NIST SPECIAL PUBLICATION 1800-26B

Data Integrity:

Detecting and Responding to Ransomware and Other Destructive Events

Volume B:
Approach, Architecture, and Security Characteristics

Jennifer Cawthra
National Cybersecurity Center of Excellence
NIST

Michael Ekstrom
Lauren Lusty
Julian Sexton
John Sweetnam
The MITRE Corporation
McLean, Virginia

December 2020

FINAL

This publication is available free of charge from
https://doi.org/10.6028/NIST.SP.1800-26.

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/building-blocks/data-integrity/detect-respond.

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified by name or company logo or other insignia in order to acknowledge their participation in this collaboration or to describe an experimental procedure or concept adequately. Such identification is not intended to imply special status or relationship with NIST or recommendation or endorsement by NIST or NCCoE; neither is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

National Institute of Standards and Technology Special Publication 1800-26B, Natl. Inst. Stand. Technol. Spec. Publ. 1800-26B, 54 pages, (December 2020), CODEN: NSPUE2

FEEDBACK

As a private-public partnership, we are always seeking feedback on our practice guides. We are particularly interested in seeing how businesses apply NCCoE reference designs in the real world. If you have implemented the reference design, or have questions about applying it in your environment, please email us ds-nccoe@nist.gov.

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

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in information technology security—the NCCoE applies standards and best practices to develop modular, adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to re-create the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Maryland.

To learn more about the NCCoE, visit https://www.nccoe.nist.gov/. To learn more about NIST, visit https://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication 1800 series) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

ABSTRACT

Ransomware, destructive malware, insider threats, and even honest mistakes present an ongoing threat to organizations that manage data in various forms. Database records and structure, system files, configurations, user files, application code, and customer data are all potential targets of data corruption and destruction.

A timely, accurate, and thorough detection and response to a loss of data integrity can save an organization time, money, and headaches. While human knowledge and expertise is an essential component of these tasks, the right tools and preparation are essential to minimizing downtime and

losses due to data integrity events. The NCCoE, in collaboration with members of the business community and vendors of cybersecurity solutions, has built an example solution to address these data integrity challenges. This project details methods and potential tool sets that can detect, mitigate, and contain data integrity events in the components of an enterprise network. It also identifies tools and strategies to aid in a security team's response to such an event.

KEYWORDS

attack vector; data integrity; malicious actor; malware; malware detection; malware response; ransomware.

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
Kyle Black	Bay Dynamics
Sunjeet Randhawa	Broadcom Inc.
Peter Romness	Cisco Systems
Matthew Hyatt	Cisco Systems
Matthew Shabat	Glasswall Government Solutions
Justin Rowland	Glasswall Government Solutions
Greg Rhein	Glasswall Government Solutions
Steve Roberts	Micro Focus
Timothy McBride	NIST
Christopher Lowde	Semperis

Name	Organization
Thomas Leduc	Semperis
Darren Mar-Elia	Semperis
Kirk Lashbrook	Semperis
Mickey Bresman	Semperis
Humphrey Christian	Symantec Corporation
Jon Christmas	Symantec Corporation
Kenneth Durbin	Symantec Corporation
Matthew Giblin	Symantec Corporation
Jim Wachhaus	Tripwire
Nancy Correll	The MITRE Corporation
Chelsea Deane	The MITRE Corporation
Sallie Edwards	The MITRE Corporation
Milissa McGinnis	The MITRE Corporation
Karri Meldorf	The MITRE Corporation
Denise Schiavone	The MITRE Corporation
Anne Townsend	The MITRE Corporation

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build this example solution. We worked with:

Technology Partner/Collaborator	Build Involvement
Symantec Corporation	Symantec Information Centric Analytics v6.5.2 Symantec Security Analytics v8.0.1
Cisco Systems	Cisco Identity Services Engine v2.4, Cisco Advanced Malware Protection v5.4, Cisco Stealthwatch v7.0.0
Glasswall Government Solutions	Glasswall FileTrust Advanced Threat Protection (ATP) for Email v6.90.2.5
Tripwire	Tripwire Log Center v7.3.1, Tripwire Enterprise v8.7
Micro Focus	Micro Focus ArcSight Enterprise Security Manager v7.0 Patch 2
Semperis	Semperis Directory Services Protector v2.7

Contents

1 Summary 1

1.1 Challenge 2

1.2 Solution 2

1.3 Benefits 3

2 How to Use This Guide 4

2.1 Typographic Conventions 5

3 Approach 6

3.1 Audience 6

3.2 Scope 6

3.3 Assumptions 7

3.4 Risk Assessment 7

3.4.1 Risk 8

3.4.2 Security Control Map 9

3.5 Technologies 13

4 Architecture 16

4.1 Architecture Description 16

4.1.1 High-Level Architecture 16

4.1.2 Architecture Components 17

5 Security Characteristic Analysis 20

5.1 Assumptions and Limitations 20

5.2 Build Testing 20

5.3 Scenarios and Findings 21

5.3.1 Ransomware via Web Vector and Self-Propagation 21

5.3.2 Destructive Malware via USB Vector 22

5.3.3 Accidental VM Deletion via Maintenance Script 23

5.3.4 Backdoor Creation via Email Vector 24

5.3.5 Database Modification via Malicious Insider 25

5.3.6 File Modification via Malicious Insider 26
5.3.7 Backdoor Creation via Compromised Update Server 27
6 Future Build Considerations 27
Appendix A List of Acronyms 29
Appendix B Glossary 30
Appendix C References 34
Appendix D Functional Evaluation 36
D.1 Data Integrity Functional Test Plan 36
D.2 Data Integrity Use Case Requirements 37
D.3 Test Case: Data Integrity DR-1 44
D.4 Test Case: Data Integrity DR-2 46
D.5 Test Case: Data Integrity DR-3 47
D.6 Test Case: Data Integrity DR-4 48
D.7 Test Case: Data Integrity DR-5 50
D.8 Test Case: Data Integrity DR-6 51
D.9 Test Case: Data Integrity DR-7 52

List of Figures

Figure 4-1 DI Detect & Respond High-Level Architecture 16

List of Tables

Table 3-1 DI Reference Design Cybersecurity Framework Core Components Map 10
Table 3-2 Products and Technologies 13
Table 6-1 Test Case Fields 36
Table 6-2 Capability Requirements 37
Table 6-3 Test Case ID: Data Integrity DR-1 44
Table 6-4 Test Case ID: Data Integrity DR-2 46
Table 6-5 Test Case ID: Data Integrity DR-3 47
Table 6-6 Test Case ID: Data Integrity DR-4 48
Table 6-7 Test Case ID: Data Integrity DR-5 50
Table 6-8 Test Case ID: Data Integrity DR-6 51
Table 6-9 Test Case ID: Data Integrity DR-7 52

1 Summary

Businesses face a near-constant threat of destructive malware, ransomware, malicious insider activities, and even honest mistakes that can alter or destroy critical data. These types of adverse events ultimately impact data integrity (DI). It is imperative for organizations to be able to detect and respond to DI attacks.

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) built a laboratory environment to explore methods to detect and respond to a data corruption event in various information technology (IT) enterprise environments. The example solution outlined in this guide describes the solution built in the NCCoE lab. It encourages detection and mitigation of DI events while facilitating analysis of these events.

The goals of this NIST Cybersecurity Practice Guide are to help organizations confidently:

- detect malicious and suspicious activity generated on the network, by users, or from applications that could indicate a DI event
- mitigate and contain the effects of events that can cause a loss of DI
- monitor the integrity of the enterprise for detection of events and after-the-fact analysis
- utilize logging and reporting features to speed response time to DI events
- analyze DI events for the scope of their impact on the network, enterprise devices, and enterprise data
- analyze DI events to inform and improve the enterprise's defenses against future attacks

For ease of use, here is a short description of the different sections of this volume.

- Section 1: Summary presents the challenge addressed by the NCCoE project with an in-depth look at our approach, the architecture, and the security characteristics we used; the solution demonstrated to address the challenge; the benefits of the solution; and the technology partners that participated in building, demonstrating, and documenting the solution. Summary also explains how to provide feedback on this guide.
- Section 2: How to Use This Guide explains how readers—business decision-makers, program managers, and IT professionals (e.g., systems administrators)—might use each volume of the guide.
- Section 3: Approach offers a detailed treatment of the scope of the project and describes the assumptions on which the security platform development was based, the risk assessment that informed platform development, and the technologies and components that industry collaborators gave us to enable platform development.

- Section 4: Architecture describes the usage scenarios supported by project security platforms, including Cybersecurity Framework [1] functions supported by each component contributed by our collaborators.
- Section 5: Security Characteristic Analysis provides details about the tools and techniques we used to perform risk assessments.
- Section 6: Future Build Considerations is a brief treatment of other data security implementations that NIST is considering consistent with Cybersecurity Framework Core Functions: Identify, Protect, Detect, Respond, and Recover.

1.1 Challenge

Thorough collection of quantitative and qualitative data is important to organizations of all types and sizes. It can impact all aspects of a business, including decision making, transactions, research, performance, and profitability. When these data collections sustain a DI attack caused by unauthorized insertion, deletion, or modification of information, such an attack can impact emails, employee records, financial records, and customer data, rendering them unusable or unreliable. Some organizations have experienced systemic attacks that caused a temporary cessation of operations. One variant of a DI attack—ransomware—encrypts data and holds it hostage while the attacker demands payment for the decryption keys.

When DI events occur, organizations should have the capabilities to detect and respond in real time. Early detection and mitigation can reduce the potential impact of events, including damage to enterprise files, infection of systems, and account compromise. Furthermore, organizations should be able to learn from DI events to improve their defenses. Analysis of malicious behavior at the network level, user level, and file level can reveal flaws in the security of the enterprise. Resolution of these flaws, though out of scope of this guide, is often only possible once they have been exploited and with the right solution in place.

1.2 Solution

The NCCoE implemented a solution that incorporates appropriate actions during and directly after a DI event. The solution is composed of multiple systems working together to detect and respond to data corruption events in standard enterprise components. These components include mail servers, databases, end-user machines, virtual infrastructure, and file share servers. Furthermore, an important function of the Respond Category of the Cybersecurity Framework is improvement of defenses—this guide includes components that aid in analysis of DI events and for improving defenses against them.

