Validating the Integrity of Computing Devices

Includes Executive Summary (A); Approach, Architecture, and Security Characteristics (B); and How-To Guides (C)

Jon Boyens
Christopher Brown
Chelsea Deane
Tyler Diamond*
Nakia Grayson
Celia Paulsen
William T. Polk
Andrew Regenscheid
Karen Scarfone
Murugiah Souppaya

Former employee; all work for this publication was done while at employer

DRAFT

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

NCCoE
NATIONAL CYBERSECURITY
CENTER OF EXCELLENCE

Validating the Integrity of Computing Devices

Includes Executive Summary (A); Approach, Architecture, and Security Characteristics (B); and How-To Guides (C)

Jon Boyens
Tyler Diamond*
Nakia Grayson
Celia Paulsen
William T. Polk
Andrew Regenscheid
Murugiah Souppaya
National Cybersecurity Center of Excellence
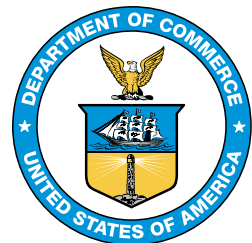*Information Technology Laboratory*

Christopher Brown
Chelsea Deane
The MITRE Corporation
McLean, Virginia

Karen Scarfone
Scarfone Cybersecurity
Clifton, Virginia

*Former employee; all work for this publication was done while at employer

DRAFT

June 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie Locasio, Under Secretary of Commerce for Standards and Technology & Director, National Institute of Standards and Technology

Validating the Integrity of Computing Devices

Volume A:
Executive Summary

Jon Boyens
Tyler Diamond*
Nakia Grayson
Celia Paulsen
William T. Polk
Andrew Regenscheid
Murugiah Souppaya
National Institute of Standards and Technology
Information Technology Laboratory

Karen Scarfone
Scarfone Cybersecurity
Clifton, Virginia

Former employee; all work for this publication was done while at employer

June 2022

DRAFT

1 Executive Summary

2 Organizations are increasingly at risk of cyber supply chain compromise, whether intentional or
3 unintentional. Cyber supply chain risks include counterfeiting, unauthorized production, tampering,
4 theft, and insertion of unexpected software and hardware. Managing these risks requires ensuring the
5 integrity of the cyber supply chain and its products and services. This project will demonstrate how
6 organizations can verify that the internal components and system firmware of the computing devices
7 they acquire are genuine and have not been unexpectedly altered during manufacturing, distribution, or
8 operational use.

9 CHALLENGE

10 Technologies today rely on complex, globally distributed and interconnected supply chain ecosystems to
11 provide highly refined, cost-effective, and reusable solutions. Most organizations' security processes
12 consider only the visible state of computing devices. The provenance and integrity of a delivered device
13 and its components are typically accepted without validating through technology that there were no
14 unexpected modifications. Provenance is the comprehensive history of a device throughout the entire
15 life cycle from creation to ownership, including changes made within the device or its components.
16 Assuming that all acquired computing devices are genuine and unmodified increases the risk of a
17 compromise affecting products in an organization's supply chain, which in turn increases risks to
18 customers and end users.

19 Organizations currently lack the ability to readily distinguish trustworthy products from others. Having
20 this ability is a critical foundation of cyber supply chain risk management (C-SCRM). C-SCRM is the
21 process of identifying, assessing, and mitigating the risks associated with the distributed and
22 interconnected nature of supply chains. C-SCRM presents challenges to many industries and sectors,
23 requiring a coordinated set of technical and procedural controls to mitigate cyber supply chain risks
24 throughout manufacturing, acquisition, provisioning, and operations.

> **This practice guide can help your organization:**
> - Avoid using compromised technology components in your products
> - Enable your customers to readily verify that your products are genuine and trustworthy
> - Prevent compromises of your own information and systems caused by acquiring and using compromised technology products

25 SOLUTION

26 To address these challenges, the NCCoE is collaborating with technology vendors to develop a prototype
27 implementation in harmony with the National Initiative for Improving Cybersecurity in Supply Chains
28 (NIICS), which emphasizes tools, technologies, and guidance focused on the developers and providers of
29 technology. NIICS' mission is to help organizations build, evaluate, and assess the cybersecurity of
30 products and services in their supply chains. This project aligns with that mission by demonstrating how

31 organizations can verify that the internal components of the computing devices they acquire are
32 genuine and have not been tampered with. This prototype relies on device vendors storing information
33 within each device and organizations using a combination of commercial off-the-shelf and open-source
34 tools that work together to validate the stored information. By doing this, organizations can reduce the
35 risk of compromise to products within their supply chains.

