating control implementation
- 1464 • **Does not** provide a ready-to-use solution but rather a blueprint for operationalization⁶⁷
- 1465 • **Does not** require all tests in the capability-specific volumes to be performed; the tests in
- 1466 each capability-specific volume are only a sampling of potential tests
- 1467 • **Does not** provide authoritative or exhaustive listings of security capabilities, sub-
- 1468 capabilities, tests, or testable controls; the capability-specific volumes provide only
- 1469 sample references to support IR 8011 implementation
- 1470 • **Does not** restrict IR 8011 solution developers and adopters⁶⁸ from developing their own
- 1471 tests

⁶⁷ Ibid.

⁶⁸ Section 1.3 describes IR 8011 solution developers and adopters.

4. Conceptual IR 8011 Implementation and Considerations

This overview volume and subsequent capability-specific volumes do not prescribe *how* to implement the IR 8011 methodology, only *what* to consider when implementing it. This section shares a vision for a conceptual IR 8011 solution to illustrate a potential operationalization of the methodology. The objective is to show the relationships between the different IR 8011 elements and the mechanics for the proposed automated control testing to convey the concepts presented in previous sections.

There are two aspects to the *implementation* of the IR 8011 methodology: (1) the operationalization of the IR 8011 methodology⁶⁹ and (2) the adoption of an operationalized solution.⁷⁰ While NIST is not engaged in the development of any solution based on the IR 8011 methodology, considerations for the design, development, and adoption of solutions based on the fundamental IR 8011 concepts are presented. Conceptual implementation examples using simplified illustrations are intended to show the dynamics of the IR 8011 methodology and how the different components in the methodology interact with one another.

The operationalization of the IR 8011 methodology focuses on the ability to develop an automated process to identify deviations between the desired state of a control implementation and its actual state. The detection of deviations can then determine whether the control is operating within an expected range or threshold or if there is non-conformance in the implementation that requires further manual investigation.

The envisioned operationalization discussions in this section are tailored with the two IR 8011 implementation groups in mind: *solution developers*⁷¹ and *solution adopters*. Organizations have the option to adopt an existing solution that can provide automated control testing based on the IR 8011 methodology, develop their own solution, or acquire a solution. Some organizations may already have many of the tools needed to implement automated control testing functionality. This discussion can help identify some of the necessary tools and functions that could help an organization leverage the IR 8011 automated control testing methodology to support its continuous monitoring strategy.

Fig. 7 depicts a simplified view to illustrate the two potential paths for implementing the IR 8011 methodology. The following subsections discuss the figure in the context of both the *solution developer* and the *solution adopter*.

⁶⁹ An example of an operationalized IR 8011 methodology is the integration of sub-capability tests within a security tool, such as a GRC application.

⁷⁰ An IR 8011 solution can be any product or service that uses automation to identify deviations between the actual state of a control implementation and its desired state. An example of an operationalized IR 8011 solution adoption is an organization's use of a GRC solution in support of its continuous monitoring program.

⁷¹ This includes solution providers and automated control test service providers.

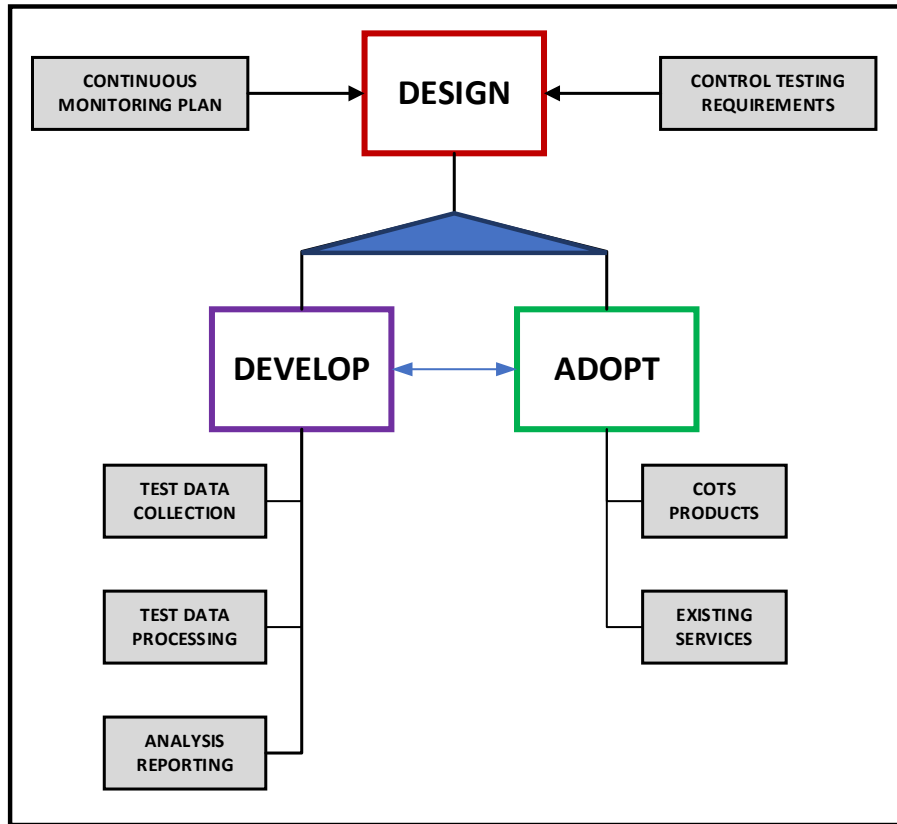


Fig. 7. Possible implementation paths

In order to achieve operationalization, the following prerequisites are considered:

- Testable controls have been identified based on the actual state data available in collection systems that were either provided by common control providers or at the system level.
 - A *collection system* (i.e., a system to collect actual and desired state specification data) is in place to support the collection of machine-readable data from test objects that *can* be subjected to automated testing.
 - *Automated methods* that can compare the desired state specification against the actual state identified by the collection system are used by the collection system to determine whether the comparison results are within risk tolerance.
- A control assessment or control test objective or procedure exists for each testable control selected.
- Control implementation is documented in a system plan.⁷²

⁷² This refers to a system security plan, system privacy plan, or cybersecurity supply chain risk management plan.

- 1517 • The organization identifies the following as part of the control implementation:
 - 1518 ○ **Specific test objects** that *can* be subjected to automated testing to determine
 - 1519 whether test objectives are met within specific thresholds
 - 1520 ○ The data to be collected as part of automated testing that demonstrates the
 - 1521 **actual state** of the implementation for the specific test objects that can be
 - 1522 subjected to automated testing
 - 1523 ○ The **variances permitted** in the actual state of the implementation for the
 - 1524 specific test objects that can be subjected to automated testing based on
 - 1525 organization- or system-defined risk tolerances

1526 **Collectors and Collection System**

1527 Once actual state and desired state data are expressed in machine-readable format, the
1528 values of the actual state and the desired state specifications can be compared via
1529 automated means.

1530 The automated control testing model proposed by IR 8011 requires that data about the
1531 desired state specification is communicated to a *collection system* by the organization
1532 managing the system. The collection of actual state values can be achieved through
1533 *collectors* (e.g., scanners, agents, clients, appliances, data ingest processes, data feeds from
1534 other devices or components). It is assumed that collectors are configured and implemented
1535 to provide reliable, valid, and accurate data that is timely and complete (see Sec. 3.1.6.2 for
1536 *timeliness* and *completeness* metrics). Additional effort may be necessary to standardize the
1537 data structures across different collectors to ensure the communication takes place.

1538 The *collection system* manages the collectors, retrieves actual state data, collects desired
1539 state data, and compares the actual state to the desired state specification to identify non-
1540 conformance (i.e., the variances and gaps defined in the sub-capability test definition for
1541 each control item being tested). A conceptual collection system, depicted in Fig. 8, illustrates
1542 the potential internal workings of an IR 8011 solution.

1543

Automated Testing Data Collection and Analysis

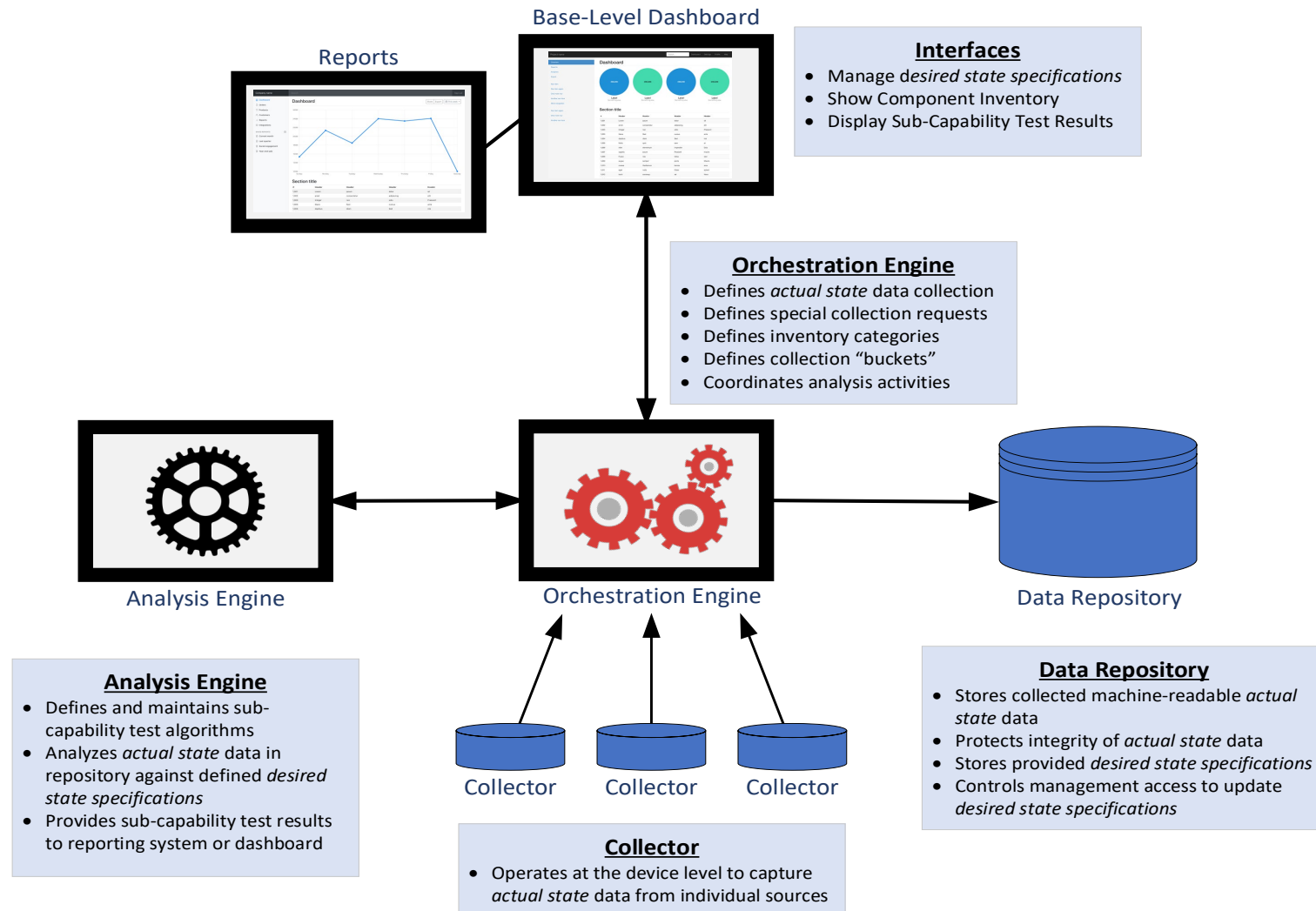


Fig. 8. Conceptual collection system

1546 A conceptual collection system includes:

- 1547 • The collector functions of the collection subsystem which capture object⁷³-specific
1548 machine-readable actual state data.
- 1549 • An orchestration engine for collector activities to retrieve time- and event-driven data
1550 and to coordinate time- and event-driven communications with a continuous
1551 monitoring dashboard or similar reporting management interface.
- 1552 • A repository to store data and protect the integrity of the stored actual state data. The
1553 data repository stores the machine-readable actual state data collected from objects.
1554 The repository also stores the machine-readable desired state specifications provided by
1555 system management through the collector system interface. Repository access controls
1556 can be used to protect the integrity of collected actual state data by enforcing read-only
1557 access to the analysis engine and interfaces. To protect the integrity of the desired state
1558 specifications, the repository and orchestration engine access controls can work
1559 together to restrict access to individuals authorized to apply changes.
- 1560 • An analysis engine to identify non-conformance and the event-driven data collection
1561 needed. The analysis engine uses the repository data to compare the actual state data
1562 against the defined desired state specifications using pre-defined algorithms. The results
1563 are made available to the reporting interface through the orchestration engine.
- 1564 • A graphical user interface⁷⁴ and reporting functions⁷⁵. Multiple user interfaces may
1565 exist, including a general dashboard to display the status of the collectors and the
1566 orchestration engine, and a reporting interface to provide reports of the sub-capability
1567 test results from the analysis engine. Authorized users may be able to generate pre-
1568 determined reports or perform ad hoc queries using the data available in the repository.