The NCCoE sought existing technologies that provided the following capabilities:

- **event detection**
- **integrity monitoring**

- **logging**
- **reporting**
- **mitigation and containment**
- **forensics/analytics**

In developing our solution, we used standards and guidance from the following, which can also provide your organization with relevant standards and best practices:

- NIST Framework for Improving Critical Infrastructure Cybersecurity (commonly known as the NIST Cybersecurity Framework [1]
- NIST Interagency or Internal Report (NISTIR) 8050: *Executive Technical Workshop on Improving Cybersecurity and Consumer Privacy* [2]
- NIST Special Publication (SP) 800-30 Rev. 1: *Guide for Conducting Risk Assessments* [3]
- NIST SP 800-37 Rev. 1: *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [4]
- NIST SP 800-39: *Managing Information Security Risk* [5]
- NIST SP 800-40 Rev. 3: *Guide to Enterprise Patch Management Technologies* [6]
- NIST SP 800-53 Rev. 4: *Security and Privacy Controls for Federal Information Systems and Organizations* [7]
- Federal Information Processing Standard 140-2: Security Requirements for Cryptographic Modules [8]
- NIST SP 800-86: *Guide to Integrating Forensic Techniques into Incident Response* [9]
- NIST SP 800-92: *Guide to Computer Security Log Management* [10]
- NIST SP 800-100: *Information Security Handbook: A Guide for Managers* [11]
- NIST SP 800-34 Rev. 1: *Contingency Planning Guide for Federal Information Systems* [12]
- Office of Management and Budget, Circular Number A-130: Managing Information as a Strategic Resource [13]
- NIST SP 800-61 Rev. 2: *Computer Security Incident Handling Guide* [14]
- NIST SP 800-83 Rev. 1: *Guide to Malware Incident Prevention and Handling for Desktops and Laptops* [15]
- NIST SP 800-150: *Guide to Cyber Threat Information Sharing* [16]
- NIST SP 800-184: *Guide for Cybersecurity Event Recovery* [17]

1.3 Benefits

The NCCoE's practice guide can help your organization:

- develop an implementation plan for detecting and responding to cybersecurity events
- facilitate detection, response, and analysis of DI events to improve defenses and mitigate impact

- maintain integrity and availability of data that is critical to supporting business operations and revenue-generating activities
- manage enterprise risk (consistent with the foundations of the NIST Cybersecurity Framework)

2 How to Use This Guide

This NIST Cybersecurity Practice Guide demonstrates a standards-based reference design and provides users with the information they need to replicate the DI detection and response solution. This reference design is modular and can be deployed in whole or in part.

This guide contains three volumes:

- NIST SP 1800-26A: *Executive Summary*
- NIST SP 1800-26B: *Approach, Architecture, and Security Characteristics* – what we built and why **(you are here)**
- NIST SP 1800-26C: *How-To Guides* – instructions for building the example solution

Depending on your role in your organization, you might use this guide in different ways:

Business decision-makers, including chief security and technology officers, will be interested in the *Executive Summary,* NIST SP 1800-26A, which describes the following topics:

- challenges that enterprises face in detecting and responding to data integrity events
- example solution built at the NCCoE
- benefits of adopting the example solution

Technology or security program managers who are concerned with how to identify, understand, assess, and mitigate risk will be interested in this part of the guide, NIST SP 1800-26B, which describes what we did and why. The following sections will be of particular interest:

- Section 3.4.1, Risk, provides a description of the risk analysis we performed.
- Section 3.4.2, Security Control Map, maps the security characteristics of this example solution to cybersecurity standards and best practices.

You might share the *Executive Summary,* NIST SP 1800-26A, with your leadership team members to help them understand the importance of adopting a standards-based solution to detect and respond to data integrity events.

IT professionals who want to implement an approach like this will find the whole practice guide useful. You can use the how-to portion of the guide, NIST SP 1800-26C, to replicate all or parts of the build created in our lab. The how-to portion of the guide provides specific product installation, configuration, and integration instructions for implementing the example solution. We do not re-create the product

manufacturers' documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create an example solution.

This guide assumes that IT professionals have experience implementing security products within the enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. Your organization can adopt this solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing parts of a DI detection and response solution. Your organization's security experts should identify the products that will best integrate with your existing tools and IT system infrastructure. We hope that you will seek products that are congruent with applicable standards and best practices. Section 3.5, Technologies, lists the products we used and maps them to the cybersecurity controls provided by this reference solution.

A NIST Cybersecurity Practice Guide does not describe "the" solution, but a possible solution. This is a draft guide. We seek feedback on its contents and welcome your input. Comments, suggestions, and success stories will improve subsequent versions of this guide. Please contribute your thoughts to ds-nccoe@nist.gov.

2.1 Typographic Conventions

The following table presents typographic conventions used in this volume.

Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For language use and style guidance, see the *NCCoE Style Guide*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit.**
`Monospace`	command-line input, onscreen computer output, sample code examples, and status codes	`mkdir`
`Monospace Bold`	command-line user input contrasted with computer output	**`service sshd start`**
blue text	link to other parts of the document, a web URL, or an email address	All publications from NIST's NCCoE are available at https://www.nccoe.nist.gov.

3 Approach

Based on key points expressed in NISTIR 8050: *Executive Technical Workshop on Improving Cybersecurity and Consumer Privacy* (2015), the NCCoE is pursuing a series of DI projects to map the Core Functions of the NIST Cybersecurity Framework. This project is centered on the Core Functions of Detect and Respond, which consist of detecting and responding to DI attacks. Compromise can come from malicious websites, targeted emails, insider threats, and honest mistakes. Monitoring solutions should be in place to detect these events. Once detected, swift response to a threat is critical to mitigate the need for recovery action after an event occurs. NCCoE engineers working with a Community of Interest (COI) defined the requirements for this DI project.

Members of the COI, which include participating vendors referenced in this document, contributed to development of the architecture and reference design, providing technologies that meet the project requirements and assisting in installation and configuration of those technologies. The practice guide highlights the approach used to develop the NCCoE reference solution. Elements include risk assessment and analysis, logical design, build development, test and evaluation, and security control mapping. This guide is intended to provide practical guidance to any organization interested in implementing a solution for detecting and responding to a cybersecurity event.

3.1 Audience

This guide is intended for individuals responsible for implementing security solutions in organizations' IT support activities. Current IT systems, particularly in the private sector, often lack the capability to comprehensively detect, mitigate, and learn from cybersecurity events. The platforms demonstrated by this project and the implementation information provided in this practice guide permit integration of products to implement a data integrity detection and response system. The technical components will appeal to system administrators, IT managers, IT security managers, and others directly involved in the secure and safe operation of business IT networks.

3.2 Scope

The guide provides practical, real-world guidance on developing and implementing a DI solution consistent with the principles in the NIST Framework for Improving Critical Infrastructure Cybersecurity Volume 1, specifically the Core Functions of Detect and Respond. Detecting emphasizes developing and implementing the appropriate activities to detect events in real time, compare the current system state to a norm, and produce audit logs for use during and after the event. Responding emphasizes real-time mitigation of events, forensic analysis during and after the event, and reporting. Examples of outcomes within these functions are integrity monitoring, event detection, logging, reporting, forensics, and mitigation.

3.3 Assumptions

This project is guided by the following assumptions:

- The solution was developed in a lab environment. The environment is based on a basic organization's IT enterprise. It does not reflect the complexity of a production environment: for example, building across numerous physical locations, accommodating extreme working conditions, or configuring systems to meet specific network/user needs. These demands can all increase the level of complexity needed to implement a DI solution.
- An organization has access to the skill sets and resources required to implement an event detection and response system.
- A DI event is taking place, and the organization is seeking to detect and mitigate the damage that an event is causing.

3.4 Risk Assessment

NIST SP 800-30 Revision 1, *Guide for Conducting Risk Assessments*, states that risk is "a measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence." The guide further defines risk assessment as "the process of identifying, estimating, and prioritizing risks to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, resulting from the operation of an information system. Part of risk management incorporates threat and vulnerability analyses, and considers mitigations provided by security controls planned or in place."

The NCCoE recommends that any discussion of risk management, particularly at the enterprise level, begins with a comprehensive review of NIST SP 800-37 Revision 2, *Risk Management Framework for Information Systems and Organizations*—publicly available material. The Risk Management Framework (RMF) guidance, as a whole, proved invaluable in giving us a baseline to assess risks, from which we developed the project, the security characteristics of the build, and this guide.

We performed two types of risk assessment:

- Initial analysis of the risk factors discussed with financial, retail, and hospitality institutions. This analysis led to creation of the DI project and the desired security posture. See NISTIR 8050, Executive Technical Workshop, for additional participant information.
- Analysis of how to secure the components within the solution and minimize any vulnerabilities they might introduce. See Section 5, Security Characteristic Analysis.

3.4.1 Risk

Using the guidance in NIST's series of publications concerning risk, we worked with financial institutions and the Financial Sector Information Sharing and Analysis Center to identify the most compelling risk factors encountered by this business group. We participated in conferences and met with members of the financial sector to define the main security risks to business operations. From these discussions came identification of an area of concern—DI. Having produced *Data Integrity: Recovering from Ransomware and Other Destructive Events,* which primarily focused on the recovery aspect of DI, we identified a need for guidance in the areas of detecting and responding to cybersecurity events in real time.

When considering risk from the perspective of detecting and responding to cybersecurity events during their execution, we must consider not only the impact of an event on an organization's assets but also the threats to those assets and the potential vulnerabilities these threats could exploit.

When discussing threats to an organization's assets from the perspective of DI, we consider these:

- malware
- insider threats
- accidents caused by human error
- compromise of trusted systems

The types of vulnerabilities we consider in relation to these threats include:

- zero-day vulnerabilities
- vulnerabilities due to outdated or unpatched systems
- custom software vulnerabilities/errors
- social engineering and user-driven events
- poor access control

Finally, the potential impact on an organization from a DI event:

- systems incapacitated
- modification/deletion of the organization's assets
- negative impact on the organization's reputation

Analysis of the threats, vulnerabilities, and potential impact to an organization has given us an understanding of the risk for organizations with respect to DI. NIST SP 800-39*, Managing Information Security Risk,* focuses on the business aspect of risk, namely at the enterprise level. This understanding is essential for any further risk analysis, risk response/mitigation, and risk monitoring activities. The following is a summary of the strategic risk areas we identified and their mitigations:

- Impact on system function–ensuring the availability of accurate data or sustaining an acceptable level of DI reduces the risk of systems' availability being compromised.
- Cost of implementation–implementing event detection and response from DI events once and using it across all systems may reduce system continuity costs.
- Compliance with existing industry standards–contributes to the industry requirement to maintain a continuity of operations plan.
- Maintenance of reputation and public image–helps reduce the damage caused by active events and facilitates the information needed to learn from the events.
- Increased focus on DI–includes not just loss of confidentiality but also harm from unauthorized alteration of data (per NISTIR 8050).

We subsequently translated the risk factors identified to security Functions and Subcategories within the NIST Cybersecurity Framework. In Table 3-1 we mapped the Categories to NIST SP 800-53 Rev. 4 controls.