36 In this approach, device vendors create an artifact within each device that securely binds the device's
37 attributes to the device's identity. The customer who acquires the device can validate the artifact's
38 source and authenticity, then check the attributes stored in the artifact against the device's actual
39 attributes to ensure they match. A similar process can be used to periodically verify the integrity of
40 computing devices while they are in use.

41 Authoritative information regarding the provenance and integrity of the components provides a strong
42 basis for trust in a computing device. Hardware roots of trust are the foundation upon which the
43 computing system's trust model is built, forming the basis in hardware for providing one or more
44 security-specific functions for the system. Incorporating hardware roots of trust into acquisition and
45 lifecycle management processes enables organizations to achieve better visibility into supply chain
46 attacks and to detect advanced persistent threats and other attacks. By leveraging hardware roots of
47 trust as a computing device traverses the supply chain, we can maintain trust in the computing device
48 throughout its operational lifecycle.

49 This project will address several processes, including:

50   ▪ how to create verifiable descriptions of components and platforms, which may be done by
51   original equipment manufacturers (OEMs), platform integrators, and even information
52   technology (IT) departments;

53   ▪ how to verify devices and components within the single transaction between an OEM and a
54   customer; and

55   ▪ how to verify devices and components at subsequent stages in the system lifecycle in the
56   operational environment.

57 This project will also demonstrate how to inspect the verification processes themselves.

58 The following is a list of the project's collaborators.

Collaborator	Security Capability or Component
ARCHER	Integrated Risk Management Platform, Incident Management, Integrating Data from Asset Discovery and Management and Security Information and Event Management (SIEM) Systems
DELL Technologies	Manufacturer, Platform Integrity Validation System
eclypsium	Platform Integrity Validation System

	Manufacturer, Platform Integrity Validation System
	Manufacturer, Platform Integrity Validation System
	Security Information and Event Management
	Manufacturer, Platform Integrity Validation System
	Certificate Authority, Platform Integrity Validation System
	Manufacturer, Platform Integrity Validation System

59 While the NCCoE is using a suite of commercial products to address this challenge, this guide does not
60 endorse these particular products, nor does it guarantee compliance with any regulatory initiatives. Your
61 organization's information security experts should identify the products that will best integrate with
62 your existing tools and IT system infrastructure. Your organization can adopt this solution or one that
63 adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and
64 implementing parts of a solution.

HOW TO USE THIS GUIDE

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

67 **Business decision makers, including chief information security and technology officers** can use this
68 part of the guide, *NIST SP 1800-34a: Executive Summary*, to understand the drivers for the guide, the
69 cybersecurity challenge we address, our approach to solving this challenge, and how the solution could
70 benefit your organization.

71 **Technology, security, and privacy program managers** who are concerned with how to identify,
72 understand, assess, and mitigate risk can use *NIST SP 1800-34b: Approach, Architecture, and Security
73 Characteristics*. It describes what we built and why, including the risk analysis performed and the
74 security/privacy control mappings.

75 **IT professionals** who want to implement an approach like this can make use of *NIST SP 1800-34c: How-
76 To Guides*. It provides specific product installation, configuration, and integration instructions for
77 building the example implementation, allowing you to replicate all or parts of this project.

DRAFT

SHARE YOUR FEEDBACK

78

79 You can view or download the draft guide at https://www.nccoe.nist.gov/supply-chain-assurance. Help
80 the NCCoE make this guide better by sharing your thoughts with us. We recognize that technical
81 solutions alone will not fully enable the benefits of our prototype implementation, so we encourage
82 organizations to share lessons learned and best practices for integrating the C-SCRM processes
83 associated with implementing this guide.

84 To provide comments, join the community of interest, or learn more about the project and example
85 implementation, contact the NCCoE at supplychain-nccoe@nist.gov.

COLLABORATORS

86

87 Collaborators participating in this project submitted their capabilities in response to an open call in the
88 Federal Register for all sources of relevant security capabilities from academia and industry (vendors
89 and integrators). Those respondents with relevant capabilities or product components signed a
90 Cooperative Research and Development Agreement (CRADA) to collaborate with NIST in a consortium to
91 build this example solution.