1569 To automate the comparison of the actual state and desired state specification, the collection
1570 system's analysis engine⁷⁶ performs the following:

- 1571 • **Accesses/reads the desired state specifications** for each sub-capability test that is
1572 applicable to each item being tested.
- 1573 • **Collects** the matching **actual state values** for each item being tested.
- 1574 • **Compares the actual state with the desired state specification data** for each
1575 combination of sub-capability test and control item to be tested with minimal human
1576 intervention.
- 1577 • **Reports** the resulting variances **to a dashboard** for prioritization and response⁷⁷.

⁷³ For example, a component device in the test boundary.

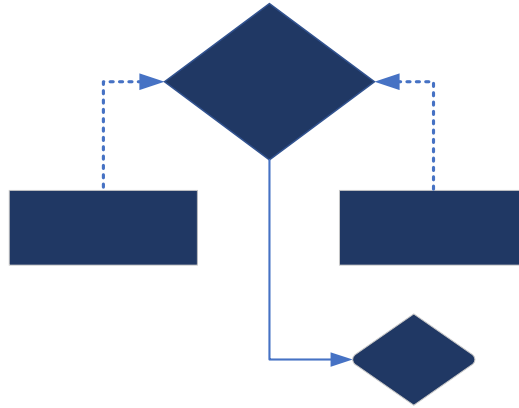
⁷⁴ This refers to a front-end interface.

⁷⁵ The reporting function is expected to be minimal by design because data is sent directly to the continuous monitoring dashboard.

⁷⁶ The analysis engine is a component of the collection system that is responsible for the actual testing.

⁷⁷ A risk scoring methodology is necessary to automate the computation of priorities and responses. Risk scoring is out of scope for this publication.

1578



1579

1580

Fig. 9. Simplified view of automated testing

1581

Although the implementation details and approaches for the implementation of a collection system are outside of the scope of this publication, the following implementation aspects can be considered:

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

- The collection system manages desired state specification data for each automated control testing implementation with access to up-to-date specification information, including organization-defined parameters (ODPs) and variations of policy due to tailored implementations and other approved policy deviations (e.g., waivers).
- Dashboards are used to consolidate reports and display results in a meaningful way.
- The collection system and organization dashboard work together to represent organization-defined desired state specifications, such as:
 - Inventories of system components (e.g., authorized devices and software) that are provided by the collection system, which provides the functionality to automatically import or enter inventory-related data
 - Values for organization-specific configuration settings that are managed (e.g., collected, processed, store, presented) by the automated test list in the organization dashboard

1597

1598

For security-related information generated by the collectors and processed by the collection system to be of maximum usefulness, all non-conformances on a system are mapped, including:

1599

1600

1601

1602

- Non-conformances in the controls implemented at the **system level**
- Non-conformances in **common controls** that the system inherits
- Non-conformances in otherwise unrelated **test objects** that allow an attack path to be established and adversely affect the system⁷⁸

1603

1604

For the collection system collectors to detect and process all three types of non-conformances, test objects being assessed/monitored are grouped into the following categories:

⁷⁸ The test boundary tends to be the entire network, including data about the most relevant test objects outside of the test boundary.

- Test objects and non-conformances **within the test boundary**
- Test objects and non-conformances **from common controls** that the system inherits

This allows the organization dashboard to compute risks from both groups.

Authorization boundaries are used to ensure that systems are distinct to facilitate security management, responsibility, and accountability. For instance, SP 800-53 control CM-08a.3, System Component Inventory, requires system components to be assigned to a specific system and ensure that system components are not duplicated in another system component inventory.

Control CM-08: System Component Inventory

Control:

- a. Develop and document an inventory of system components that:
 1. Accurately reflects the system;
 2. Includes all components within the system;
 3. **Does not include duplicate accounting of components or components assigned to any other system;**
 4. Is at the level of granularity deemed necessary for tracking and reporting; and
 5. Includes the following information to achieve system component accountability: [Assignment: organization-defined information deemed necessary to achieve effective system component accountability]; and
- b. Review and update the system component inventory [Assignment: organization-defined frequency].

Identifying and Communicating Requirements

A critical factor in systems engineering is the identification and communication of requirements. Both developers and adopters agree on the requirements related to the solution, the system or enterprise architecture, and other factors that can enable or prevent the testing and/or monitoring of controls. The IR 8011 solution is likely to be implemented organization-wide, so identifying and communicating requirements can help ensure a smooth integration. Such communication is important when implementing an in-house-developed solution, such as ensuring that the IR 8011 solution can reliably test controls using sub-capability tests that are based on the right versions of the control catalog and assessment procedures in use by the system/organization.

There are specific requirements for any IR 8011 solution to work:

- The ability of the collection system to collect machine-readable data formats

- The standardization of the data formats across all collectors within the collection system to support the analyses
- The ability of associated test objects to be tested using automated methods
- The ability of the collection system to capture operational data from all objects within the test boundary or within an organization so that automated control testing can remain effective

From an IR 8011 solution development standpoint, assessment objectives for controls and control items are necessary to formulate sub-capability tests.

The IR 8011 solution may also consider data collection from stand-alone devices, such as Internet of Things (IoT) implementations, operational technology (OT) enclaves, and operational environments. Although these devices are within the scope of RMF or control implementation, they may be physically or logically isolated from the organizational architecture, which could result in being excluded from the test boundary.

4.1. IR 8011 Solution Developer's Perspective

The solution developer considers the implementation of the methods designed to capture, analyze, and report on the results associated with testable control items. They focus on either (1) building an entire automated control testing solution or (2) providing customized solutions that are designed to be integrated with existing risk management tools. For instance, vendors may develop products designed to provide adoptable solutions that can integrate with a variety of collection systems and provide standardized analysis and reporting functions.

The solution development discussed in this section is based on the efforts to model the concepts associated with the automated control testing methodology. Discussion of development strategies based on the IR 8011 methodology is encouraged to arrive at solutions that best serve a variety of use cases across industries and economic sectors.

4.1.1. Build a Custom IR 8011 Solution

When building a custom IR 8011 solution, it is necessary to establish the data relationships between the testable control items and to capture actual control implementation results. The solution then leverages sub-capability tests that verify whether control implementation is within acceptable limits. Applications of this custom solution include control testing on a per security capability basis and control testing on a smaller scale, such as in support of internal automated control testing.

4.1.1.1. Design for Automated Control Testing

To automate the control testing process based on the IR 8011 methodology, a data collection system is necessary to collect machine-readable data from testable objects. The solution developer defines the analytical functions for comparing the desired state specification against the actual state values obtained by the collection system with consideration for any variances,

thresholds, and other acceptable ranges. The analyses support the ability to report deviations between the actual state of a control implementation and the desired state expectation using machine-readable and automated methods. The automated analysis of the deviations using sub-capability parameters can help determine whether the control is operating within a defined expected range or whether there is a weakness in the implementation that requires investigation and possible remediation. The resulting report advises the system owner and other organizational risk management personnel on whether the comparison results are within risk tolerances.

The elements of the solution development are illustrated in Fig. 10:

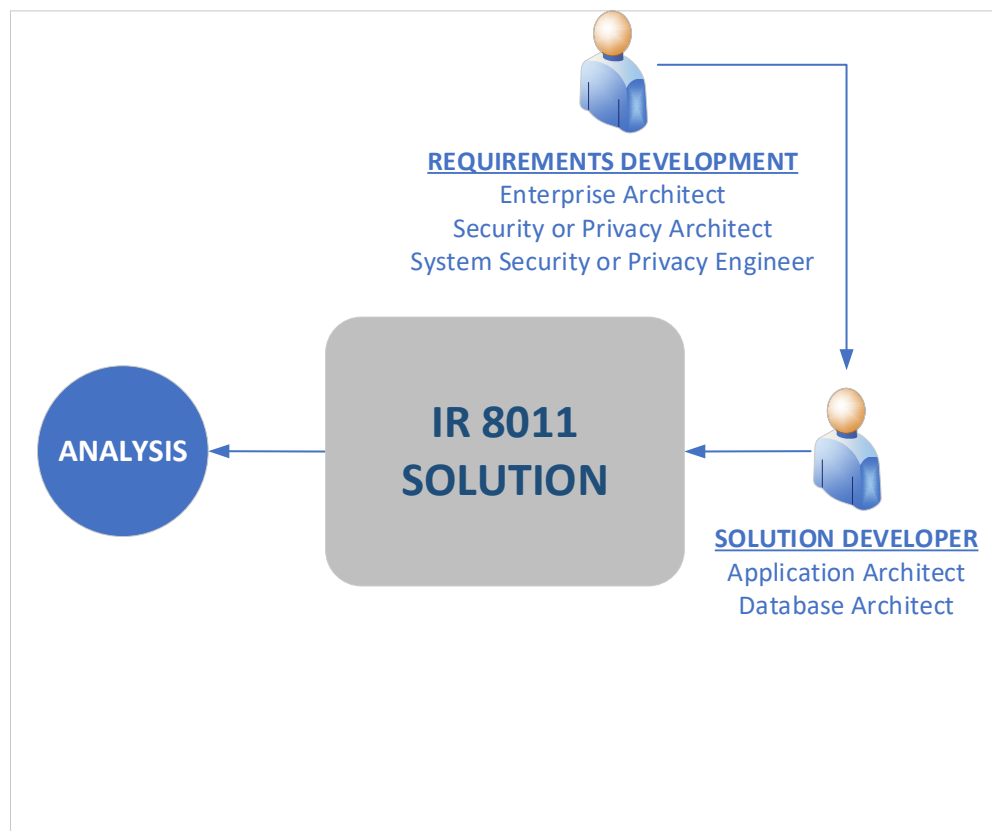


Fig. 10. Solution development elements

The Control Testing Process

Even if only designing a component of an overall system, it is critical to have a full understanding of the IR 8011 methodology to ensure maximum compatibility if and when third-party implementations are integrated into the same continuous monitoring system. The interaction between key IR 8011 elements within the control testing process is illustrated in Fig. 11:

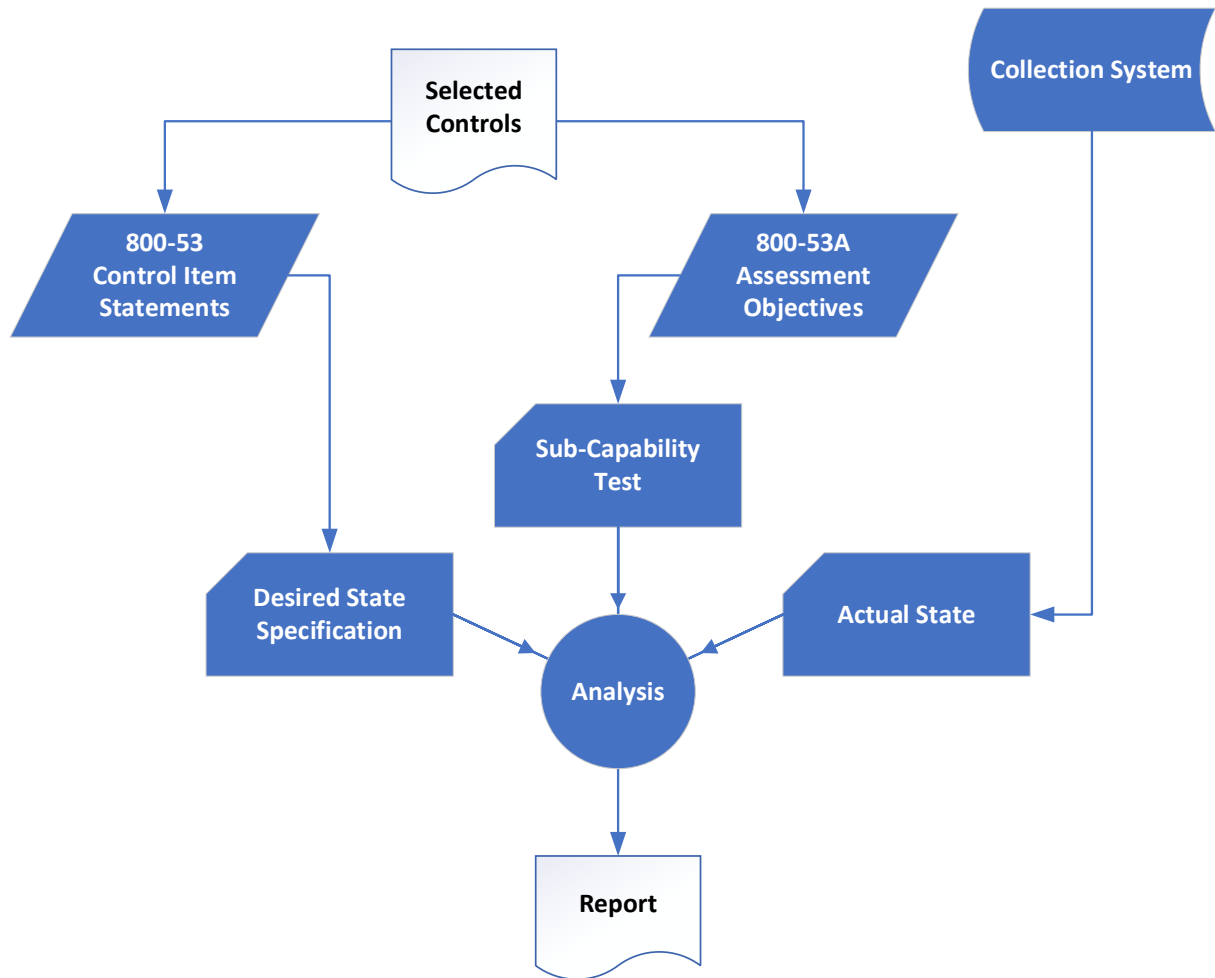


Fig. 11. Control testing process overview

First, the organization selects and allocates controls from the SP 800-53 catalog and identifies the controls in an appropriate system plan. The system plan also documents the relationship of the control implementations, whether they are inherited from a common control provider, system-specific, or involve a hybrid implementation [SP800-53]. For each control statement in the selected controls, the control implementation details are identified in the system plan.