3.4.2 Security Control Map

As explained in Section 3.4.1, we identified the Cybersecurity Framework security Functions and Subcategories that we wanted the reference design to support through a risk analysis process. This was a critical first step in drafting the reference design and example implementation to mitigate the risk factors. Table 3-1 lists the addressed Cybersecurity Framework Functions and Subcategories and maps them to relevant NIST standards, industry standards, and controls and best practices. The references provide solution validation points in that they list specific security capabilities that a solution addressing the Cybersecurity Framework Subcategories would be expected to exhibit. Organizations can use Table 3-1 to identify the Cybersecurity Framework Subcategories and NIST SP 800-53 Rev. 4 controls that they are interested in addressing.

When cross-referencing Functions of the Cybersecurity Framework with product capabilities used in this practice guide, it is important to consider:

- This practice guide, though primarily focused on Detect/Respond capabilities, also uses PR.DS-6, a Protect Subcategory. This is primarily because creation of integrity baselines is used for comparison when detecting attacks but is created prior to the start of an attack.
- Not all the Cybersecurity Framework Subcategories guidance can be implemented using technology. Any organization executing a DI solution would need to adopt processes and organizational policies that support the reference design. For example, some of the Subcategories within the Cybersecurity Framework Function called Respond are processes and policies that should be developed prior to implementing recommendations.

Table 3-1 DI Reference Design Cybersecurity Framework Core Components Map

Cybersecurity Framework v1.1				Standards & Best Practices	
Function	Category	Subcategory	NIST SP 800-53 R4	ISO/IEC 27001:2013	NIST SP 800-181
PROTECT (PR)	Data Security (PR.DS)	PR.DS-6: Integrity checking mechanisms are used to verify software, firmware, and information integrity.	SC-16, SI-7	A.12.2.1, A.12.5.1, A.14.1.2, A.14.1.3, A.14.2.4	OM-DTA-001
DETECT (DE)	Anomalies and Events (DE.AE)	DE.AE-1: A baseline of network operations and expected data flows for users and systems is established and managed.	AC-4, CA-3, CM-2, SI-4	A.12.1.1, A.12.1.2, A.13.1.1, A.13.1.2	SP-ARC-001
		DE.AE-2: Detected events are analyzed to understand attack targets and methods.	AU-6, CA-7, IR-4, SI-4	A.12.4.1, A.16.1.1, A.16.1.4	PR-CDA-001
		DE.AE-3: Event data are collected and correlated from multiple sources and sensors.	AU-6, CA-7, IR-4, IR-5, IR-8, SI-4	A.12.4.1, A.16.1.7	CO-OPS-001, PR-CIR-001
		DE.AE-4: Impact of events is determined.	CP-2, IR-4, RA-3, SI-4	A.16.1.4	PR-INF-001

Cybersecurity Framework v1.1				Standards & Best Practices	
Function	Category	Subcategory	NIST SP 800-53 R4	ISO/IEC 27001:2013	NIST SP 800-181
		DE.AE-5: Incident alert thresholds are established.	IR-4, IR-5, IR-8	A.16.1.4	PR-CIR-001
	Security Continuous Monitoring (DE.CM)	DE.CM-1: The network is monitored to detect potential cybersecurity events.	AC-2, AU-12, CA-7, CM-3, SC-5, SC-7, SI-4		OM-NET-001
		DE.CM-3: Personnel activity is monitored to detect potential cybersecurity events.	AC-2, AU-12, AU-13, CA-7, CM-10, CM-11	A.12.4.1, A.12.4.3	AN-TWA-001
		DE.CM-4: Malicious code is detected.	SI-3, SI-8	A.12.2.1	SP-DEV-001
		DE.CM-5: Unauthorized mobile code is detected.	SC-18, SI-4, SC-44	A.12.5.1, A.12.6.2	SP-DEV-001
		DE.CM-7: Monitoring for unauthorized personnel, connections, devices, and software is performed.	AU-12, CA-7, CM-3, CM-8, PE-3, PE-6, PE-20, SI-4	A.12.4.1, A.14.2.7, A.15.2.1	AN-TWA-001
	Detection Processes (DE.DP)	DE.DP-2: Detection activities comply with all applicable requirements.	AC-25, CA-2, CA-7, SA-18, SI-4, PM-14	A.18.1.4, A.18.2.2, A.18.2.3	PR-CDA-001
RESPOND (RS)	Response Planning (RS.RP)	RS.RP-1: Response plan is executed during or after an incident.	CP-2, CP-10, IR-4, IR-8	A.16.1.5	PR-CIR-001

Cybersecurity Framework v1.1				Standards & Best Practices	
Function	Category	Subcategory	NIST SP 800-53 R4	ISO/IEC 27001:2013	NIST SP 800-181
	Communica-tions (RS.CO)	RS.CO-2: Incidents are reported consistent with established criteria.	AU-6, IR-6, IR-8	A.6.1.3, A.16.1.2	IN-FOR-002
	Analysis (RS.AN)	RS.AN-1: Notifications from detection systems are investigated.	AU-6, CA-7, IR-4, IR-5, PE-6, SI-4	A.12.4.1, A.12.4.3, A.16.1.5	PR-CDA-001
		RS.AN-2: The impact of the incident is under-stood.	CP-2, IR-4	A.16.1.4, A.16.1.6	PR-CIR-001
		RS.AN-3: Forensics are performed.	AU-7, IR-4	A.16.1.7	IN-FOR-002
		RS.AN-4: Incidents are categorized consistent with response plans.	CP-2, IR-4, IR-5, IR-8	A.16.1.4	PR-CIR-001
	Mitigation (RS.MI)	RS.MI-1: Incidents are contained.	IR-4	A.12.2.1, A.16.1.5	PR-CIR-001
		RS.MI-2: Incidents are mitigated.	IR-4	A.12.2.1, A.16.1.5	PR-CIR-001

3.5 Technologies

Table 3-2 lists all of the technologies used in this project and provides a mapping among the generic application term, the specific product used, and the security control(s) the product provides. Refer to Table 3-1 for an explanation of the NIST Cybersecurity Framework Subcategory codes.

Table 3-2 Products and Technologies

Component	Product	Function	Cybersecurity Framework Subcategories
Integrity Monitoring	Tripwire Enterprise v8.7 Semperis Directory Services Protector (DSP) v2.7	• Provides file hashes and integrity checks for files and software, regardless of file type. • Provides integrity monitoring for data. • Provides integrity monitoring for Active Directory.	PR.DS-6, DE.AE-1, DE.CM-3, DE.CM-7
Event Detection	Cisco Advanced Malware Protection (AMP) v5.4 Glasswall FileTrust ATP for Email v6.90.2.5 Cisco Stealthwatch v7.0.0	• Provides the ability to receive information about new threats. • Provides the ability to statically detect malicious software.	DE.AE-3, DE.CM-1, DE.CM-4, DE.CM-5, DE.CM-7

Component	Product	Function	Cybersecurity Framework Subcategories
	Semperis DSP v2.7	• Provides ability to dynamically detect malicious software. • Provides ability to detect malicious email attachments. • Provides ability to scan the network for anomalies. • Provides the ability to monitor user behavior for anomalies. • Provides ability to scan email attachments for deviations from file type specifications or organizational policy.	
Logging	Micro Focus ArcSight Enterprise Security Manager (ESM) v7.0 Patch 2	• Provides auditing and logging capabilities configurable to organizational policy. • Correlates logs of cybersecurity events with user information. • Provides automation for logging.	DE.AE-1, DE.AE-3, DE.AE-4, DE.CM-1, DE.CM-3, DE.CM-7, RS.AN-2
	Tripwire Log Center v7.3.1		
Forensics/Analytics	Cisco AMP v5.4	• Provides forensics to track effects of malware retrospectively. • Provides network traffic analysis. • Provides ability to analyze files sent over the network. • Provides analysis capabilities for finding anomalies in enterprise activity.	DE.AE-2, DE.AE-4, DE.CM-1, RS.RP-1, RS.AN-1, RS.AN-2, RS.AN-3
	Symantec Security Analytics v8.0.1		
	Micro Focus ArcSight ESM v7.0 Patch 2		
	Symantec Information Centric Analytics (ICA) v6.5.2		
	Cisco AMP v5.4		

Component	Product	Function	Cybersecurity Framework Subcategories
Mitigation and Containment	Cisco Identity Services Engine (ISE) v2.4 Glasswall FileTrust ATP for Email v6.90.2.5 Semperis DSP v2.7	• Provides ability to sandbox files locally. • Provides ability to enforce policy across the enterprise. • Provides ability to quarantine devices across the enterprise. • Provides ability to sanitize files through file reconstruction. • Provides ability to revert changes to domain services.	DE.CM-5, RS.RP-1, RS.MI-1, RS.MI-2
Reporting	Micro Focus ArcSight ESM v7.0 Patch 2	• Provides ability to send security alerts based on organizational policy. • Provides ability to provide reports of enterprise health. • Provides ability to provide reports of malware detection across the enterprise.	DE.AE-5, RS.RP-1, RS.CO-2

4 Architecture

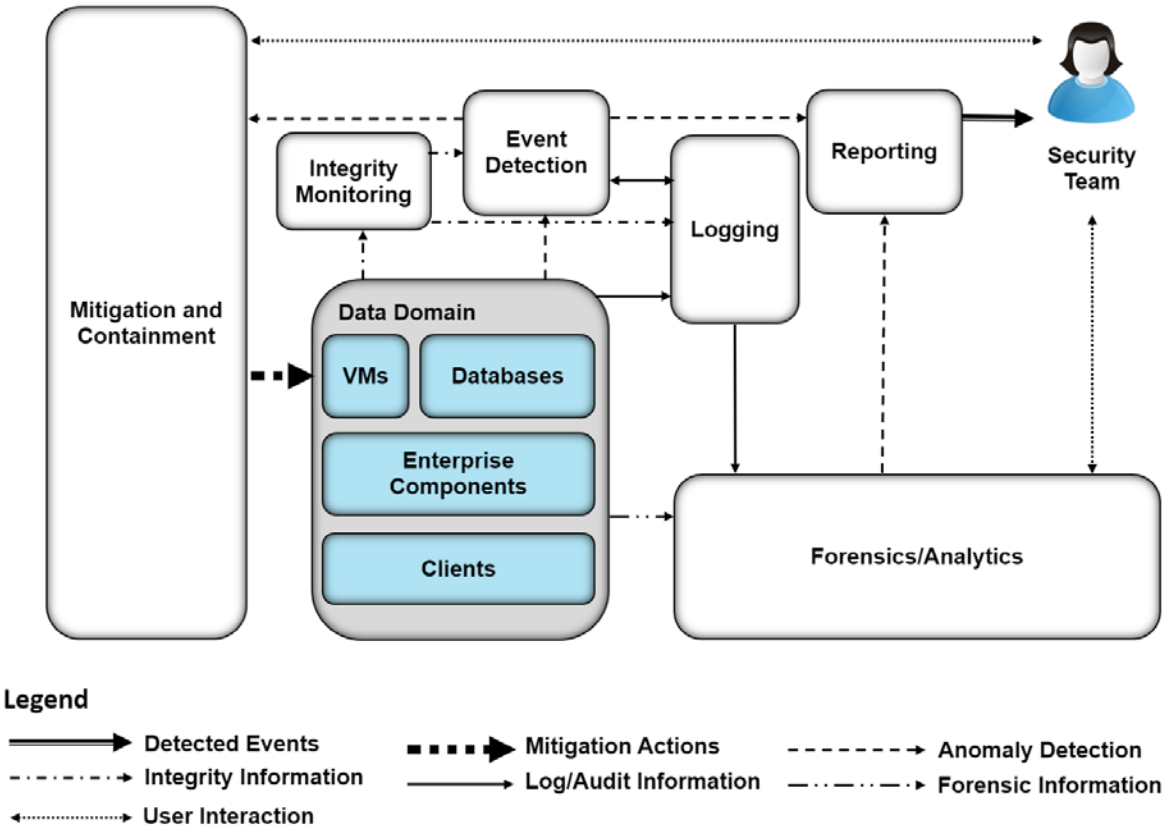
This section presents the high-level architecture used for implementation of a DI solution that detects and responds to ransomware and other destructive events.