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

DRAFT

Validating the Integrity of Computing Devices

Tyler Diamond*
Nakia Grayson
William T. Polk
Andrew Regenscheid
Murugiah Souppaya
National Institute of Standards and Technology
Information Technology Laboratory

Christopher Brown
The MITRE Corporation
McLean, Virginia

Karen Scarfone
Scarfone Cybersecurity
Clifton, Virginia

*Former employee; all work for this publication was done while at employer

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.

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to: supplychain-nccoe@nist.gov.

Public comment period: June 23, 2022 through July 25, 2022

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 at supplychain-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

Organizations are increasingly at risk of cyber supply chain compromise, whether intentional or unintentional. Cyber supply chain risks include counterfeiting, unauthorized production, tampering, theft, and insertion of unexpected software and hardware. Managing these risks requires ensuring the integrity of the cyber supply chain and its products and services. This project will demonstrate how organizations can verify that the internal components of the computing devices they acquire, whether laptops or servers, are genuine and have not been tampered with. This solution relies on device vendors storing information within each device, and organizations using a combination of commercial off-the-shelf and open-source tools that work together to validate the stored information. This NIST Cybersecurity Practice Guide provides a draft describing the work performed so far to build and test the full solution.

63 KEYWORDS

64 *computing devices; cyber supply chain; cyber supply chain risk management (C-SCRM); hardware root of*
65 *trust; integrity; provenance; supply chain; tampering.*

66 ACKNOWLEDGMENTS

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

Name	Organization
Jason Ajmo	The MITRE Corporation
Chelsea Deane	The MITRE Corporation
Spike E. Dog	The MITRE Corporation
Joe Sain	The MITRE Corporation
Thomas Walters	The MITRE Corporation
Andrew Medak	National Security Agency (NSA)
Lawrence Reinert	NSA
Manuel Offenberg	Seagate
David Kaiser	Seagate
Paul Gatten	Seagate
Simon Phatigaraphong	Seagate
Bill Downer	Seagate Government Solutions
Jack Fabian	Seagate Government Solutions

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

Technology Partner/Collaborator	Build Involvement
Archer	Archer Suite 6.9
Dell Technologies	PowerEdge R650, Secured Component Verification tool; Precision 3530, CSG Secured Component Verification tool
Eclypsium	Eclypsium Analytics Service, Eclypsium Device Scanner

Technology Partner/Collaborator	Build Involvement
HP Inc.	(2) Elitebook 840 G7, HP Sure Start, HP Sure Recover, Sure Admin, HP Client Management Script Library (CMSL), HP Tamperlock
Hewlett Packard Enterprise	Proliant DL360 Gen 10, Platform Certificate Verification Tool (PCVT)
IBM	QRadar SIEM
Intel	HP Inc. Elitebook 360 830 G5, Lenovo ThinkPad T480, Transparent Supply Chain Tools, Key Generation Facility, Cloud Based Storage, TSCVerify and AutoVerify software tools
National Security Agency (NSA)	Host Integrity at Runtime and Start-Up (HIRS), Subject Matter Expertise
Seagate Government Solutions	(3) 18TB Exos X18 hard drives, 2U12 Enclosure, Firmware Attestation API, Secure Device Authentication API

DOCUMENT CONVENTIONS

72

73 The terms "shall" and "shall not" indicate requirements to be followed strictly to conform to the
74 publication and from which no deviation is permitted. The terms "should" and "should not" indicate that
75 among several possibilities, one is recommended as particularly suitable without mentioning or
76 excluding others, or that a certain course of action is preferred but not necessarily required, or that (in
77 the negative form) a certain possibility or course of action is discouraged but not prohibited. The terms
78 "may" and "need not" indicate a course of action permissible within the limits of the publication. The
79 terms "can" and "cannot" indicate a possibility and capability, whether material, physical, or causal.

CALL FOR PATENT CLAIMS

80

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

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

89 a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not
90 currently intend holding any essential patent claim(s); or

91 b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring
92 to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft
93 publication either:

94   1. under reasonable terms and conditions that are demonstrably free of any unfair discrimination;
95   or
96   2. without compensation and under reasonable terms and conditions that are demonstrably free
97   of any unfair discrimination.