Organizations have the flexibility to create additional controls outside of the SP 800-53 control catalog as part of the [SP800-53B] control baseline tailoring process. These organization-defined controls may supplement the security, privacy, or cybersecurity supply chain risk management plans or be included in a control overlay that addresses a specific technology or type of operational environment.

[SP800-53A] provides a methodology for creating assessment objectives for organization-defined controls. Information in the IR 8011 volumes provides a model for defining the applicable sub-capability tests associated with the assessment objective for organization-defined controls.

In addition to the continuous monitoring plan and the control assessment plan, the organization identifies a control test plan⁷⁹ while considering the following factors:

- The scope of the automated control testing functionality across the entire organization, for specific systems, or limited to specific component types
- The security capabilities and sub-capabilities that correspond to the attack and defend steps used within the scope of the automated control testing functionality
- The control testing requirements for generating and standardizing data to be collected as part of automated testing
- The testable objects⁸⁰ to determine whether test objectives are met within specific thresholds
- The assessment objectives for the control implementation, with consideration for the testable objects
- The collectable data elements that demonstrate the actual state of implementation for the testable objects
- The sub-capability tests applied based on defined use cases associated with each test objective and collectable data element
- The variances permitted in the actual state of the implementation for the specific objects that can be subjected to automated testing based on organization- or system-defined risk tolerances

Further research within the community may provide insight into improved designs for automating control testing and recommended practices for maximizing efficiency and effectiveness.

Implementation Strategy

After determining the design and approach for implementing an IR 8011 solution, the organization decides whether to adopt an existing solution, develop an internal solution, or use a hybrid solution.

4.1.1.2. Determine Necessary Data Sources

Determine what machine-readable data is necessary to support the analysis of the test results, identify the sources of the data, and periodically review data sources for any changes.

A typical solution may include the following machine-readable data:

- **Control Statements** — Objectives of the control (i.e., what the control is intended to protect). Controls change over time, and as the organization tailors controls, control baselines, or creates new organization-specific controls, it is important to keep the IR 8011 implementation up to date with current control information,⁸¹ including changes

⁷⁹ This term is specific to IR 8011.

⁸⁰ Testable objects are specific objects that can be subjected to automated testing.

⁸¹ How IR 8011 keeps datasets up to date is outside of the scope of this publication.

- in how desired state specifications are identified, how sub-capability tests are determined, and how changes in the collection system impact the analysis.
- **Assessment Objectives** —Determination statements based on control statements. Not all control-based frameworks provide associated assessment procedures with assessment objectives that are specific to the controls. This is taken into consideration when developing solutions for control-based frameworks other than the NIST RMF and SP 800-53.
 - **Sub-Capability Test** — The actual tests that can be automated when testing controls, specifically control items. Sub-capability tests are derived from control statements and control assessment objectives.

4.1.1.3. Define Data Relationships

Organizations determine which data management platforms best support their automated control testing solution. For simplicity, the focus is set on the relationships between the control statements and the control assessment objectives from the SP 800-53 and SP 800-53A datasets. Fig. 12 provides an example of a defined data relationship between these elements.⁸²

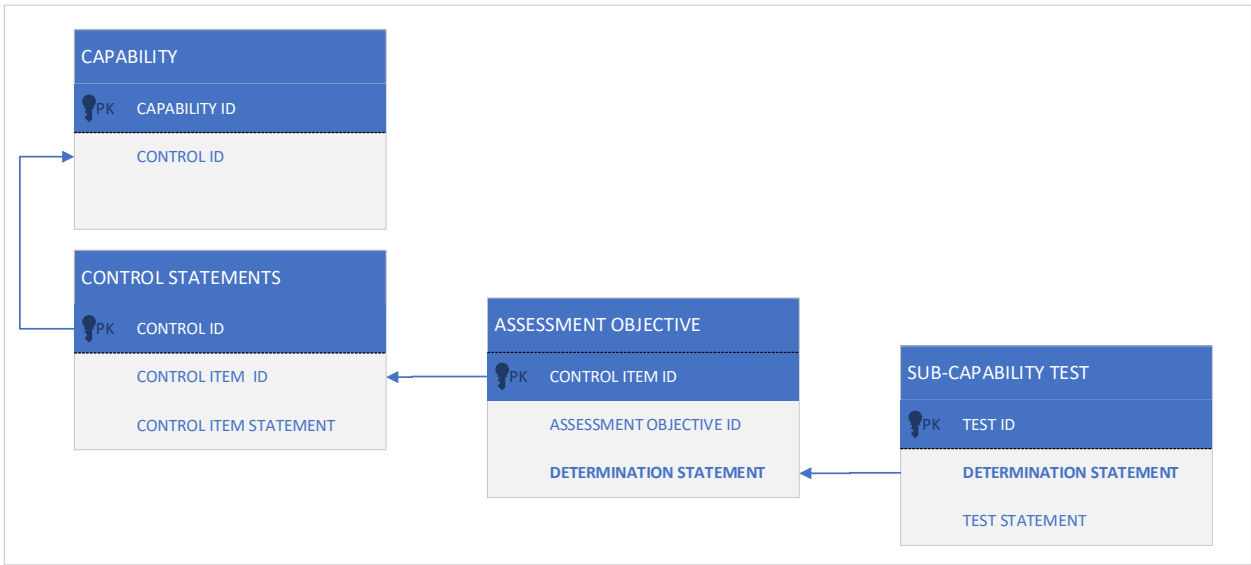


Fig. 12. Sample data relationships

These data relationships are reviewed periodically to consider changes to the data sources and any tailoring performed by the organization or system to control statements or assessment objectives.

Data structures that express the desired state specification and sub-capability test variances are compared with outputs from the collection system that provides actual state information about

⁸² In Fig. 12, PK represents the *primary key*.

system operations. Recognizing that the collection system may involve distinct data outputs from the system's endpoints, developers may require additional methods to transform the machine-readable actual state data from multiple sources to match the data structures that express the desired state specifications. The initial creation of these data transform methods is intended to standardize the data so that the data elements can be compared, which may require significant manual effort to ensure that the query results that support the comparison analysis align with the expected results of the automated process. If the desired state specifications and actual state data structures and formats are mismatched, the analysis process is more likely to produce inaccurate results about the state of the control implementations being tested. Periodic reviews of the data sources and relationships are necessary to address updates made to the applications and services that comprise the collection system.

Table 35 summarizes⁸³ the data relationship between IR 8011 elements:

Table 35. Data Relationship Between IR 8011 Elements

IR 8011 Element	Data Relationship
Sub-Capability Tests	1 sub-capability test supports 1 sub-capability (1:1)
Determination Statements	1 determination statement is supported by 1 sub-capability test (1:1) 1 determination statement can support 1 sub-capability (1:1)
Control Items	1 control/control item can support multiple security capabilities (1:∞) 1 control/control item can support multiple sub-capabilities (1:∞) 1 control item can have multiple determination statements (1:∞) 1 control can have multiple control items (1:∞)
Sub-Capabilities	1 sub-capability can be supported by many controls/control items (1:∞) 1 sub-capability can support 1 security capability (1:1)
Attack Steps	1 attack step can be addressed by multiple security capabilities (1:∞) 1 attack step can include multiple attack actions (1:∞) 1 attack step can be addressed by 1 defend step (1:1) 1 attack action can be supported by multiple defend actions (1:∞)
Defend Steps	1 defend step can include multiple defend actions (1:∞) 1 defend action is translated into 1 sub-capability (1:1)
Security Capability	1 security capability can be supported by multiple sub-capabilities (1:∞) 1 security capability can be supported by multiple controls/control items (1:∞) Multiple security capabilities can address multiple attack steps (∞:∞) 1 security capability can address multiple defend steps (1:∞)

4.1.1.4. Define Solution Functionalities

Based on the requirements for the automated control testing solution, the developer defines specific functionalities as features of the IR 8011 solution. The following are some conceptual

⁸³ Non-prescriptive: implementers have the flexibility to propose alternative relationships between IR 8011 elements.

examples of functionality that could support the basic intent of IR 8011 and potential new capabilities.

Security Capability Narrative

A potential feature of an IR 8011 solution is the ability to issue or display narratives for each security capability. These narratives can automatically feed, for instance, a security management application (e.g., a GRC application or a dedicated control assessment tool). A conceptual example of an interface that provides narratives for a security capability is illustrated in Fig. 13⁸⁴:

CAPABILITY NARRATIVE	
CAPABILITY	HWAM
DESCRIPTION	Hardware Asset Management
DESIRED RESULTS	Ensure that unauthorized and unmanaged devices are identified to prevent use by attackers as a platform from which to extend compromise of information systems.
CONSIDERATIONS	Maintain a list of authorized hardware and who manages it. Treat other hardware discovered within the authorization boundary as non-conformance.

Fig. 13. Sample capability narrative in data

Control/Control Item Narrative

The NIST [CPRT] is used as the source for the [SP800-53], [SP800-53A], and [SP800-53B] machine-readable datasets. These datasets can be integrated into an IR 8011 solution to provide the control/control item narratives which include the control statements and assessment objectives necessary to define desired state specification and sub-capability test parameters.

As NIST updates the SP 800-53 control catalog, the SP 800-53A assessment objectives and procedures, and the SP 800-53B control baselines, the updated machine-readable files available from the [CPRT] can be ingested into the developed solution or component, such as a database that supports the IR 8011 solution.

Organizations may use other sources for control statements and assessment objectives based on applicable requirements.

⁸⁴ These screenshots serve as illustrations only. Actual design and approach for displaying and sharing narratives are at the discretion of the developer.

1808 Using standardized datasets provides for a consistent set of control information that relates
1809 to the organization's definitions for security capabilities, sub-capabilities, and sub-capability
1810 tests that enable automation for supporting control testing.

1811 Sub-Capability and Sub-Capability Test Narrative

1812 A conceptual example of an interface that provides narratives for sub-capabilities and sub-
1813 capability tests is illustrated in Fig. 14:

SUB-CAPABILITY and SUB-CAPABILITY TEST Description

Prevent unauthorized devices Sub-Capability and Sub-Capability Test HWAM-F01-Test

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

Sub-Cap Name	SubCap-Desired Result
Prevent unauthorized devices	Prevent or reduce the presence of unauthorized devices, thus reducing the number of potentially malicious or high-risk devices.

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

SubCap Test ID	SubCap Test Name	Test Criteria Summary	Test Criteria Notes
HWAM-F01-Test	Test for Unauthorized Devices	Device is present in an authorization boundary for which it has not been authorized.	1) The desired state specification is a list of all devices within an authorization boundary. 2) The actual state of the device or component inventory within the organizationally-defined tolerance is determined by the collection system. 3) Non-conformance is identified where a device is in the actual state but not in the desired state specification, indicating an unauthorized device in an authorization boundary.

Fig. 14. Sub-capability and sub-capability test description

1816 Like the control item narrative, the sub-capability and sub-capability test narratives could be
1817 part of the same interface as the capability narrative.

1818 Queries for Identifying Testable Controls and Control Items on a Per Security Capability Basis, 1819 Control Family Basis, and Control Baseline Basis

1820 The IR 8011 methodology provides an approach for identifying testable controls and control
1821 items on a per security capability basis. An implementation of IR 8011 may be expanded to
1822 offer the ability to identify testable controls and control items on a per control family and
1823 control baseline basis. The idea is to utilize the existing implementation to group testable
1824 controls by control family and/or control baselines. While the grouping of testable controls
1825 on a per security capability basis supports continuous monitoring, grouping testable controls
1826 on a control family and/or control baseline basis can also support control assessments.

4.1.1.5. Analysis and Reporting

This overview volume focuses on the methodology for developing sub-capability tests to identify testable controls in support of a specific security capability. In operationalizing the methodology, developers elaborate the logic that can turn these sub-capability tests into an *operational engine* that processes the sub-capability tests. This operational engine executes the tests by computing actual state values and desired state specifications and reporting results.