4.1 Architecture Description

4.1.1 High-Level Architecture

The DI solution is designed to address the security Functions and Subcategories described in Table 3-1 and is composed of the capabilities illustrated in Figure 4-1.

Figure 4-1 DI Detect & Respond High-Level Architecture



- **Integrity monitoring** provides capabilities for comparing current system states against established baselines.

- **Event detection** provides capabilities for detecting ongoing events and can be composed of intrusion detection, malware detection, user anomaly detection, and others, depending on the established threat model of the organization.
- **Logging** records and stores all the log files produced by components within the enterprise.
- **Forensics/analytics** provides the capability to probe/analyze logs and machines within the enterprise to learn from DI events.
- **Mitigation and containment** allows responding to DI events by containing and limiting the threat's ability to affect the system.
- **Reporting** provides the capability to report on all activities within the enterprise and within the reference architecture for analysis by a security team.

These capabilities work together to provide the Detect and Respond Functions for DI. The integrity monitoring capability collects integrity information prior to attacks so that when an attack happens, records of all file/system changes are preserved. In combination with event detection, these records not only function as a tool to inform recovery but also as early indicators of compromise. Event detection uses these records and its own mechanisms to actively detect events as they happen and to take appropriate action through other components of the reference architecture. Logging collects information from event detection and integrity monitoring for use in response functions. Mitigation and containment provides capabilities to stop ongoing attacks and limit their effect on the system. Forensics/analytics allow analysis of logs and threat behavior to aid the organization in learning from the attack. Reporting provides capabilities for reporting information from analysis and logging to the appropriate parties both during and after an attack. The information gained from these attacks can be used to inform products that fall in the Identify Function of the Cybersecurity Framework to indicate vulnerabilities in the enterprise that need to be remediated.

4.1.2 Architecture Components

4.1.2.1 Integrity Monitoring

The integrity monitoring component provides the ability to test, understand, and measure attacks that occur on files and components within the enterprise. When considering DI from the perspective of detecting and responding to an active attack, being able to track changes to files is critical. Asset integrity changes can provide an early detection mechanism by tracking changes made at abnormal times or by tracking users who typically do not make such changes. Furthermore, the changes tracked during a DI event can be used to inform the recovery process; they provide information about what changes happened, when changes began to take place, as well as what programs were involved in the changes.

Integrity monitoring typically requires an operation baseline to be taken prior to the start of a DI event—this baseline is used for comparison against the system's state during an attack.

For the integrity monitoring capability, we use a combination of two tools: Tripwire Enterprise and Semperis DSP. Once a baseline is taken prior to an attack, Tripwire Enterprise stores integrity information for selected data across all systems. When a "check" is run, Tripwire collects all the changes that occurred to monitored files on those systems. These changes are forwarded to the logging component, which can then report and alert on them, becoming an indicator of a DI event. Furthermore, these collected changes can be used to help remediate the effects of malware on a system.

Semperis DSP provides a similar function but with a focus on Active Directory. Changes to Active Directory users, groups, and other services are collected and can be used to notify administrators of potentially malicious activity. Given the sensitive nature of Active Directory, Semperis DSP does not rely on a single source of information but instead monitors multiple aspects of Active Directory. This helps ensure that any change to permissions or privileged credentials is captured, including changes that attackers attempt to hide (for example, by circumventing security auditing).

4.1.2.2 Event Detection

The event detection component provides the ability to detect events as they happen. This can be achieved through a combination of mechanisms, depending on the needs of the organization. Analysis of integrity monitoring logs can indicate malicious activity. Malware detection, behavior-based anomaly detection, and intrusion detection are all potential examples of event detection. The goal of this component is to detect events as they happen, to trigger the appropriate responses, and to provide information about the attack to the security team.

For the event detection capability, we use a combination of tools. Cisco AMP is used to detect malicious files. Glasswall FileTrust ATP for Email is used to identify malicious email attachments that do not conform to file standards and organizational policies. Cisco Stealthwatch is used to detect malicious network activity. Finally, Semperis DSP is used to detect changes in Active Directory. Information from these four can be correlated to identify malicious patterns of behavior from users.

4.1.2.3 Logging

Logging from each component serves several functions in an architecture that aims to detect and respond to active DI events. Logs are produced through integrity monitoring and event detection, which aid other components in responding to active events. Both mitigation and containment and forensics/analytics use logs to inform their actions—logs tell them what systems are being affected and what programs are causing the event. Further, these logs help decide what steps should be taken to remediate the attack and protect against it going forward.

For the logging capability, we use a combination of two tools: Micro Focus ArcSight and Tripwire Log Center. While Tripwire Log Center's purpose in this build is primarily to collect, transform, and forward logs from Tripwire Enterprise to ArcSight, ArcSight performs a wider function. ArcSight collects logs from

various sources in the enterprise, such as event detection and integrity monitoring, as well as Windows event logs and Ubuntu syslogs. The goal of this widespread collection is to provide a base for the forensics/analytics component.

4.1.2.4 Mitigation and Containment

The mitigation and containment component provides the ability to limit a destructive event's effect on the enterprise. This component may be able to interact with a security team for greater effectiveness and may have the option to provide automated response to certain DI events. This response can involve stopping execution of associated programs, disabling user accounts, disconnecting a system from the network, and more, depending on the threat. Other actions may involve removing software from a system, restarting services, or copying the threat to a safe environment for analysis.

For the mitigation and containment capability, we use a combination of tools. Cisco AMP provides the ability to remove malicious files on sight—combined with its event detection capability, this can be leveraged to immediately respond to malware on user systems. Cisco ISE provides quarantine functions that can be used to respond to detected malware and poor machine posture as well as to network events in Stealthwatch. Semperis DSP provides the ability to immediately and automatically revert detected changes in Active Directory, mitigating the use of backdoors and other malicious domain changes. Semperis DSP can also disable user accounts to prevent further changes from compromised or maliciously created accounts. Glasswall provides the ability to sanitize malicious or noncompliant email attachments before they ever reach the user's inbox, thereby eliminating malicious content in email attachments.

4.1.2.5 Forensics/Analytics

The forensics/analytics component uses the logs generated by event detection and the enterprise to discover the source and effects of the DI event and learn about how to prevent similar events in the future, if possible. This component will typically allow an organization to analyze malware or logs related to the malware's execution and produce information such as: the servers that the malware communicates with, or the executable's signature, to improve detection of the malware in the future. Furthermore, the ability to examine machines affected by malware for lasting effects may be desirable. The information gained from forensic analysis can also be used to enhance the organization's protections against malware and potentially reform policy in the organization.

For the forensics/analytics capability, we use a combination of tools. Cisco AMP provides the ability to review the history of malicious files to determine the source and movement across the enterprise. Symantec Security Analytics provides the ability to analyze network traffic in a similar manner. ArcSight ESM provides event correlation capabilities for logs collected from almost all the other capabilities, allowing processing of events before they are reported to the security team. Symantec ICA provides additional analysis capabilities for logs as well as aggregation and visualization of certain potentially

malicious movements within the enterprise. These products aid in the future prevention of such attacks as well as determine the scope of the event's effect on the system.

4.1.2.6 Reporting

The reporting component is primarily an interface between various components of the architecture and the security team. It allows alerting based on events through email and dashboards, depending on the organization's need. The reporting capabilities are best used throughout the entirety of an event—they can be used to alert the security team when an event starts as well as to provide regular status updates when events are not happening or have just finished.

For the reporting capability, we use Micro Focus ArcSight. ArcSight can send email alerts and generate reports based on the log correlation and analysis that it performs. By ensuring integration of as many relevant logs as possible with ArcSight's logging capabilities, we can use various indicators to trigger alerts when certain logs or sets of logs are received by ArcSight.

5 Security Characteristic Analysis

The purpose of the security characteristic analysis is to understand the extent to which the project meets its objective of demonstrating a DI detect-and-respond solution. In addition, it seeks to understand the security benefits and drawbacks of the example solution.

5.1 Assumptions and Limitations

The security characteristic analysis has the following limitations:

- It is neither a comprehensive test of all security components nor a red-team exercise.
- It cannot identify all weaknesses.
- It does not include the lab infrastructure. It is assumed that devices are hardened. Testing these devices would reveal only weaknesses in implementation that would not be relevant to those adopting this reference architecture.

5.2 Build Testing

The purpose of the security characteristic analysis is to understand the extent to which the building block meets its objective of detecting and responding to DI events. Furthermore, the project aims to facilitate analysis of these events during and after an attack. In addition, it seeks to understand the security benefits and drawbacks of the reference design.

5.3 Scenarios and Findings

One aspect of our security evaluation involved assessing how well the reference design addresses the security characteristics that it was intended to support. The Cybersecurity Framework Subcategories were used to provide structure to the security assessment by consulting the specific sections of each standard that are cited in reference to a Subcategory. The cited sections provide validation points that the example solution would be expected to exhibit. Using the Cybersecurity Framework Subcategories as a basis for organizing our analysis allowed us to systematically consider how well the reference design supports the intended security characteristics.

Below are the scenarios created to test various aspects of this architecture. More detailed resolutions and mappings of these scenarios' requirements to the Cybersecurity Framework can be found in Appendix D.