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

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

105 Such statements should be addressed to: supplychain-nccoe@nist.gov

Contents

List of Figures

194   ## List of Tables

1   Summary

Organizations are increasingly at risk of cyber supply chain compromise, whether intentional or unintentional. Cyber supply chain risks include counterfeiting, unauthorized production, tampering, theft, and insertion of unexpected software and hardware. Managing these risks requires ensuring the integrity of the cyber supply chain and its products and services. This prototype implementation will demonstrate how organizations can verify that the internal components of the computing devices they acquire are genuine and have not been unexpectedly altered during manufacturing or distribution processes.

This is an initial public draft version of the document which addresses gaps in the preliminary draft content (see Future Build Considerations in the preliminary draft). This draft may be updated in the future to address public comments or significant advances in the technology.

Further, this guide includes proof-of-concept software tools and services which have not been commercialized by our partner collaborators. We encourage early adopters to experiment with the guidelines in a test or development environment, with the understanding that they will identify gaps and challenges. The National Institute of Standards and Technology (NIST) welcomes early informal feedback and comments, which will be adjudicated after the specified public comment period.

This project has been conducted in two phases: laptop and server builds. The preliminary draft focused on validating the integrity of laptop hardware contributed by our technology partners. In this version of the publication, we incorporate hardware from our server manufacturing and component partners. The server build leverages and extends much of the laptop build architecture that is documented in the preliminary draft. In this update, we have also added a Security Information and Event Management (SIEM) component to the architecture that enhances our ability to monitor and detect unauthorized component swaps and firmware changes. We hope that this approach will provide organizations with a holistic methodology for managing supply chain risk.

For ease of use, the following provides a short description of each section in this volume.

Section 1, Summary, presents the challenge addressed by this National Cybersecurity Center of Excellence (NCCoE) project, including our approach to addressing the challenge, the solution demonstrated, and the benefits of the solution.

Section 2, How to Use This Guide, explains how business decision makers, program managers, and information technology (IT) and operational technology (OT) professionals might use each volume of the guide.

Section 3, Approach, offers a detailed treatment of the scope of the project, the risk assessment that informed the solution, and the technologies and components that industry collaborators supplied to build the example solution.

236   Section 4, Architecture, specifies the components of the prototype implementation and details how data
237   and communications flow between validation systems.

238   Section 5, Security Characteristic Analysis, provides details about the tools and techniques used to test
239   and understand the extent to which the project prototype implementation meets its objective:
240   demonstrating how organizations can verify that the components of their acquired computing devices
241   are genuine and have not been tampered with or otherwise modified throughout the devices' life cycles.

242   Section 6, Future Build Considerations, conveys the technical characteristics we plan to incorporate as
243   we continue to prototype with our collaborators.

244   Appendices A through C provide acronyms, a list of references cited in this volume, and project scenario
245   sequence diagrams, respectively.

1.1   Challenge

247   Technologies today rely on complex, globally distributed, and interconnected supply chain ecosystems
248   to provide highly refined, cost-effective, and reusable solutions. Most organizations' security processes
249   consider only the visible state of computing devices. The provenance and integrity of a delivered device
250   and its components are typically accepted without validating through technology that there have been
251   no unexpected modifications. *Provenance* is the comprehensive history of a device throughout the
252   entire life cycle from creation to ownership, including changes made within the device or its
253   components. Assuming that all acquired computing devices are genuine and unmodified increases the
254   risk of a compromise affecting products in an organization's supply chain, which in turn increases risks to
255   customers and end users, as illustrated in Figure 1-1. Mitigating this risk is not addressed at all in many
256   cases.