The collection system [Fig. 8] obtains and interprets machine-readable actual state data with the potential to perform data validation checks as the data is processed. A logic structure of the elements that support the *analysis* portion of the methodology and a decision flow with potential questions to be asked as part of the automated testing of a control are illustrated in Fig. 15:

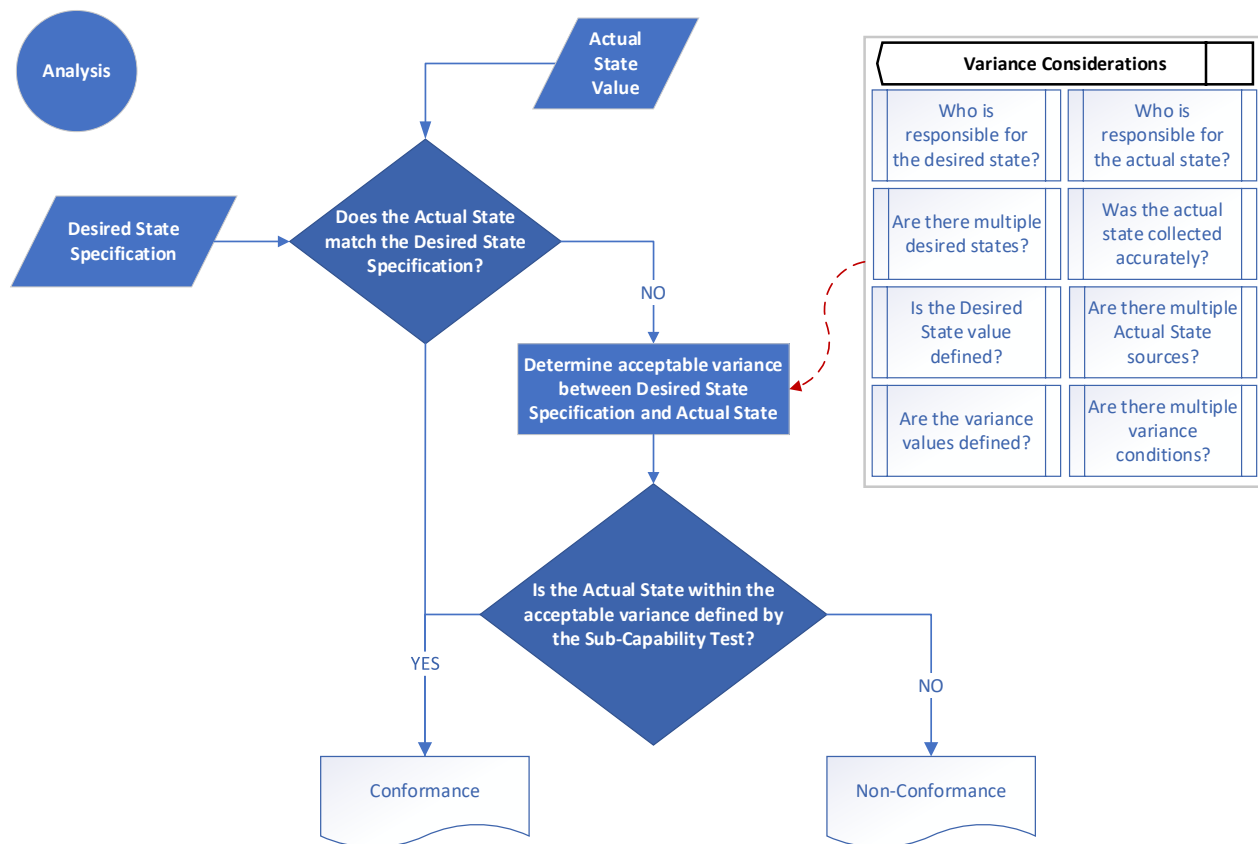


Fig. 15. Sample analysis logic

The analysis engine may be a suite of applications that generate, transmit, and store different datasets that may require additional methods to normalize and standardize the data to support the analysis of the datasets using organization-defined data definitions.

For example, if the sub-capability test determines that the desired state and the actual state do not match, the automated analysis can evaluate the variance based on the following considerations:

1846 • Who is responsible for the desired state and the acceptable variances?

1847 ○ Is the desired state defined?

1848 ○ Are there multiple desired state values being evaluated?

1849 ○ Are the acceptable variance values defined?

1850 • Who is responsible for the actual state?

1851 ○ Was the actual state information collected accurately?

1852 ○ Are there multiple actual state sources?

1853 ○ Are there multiple acceptable variance values?

1854 The analysis can use machine-readable values for the acceptable variance values as thresholds
1855 to determine whether the actual state value conforms to the approved organization risk
1856 requirements for the operating environment. If the actual state value exceeds the threshold of
1857 the acceptable variance value, the test would be expected to report a non-conformance.

1858 The organization dashboard or equivalent management interface provides the required
1859 documentation of the test results. A dashboard or management interface includes a grouping
1860 of tested objects by boundary and inherited common controls. The report⁸⁵ provided by the
1861 dashboard includes:

1862 • Detailed lists of non-conformances by the system, responsible party, and/or device

1863 • Detailed lists of the non-conformances that contribute the most overall risk

1864 • Organization-defined prioritization of which non-conformances to address first

1865 • Summary levels of risk by security capability, risk response manager, and system

1866 • Estimated consequences of the given level of risk to facilitate risk management,
1867 investment, and other business/mission decisions

1868 **4.1.1.6. Develop, Use, and Maintain an IR 8011 Database**

1869 Depending on the operational application of the IR 8011 methodology, an IR 8011 database can
1870 be a valuable tool for developing sub-capability tests and maintaining the necessary narratives
1871 to support control testing or automated control assessments. Examples of database use
1872 include:

1873 • Data store for SP 800-53 control statements and discussion text

1874 • Data store for SP 800-53A assessment procedures

1875 • Data store for SP 800-53B control baselines

1876 • Control search via keywords

⁸⁵ The test report information generated by the organization dashboard is acceptable whether it is printed on paper or presented electronically.

- Assessment procedures search via keywords
- Queries for identifying testable controls and control items:
 - On a per security capability basis
 - On a control family basis
 - On a control baseline basis
- Data store for security capability, control item, and sub-capability narratives

To ensure consistency with existing RMF materials, public datasets for SP 800-53, SP 800-53A, and SP 800-53B from the [CPRT]⁸⁶ can be imported into a database to create a testbed to explore and test the methodology. By starting with the NIST-provided data that follows a well-defined data structure, changes to the control catalog can be easily imported to replace the outdated catalog without having to change the table, query, form, or report definitions as part of the database maintenance process.

Developers can create additional data relationships and queries to further parse their local datasets to arrive at a representation of the controls, control items, and control assessment determination statements that support the organization's implementation of an automated control testing solution.

IR 8011 Database

NIST does not provide an IR 8011 database as supplemental material to this publication. The SP 800-53 control catalog, the SP 800-53A assessment objectives, and the SP 800-53B control baselines are provided as human- and machine-readable datasets via the [CPRT]. These datasets can be used to identify sub-capabilities, controls associated with the capability and sub-capability, and sub-capability tests by solution developers in support of their IR 8011 operationalization efforts.

Additional tables can be defined to store the organizationally defined security capabilities, sub-capabilities, sub-capability tests, and other data elements. Relationships can also be defined, such as the one-to-one relationship between sub-capabilities and sub-capability tests. With an appropriately designed schema, queries can be developed to output the sample data tables that are referenced in the capability-specific volumes. These queries can be used to create the control and capability narrative forms/reports that demonstrate the relationships between the elements of the automated control testing capability.

4.1.1.7. Control Search (via Keywords)

Keyword searches can be used to identify the control items, or parts of control items, associated with specific attacker and defender steps and actions. Keyword identification is

⁸⁶ The CPRT provides datasets in spreadsheet and JSON formats that can be imported/ingested for manipulation using database management and other applications.

generally a manual process to (1) identify the associated control items and (2) validate that the resulting control items are relevant to a specific security capability. The developer responsible for the identification of keywords possesses sufficient knowledge of controls, control families and the relationships between them, control assessment procedures, threats, risks, and other factors that influence the selection of keywords to greatly increase the probability of finding the right controls to support a security capability.

Defining the scope of the search is important. In the SP 800-53 control catalog, the listing of security and privacy controls is preceded by general guidance and proceeded by references, glossary, acronyms, and other supporting information. Searching the entire SP 800-53 control catalog in portable document format (PDF) may skew the results of a search. For better results, the scope of the search is only on the control statements and on the guidance text in the Discussion portion of the control. Rather than using the PDF version of SP 800-53 for control searches by keyword, consider using the NIST control datasets that can be downloaded from the [CPRT] and imported into a database or spreadsheet. The datasets exclude text from the publication's front matter and appendix content.

One of the challenges of using keywords to find controls is the fact that a positive match only occurs if the keyword⁸⁷ is found somewhere in the control statement and in the guidance within the control discussion text.⁸⁸ To increase the chances of producing accurate results, keywords are selected to describe a specific context or a situation, a threat, an object, an activity, and any descriptor that can be used to identify the necessary protection and to ensure a security capability is in place.

Boolean-based search tools⁸⁹ can enhance the control searches through the use of logic statements. The order of Boolean operators and the grouping of search expressions can also impact the result set, where "(Expression 1 AND Expression 2) OR Expression 3" provides different results than "Expression 1 AND (Expression 2 OR Expression 3)."

Achieving the goal of producing results with minimal errors – both false positives and false negatives – means identifying controls that can be effective in meeting the defense objectives of a security capability. To produce accurate results, the keywords chosen are relevant not only to the defense actions but also to the attack actions identified for the security capability. Both the quantity and the quality of the keywords used can impact the accuracy of the results, **as does the subject matter expertise of the implementer to identify and select additional controls.** A larger number of keywords can reduce the number of accurate results, but a small number of keywords can fail to produce relevant control items in the search results or return a large result set that requires greater manual effort to review.

Variations of the keywords are considered to increase the probability of greater positive matches. The IR 8011 solution could orient the user, for example, to consider using synonyms to search by keywords or automatically identify keyword synonyms for the user to consider as additional keywords.

⁸⁷ Either the exact keyword or a variation of a keyword; for example, using wildcards.

⁸⁸ This is specific to controls in the SP 800-53 catalog.

⁸⁹ For instance, a manual search may be supported by electronic functions in office productivity software, such as "search" or "find," as opposed to visually searching controls which is time-consuming and may not yield accurate results.

Control searches by keyword have limitations. A control or control item may be missed if the keyword selected is either not applicable or is a potential variant of an existing word in the catalog. Developers are encouraged to consider synonyms and keyword variations to increase the probability of finding the correct control/control enhancement. For example, possible variations of POA&M may include:

- POAM
- poam
- plan of action
- plan of actions
- milestones
- plan of action and milestones
- plans of action

The testable control sample set in the security capability-specific volumes are identified using Boolean operators with manually determined keywords to obtain a sampling of controls that can be tested via automated means and in support of a specific security capability. Developers may arrive at different results based on the keywords and the logic used. As technologies evolve, developers may be able to leverage machine-learning and natural language processing models to help identify controls within a catalog for a specific security capability.⁹⁰

4.1.2. Integrate IR 8011 Sub-Capability Tests into Existing Solutions

Developers determine whether the developed solution can be integrated or interconnected with an existing management application, such as a GRC system. An effective GRC application can help manage security and privacy risks by supporting the implementation and monitoring of certain controls. GRC applications often provide a central repository of security-relevant data, including desired and actual state data, so they can be excellent candidates for sub-capability test integration. In fact, the integration of sub-capability tests into GRC and continuous monitoring applications may be the most common IR 8011 operationalization method.

Whether the integrated sub-capability tests are organized by security capability or not, authorized users of the GRC application could manually run sub-capability tests or schedule them to be automatically executed according to a predefined schedule. Depending on the role of the individual or service executing the sub-capability test and application use, the automated testing can support self-assessment activities, external or independent assessment activities, and internal monitoring activities. A continuous monitoring solution can provide an organization with the ability to automatically test controls throughout the system life cycle.

⁹⁰ These models have not been applied to the IR 8011 project at this point. Further community research on the subject is encouraged to improve the identification of controls for the security capabilities identified in the IR 8011 project.

1984 When considering the integration of automated testing features or functions into a GRC
1985 application, developers can consider the IR 8011 methodology for determining the
1986 requirements for collecting, analyzing, and reporting data to identify non-conformances within
1987 the scope of the GRC application.

1988 **4.1.3. Derive Sub-Capability Tests Outside of IR 8011 Scope**

1989 IR 8011 primarily supports the RMF *Monitor* step, but the methodology can be used to support
1990 some of the activities more closely associated with the RMF *Assess* step, such as assessing
1991 specific controls via the test method.

1992 The IR 8011 sub-capability test development process can be used to derive sub-capability tests
1993 for any control family or to support a new security capability that is not covered by the IR 8011
1994 series. By providing an IR 8011 solution that can support the testing of controls on a control
1995 family basis, for example, the developer can support control assessments in general, allowing
1996 for greater customization and additional efficiencies for control assessments.

1997 **4.1.4. Control Testing as a Service**

1998 The potential implementation of the IR 8011 methodology can also offer control testing as a
1999 service. Similar to cloud service offerings, *control testing as a service* focuses on providing
2000 software, platform, and infrastructure services to adopters. Whether the service offers an
2001 entire infrastructure required for the proper functioning of assessment and monitoring tasks or
2002 focuses only on sub-capability testing as a simple test or assessment activity, developers
2003 consider all IR 8011 components and their relationships when integrating with the adopter's
2004 infrastructure.

2005 Potential approaches to control testing as a service include:

- 2006 • **Control testing software as a service** (e.g., GRC application integration using the
2007 adopter's platform)
- 2008 • **Control testing platform as a service** (e.g., separate collection system, analysis engine,
2009 dashboards, and other IR 8011 components using the adopter's infrastructure)
- 2010 • **Control testing infrastructure as a service** (e.g., entire IR 8011 off-premises architecture
2011 offering)

2012 Each of these approaches have advantages and disadvantages. For example, control testing
2013 infrastructure as a service does not require an external independent assessor or assessment
2014 team to conduct automated assessments within an organization's boundary if the desired state
2015 specifications and actual state values are reported to the provider. For each approach,
2016 developers consider:

- 2017 • The best way to collect and exchange desired state specifications and actual state data
- 2018 • The location of the collection system (e.g., at the adopter's site, at the service provider's
2019 site)

- 2020 • The location of the dashboard (e.g., at the adopter's site, at the service provider's site)
- 2021 • Architectural changes at the adopter's site

2022 **4.2. IR 8011 Solution Adopter's Perspective**

2023 The solution adopter focuses on the implementation of the methods designed to capture,
2024 analyze, and report on the results associated with testable control items. For illustration
2025 purposes, this section assumes that an IR 8011 solution exists and is ready to be adopted.