5.3.1 Ransomware via Web Vector and Self-Propagation

5.3.1.1 Scenario

The following scenario was simulated to test the architecture's defense against ransomware.

A user mistakenly downloads ransomware from an external web server. When the user executes this malicious software, it generates a cryptographic key, which is sent back to the external web server. The malware then utilizes a privilege escalation exploit to propagate across the network. The malicious software encrypts files on the machines to which it propagated and demands payment in exchange for decryption of these files.

5.3.1.2 Resolution

The build provides a significant defense in depth against this use case.

The **event detection** capability provides the ability to detect malicious software on the system and generate logs and alerts based on this activity. It also allows for the detection of suspicious network behavior, such as propagation.

The **mitigation and containment** capability provides the ability to halt execution of the ransomware and remove it from the system. Furthermore, it allows quarantine of the affected machine(s) from the network after detection of malicious activity.

The **integrity monitoring** capability provides the ability to collect changes to files, including changes made by the ransomware as well as the ransomware's first creation or download onto the system.

When forwarded to the **logging** capability, these logs in combination with others can be used to identify the scope of the attack.

The **reporting** capability uses logs from the above capabilities to report on malicious activity and to increase response time.

The **forensics/analytics** capability analyzes logs related to the event to provide information that can be used to strengthen defenses against the attack in the future. This includes the websites it communicated with or was downloaded from, the signature of the executable, and the scope of the attack.

5.3.1.3 Other Considerations

Because malware comes in many forms, it is imperative to have multiple layers of defense against it while also working to actively improve these defenses. An early defense against malware means denylisting known malicious sites. However, because this must be done entirely before the attack takes place, it is out of scope of this build.

This build suggests a forensics/analytics capability specifically for informing and strengthening the enterprise's defenses against future attacks. This is a function of the Respond Category—learning from attacks can inform defense of such attacks in the future, both in the Protect and Detect phases of the attack. Denylisting is one such defense that can be informed by the Respond Category, and event detection is another.

5.3.2 Destructive Malware via USB Vector

5.3.2.1 Scenario

The following scenario was simulated to test the architecture's defense against destructive malware.

A user finds an unmarked Universal Serial Bus (USB) device and inserts it into his or her system. The USB device contains malicious software that may run automatically or with user interaction. The malicious software modifies and deletes the user's files, removing text from text files and entirely deleting any media files it finds. The software does not offer a recovery mechanism as ransomware might, aiming only to corrupt files.

5.3.2.2 Resolution

The build provides several mechanisms to detect and mitigate this use case.

The **integrity monitoring** capability provides the ability to detect changes to the file system, allowing the changes and deletions to be detected and logged. Furthermore, information about what program (and by extension, where the program was located—that is, on a USB drive) is included in the logs.

The **logging** capability is used to collect logs from the integrity monitoring capability for posterity, as well as from Windows event logs to monitor usage of external drives in comparison to normal usage.

The **event detection** capability provides the ability to detect malicious files on the USB inserted into the system. It also can detect execution of these files.

The **mitigation and containment** capability provides the ability to stop malicious files from executing as well as delete the files on the USB drive.

5.3.2.3 Other Considerations

USB attacks do not always come in the form of disguised file-based malware. As USB attacks allow direct interfacing with the hardware of the system, they can aim to destroy the system via electrical attacks or involve impersonation of a keyboard or other devices to avoid detection and gain privileges. These attacks may be better mitigated through a thorough physical security policy and restrictions on the types of allowed connected devices. Advanced attacks that involve manipulation of hardware can become increasingly difficult to detect once plugged into the system. A prevention solution involving backups, physical security, and employee education is often more effective.

5.3.3 Accidental VM Deletion via Maintenance Script

5.3.3.1 Scenario

The following scenario was simulated to test the architecture's defense against data integrity events that occur on virtual machines.

A routine maintenance script on the system causes an error. During a move operation in the Hyper-V system, the script deletes an important virtual machine (VM). A maintenance script with an error of this type could be a side effect of a normal system function or an error made by a member of the organization. It is expected that the build will mitigate the damage caused to virtual machines in such an incident.

5.3.3.2 Resolution

The build provides several methods for detecting and analyzing this use case. Errors in custom code are often difficult to detect at run time and because they are usually run by privileged programs. Classifying them as malware or even as "unintended" changes is often undesirable.

The **integrity monitoring** capability provides the ability to detect changes to VM configurations, allowing the VM deletion to be detected and logged. Furthermore, information about what program (i.e., the routine maintenance script) is included in the logs.

The **logging** capability provides the ability to collect these events for posterity.

The **forensics/analytics** capability provides the ability to analyze the events after the fact to enable the security team to understand the impact, resolve the error in the script, and inform the restoration process.

5.3.3.3 Other Considerations

This solution will aid in identifying the script that causes a configuration change or deletion, but ultimately some things cannot be automated by the solution. Understanding the impact of the event requires a security team, and this build aims to provide the tools for a security team to do so.

Resolving an error in a maintenance script will also typically require effort on the part of the system administrators. Judgment on whether a script should be deleted, disabled, or left running during the remediation process is necessary and can depend on the size of the script, the affected assets, and the availability of resources to put toward resolving the error. Because of these considerations, the organization is left to decide whether a malfunctioning script should be treated like malware (see other scenarios that deal with malware) or as a part of the enterprise as it is possible that the remediation process is lengthy and exceeds the scope of the Detect/Respond Categories of the NIST Cybersecurity Framework.

5.3.4 Backdoor Creation via Email Vector

5.3.4.1 Scenario

The following scenario was simulated to test the architecture's defense against malicious email attachments.

A user unknowingly opens a malicious attachment that was received in an email. When opened, the attachment quietly fetches files from an external web server. It then creates several unapproved backdoor accounts on the authentication server. It is expected that the build will mitigate the impacts of such an incident.

5.3.4.2 Resolution

The build provides several layers of defense against this use case. The **integrity monitoring** capability forwards logs of file changes and Active Directory changes to the logging capability, allowing recording and detection of both the malicious attachment's download and the changes it makes to the system account structure.

The **logging** and **reporting** capabilities provide the ability to generate alerts based on events for the security team to quickly take action to resolve them.

The **event detection** capability provides detection at two points in time—both before the attachment reaches the user's inbox and, should this fail, after the attachment downloads to the system.

The **mitigation and containment** capability provides mitigation before the attachment reaches the user's inbox, as well as when it is on the user's system.

The **forensics/analytics** capability provides the ability to view the network traffic generated by the attachment when fetching its malicious files from the web server. This can inform defense of the enterprise in the Protect Category of the Cybersecurity Framework before any similar events happen in the future.

5.3.4.3 Other Considerations

Another defense that can partially prevent this use case is detection of the email as spam. However, as this is often a function of the email provider and not a separate security solution, it is out of scope for this build.

This build suggests a forensics/analytics capability specifically for informing and strengthening the defenses of the enterprise against future attacks. This is a function of the Respond Category—learning from attacks can inform the defense of such attacks in the future, both in the Protect and Detect phases of the attack.

5.3.5 Database Modification via Malicious Insider

5.3.5.1 Scenario

The following scenario was simulated to test the architecture's defense against unwanted database modification.

A malicious insider has access to an enterprise database through a web page. The insider leverages a vulnerability in the web page to delete a large portion of the database. Though this scenario deals with a web vulnerability, other vulnerabilities could be used to modify the database undesirably. It is expected that the build will mitigate the impact that a user can have on the database.

5.3.5.2 Resolution

The build provides several layers of defense against this use case. The **integrity monitoring** capability is used to detect changes to the database.

These changes are forwarded to the **logging** capability, which also collects information about web requests.

The **reporting** capability provides the ability to generate alerts and quickly inform the security team of an anomaly, based on the logs.

The **forensics/analytics** capability is used to investigate the malicious access as well as identify the page with the vulnerability. Because this vulnerability is a vulnerability in custom code, it is important for information-gathering mechanisms to be in place to provide ample information for the resolution of this vulnerability.

5.3.5.3 Other Considerations

This use case highlights the need for a response-oriented build to collaborate with an identify-oriented build. Identification and resolution of vulnerabilities in custom code are sometimes feasible only through gathering information after the vulnerability has been exploited. This build provides the mechanisms to gather such information, but it is ultimately up to the security team to resolve the vulnerability and learn from the attack.

5.3.6 File Modification via Malicious Insider

5.3.6.1 Scenario

The following scenario was simulated to test the architecture's defense against malicious file and backup modification.

A malicious insider is assumed to have stolen administrator-level credentials through non-technical means. The insider, using these credentials, uses remote Windows PowerShell sessions to uniformly modify employee stock information to their benefit across several machines. This attack will also target the enterprise's backup system to modify all records of the previous stock information. It is expected that the aspects of the build described above will mitigate the ability of the user to target and modify enterprise data and backups. The method of securing administrator credentials will be considered out of scope for this solution.

5.3.6.2 Resolution

The build has several layers of defense against this use case. The **integrity monitoring** capability detects changes to files and backups caused by a malicious insider.

When forwarded to the **logging** and **reporting** capabilities, the build can report on these changes. Irregularities or differences from the normal backup schedule are important indicators of a compromise.

When the security team is alerted to a malicious insider, they can use the **mitigation and containment** capability to disable the insider's access.

5.3.6.3 Other Considerations

Malicious insiders are powerful adversaries, because they already have some level of access to the system. The existence of malicious insiders widens the threat surface of an enterprise to needing defense against internal machines as well as external machines. For this reason, this build includes mitigations against threats already present inside the enterprise and not just threats that originate externally. This includes the ability to disable user accounts, quarantine machines, and monitor network traffic originating from within the enterprise.

5.3.7 Backdoor Creation via Compromised Update Server

5.3.7.1 Scenario

The following scenario was simulated to test the architecture's defense against compromised update servers.

An update server that services an enterprise machine is compromised and provides an update to the enterprise machine that contains a backdoor. The update contains a vulnerable version of vsftpd, allowing an attacker root access into the machine updated by the compromised server. It is expected that the build will mitigate the impact of a compromised update server.

5.3.7.2 Resolution

The build has several layers of defense against this use case. **Integrity monitoring** detects changes to programs, providing information about how and when the program was changed. It also detects changes to any files made by an intruder.

The **event detection** capability is used to detect the malicious update through signature detection. Furthermore, it detects the connection to the open port by an attacker.

The **mitigation and containment** capability is used to delete/quarantine the malicious update, stopping the port from being accessible. It can also be used to quarantine the machine from the network, to prevent the spread of the intrusion and remove the attacker's access.

5.3.7.3 Other Considerations

The use of the event detection capability to detect largely assumes that the update has been reported as vulnerable, either through a well-known history of being vulnerable or through intelligence-sharing channels. As such, an event detection capability would, in some cases of new custom attacks, be unable to detect this at first sight. However, the build provides other tools, such as monitoring network activity, that can alert security staff to such attacks.

Using a data integrity identify-and-protect build to incorporate denylisting and network protection as part of the defense is beneficial, as a use case that involves connecting to an unused port would be entirely defeated by a network protection allowlist of approved ports.

6 Future Build Considerations

The NCCoE is creating an overarching guide to combining the architectures of the various DI projects: Identify and Protect, Detect and Respond, and Recover. These architectures share some commonalities, such as integrity monitoring, as well as some potential integrations and cycles that could not be expressed in just one of the practice guides. The different Functions of the Cybersecurity Framework are intended to prepare and inform one another, and the overarching guide addresses those issues.

The NCCoE is also considering additional data security projects that map to the Cybersecurity Framework Core Functions of Identify, Protect, Detect, Respond, and Recover. These projects will focus on data confidentiality—the defense of enterprise systems from attacks that would compromise the secrecy of data.

Appendix A List of Acronyms

AMP	Advanced Malware Protection
ATP	Advanced Threat Protection
COI	Community of Interest
DE	Detect
DI	Data Integrity
DSP	Directory Services Protector
ESM	Enterprise Security Manager
ICA	Information Centric Analytics
ISE	Identity Services Engine
IT	Information Technology
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
NISTIR	NIST Interagency or Internal Report
PR	Protect
RMF	Risk Management Framework
RS	Respond
SP	Special Publication
USB	Universal Serial Bus
VM	Virtual Machine
vsftpd	Very Secure File Transfer Protocol Daemon

Appendix B Glossary

Access Control	The process of granting or denying specific requests to: 1) obtain and use information and related information processing services; and 2) enter specific physical facilities (e.g., federal buildings, military establishments, border crossing entrances) SOURCE: Federal Information Processing Standard (FIPS) 201; CNSSI-4009
Architecture	A highly structured specification of an acceptable approach within a framework for solving a specific problem. An architecture contains descriptions of all the components of a selected, acceptable solution, while allowing certain details of specific components to be variable to satisfy related constraints (e.g., costs, local environment, user acceptability). SOURCE: FIPS 201-2
Audit	Independent review and examination of records and activities to assess the adequacy of system controls and ensure compliance with established policies and operational procedures. SOURCE: CNSSI 4009-2015
Backdoor	An undocumented way of gaining access to a computer system. A backdoor is a potential security risk. SOURCE: National Institute of Standards and Technology (NIST) Special Publication (SP) 800-82 Rev. 2
Backup	A copy of files and programs made to facilitate recovery if necessary. SOURCE: NIST SP 800-34 Rev. 1
Compromise	Disclosure of information to unauthorized persons, or a violation of the security policy of a system in which unauthorized intentional or unintentional disclosure, modification, destruction, or loss of an object may have occurred. SOURCE: NIST SP 800-32

Continuous Monitoring	Maintaining ongoing awareness to support organizational risk decisions. SOURCE: NIST SP 800-137
Cybersecurity	Prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation. SOURCE: CNSSI 4009-2015 (NSPD-54/HSPD-23)
Data	A subset of information in an electronic format that allows it to be retrieved or transmitted. SOURCE: CNSSI-4009
Data Integrity	The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. SOURCE: CNSSI-4009
Information Security	The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability. SOURCE: FIPS 199 (44 U.S.C., Sec. 3542)
Information Security Risk	The risk to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation due to the potential for unauthorized access, use, disclosure, disruption, modification, or destruction of information and/or information systems. SOURCE: CNSSI 4009-2015 (NIST SP 800-30 Rev. 1)
Information System	A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information. SOURCE: FIPS 200 (44 U.S.C., Sec. 3502)
Insider	An entity inside the security perimeter that is authorized to access system resources but uses them in a way not approved by those who granted the authorization.

	SOURCE: NIST SP 800-82 Rev. 2 (RFC 4949)
Kerberos	An authentication system developed at the Massachusetts Institute of Technology (MIT). Kerberos is designed to enable two parties to exchange private information across a public network. SOURCE: NIST SP 800-47
Log	A record of the events occurring within an organization's systems and networks. SOURCE: NIST SP 800-92
Malware	A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system. SOURCE: NIST SP 800-111
Privacy	Assurance that the confidentiality of, and access to, certain information about an entity is protected. SOURCE: NIST SP 800-130
Risk	The level of impact on organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals, resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring. SOURCE: FIPS 200
Risk Assessment	The process of identifying the risks to system security and determining the probability of occurrence, the resulting impact, and additional safeguards that would mitigate this impact. Part of Risk Management and synonymous with Risk Analysis. SOURCE: NIST SP 800-63-2
Risk Management Framework	The Risk Management Framework (RMF), presented in NIST SP 800-37, provides a disciplined and structured process that integrates information security and risk management activities into the system development life cycle. SOURCE: NIST SP 800-82 Rev. 2 (NIST SP 800-37)

Security Control	A protection measure for a system. SOURCE: NIST SP 800-123
Virtual Machine	Software that allows a single host to run one or more guest operating systems. SOURCE: NIST SP 800-115
Vulnerability	Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. SOURCE: FIPS 200 (adapted from CNSSI 4009)

Appendix C References

[1] A. Sedgewick, *Framework for Improving Critical Infrastructure Cybersecurity,* Version 1.1, National Institute of Standards and Technology, Gaithersburg, Maryland, Apr. 2018, 55 pp. Available: https://www.nist.gov/cyberframework/framework.

[2] L. Kauffman, N. Lesser and B. Abe, *Executive Technical Workshop on Improving Cybersecurity and Consumer Privacy*, NISTIR 8050, National Institute of Standards and Technology, Gaithersburg, Maryland, April 2015, 155pp. Availabe: https://nccoe.nist.gov/sites/default/files/library/nistir-8050-draft.pdf

[3] G. Stoneburner, *et al*., *Guide for Conducting Risk Assessments*, NIST Special Publication (SP), 800-30 Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, September 2012, 95 pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-30r1.

[4] R. Ross, *et al*., *Guide for Applying the Risk Management Framework to Federal Information Systems*, NIST Special Publication (SP) 800-37, National Institute of Standards and Technology, Gaithersburg, Maryland, February 2010, 101pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-37r1.

[5] R. Ross *et al*., *Managing Information Security Risk*, NIST Special Publication (SP) 800-39, National Institute of Standards and Technology, Gaithersburg, Maryland, March 2011, 87pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-39.

[6] M. Souppaya *et al.*, *Guide to Enterprise Patch Management Technologies*, NIST Special Publication (SP) 800-40 Revision 3, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2013, 25pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-40r3.

[7] R. Ross *et al.*, *Security and Privacy Controls for Federal Information Systems and Organizations*, NIST Special Publication (SP) 800-53 Revision 4, National Institute of Standards and Technology, Gaithersburg, Maryland, April 2013, 461pp. Available: https://doi.org/10.6028/NIST.SP.800-53r4.

[8] U.S. Department of Commerce. *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards (FIPS) Publication 140-3, Mar. 2019, 65pp. Available: https://csrc.nist.gov/publications/detail/fips/140/3/final.

[9] K. Kent *et al.*, *Guide to Integrating Forensic Techniques into Incident Response*, NIST Special Publication (SP) 800-86, National Institute of Standards and Technology, Gaithersburg, Maryland, August 2006, 121pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-86.

[10] K. Kent and M. Souppaya, *Guide to Computer Security Log Management*, NIST Special Publication (SP) 800-92, National Institute of Standards and Technology, Gaithersburg, Maryland, September 2006, 72pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-92.

[11] P. Bowen *et al.*, *Information Security Handbook: A Guide for Managers*, NIST Special Publication (SP) 800-100, National Institute of Standards and Technology, Gaithersburg, Maryland, October 2006, 178pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-100.

[12] M. Swanson *et al.*, *Contingency Planning Guide for Federal Information Systems*, NIST Special Publication (SP) 800-34 Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, May 2010, 148pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-34r1.

[13] Office of Management and Budget (OMB), *Management of Federal Information Resources*, OMB Circular No. A-130, November 2000. Available: https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/OMB/circulars/a130/a130revised.pdf.

[14] P. Cichonski *et al.*, *Computer Security Incident Handling Guide*, NIST Special Publication (SP) 800-61 Revision 2, National Institute of Standards and Technology, Gaithersburg, Maryland, August 2012, 79pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-61r2.

[15] M. Souppaya and K. Scarfone, *Guide to Malware Incident Prevention and Handling for Desktops and Laptops*, NIST Special Publication (SP) 800-83 Revision 1, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2013, 46pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-83r1.

[16] C. Johnson *et al.*, *Guide to Cyber Threat Information Sharing*, NIST Special Publication (SP) 800-150, National Institute of Standards and Technology, Gaithersburg, Maryland, October 2016, 42pp. Available: http://dx.doi.org/10.6028/NIST.SP.800-150.

[17] M. Bartock *et al.*, *Guide for Cybersecurity Event Recovery*, NIST Special Publication (SP) 800-184, National Institute of Standards and Technology, Gaithersburg, Maryland, December 2016, 52pp. http://dx.doi.org/10.6028/NIST.SP.800-184.

Appendix D Functional Evaluation

A functional evaluation of the data integrity (DI) example implementation, as constructed in our laboratory, was conducted to verify that it meets its objective of detecting and responding to DI events. Furthermore, this project aims to analyze the events to aid recovery and protection of the enterprise against future attacks. The evaluation verified that the example implementation could perform the following functions:

- Detect malicious network activity, malicious mobile code, malicious code execution, and unauthorized user behavior.
- Contain and analyze these types of incidents.
- Mitigate the impact of these incidents as they occur.
- Report relevant details for use in mitigation and protection against future events.

Section D.1 describes the format and components of the functional test cases. Each functional test case is designed to assess the capability of the example implementation to perform the functions listed above and detailed in Section D.1.

D.1 Data Integrity Functional Test Plan

One aspect of our security evaluation involved assessing how well the reference design addresses the security characteristics that it was intended to support. The Cybersecurity Framework Subcategories were used to provide structure to the security assessment by consulting the specific sections of each standard that are cited in reference to that Subcategory. The cited sections provide validation points that the example solution is expected to exhibit. Using the Cybersecurity Framework Subcategories as a basis for organizing our analysis allowed us to systematically consider how well the reference design supports the intended security characteristics.

This plan includes the test cases necessary to conduct the functional evaluation of the DI example implementation, which is currently deployed in a lab at the National Cybersecurity Center of Excellence. The implementation tested is described in Section 4.