2026 For the solution adopter, the primary benefit of IR 8011 operationalization is the ability to
2027 support the automated testing of control items for security capabilities, which reduces the time
2028 for monitoring controls.⁹¹ The long-term objective of adopting an automated control testing
2029 solution is to support the organization's continuous monitoring program. If an IR 8011 solution
2030 is operational when a new system is in development, specifically during the RMF *Implement*
2031 step, the analysis results from the IR 8011 solution may inform system personnel of what
2032 controls are already in place. For example, testable controls associated with system or
2033 application security hardening may be in place by default. An initial test may identify what
2034 control or part of a control is already in place, facilitating the implementation of controls.

2035 The solution adopter relies on tools for the functions that capture, analyze, and report on the
2036 results from testing controls and control items, including tools that comprise the *collection*
2037 *system* for the capture of actual state information from the components implemented within
2038 the scope of the assessment or monitoring. After the tools are in place, the emphasis is on the
2039 review and analysis of the test results. The test results only represent the status of control
2040 implementations at the control item level that can be tested; other parts of the control are still
2041 assessed through other methods. A simplified view from an adopter's perspective is illustrated
2042 in Fig. 16:

⁹¹ The testable controls covered by IR 8011 solutions are limited to select control items within a control and select controls that support a security capability. The assessment of any non-testable control items requires the use of other assessment methods (e.g., examination and interview) to supplement the automated testing. How systems and organizations supplement these tests is outside of the scope of the IR 8011 series.

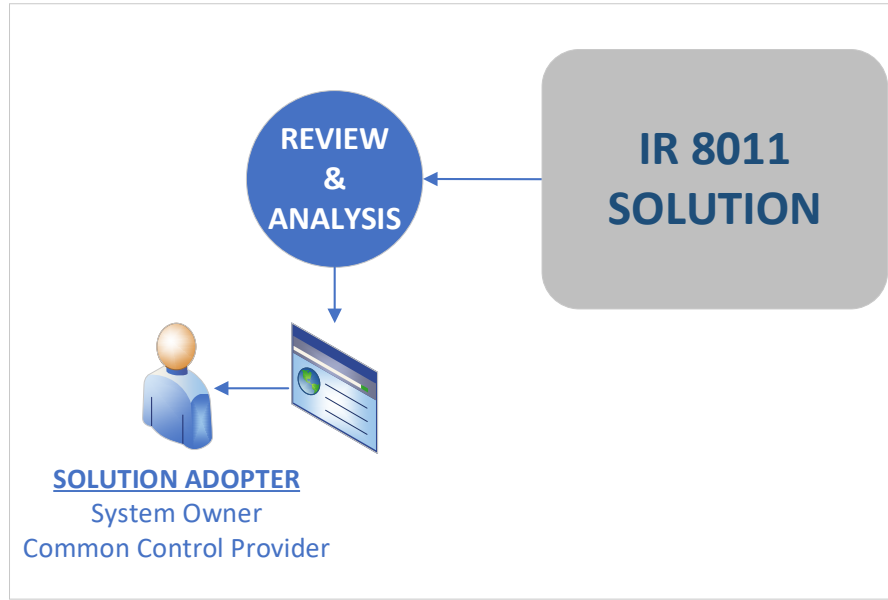


Fig. 16. Solution adoption

4.2.1. Roles and Responsibilities

IR 8011 provides an operational approach to implementing automated control testing. Here, operational roles and responsibilities are defined in addition to the responsibilities associated with risk management processes.

SP 800-37-Defined Management Responsibilities

The security and privacy risk management roles and responsibilities defined in [SP800-37] indicate who has the responsibility and authority to oversee the security of a system and ensure that the security and privacy requirements documented in the system security and privacy plans are met. Responsibility for the operational task of finding and responding to non-conformance on the system is not specified, but the personnel who perform operational roles typically report to management-level roles.

[SP800-37] assigns the management responsibility to discover and respond to security non-conformance at the system level to the system owner and to the system security or privacy officer, as shown in Table 36.

Table 36. System owner and security or privacy officer responsibilities

Role	Responsibilities
System owner	The system owner is an organizational official responsible for the procurement, development, integration, modification, operation, maintenance, and disposal of a system. The system owner is responsible for addressing the operational interests of the user community — including users who require access to the system to satisfy mission, business, or operational requirements — and for ensuring compliance with security requirements.

Role	Responsibilities
System security or privacy officer	The system security or privacy officer is an individual responsible for ensuring that the security and privacy posture is maintained for an organizational system and works in close collaboration with the system owner. The security or privacy officer also serves as principal advisor on all matters, technical and otherwise, involving the controls for the system.

Operational Roles and Responsibilities

The operational roles supplement the management roles defined in [SP800-37]. Additional details are provided with each security capability to clarify how to operationalize automated control testing in addition to the conceptual implementation examples in this section. Each organization has the flexibility to decide the management roles to which personnel performing the operational roles report.

The system owner and security or privacy officer are unlikely to perform the daily operational tasks by which most endpoint security non-conformances are managed (e.g., connect devices to the network, install software, set configuration values, patch software). While they have overall management responsibility for the system and its security and privacy posture, the system owner and security or privacy officer roles can be supplemented by more detailed operational roles as needed in order to execute day-to-day tasks.

Continuous Monitoring Operational Responsibilities

Continuous monitoring operational roles and responsibilities, as shown in Table 37, illustrate operational roles for completing tasks that risk management roles would typically delegate to others (see Table 37).⁹²

Table 37. Example of continuous monitoring operational roles for the HWAM security capability

Role Code	Role Title	Role Description
DeviceMgr	Device manager	Assigned to a specific device or group of devices, responsible for adding/removing devices from the boundary and configuring the hardware of each device (i.e., adding and removing hardware device sub-components), specified in the desired state inventory specification, and may be a person or a group with a group manager.
DesiredStateMgr	Desired state manager	Ensures that data specifying the desired state of the relevant capability is entered into the continuous monitoring system's desired state data; is available to guide the actual state collection subsystem and to identify non-conformance; is needed for both the test boundary and each test object; and resolves any ambiguity about any authorization boundary that presents non-conformances. Authorizers share some of the DSM responsibilities by authorizing specific items (e.g., devices, software products, settings) and defining the desired state. The DSM oversees and organizes this activity.

⁹² For the purpose of this example, not all roles are shown. See the relevant capability volume for a list of sample roles.

2077

2078 The roles defined here are examples to help implement automated testing and response and
2079 to maintain the desired security and privacy posture. The ultimate goal is to ensure that
2080 operational duties are assigned to roles and then to individuals or teams with the capacity to
2081 perform those roles. Depending on the size and complexity of the system, the operational
2082 roles may be full-time positions or performed along with other duties. For example,
2083 organizations may want to subdivide, rename, and/or combine the roles to reflect local
2084 practice. Organizations may also decide to assign continuous monitoring operational roles to
2085 the system owner or security or privacy officer.

2086 A primary output of continuous monitoring is a list of non-conformances that require a
2087 response. Each non-conformance in the list is assigned to predetermined operational roles
2088 and/or teams. The continuous monitoring dashboard can be configured to efficiently
2089 allocate response actions to the appropriate roles/teams given the correct operational role
2090 information to ensure that appropriate response actions are taken. Potential response
2091 actions are suggested in the non-conformance tables but may require the input or approval
2092 of the system owner, security or privacy officer, or other authorizing official if there is a need
2093 for risk acceptance.

2094 Finally, some of the operational roles address non-conformance that cannot be assigned to a
2095 specific system. For example, the system assignment of unauthorized devices detected in
2096 the authorization boundary may be unknown. A specific role is defined at the network level
2097 to manage unassigned non-conformances.

2098 **4.2.2. Buy or Build Considerations**

2099 A key consideration for adopting a solution (i.e., product or service) that supports automated
2100 control testing is whether to *buy* an existing package that provides the functionalities necessary
2101 to support the organization or to *develop* a customized solution to address specific
2102 organizational requirements. Some factors to consider include:

- 2103 • Acquiring a commercial-off-the-shelf (COTS) IR 8011 solution
 - 2104 ○ May be part of or able to be incorporated into GRC applications
 - 2105 ○ Support for both assessment and monitoring activities
 - 2106 ○ Leverage product maintenance and support provided by the solution
 - 2107 developer/provider
- 2108 • Acquiring a custom solution-developed IR 8011 solution
 - 2109 ○ When a COTS solution is not available
 - 2110 ○ When a COTS solution is not appropriate or effective for implementation within
 - 2111 an adopter's environment
 - 2112 ○ Ensure that a custom-developed solution is maintained and supported

- 2113 • Building an IR 8011 solution in-house
 - 2114 ○ Support of internal automated control assessments (e.g., on a smaller scale than
 - 2115 automating control assessments on a per security capability basis)
 - 2116 ○ Ensure that an in-house-developed solution is maintained and supported
- 2117 • Hybrid buy-build IR 8011 solution
 - 2118 ○ Acquired solution is supplemented with an in-house-developed solution

2119 Whether building or buying a solution:

- 2120 • Ensure that the solution has the ability to update the controls and assessment
- 2121 objectives as they are updated by the source while retaining the history necessary for
- 2122 trending
- 2123 • Ensure that ODP values are captured as desired state specifications
- 2124 • Ensure that the same SP 800-53 revision number is used

2125 Organizations have the flexibility to create additional controls outside of the SP 800-53 control
2126 catalog. These organization-developed controls may be tailored into the security, privacy, or
2127 cybersecurity supply chain risk management plans or be included in an overlay that addresses a
2128 specific technology or type of operational environment. [SP800-53A] provides a methodology
2129 for creating assessment objectives for these organization-developed controls, and the IR 8011
2130 volumes provide a model for defining the applicable sub-capability tests associated with the
2131 test objective.

2132 4.2.3. Support for Internal Automated Control Testing

2133 Tests can be individual scripts that are bundled to assess controls on a control item-by-control
2134 item basis, on a control-by-control basis, on a control family basis, or on a security capability
2135 basis. These tests can support internal automated control testing as long as desired state
2136 specifications exist and actual state data can be collected. The collection system or the
2137 component that performs the tasks of a collection system need not be complex. The data
2138 collection, analysis, and reporting processes may not even be fully automated if their purpose is
2139 to support control assessment or monitoring activities.

2140 Table 38 provides a conceptual implementation example of the IR 8011 methodology to
2141 illustrate the operationalization of a sub-capability test to automate the testing of a control or
2142 control item and to show how a foundational sub-capability test could be performed.

2143 *Table 38. Implementation example of HWAM-F01-Test*

Element	Test Object Example(s)
Desired state specification	An inventory list of the MAC addresses of all authorized devices that can be admitted to the network
Actual state	The MAC addresses of all devices on the network

Element	Test Object Example(s)
Response⁹³	Remove unauthorized devices that are discovered in the boundary. Investigate root cause for why unauthorized devices were present in the boundary and how.
Implementation	Software-based network sensors are placed within the test boundary to detect all devices on the network and collect actual state data. Detection of the devices requires the identification of all of the wired or wireless devices that are already present on the network as well as any new devices that join the network. A simple check involves comparing the MAC addresses of the devices on the network to an existing list of approved MAC addresses. If a detected MAC address is not listed on the approved list, then automatically remove the device by blocking or rejecting the MAC or IP address of the unauthorized device. The comparison can be scripted using regular expressions, and the device can be blocked automatically via network utility software. Notifications and (human) verification follow.

2144 Organizations may choose to develop internal tests to save costs and/or allow for greater
2145 customization. However, it is still necessary to enforce and maintain rigor in the development
2146 and maintenance of the tests and supporting information technology, including their design,
2147 testing, configuration management, maintenance, and other important development and
2148 maintenance aspects.

2149 **4.2.4. Support for External Independent Automated Control Testing**

2150 External independent automated control testers may take advantage of any on-premises
2151 implementation of an automated control test system at the adopter's site and any operational
2152 tests utilized by the external independent assessor or assessment team. There may be
2153 challenges, and additional risks to the organization, associated with allowing external
2154 independent testers to load and execute their own automated control testing (e.g., access
2155 control, permissions, integration, and other difficulties that are inherent to connecting an
2156 external resource to an internal resource). The security of the system architecture is reviewed
2157 and analyzed, and policies and procedures are reviewed before allowing an external entity to
2158 access internal networks and resources.

2159 In its most simplistic approach, the external independent tester would use the adopter's
2160 existing resources to validate and verify the test execution and results. An example of an
2161 existing resource is the adopter's GRC application/repository with or without an integrated IR
2162 8011 functionality, such as support for running sub-capability tests. A more complex
2163 implementation would require the external independent assessor to run their own
2164 implementation of the IR 8011 methodology, which may include the use of their own data
2165 repositories hosting the adopter's desired state and actual state information and their own
2166 implementation of and mechanisms for actual state collection and state analysis.