Each test case consists of multiple fields that collectively identify the goal of the test, the specifics required to implement the test, and how to assess the results of the test. Table 6-1 describes each field in the test case.

Table 6-1 Test Case Fields

Test Case Field	Description
Parent requirement	Identifies the top-level requirement or the series of top-level require-ments leading to the testable requirement.

Test Case Field	Description
Testable requirement	Drives the definition of the remainder of the test case fields. Specifies the capability to be evaluated.
Description	Describes the objective of the test case.
Associated Cybersecurity Framework Subcategories	Lists the National Institute of Standards and Technology Special Publication 800-53 rev 4 controls addressed by the test case.
Preconditions	The starting state of the test case. Preconditions indicate various starting state items, such as a specific capability configuration required or specific protocol and content.
Procedure	The step-by-step actions required to implement the test case. A procedure may consist of a single sequence of steps or multiple sequences of steps (with delineation) to indicate variations in the test procedure.
Expected results	The expected results for each variation in the test procedure.
Actual results	The observed results.
Overall result	The overall result of the test as pass/fail. In some test-case instances, the determination of the overall result may be more involved, such as determining pass/fail based on a percentage of errors identified.

D.2 Data Integrity Use Case Requirements

Table 6-2 identifies the DI functional requirements addressed in the test plan and associated test cases.

Table 6-2 Capability Requirements

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 1	The DI example implementation shall detect and respond to malware that encrypts files and displays notice demanding payment.		Data Integrity DR-1
CR 1.a		File integrity changes are collected and logged.	Data Integrity DR-1
CR 1.b		Access is halted.	Data Integrity DR-1

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 1.c		Executable is identified as malicious, using a denylist.	Data Integrity DR-1
CR 1.d		Executable is identified as malicious through analysis, and denylist is updated.	Data Integrity DR-1
CR 1.e		Execution is halted.	Data Integrity DR-1
CR 1.f		Downloads are identified as malicious, using a denylist.	Data Integrity DR-1
CR 1.g		Downloads are identified as malicious through analysis, and denylist is updated.	Data Integrity DR-1
CR 1.h		Downloads are prevented.	Data Integrity DR-1
CR 1.i		Attempts to propagate are detected.	Data Integrity DR-1
CR 1.j		Machines attempting to propagate are prevented from propagating.	Data Integrity DR-1
CR 1.k		Suspicious network traffic is detected, and denylist is updated.	Data Integrity DR-1

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 2	The DI example implementation shall detect and respond to malware inserted via Universal Serial Bus (USB) that modifies and deletes user data.		Data Integrity DR-2
CR 2.a		File integrity changes are collected and logged.	Data Integrity DR-2
CR 2.b		The insertion of a USB device is detected and logged.	Data Integrity DR-2
CR 2.c		The executable is identified as malicious, using a denylist.	Data Integrity DR-2
CR 2.d		The executable is identified as malicious through analysis, and the denylist is updated.	Data Integrity DR-2
CR 2.e		Malicious executable is halted or deleted.	Data Integrity DR-2
CR 3	The DI example implementation shall detect and respond to virtual machine deletion.		Data Integrity DR-3
CR 3.a		Virtual machine integrity changes are collected and logged.	Data Integrity DR-3

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 3.b		The event causing deletion of the virtual machine is analyzed.	Data Integrity DR-3
CR 4	The DI example implementation shall detect and respond to malware received via phishing email.		Data Integrity DR-4
CR 4.a		Configuration integrity changes are collected and logged.	Data Integrity DR-4
CR 4.b		Email is identified as malicious, using a denylist.	Data Integrity DR-4
CR 4.c		Email is identified as malicious through analysis, and the denylist is updated.	Data Integrity DR-4
CR 4.d		Email is deleted or sorted into spam.	Data Integrity DR-4
CR 4.e		The attachment is identified as malicious, using a denylist.	Data Integrity DR-4
CR 4.f		The attachment is identified as malicious through analysis, and the denylist is updated.	Data Integrity DR-4
CR 4.g		Execution of the spreadsheet is stopped, and the denylist is updated if necessary.	Data Integrity DR-4

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 4.h		The downloads are identified as malicious, using a denylist.	Data Integrity DR-4
CR 4.i		The downloads are identified as malicious through analysis, and the denylist is updated.	Data Integrity DR-4
CR 4.j		The malicious executable is halted or deleted.	Data Integrity DR-4
CR 4.k		Suspicious network traffic is detected, and denylist is updated.	Data Integrity DR-4
CR 5	The DI example implementation shall detect and respond to changes to the database made through a web server vulnerability in custom code.		Data Integrity DR-5
CR 5.a		Database integrity changes are collected and logged.	Data Integrity DR-5
CR 5.b		Information about the client interacting with the web service is collected and logged.	Data Integrity DR-5
CR 5.c		Information from the attack is reported for use in protection against future events.	Data Integrity DR-5

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 6	The DI example implementation shall detect and respond to targeted modification by malicious insiders with elevated privileges.		Data Integrity DR-6
CR 6.a		File integrity changes are collected and logged.	Data Integrity DR-6
CR 6.b		Backup integrity changes are collected and logged.	Data Integrity DR-6
CR 6.c		Detected changes are reported.	Data Integrity DR-6
CR 6.d		Associated user accounts are contained.	Data Integrity DR-6
CR 7	The DI example implementation shall detect and respond to an intrusion via compromised update server.		Data Integrity DR-7
CR 7.a		Program integrity changes are collected and logged.	Data Integrity DR-7
CR 7.b		The downloaded service is identified as malicious, using a denylist.	Data Integrity DR-7
CR 7.c		The downloaded service is identified as malicious through analysis, and the denylist is updated.	Data Integrity DR-7

Capability Requirement (CR) ID	Parent Requirement	Sub Requirement 1	Test Case
CR 7.d		The service is halted and reverted or deleted.	Data Integrity DR-7
CR 7.e		The download site is temporarily added to the denylist.	Data Integrity DR-7
CR 7.f		The port opened by the service is detected.	Data Integrity DR-7
CR 7.g		The opened port is closed.	Data Integrity DR-7
CR 7.h		The intrusion into the infected machine is detected.	Data Integrity DR-7
CR 7.i		The intrusion into the infected machine is contained.	Data Integrity DR-7

D.3 Test Case: Data Integrity DR-1

Table 6-3 Test Case ID: Data Integrity DR-1

Parent requirement	(CR 1) The DI example implementation shall detect and respond to malware that encrypts files and displays notice demanding payment.
Testable requirement	(CR 1.a) Integrity Monitoring, Logging, Reporting, (CR 1.c, CR 1.d, CR 1.f, CR 1.g, CR 1.i) Event Detection, (CR 1.b, CR 1.e, CR 1.j) Mitigation and Containment, (CR 1.h, CR 1.k) Forensics and Analytics
Description	Show that the DI solution has capabilities to detect behaviors typical of ransomware, and mitigate these behaviors appropriately.
Associated Cybersecurity Framework Subcategories	PR.DS-6, DE.AE-5, DE.CM-5, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2,DE.CM-4, DE.CM-7, DE.DP-2, DE.AE-1, DE.CM-1
Preconditions	User navigates to a malicious website and clicks on an ad for a virus cleaner. The virus cleaner is ransomware, which propagates across the domain and encrypts user files.
Procedure	The **integrity monitoring** capability is used to monitor and log changes to the integrity of files. The **logging** capability and the **reporting** capability are used to notify the security team of changes to the integrity of files and of potentially malicious events. The **event detection** capability is used to detect the ransomware in real time before or during its execution. It is also used to detect propagation of the ransomware. The **mitigation and containment** capability is used to halt the ransomware's execution and delete it from the system. It is also used to quarantine affected machines once a breach is discovered. The **forensics/analytics** capability is used to discover malicious hosts and websites accessed by the ransomware.
Expected Results (pass)	The build can monitor and report changes to the integrity of files (CR 1.a). The machine is quarantined when malware is detected (CR 1.b).

	Malicious executables are identified through signature detection or analysis (CR 1.c, CR 1.d). Malicious executables are prevented from executing (CR 1.e). Malicious downloads are identified through signature detection or analysis (CR 1.f, CR 1.g). Malicious downloads are prevented (CR 1.h). Propagation of malicious executables is detected (CR 1.i). Propagation of malicious executables is prevented (CR 1.j). Network traffic is captured and analyzed for suspicious activity (CR 1.k).
Actual Results	**Tripwire Enterprise (integrity monitoring)** is used to successfully detect changes to files on the affected systems. **ArcSight ESM (logging)** is used to successfully log events from event detection and integrity monitoring for use in reporting and forensics/analytics. **ArcSight ESM (reporting)** is used to successfully report on malicious activity detected in logs. **Cisco AMP (event detection)** is used to successfully detect the malicious executable. **Cisco AMP (mitigation and containment)** is used to successfully remove malicious executables from the affected systems. **Cisco Stealthwatch (event detection)** is used to successfully capture malicious or suspicious network traffic from the executable. **Cisco ISE (mitigation and containment)** is used to successfully quarantine affected machines. **Symantec Security Analytics (forensics/analytics)** is used to successfully review network traffic generated by the ransomware for potentially malicious hosts and websites.

	Symantec ICA (forensics/analytics) successfully displays relevant events from ArcSight for analysis to aid in identifying the malicious files for use in future event detection as well as for removal by the security team.
Overall Result	Pass. All requirements for this use case are met.

D.4 Test Case: Data Integrity DR-2

Table 6-4 Test Case ID: Data Integrity DR-2

Parent requirement	(CR 2) The DI example implementation shall detect and respond to malware inserted via USB that modifies and deletes user data.
Testable requirement	(CR 2.a) Integrity Monitoring, (CR 2.b, CR 2.c) Event Detection, (CR 2.d) Forensics and Analytics, (CR 2.e) Mitigation and Containment
Description	Show that the DI solution can detect behaviors of destructive malware and can mitigate these behaviors appropriately.
Associated Cybersecurity Framework Subcategories	DE.AE-5, DE.CM-4, DE.CM-7, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2
Preconditions	A user inserts an unidentified USB drive into their computer. They click on a file on the drive, which immediately destroys any files on their machine.
Procedure	The **integrity monitoring** capability is used to monitor integrity changes to the system. The **logging** capability is used to collect logs from the integrity monitoring capability. The **event detection** capability is used to detect malicious files on the USB inserted into the system. The **mitigation and containment** capability is used to prevent malicious files from executing.
Expected Results (pass)	The build can monitor and report changes to the integrity of files (CR 2.a). The build can detect insertion of a USB (CR 2.b). Malicious executables are identified through signature detection or analysis (CR 2.c, CR 2.d).