⁹³ Action or activity should a non-conformance occur.

4.2.5. Integration Into Existing Continuous Monitoring Programs

Whether an IR 8011 solution is bought or built, the IR 8011 implementation focuses on supporting the organization's continuous monitoring strategy or program. As IR 8011 solutions or processes are adopted, the organization updates its risk management strategy, continuous monitoring strategy, and other approaches for managing risks to achieve processes for ongoing assessments⁹⁴ and authorizations.⁹⁵ These processes may include automated control testing and reporting at predetermined intervals and procedures on how to assess the other non-testable portions of the controls and control items.

4.3. Understanding Limitations to IR 8011 Operationalization

Both developers and adopters are advised to understand the limitations of operationalizing the IR 8011 methodology. Even if the solution developer implements all of the sub-capability tests from the capability-specific volumes in the IR 8011 series, the testable controls provided in the capability-specific volumes only represent a sample set of controls that can defend against the sample of attacks addressed in the methodology. Alternative attack and defend models provide for other attack types and defense strategies that are not addressed in these IR 8011 volumes.

Effective analysis and suitable data quality measures can contribute to a successful execution of sub-capability tests. While the automated testing of control items may provide a degree of efficiency to the monitoring process, considerations are given to the application of the examine and/or interview assessment methods to fully understand test results within the context of the risk management strategy. This includes determining whether the organization and system artifacts demonstrate an understanding of how the automated test results can be used, and whether system managers and administrators understand how to respond to risks that may not be captured through automated testing. IR 8011 operationalization can support limited available resources to focus on maintaining adequate security by highlighting specific areas where additional effort may be necessary based on the identification of non-conformance in the security implementations across the organization. This may include identifying the most frequent non-conformances and their location and origin, aggregating non-conformances by responsible risk response party, and analyzing data reporting so that proper and speedy responses can be made.

4.4. Implementation Validation

Implementers have the flexibility to operationalize the IR 8011 methodology in a variety of ways. There is no one prescriptive way to implement the IR 8011 methodology. This freedom and flexibility give implementers the autonomy to develop and adopt an IR 8011 solution that meets an organization's needs for automating the continuous monitoring of specific controls. It is prudent for implementers of the methodology to validate the implementation to ensure that the organization's requirements for continuous monitoring are satisfied. The entity performing

⁹⁴ "Ongoing assessment of [control] effectiveness supports a system's [authorization] over time in highly dynamic environments of operation with changing threats, vulnerabilities, technologies, and [mission]/business processes" [SP800-137].

⁹⁵ [SP800-137] provides information regarding the use of continuous monitoring in support of ongoing system authorization.

2203 the validation is identified by the system or organization, whether as developers of the IR 8011
2204 solution or as adopters. To support implementation validation efforts, a non-exhaustive list of
2205 validation considerations is provided in Appendix E. These are not intended to be quantitative
2206 criteria for validation but rather a qualitative approach to ensure the trustworthiness of the
2207 solution and its implementation.

2208 References

- 2209 [CPRT] National Institute of Standards and Technology (2025), NIST Cybersecurity
2210 and Privacy Reference Tool (CPRT). Available at
2211 <https://csrc.nist.gov/projects/cprt>
- 2212 [CSWP30] Takamura E, Licata J, Pillitteri VY (2023) Automation Support for Control
2213 Assessments: Project Update and Vision. (National Institute of Standards and
2214 Technology, Gaithersburg, MD), NIST Cybersecurity White Paper (CSWP) NIST
2215 CSWP 30. <https://doi.org/10.6028/NIST.CSWP.30>
- 2216 [IR8011v1] Dempsey KL, Eavy P, Moore G (2017) Automation Support for Security
2217 Control Assessments: Volume 1: Overview. (National Institute of Standards
2218 and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR)
2219 NIST IR 8011v1. <https://doi.org/10.6028/NIST.IR.8011-1>
- 2220 [IR8011v2] Dempsey KL, Eavy P, Moore G (2017) Automation Support for Security
2221 Control Assessments: Volume 2: Hardware Asset Management. (National
2222 Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency
2223 or Internal Report (IR) NIST IR 8011v2.
2224 <https://doi.org/10.6028/NIST.IR.8011-2>
- 2225 [IR8011v3] Dempsey KL, Goren N, Eavy P, Moore G (2018) Automation Support for
2226 Security Control Assessments: Software Asset Management. (National
2227 Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency
2228 or Internal Report (IR) NIST IR 8011v3.
2229 <https://doi.org/10.6028/NIST.IR.8011-3>
- 2230 [IR8011v4] Dempsey KL, Takamura E, Eavy P, Moore G (2020) Automation Support for
2231 Security Control Assessments: Software Vulnerability Management. (National
2232 Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency
2233 or Internal Report (IR) NIST IR 8011v4.
2234 <https://doi.org/10.6028/NIST.IR.8011-4>
- 2235 [NVD] National Institute of Standards and Technology (2025), National Vulnerability
2236 Database (NVD). Available at <https://nvd.nist.gov>
- 2237 [OMBA130] Office of Management and Budget, Circular A-130, *Managing Information as*
2238 *a Strategic Resource*, July 2016. [https://www.whitehouse.gov/wp-](https://www.whitehouse.gov/wp-content/uploads/legacy_drupal_files/omb/circulars/A130/a130revised.pdf)
2239 [content/uploads/legacy_drupal_files/omb/circulars/A130/a130revised.pdf](https://www.whitehouse.gov/wp-content/uploads/legacy_drupal_files/omb/circulars/A130/a130revised.pdf)
- 2240 [RMF] National Institute of Standards and Technology (2025), NIST Risk
2241 Management Framework (RMF) Project. Available at <https://nist.gov/rmf>
- 2242 [SP800-30] Joint Task Force Transformation Initiative (2012) Guide for Conducting Risk
2243 Assessments. (National Institute of Standards and Technology, Gaithersburg,
2244 MD), NIST Special Publication (SP) NIST SP 800-30r1.
2245 <https://doi.org/10.6028/NIST.SP.800-30r1>
- 2246 [SP800-37] Joint Task Force (2018) Risk Management Framework for Information
2247 Systems and Organizations: A System Life Cycle Approach for Security and
2248 Privacy. (National Institute of Standards and Technology, Gaithersburg, MD),
2249 NIST Special Publication (SP) NIST SP 800-37r2.
2250 <https://doi.org/10.6028/NIST.SP.800-37r2>

- 2251 [SP800-39] Joint Task Force Transformation Initiative (2011) Managing Information
2252 Security Risk: Organization, Mission, and Information System View. (National
2253 Institute of Standards and Technology, Gaithersburg, MD), NIST Special
2254 Publication (SP) NIST SP 800-39. <https://doi.org/10.6028/NIST.SP.800-39>
- 2255 [SP800-53] Joint Task Force (2020) Security and Privacy Controls for Information Systems
2256 and Organizations. (National Institute of Standards and Technology,
2257 Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53r5. Includes
2258 updates as of December 10, 2020. <https://doi.org/10.6028/NIST.SP.800-53r5>
- 2259 [SP800-53A] Joint Task Force (2022) Assessing Security and Privacy Controls in Information
2260 Systems and Organizations. (National Institute of Standards and Technology,
2261 Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53Ar5.
2262 <https://doi.org/10.6028/NIST.SP.800-53Ar5>
- 2263 [SP800-53B] Joint Task Force (2020) Control Baselines for Information Systems and
2264 Organizations. (National Institute of Standards and Technology,
2265 Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53B, Includes
2266 updates as of December 10, 2020. <https://doi.org/10.6028/NIST.SP.800-53B>
- 2267 [SP800-115] Scarfone KA, Souppaya MP, Cody A, Orebaugh AD (2008) Technical Guide to
2268 Information Security Testing and Assessment. (National Institute of
2269 Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP)
2270 NIST SP 800-115. <https://doi.org/10.6028/NIST.SP.800-115>
- 2271 [SP800-137] Dempsey KL, Chawla NS, Johnson LA, Johnston R, Jones AC, Orebaugh AD,
2272 Scholl MA, Stine KM (2011) Information Security Continuous Monitoring
2273 (ISCM) for Federal Information Systems and Organizations. (National Institute
2274 of Standards and Technology, Gaithersburg, MD), NIST Special Publication
2275 (SP) NIST SP 800-137. <https://doi.org/10.6028/NIST.SP.800-137>
- 2276 [SP800-137A] Dempsey KL, Pillitteri VY, Baer C, Niemeyer R, Rudman R, Urban S (2020)
2277 Assessing Information Security Continuous Monitoring (ISCM) Programs:
2278 Developing an ISCM Program Assessment. (National Institute of Standards
2279 and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP
2280 800-137A. <https://doi.org/10.6028/NIST.SP.800-137A>
- 2281 [SP800-160v1] Ross RS, McEvilley M, Winstead M (2022) Engineering Trustworthy Secure
2282 Systems. (National Institute of Standards and Technology, Gaithersburg, MD),
2283 NIST Special Publication (SP) NIST SP 800-160v1r1.
2284 <https://doi.org/10.6028/NIST.SP.800-160v1r1>
- 2285 [SP800-160v2] Ross RS, Pillitteri VY, Graubart R, Bodeau D, McQuaid R (2021) Developing
2286 Cyber Resilient Systems: A Systems Security Engineering Approach. (National
2287 Institute of Standards and Technology, Gaithersburg, MD), NIST Special
2288 Publication (SP) NIST SP 800-160v2r1. <https://doi.org/10.6028/NIST.SP.800-160v2r1>
- 2289

2290 **Appendix A. Glossary**

2291 **actual state**

2292 The observable state or behavior of an entity (e.g., device, software, person, credential, account) at the point in
2293 time when the collector generates security-related information. In particular, the actual state includes the states or
2294 behaviors that might indicate non-conformance.

2295 **anomalous event response and recovery management**

2296 *See Capability, anomalous event response and recovery management.*

2297 **asset**

2298 Resources of value that an organization possesses or employs.

2299 **behavior management**

2300 *See Capability, behavior management.*

2301 **capability**

2302 *See Capability, security.*

2303 **Capability, anomalous event detection management**

2304 A security capability for continuous monitoring that identifies routine and unexpected events that can compromise
2305 security within a time frame that prevents or reduces the impact and consequences of the events to the extent
2306 possible.

2307 **Capability, behavior management**

2308 A security capability for continuous monitoring that ensures that people are aware of expected security-related
2309 behavior and are able to perform their duties to prevent advertent and inadvertent behavior that compromises
2310 information.

2311 **Capability, boundary management (filters)**

2312 A security capability for continuous monitoring that ensures that traffic into and out of the network (and out of the
2313 physical facility protection) does not compromise security. Do the same for enclaves that subdivide the network.

2314 **Capability, boundary management (other)**

2315 A security capability for continuous monitoring that ensures that information is protected (with adequate strength)
2316 when needed to protect confidentiality and integrity, whether that information is in transit or at rest.

2317 **Capability, boundary management (physical)**

2318 A security capability for continuous monitoring that ensures that movement (e.g., of people, media, equipment)
2319 into and out of the physical facility does not compromise security.

2320 **Capability, configuration settings management**

2321 A security capability for continuous monitoring that identifies configuration settings (i.e., Common Configuration
2322 Enumerations [CCEs]) on devices that are likely to be used by attackers to compromise a device and use it as a
2323 platform from which to compromise the network.

2324 **Capability, credentials and authentication management**

2325 A security capability for continuous monitoring that ensures that people only have the necessary credentials and
2326 authentication methods to perform their duties.

2327 **Capability, event preparation management**

2328 A security capability for continuous monitoring that ensures that procedures and resources are in place to respond
2329 to both routine and unexpected events that can compromise security, including both actual attacks and
2330 contingencies (e.g., natural disasters).

- 2331 **Capability, hardware asset management**
2332 A security capability for continuous monitoring that identifies unmanaged devices that are likely to be used by
2333 attackers as a platform from which to compromise the network.
- 2334 **Capability, manage and assess risk**
2335 A security capability for continuous monitoring that reduces the successful exploits of other non-meta capabilities
2336 that occur because the risk management process fails to correctly identify and prioritize the actions and
2337 investments needed to lower the risk profile.
- 2338 **Capability, perform resilient systems engineering**
2339 A security capability for continuous monitoring that reduces the successful exploits of other non-meta capabilities
2340 that occur because there was inadequate design, engineering, implementation, testing, and/or other technical
2341 issues in implementing and/or monitoring the controls related to the other non-meta capabilities. It also reduces
2342 the successful exploits that occur because there were inadequately defined requirements, policy, planning, and/or
2343 other management issues in implementing and/or monitoring the controls related to other non-meta capabilities.
- 2344 **Capability, privilege and account management**
2345 A security capability for continuous monitoring that ensures that people only have the necessary privileges to
2346 perform their duties.
- 2347 **Capability, security**
2348 A set of mutually reinforcing controls implemented by technical, physical, and procedural means. Such controls are
2349 typically selected to achieve a common information security- or privacy-related purpose. [SP800-53A]
- 2350 **Capability, software asset management**
2351 A security capability for continuous monitoring that identifies unauthorized software on devices that is likely to be
2352 used by attackers as a platform from which to compromise the network.
- 2353 **Capability, trust management**
2354 A security capability for continuous monitoring that prevents insider attacks by ensuring that untrustworthy
2355 persons are not granted network access.
- 2356 **Capability, vulnerability management**
2357 A security capability for continuous monitoring that identifies vulnerabilities (i.e., Common Vulnerabilities and
2358 Exposures [CVEs]) on devices that are likely to be used by attackers to compromise a device and use it as a
2359 platform from which to compromise the network.
- 2360 **collection system**
2361 A system that collects actual state data and compares it to the desired state specification to find security non-
2362 conformance.
- 2363 **collector**
2364 Typically, an automated sensor that gathers actual state data. Part of the *collection system*.
- 2365 **configuration settings management**
2366 See *Capability, configuration settings management*.
- 2367 **continuous monitoring capability**
2368 See *Capability*.
- 2369 **continuous monitoring dashboard**
2370 A hierarchy of dashboards to facilitate the reporting of appropriate security-related information at multiple
2371 organizational levels.

- 2372 **control item**
2373 All or part of an SP 800-53 control requirement expressed as a statement for implementation and assessment.
2374 Both controls and control enhancements are treated as control items. Controls and control assessments are
2375 further subdivided if multiple security requirements within the control or control enhancement in [SP800-53] are
2376 in listed format (e.g., a, b, c).
- 2377 **control test plan**
2378 The objectives for the control testing and a detailed roadmap of how to conduct such testing.
- 2379 **dashboard**
2380 *See organization dashboard.*
- 2381 **desired state**
2382 *See desired state specification.*
- 2383 **desired state specification**
2384 A defined value, list, or rule (i.e., specification) that states or allows for the computation of the state that the
2385 organization desires in order to reduce information security risk. Desired state specifications are generally
2386 statements of policy.
- 2387 **device**
2388 In automated testing, a type of testable object that is an IP addressable component or equivalent within a
2389 boundary or a removable component that is of security significance.
- 2390 **device role**
2391 A group of devices with the same rules. For example, the list of permitted software for a server is likely different
2392 from that for a workstation which causes servers and devices to have separate device roles. Roles can be defined
2393 by the organization or by an external entity such as a sector, community or another source. Examples of high-level
2394 roles include user-endpoint, server, networking device, cellular device, mobile device, and other devices. Each
2395 might be further subdivided. For instance, servers might be divided into many sub-categories such as database
2396 server, email server, file server, DNS server, DHCP server, and authentication server. A device role is needed
2397 whenever the organization wants a group of devices to have different rules for authorized software, settings
2398 and/or patching.
- 2399 **foundational sub-capability tests**
2400 Sub-capability tests that expose the ineffectiveness of controls that are fundamental to the purposes of the
2401 capability in which the sub-capability test appears.
- 2402 **hardware asset management**
2403 *See Capability, hardware asset management.*
- 2404 **identifier**
2405 Data that identifies an entity of interest (e.g., a sub-capability, a sub-capability test). In database terms, it is a
2406 primary or candidate key that can be used to uniquely identify or reference a testable object so that it is not
2407 confused with other objects.
- 2408 **information security continuous monitoring system**
2409 The system that collects, analyzes, and displays ISCM security-related information (e.g., an IR 8011 solution).
- 2410 **Limit, specification**
2411 A condition indicating that risk has exceeded acceptable levels and that immediate action is needed to reduce the
2412 risk or the system/testable object may need to be removed from operations or lose the authorization to operate.

- 2413 **local sub-capability tests**
2414 The sub-capability tests that an organization adds to *foundational sub-capability tests* based on an assessment of
2415 its own needs and risk tolerance. A local sub-capability test supports or strengthens foundational sub-capability
2416 tests. Agencies may choose not to apply a given local sub-capability test if the supporting controls have not been
2417 selected or implemented.
- 2418 **manage and assess risk**
2419 See *Capability, manage and assess risk*.
- 2420 **manage boundaries**
2421 See *Capability, boundary management*.
- 2422 **manage credentials and authentication**
2423 See *Capability, credentials and authentication management*.
- 2424 **manage privileges**
2425 See *Capability, privilege and account management*.
- 2426 **non-conformance**
2427 Indicates a weakened state of security that increases risks due to one or more unmet requirement.
- 2428 **non-conformance type**
2429 A non-conformance that could occur on many testable objects. Generally, a sub-capability test checks for the
2430 presence or absence of a non-conformance type.
- 2431 **object**
2432 See *Object, testable*.
- 2433 **Object, testable**
2434 Testable objects identify the specific items being tested. Testable objects include *specifications, mechanisms,*
2435 *activities, and individuals*, which in turn may include devices, software products, software executables, credentials,
2436 accounts, account privileges, and things to which privileges are granted (including data and physical facilities).
- 2437 **ongoing assessment**
2438 The continuous evaluation of the effectiveness of control implementation. A subset of continuous monitoring
2439 activities.
- 2440 **organization dashboard**
2441 An organization-level dashboard that a) collects data from a collection system and b) shows detailed testable
2442 object-level data and testable object-level non-conformance indicators to organizationally authorized personnel.
- 2443 **prepare for events**
2444 See *Capability, event preparation management*.
- 2445 **regular expression**
2446 A sequence of characters or words that forms a search pattern, mainly for use in pattern matching with strings or
2447 string matching.
- 2448 **risk**
2449 A measure of the extent to which an organization is threatened by a potential circumstance or event, the adverse
2450 impacts that would arise if the circumstance or event occurs, and the likelihood of occurrence. Likelihood is
2451 influenced by the ease of exploitation and the frequency with which a testable object is being attacked at present.
2452 [OMBA130]
- 2453 **security capability**
2454 See *Capability, security*.

- 2455 **security control item**
2456 *See Control item.*
- 2457 **specification limit**
2458 *See Limit, specification.*
- 2459 **software asset management**
2460 *See Capability, software asset management.*
- 2461 **sub-capability**
2462 A capability that supports the achievement of a larger capability. In the IR 8011 series, each defined capability is
2463 decomposed into the set of sub-capabilities that are necessary and sufficient to support the purpose of the larger
2464 capability.
- 2465 **sub-capability test**
2466 A way to verify determination statements. It is stated as a test (wherever appropriate), can be automated, and
2467 explicitly defines a particular desired state specification that is then compared to the corresponding actual state to
2468 determine the test result. A sub-capability test provides information that may help determine the degree of
2469 control effectiveness and/or level of risk that is acceptable. Sub-capability tests also suggest risk response options
2470 and assesses a corresponding sub-capability.
- 2471 **target**
2472 The system or organization under attack.
- 2473 **test boundary**
2474 The range, scope or coverage of a test. It may encompass one or more environments of operation and be inclusive
2475 of more systems and components than a single authorization boundary.
- 2476 **test completeness**
2477 The degree to which the continuous monitoring-generated, security-related information is collected on all testable
2478 objects for all applicable sub-capability tests within a defined period of time.
- 2479 **test criteria**
2480 Rules of logic to allow for the automated or manual detection of non-conformance. Typically, the test criteria in
2481 continuous monitoring define what in the desired state specification is compared to what in the actual state and
2482 the conditions that indicate non-conformance.
- 2483 **test timeliness**
2484 The degree to which the continuous monitoring-generated, security-related information is collected within the
2485 specified period of time (or frequency).
- 2486 **testable object**
2487 *See Object, testable.*
- 2488 **trust**
2489 *See Capability, trust management.*
- 2490 **trust management**
2491 *See Capability, trust management.*
- 2492 **unmanaged device**
2493 A device inside of the authorization boundary that is either unauthorized or, if authorized, not assigned to a person
2494 to administer.
- 2495 **vulnerability management**
2496 *See Capability, vulnerability management.*

2497 **Appendix B. List of Abbreviations and Acronyms**

2498 **BEHAVE**

2499 Security-Related Behavior Management

2500 **BOUND-N**

2501 Network Boundary Management (filters)

2502 **BOUND-O**

2503 Other Boundary Management

2504 **BOUND-P**

2505 Physical Boundary Management

2506 **CAT**

2507 Control Allocation Table

2508 **CI**

2509 Control Item

2510 **CM**

2511 Configuration Management

2512 **COTS**

2513 Commercial-Off-the-Shelf

2514 **CPRT**

2515 Cybersecurity and Privacy Reference Tool

2516 **CRED**

2517 Credentials and Authentication Management

2518 **CSM**

2519 Configuration Settings Management

2520 **CSP**

2521 Cloud Service Provider

2522 **CUI**

2523 Controlled Unclassified Information

2524 **CVE**

2525 Common Vulnerabilities and Exposures

2526 **CWE**

2527 Common Weakness Enumeration

2528 **DB**

2529 Database

2530 **DBMS**

2531 Database Management System

2532 **DDoS**



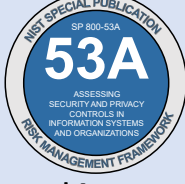



2533 Distributed Denial of Service

2534	DesiredStateMgr
2535	Desired State Manager
2536	DeviceMgr
2537	Device Manager
2538	DMZ
2539	Demilitarized Zone
2540	DS
2541	Determination Statement
2542	EVENT-DETECT
2543	Anomalous Event Detection Management
2544	EVENT-RESPOND
2545	Event Response and Recovery Management
2546	GRC
2547	Governance, Risk, and Compliance
2548	HWAM
2549	Hardware Asset Management
2550	IoT
2551	Internet of Things
2552	IR
2553	NIST Interagency or Internal Report
2554	ITL
2555	Information Technology Laboratory
2556	ISCM
2557	Information Security Continuous Monitoring
2558	ISCM-Sys
2559	Information Security Continuous Monitoring System
2560	ISCM-TB
2561	Information Security Continuous Monitoring Test boundary
2562	JSON
2563	JavaScript Object Notation
2564	MAC
2565	Media Access Control
2566	NIST
2567	National Institute of Standards and Technology
2568	NVD
2569	National Vulnerability Database
2570	OMB
2571	Office of Management and Budget

2572	OT
2573	Operational Technology
2574	PII
2575	Personally Identifiable Information
2576	POA&M
2577	Plan of Action and Milestones
2578	PREP
2579	Event (Incident and Contingency) Preparation Management
2580	PRIV
2581	Privilege and Account Management
2582	RISK
2583	Manage and Assess Risk
2584	RiskExec
2585	Risk Executive (Function)
2586	RMF
2587	Risk Management Framework
2588	SCRM
2589	Supply Chain Risk Management
2590	SE
2591	Systems Engineering
2592	SP
2593	Special Publication
2594	SWAM
2595	Software Asset Management
2596	TRUST
2597	Trust Management
2598	VPN
2599	Virtual Private Network
2600	VUL
2601	Software Vulnerability Management

Appendix C. NIST RMF-Related Publications and Their Relationships to IR 8011

Table 39. NIST RMF-related publications and their relationships to IR 8011

RMF-Related Technical Publication	Relationship to IR 8011
 <p>Risk Management Methodology</p>	<p>SP 800-37, <i>Risk Management Framework for Information Systems and Organizations</i> (foundational to the understanding of the IR 8011 methodology)</p> <p>Describes the seven-step NIST Risk Management Framework (RMF) methodology for managing security and privacy risks, including the use of controls and control baselines for reducing security and privacy risks and the assessment and monitoring of implemented controls.</p>
 <p>Control Catalog</p>	<p>SP 800-53, <i>Security and Privacy Controls for Information Systems and Organizations</i> (source of security and privacy controls)</p> <p>Comprehensive catalog of security and privacy controls that includes guidance to facilitate control implementation. Each control in the control catalog can be broken down into control items, which are granular parts of a control that may be individually tested.</p>
 <p>Control Assessment Procedures</p>	<p>SP 800-53A, <i>Assessing Security and Privacy Controls in Information Systems and Organizations</i> (guide for assessing security and privacy controls from SP 800-53)</p> <p>Contains assessment procedures for the controls in the SP 800-53 control catalog in addition to an assessment methodology and additional guidance. The assessment procedures in SP 800-53A are granularized to support the assessment and automated testing of specific control items to facilitate the development of sub-capability tests.</p>
 <p>Control Baselines</p>	<p>SP 800-53B, <i>Control Baselines for Information Systems and Organizations</i> (security and privacy control baseline and control tailoring reference; source for baselines)</p> <p>Select SP 800-53 controls are allocated to security and privacy control baselines based on impact.</p>
 <p>ISCM Strategy and Program Development</p>	<p>SP 800-137, <i>Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations</i> (reference)</p> <p>NIST guidance on continuous monitoring. Many of the IR 8011 concepts are derived from SP 800-137.</p>
 <p>ISCM Program Assessment</p>	<p>SP 800-137A, <i>Assessing Information Security Continuous Monitoring (ISCM) Programs: Developing an ISCM Program Assessment</i> (reference)</p> <p>NIST guidance on assessing continuous monitoring programs.</p>

2605 **Appendix D. Benefits of Breaking Down Security Capabilities Into Elements**

2606 In support of objective #1 in the IR 8011 methodology (see Sec. 3.1), security capabilities are
2607 broken down into individual elements⁹⁶ above the control level to reach the most appropriate
2608 level to focus automated testing on: the sub-capability element. Breaking down security
2609 capabilities into elements offers other benefits as well, such as:

- 2610 • Support for the strong systems engineering of security capabilities
- 2611 • Support for control selection guidance
- 2612 • Simplification of the overall protection process
- 2613 • Ability to test control outcomes at a higher level than individual controls
- 2614 • Improved risk management by measuring control outcomes that are more closely
2615 aligned with desired business results
- 2616 • Helps organizations address organizational and mission/business risk

2617 **D.1. Supports the Strong Systems Engineering of Security Capabilities**

2618 In typical systems engineering, the engineering process begins with general business
2619 requirements at a fairly high level of abstraction. More detailed technical requirements are
2620 then derived from the business requirements. Traditionally, predefined control sets provide
2621 detailed technical requirements without documenting the traceability of control items to more
2622 general requirements.⁹⁷ An unintended and undesirable consequence of this has been that
2623 many security programs focus on the individual controls as a compliance checklist with little
2624 consideration for how the controls work together to protect the confidentiality, integrity, and
2625 availability of information and systems.