	Malicious executables are prevented from executing (CR 2.e).
Actual Results	**Tripwire Enterprise (integrity monitoring)** successfully detects changes made by an executable running from a USB. **ArcSight ESM (logging)** successfully collects logs from the integrity monitoring capability. Furthermore, USB insertions can be collected by using Windows group policy. **Cisco AMP (event detection)** successfully detects malicious files on the USB drive. **Cisco AMP (mitigation and containment)** immediately deletes these malicious files on the system if they are copied. It also prevents execution if the file is run from the USB drive.
Overall Result	Pass (partial). Cisco AMP does not immediately delete the file from the USB drive when it is plugged in if the user does not make any action (copy or execution). However, because both these actions trigger deletion, this is not a significant shortcoming as the file is otherwise harmless.

D.5 Test Case: Data Integrity DR-3

Table 6-5 Test Case ID: Data Integrity DR-3

Parent requirement	(CR 3) The DI example implementation shall detect and respond to virtual machine deletion.
Testable requirement	(CR 3.a) Integrity Monitoring, (CR 3.b) Forensics and Analytics
Description	Show that the DI solution can detect and analyze DI events that involve virtual machines.
Associated Cybersecurity Framework Subcategories	DE.AE-5, DE.CM-3, DE.CM-7, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2
Preconditions	A routine maintenance script contains an error that accidentally deletes a virtual machine.
Procedure	The **integrity monitoring** capability is used to monitor integrity changes to the system. The **logging** capability is used to collect logs from the integrity monitoring capability.

	The **forensics/analytics** capability is used to analyze logs and determine the cause of integrity events.
Expected Results (pass)	The build can monitor and report changes to the integrity of virtual machines (CR 3.a). The build can analyze the impact of DI events (CR 3.b).
Actual Results	**Tripwire Enterprise (integrity monitoring)** successfully monitors and logs changes to configurations of virtual machines. **ArcSight ESM (logging)** successfully collects logs and reports on the events generated by the integrity monitoring capability, enabling faster response time. **Symantec ICA (forensics/analytics)** successfully displays relevant events from ArcSight for analysis to aid in identifying the file that causes the deletion.
Overall Result	Pass. All requirements for this use case are met.

D.6 Test Case: Data Integrity DR-4

Table 6-6 Test Case ID: Data Integrity DR-4

Parent requirement	(CR 4) The DI example implementation shall detect and respond to malware received via phishing email.
Testable requirement	(CR 4.a) Integrity Monitoring and Logging, (CR 4.b, CR4.e, CR 4.h, CR 4.k) Event Detection, (CR 4.c, CR 4.f, CR 4.i) Forensics and Analytics, (CR 4.d, CR 4.g, CR 4.j) Mitigation and Containment
Description	Show that the DI solution can detect malicious attachments and respond to malicious configuration changes.
Associated Cybersecurity Framework Subcategories	PR.DS-6, DE.AE-5, DE.CM-5, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2
Preconditions	The user receives a phishing email with a malicious spreadsheet attached. The spreadsheet is downloaded and opened, causing account changes in Active Directory.
Procedure	The **integrity monitoring** capability is used to detect and log the account creation.

	This information is forwarded to the **logging** capability, along with other available Active Directory information. The email attachment is detected as malicious by the **event detection** capability and mitigated by the **mitigation and containment** capability, both when the file is in the inbox and when it is on the user's system. The solution can review the network traffic generated by the file when it calls out to the malicious web server to download files through **forensics/analytics**.
Expected Results (pass)	The build can monitor and report changes to the integrity of configurations (CR 4.a). Malicious emails are identified through signature detection or analysis (CR 4.b, CR 4.c). Emails identified as malicious are sorted into spam or deleted (CR 4.d). Malicious attachments are identified through signature detection or analysis (CR 4.e, CR 4.f). Malicious attachments are prevented from executing (CR 4.g). Malicious downloads are identified through signature detection or analysis (CR 4.h, CR 4.i). Malicious executables are prevented from executing (CR 4.j). Network traffic is captured and analyzed for suspicious activity (CR 4.k).
Actual Results	**Semperis DSP (integrity monitoring)** successfully monitors and logs changes to Active Directory. **ArcSight ESM (logging)** successfully collects logs and reports on the events generated by the integrity monitoring capability, enabling faster response time. **Glasswall FileTrust (event detection)** successfully identifies the malicious attachment before it reaches the user's inbox.

	Glasswall FileTrust (mitigation and containment) successfully mitigates the malicious attachment before it reaches the user's inbox. The malicious file is successfully uploaded to **Cisco AMP (event detection)** for signature detection. **Cisco AMP (event detection)** successfully mitigates the file when found on user workstations. **Symantec Security Analytics (forensics/analytics)** is used to successfully detect network traffic involving download of files from the malicious server.
Overall Result	Pass (partial). Emails are not sorted into spam (CR 4.b–d); rather, the attachment is mitigated before reaching the user's inbox. Sorting emails into spam is often a function of the email infrastructure.

D.7 Test Case: Data Integrity DR-5

Table 6-7 Test Case ID: Data Integrity DR-5

Parent requirement	(CR 5) The DI example implementation shall detect and respond to changes to the database made through a web server vulnerability in custom code.
Testable requirement	(CR 5.a) Integrity Monitoring, (CR 5.b) Logging, (CR 5.c) Reporting
Description	Show that the DI solution can detect and respond to an exploitation a vulnerability in custom code that leads to an attack on the database.
Associated Cybersecurity Framework Subcategories	DE.AE-5, DE.CM-3, DE.CM-7, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2
Preconditions	A vulnerability in the source code of an intranet web page is discovered by a malicious insider. The insider exploits this vulnerability to delete significant portions of the database.
Procedure	The **integrity monitoring** capability is used to detect changes to the database. The **logging** capability is used to monitor changes to the database and to log web requests.

	The **reporting** capability is used to alert the security team of significant changes to the database. The **forensics/analytics** capability is used to investigate the malicious access as well as identify the page with the vulnerability.
Expected Results (pass)	The build can monitor and report changes to the integrity of the database (CR 5.a). Malicious interaction with the web server is detected (CR 5.b). Information about the attack is reported for use in maintaining the enterprise systems (CR 5.c).
Actual Results	**Tripwire Enterprise (integrity monitoring)** successfully monitors changes to the database configuration. **ArcSight ESM (logging)** successfully logs changes to the database and web requests. **ArcSight ESM (reporting)** successfully alerts the security team of changes to the database. **Symantec Security Analytics (forensics/analytics)** allows identification of web requests that could have caused the deletion, helping identify the web server's vulnerability in custom code.
Overall Result	Pass. All requirements for this use case are met.

D.8 Test Case: Data Integrity DR-6

Table 6-8 Test Case ID: Data Integrity DR-6

Parent requirement	(CR 6) The DI example implementation shall detect and respond to targeted modification by malicious insiders with elevated privileges.
Testable requirement	(CR 6.a, 6.b) Integrity monitoring, (CR 6.c) Reporting, (CR 6.d) Mitigation and Containment
Description	Show that the DI solution can detect and respond to targeted modification of assets and backups by malicious insiders.
Associated Cybersecurity Framework Subcategories	DE.AE-5, DE.CM-3, DE.CM-7, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2

Preconditions	A malicious insider attempts to modify targeted information in both the enterprise systems and the backup systems by using elevated credentials obtained extraneously.
Procedure	The **integrity monitoring** capability is used to detect changes to the file system. The **reporting** capability is used to notify the security team of changes to critical data assets. The **mitigation and containment** capability is used to prevent the malicious user from making further modifications.
Expected Results (pass)	The build can monitor and report changes to the integrity of files and backups (CR 6.a, CR 6.b). Information about the attack is reported for use in responding to the threat (CR 6.c). User accounts associated with the attack are contained (CR 6.d).
Actual Results	**Tripwire Enterprise (integrity monitoring)** successfully detects changes to files and backups caused by a malicious insider. **ArcSight ESM (reporting)** successfully reports and alerts administrators via email on changes made to files by a malicious insider. **Semperis DSP (mitigation and containment)** successfully disables the user accounts associated with malicious insider activity.
Overall Result	Pass. All requirements for this use case are met.

D.9 Test Case: Data Integrity DR-7

Table 6-9 Test Case ID: Data Integrity DR-7

Parent requirement	(CR 7) The DI example implementation shall detect and respond to an intrusion via compromised update server.
Testable requirement	(CR 7.a) Integrity Monitoring, (CR 7.b) Event Detection, (CR 7.c) Forensics and Analytics, (CR 7.d, CR 7.e) Mitigation and Containment
Description	Show that the DI solution can detect a malicious update from a compromised update server as well as detect and respond to a resulting intrusion.

Associated Cybersecurity Framework Subcategories	PR.DS-6, DE.AE-5, DE.CM-5, DE.DP-2, RS.CO-2, DE.AE-2, DE.AE-3, DE.AE-4, RS.AN-1, RS.AN-2, RS.AN-3, RS.AN-4, RS.RP-1, RS.MI-1, RS.MI-2, DE.CM-4, DE.CM-7, DE.AE-1, DE.CM-1,
Preconditions	An external update server has been compromised, and a user workstation attempts to update from this server.
Procedure	The **integrity monitoring** capability is used to detect changes to the integrity of programs and files. The **event detection** capability is used to detect the malicious update. It is also used to detect the connection to the machine. The **mitigation and containment** capability is used to halt execution of the update and delete it. It is also used to contain the intrusion.
Expected Results (pass)	The build can monitor and report changes to the integrity of programs (CR 7.a). The malicious update is identified through signature detection or analysis (CR 7.b, CR 7.c). The malicious service is halted and reverted or deleted (CR 7.d). Other users are temporarily prevented from accessing this update server (CR 7.e). The port opened by the service is detected (CR 7.f). The port opened by the service is closed (CR 7.g). The intrusion is detected (CR 7.h). The intrusion is contained (CR 7.i).
Actual Results	**Tripwire Enterprise (integrity monitoring)** is used to identify changes in programs on the system as well as any changes made by the attacker. **Cisco AMP (event detection)** is used to detect the malicious update. **Cisco Stealthwatch (event detection)** is used to detect a connection to the machine via an unusual port.

	Cisco AMP (mitigation and containment) is used to halt the execution of the file and delete it, thereby closing the vulnerable port. **Cisco ISE (mitigation and containment)** is used to disconnect the affected machines from the network to prevent the spread of the intrusion.
Overall Result	Pass (partial). Cisco AMP does not seem to support network blocking for Unix machines at the time this practice guide was written—it supports only detection (it does support network blocking for Windows use cases, though, so a similar use case on Windows machines would potentially work). Instead, we rely on network protection, a DI Protect capability, to prevent further access to the update server; and on Cisco AMP's mitigation capabilities to remedy any known malicious files downloaded from the server.