2626 The set of elements in the IR 8011 methodology supports integrated systems engineering by
2627 making the desired results of a security program clear and measurable at a concrete level,
2628 which in turn makes the results more understandable to non-security experts and easier to link
2629 to desired business and mission results. Maintaining an awareness of the desired results to be
2630 produced facilitates better security engineering and enables control designers to look at
2631 controls as parts of a system designed to achieve an overall purpose.

2632 **D.2. Supports Guidance for Control Selection**

2633 Informed and judicious decision-making in control selection requires an understanding of how
2634 controls work together to respond to attack steps and achieve broader security protections
2635 commensurate with risk. The concept of a security capability is a construct that recognizes that
2636 the protection of information being processed, stored, or transmitted by systems seldom

⁹⁶ See Fig. 1. IR 8011 methodology elements.

⁹⁷ See [SP800-160v1] for guidance on the systems engineering of information security for mission assurance.

2637 derives from a single control. In most cases, such protection results from the selection and
2638 implementation of a set of mutually reinforcing controls.

2639 **D.3. Simplifies Understanding of the Overall Protection Process**

2640 Defining security capabilities can simplify how a protection problem is viewed conceptually.
2641 Security capabilities provide a method for grouping controls that are selected and implemented
2642 for a common purpose or to achieve a common objective. Placing controls into groups that
2643 support attack steps, capabilities, and sub-capabilities facilitates better comprehension of
2644 security and privacy requirements and implementations. The grouping of controls into
2645 capabilities increases awareness of the results that controls are expected to produce.

2646 **D.4. Enables Testing of Control Outcomes at a Higher Level Than Individual Controls**

2647 Selecting the most appropriate level of all of the elements in the IR 8011 methodology to test
2648 the effectiveness of control implementations involves trade-offs. If testing is too detailed, the
2649 parts may work individually but not collectively. However, if results are assessed at a higher
2650 level of abstraction and an *other than satisfied* control is detected at that level, then root cause
2651 analysis is needed to identify the supporting control items that are not working. As noted in
2652 [SP800-53A]:

2653 Traditionally, assessments have been conducted on a control-by-control
2654 basis and produce results that are characterized as pass (i.e., control
2655 satisfied) or fail (i.e., control not satisfied). However, the failure of a
2656 single control or, in some cases, the failure of multiple controls may not
2657 affect the overall security and privacy capability required by an
2658 organization. This is not to say that such controls do not contribute to
2659 the security or privacy of the system and/or organization (as defined by
2660 the security requirements and privacy requirements during the
2661 initiation phase of the system development life cycle), but rather that
2662 such controls may not support the particular security and privacy
2663 capability. Furthermore, every implemented control and privacy control
2664 may not necessarily support the need to support an organization-
2665 defined capability.

2666 As discussed in Sec. 3.1.3:

The sub-capability element is the most appropriate level of all of the elements in the methodology on which to focus automated testing for continuous monitoring. The sub-capability layer is closer to control outcomes and easier to automate. When non-conformances are found, root cause analysis can be used to find the specific control items causing the non-conformance.

2667

2668 **D.5. Improves Risk Management by Measuring Control Outcomes**

2669 NIST guidance on information security risk management, [SP800-30]and [SP800-39],
2670 emphasizes both system-level and mission-level risks. Additionally, [SP800-37], [SP800-53], and
2671 [SP800-115] focus on assessing and analyzing results in addition to control effectiveness.
2672 [SP800-39] recommends a multi-layered “approach to risk management that addresses risk-
2673 related concerns at: (i) the organization level; (ii) the mission/business process level; and (iii)
2674 the system level.” Controls largely exist at the system level, and business and security outcomes
2675 are most visible at the organization and mission/business process level. As noted in
2676 [SP800-53A]:

2677 Ultimately, authorization decisions (i.e., risk acceptance decisions) are
2678 made based on the degree to which the desired security and privacy
2679 capabilities have been effectively achieved and are meeting the security
2680 and privacy requirements defined by an organization. Risk-based
2681 decisions are directly related to organizational risk tolerance that is
2682 defined as part of an organization’s risk management strategy.

2683 Dissecting the methodology into individual elements allows for a closer alignment to the
2684 organization’s mission and makes it easier for analysts to trace specific requirements. Mission-
2685 specific layers are added by each organization based on the contributions of the systems being
2686 managed to support a specific mission. The attack step and security capability elements are
2687 provided to make it easier to trace controls to the organization’s mission.

2688 **D.6. Helps Organizations Address Organizational, Mission, and Business Risks**

2689 To manage risks for systems as defined in [SP800-37], devices are grouped by authorization
2690 boundary to allow for the analysis of system-level risks. The security-related information
2691 produced by automated control assessment across the larger test boundary gives the risk
2692 executive the ability to consider risks for other groupings of devices and better identify risk
2693 concentrations and aggregate risk. Groupings that might be useful include devices that are:

- 2694 • Identified as mission-critical
- 2695 • Necessary for an integrated business function
- 2696 • Managed by a separate business partner
- 2697 • Supporting a specific mission across the entire organization
- 2698 • Supporting a particular customer

2699 Looking at risks with organization-defined thresholds across such large groupings of devices
2700 helps the organization address organizational, mission, and business risks, as described in
2701 [SP800-39].

Appendix E. Considerations for IR 8011 Implementation Validation

The considerations and sample questions in Table 40 are intended to facilitate the validation of an IR 8011 solution by a developer or adopter (NIST does not validate or comment on the implementation of its technical publications). These sample questions are meant to raise awareness of the scope of the solution; address maintenance and adherence to the latest revisions of RMF-supporting publications; and determine the completeness and/or sufficiency of sub-capability tests⁹⁸ among other related criteria. The listing in Table 40 is not exhaustive nor a compliance checklist, and additional considerations and questions may be necessary as determined by the organization.

Table 40. Sample considerations for validating the operationalization of IR 8011

Considerations	Sample Questions
Security Capability Offering: <i>Determine what security capability is supported.</i>	What security capabilities are supported in the IR 8011 implementation?
Deviation from the IR 8011 Methodology: <i>Determine whether there has been any deviation from the IR 8011 methodology during the operationalization process.</i>	Has there been any deviation from the IR 8011 methodology during the operationalization process?
Control Catalog Version: <i>Determine whether the IR 8011 implementation utilizes the same version of controls that the system or organization is expected to implement.</i>	Is the IR 8011 implementation up to date with the latest control catalog version? How is the IR 8011 implementation kept up to date with the latest control catalog version?
Assessment Objectives Exist: ⁹⁹ <i>Determine whether determination statements from assessment objectives exist.</i>	Are determination statements from assessment objectives available for each control set for a security capability?
Organization-Defined Parameter Import: <i>Determine whether the correct ODP values are in use.</i>	Are sub-capability tests updated with current ODP values?
Appropriateness of Sub-Capabilities: <i>Determine whether the identified sub-capabilities are appropriate.</i>	Are the identified sub-capabilities appropriate? Is there a need to complement the existing sub-capabilities with additional sub-capabilities? Are there sub-capabilities that may not be applicable or appropriate?
Sufficiency of Sub-Capability Tests: <i>Determine whether the sub-capability tests are sufficient.</i>	Do the sub-capability tests provide the necessary assurance that the results give a clear picture of the security capability?

⁹⁸ Section 3.1.6.4 addresses concerns regarding test failures and discusses ways to identify root causes for the failures.

⁹⁹ This is primarily for IR 8011 solution developers.

Considerations	Sample Questions
Correlation With Selected Controls: <i>Determine whether the testable controls identified for a security capability correlate with selected controls.</i>	Is there any testable control for the security capability that has not been selected for implementation? ¹⁰⁰
Alignment with or Variation from the IR 8011 Methodology or Capability Volume Content: <i>Determine whether there have been deviations from the IR 8011 methodology or capability volume content.</i>	Have there been deviations from the IR 8011 methodology? Have there been deviations from the capability volume content?
Level of Automation: <i>Determine whether the level of automation is greater than the level of manual effort.¹⁰¹</i>	How do automated processes compare to manual or procedural processes after the implementation of an IR 8011 solution? Is there a balance between automated processes and manual/procedural processes?
Integration with Existing Continuous Monitoring Program: <i>Determine whether the IR 8011 solution fulfills the continuous monitoring program's objectives.</i>	Does the IR 8011 solution meet the organization's continuous monitoring objectives? Has the organization addressed how it will address gaps in the non-testable portions of the controls?

2712

¹⁰⁰ This can occur when the baseline is tailored by the organization or when the organization generates its own control baseline.

¹⁰¹ The initial effort for promoting any automation requires extensive manual preparation.

2713 Appendix F. Change Log

2714 The following is a summary of changes in Revision 1:

- 2715 • The IR 8011 series title changed from “Automation Support for Security Control
2716 Assessments” to “Testable Controls and Security Capabilities for Continuous
2717 Monitoring.” This change is intended to shift the focus from assessments to monitoring
2718 to better align with the IR 8011 scope to support continuous monitoring. As a result,
2719 most references to “automated control assessments” have been replaced with
2720 “automated control testing” throughout the publication. There is also greater emphasis
2721 on continuous monitoring capabilities than on control assessments.
- 2722 • The report was reorganized into three major parts: (1) IR 8011 overview, including
2723 foundational concepts; (2) IR 8011 methodology; and (3) potential IR 8011 methodology
2724 implementation/operationalization. Section 1 describes the basics and scope of IR 8011
2725 as well as what it can and cannot do. Section 2 describes the foundational concepts of IR
2726 8011. Section 3 describes the IR 8011 methodology with guidance, additional diagrams,
2727 and other visual aids to facilitate understanding of the model. Section 4 is new and
2728 dedicated to a conceptual operationalization of IR 8011, including conceptual
2729 implementation examples of the IR 8011 methodology.
- 2730 • The IR 8011 audience groups were refined and categorized as “IR 8011 Developer” and
2731 “IR 8011 Adopter.” A third group was identified as “Cybersecurity Researchers.”
2732 Guidance throughout the publication was written with these specific groups in mind.
- 2733 • Plain language was used to improve readability and facilitate understanding.
- 2734 • The two major objectives of the IR 8011 methodology were clearly delineated with
2735 processes and elements that support the objectives described and exemplified.
- 2736 • All controls and control assessment objectives were updated from Revision 4 to Revision
2737 5 of [SP800-53] and [SP800-53A]. Control baselines and related references now point to
2738 [SP800-53B].
- 2739 • The Sortable Control Item Code has been removed. Starting with [SP800-53] Update
2740 5.1.1 and [SP800-53A] Update 5.1.1, leading zeros¹⁰² have been added to both control
2741 and assessment procedure identifiers, which solves sorting issues.
- 2742 • Elements from [SP800-53A] have been incorporated into the text, including how
2743 assessment procedures and objectives apply to the IR 8011 methodology and new
2744 numbering schema.
- 2745 • The *defend steps* described in this volume were previously referred to as *block steps*,
2746 whose objective was to block or delay an attack. However, blocking or delaying are just
2747 a few of the potential responses to an attack. *Defend* is used as a general response and
2748 to better describe the attack-defend model.

¹⁰² The notation with leading zeros substitutes the original (now legacy) IR 8011 numbering scheme that used a “z” as part of the control item representation. The “z” notation is also mitigated by the improved alignment of the control statements in [SP800-53] and the determination statements in [SP800-53A].

- 2749 • The terms *actual state* and *desired state specification* are used instead of *actual*
2750 *behavior* and *expected behavior* since a *state* is a stronger descriptor of a condition or
2751 value rather than a behavior or action. Likewise, *desired state specification* provides a
2752 more descriptive term for the acceptable specification, which can take shape as a value
2753 rather than a stance, an action, or lack thereof as *behavior* suggests.
- 2754 • New and improved visual aids have been added to facilitate understanding, navigation,
2755 and overall reading experience, including refreshed graphics, higher resolution images,
2756 and accessible visual aids.
- 2757 • Volume 1 more clearly states that the controls and control items that support specific
2758 security capabilities are a non-exhaustive sampling of controls and control items.
- 2759 • Language on the potential application of the IR 8011 methodology using non-RMF and
2760 non-SP 800-53 frameworks has been added along with identified conditions for such
2761 flexibility.
- 2762 • This revision includes a conceptual use of NIST-provided SP 800-53, SP 800-53A, and SP
2763 800-53B datasets that can be leveraged for developing, using, and maintaining an IR
2764 8011 database for sub-capability test development and testable control identification
2765 purposes.
- 2766 • Considerations for validating IR 8011 methodology applications have been provided to
2767 assist implementers in evaluating implementations.
- 2768 • The glossary and acronyms lists were updated and expanded.