257   **Figure 1-1 Supply Chain Risk**
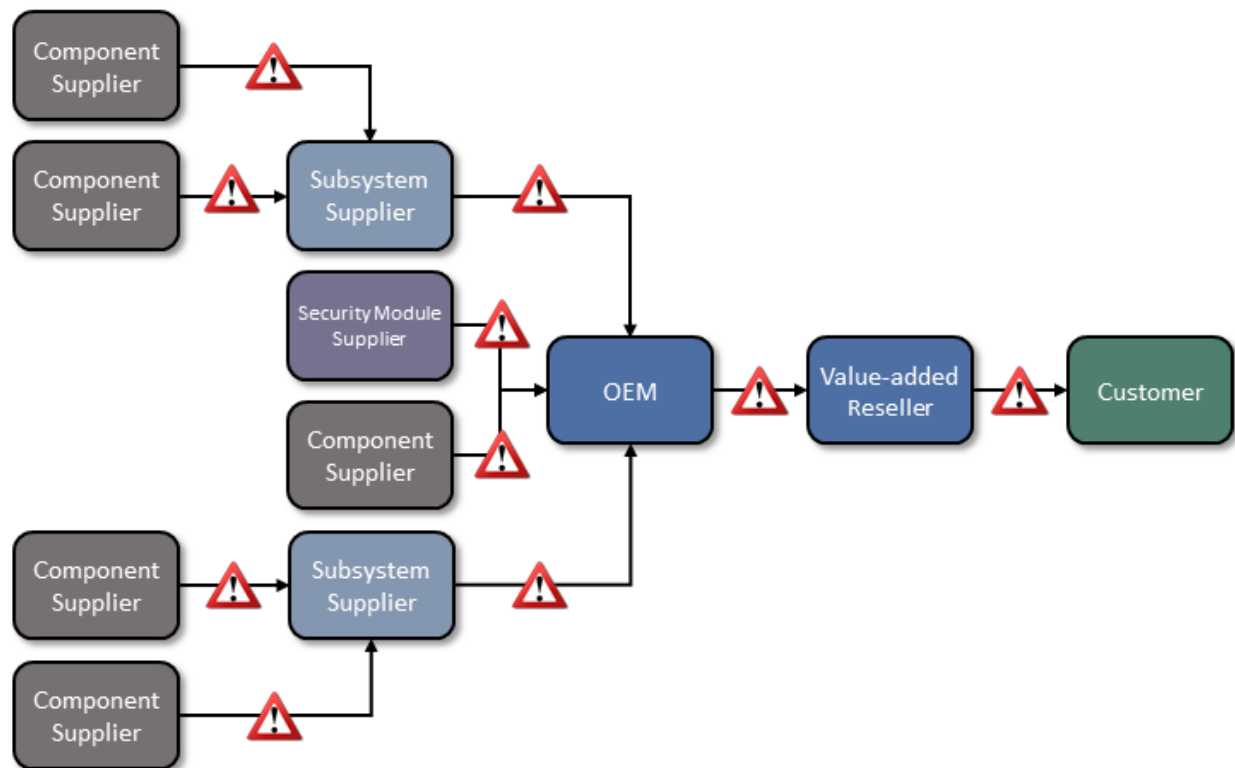
258   Organizations currently lack the ability to readily distinguish trustworthy products from others. At best,
259   government organizations could access an information source on counterfeit components such as the
260   Government-Industry Data Exchange Program (GIDEP), which contains information on equipment, parts,
261   and assemblies that are suspected to be counterfeit. Additionally, organizations with sufficient
262   resources could have acquisition quality assurance programs that examine manufacturer supply chain
263   practices, perform spot-checks of deliveries, and/or require certificates of conformity.

264   Having this ability is a critical foundation of cyber supply chain risk management (C-SCRM). *C-SCRM*
265   is the process of identifying, assessing, and mitigating the risks associated with the distributed and
266   interconnected nature of supply chains. C-SCRM presents challenges to many industries and sectors,
267   requiring a coordinated set of technical and procedural controls to mitigate cyber supply chain risks
268   throughout manufacturing, acquisition, provisioning, and operations.

269   1.2   Solution

270   To address these challenges, the NCCoE is collaborating with technology vendors to develop a prototype
271   implementation. Once completed, this project [1] will demonstrate how organizations can verify that
272   the internal components of the computing devices they acquire are genuine and have not been

273 tampered with. This solution relies on device vendors storing information within each device, and
274 implementers using a combination of commercial off-the-shelf and open-source tools that work
275 together to validate the stored information. By doing this, organizations can reduce the risk of
276 compromise to products within their supply chains.

277 In this approach, device vendors create one or more artifacts within each device that securely bind
278 the device's attributes to the device's identity. An organization that acquires the device can validate the
279 artifacts' source and authenticity, then check the attributes stored in the artifacts against the device's
280 actual attributes to ensure they match before fielding the device to the end user. A similar process can
281 be used to periodically verify the integrity of computing devices while they are in use.

282 Hardware roots of trust are a central technology in our approach to enable the use of authoritative
283 information regarding the provenance and integrity of the components, which provide a strong basis
284 for trust in a computing device. A hardware root of trust is comprised of highly reliable firmware and
285 software components that perform specific, critical security functions. Hardware roots of trust are the
286 foundation upon which the computing system's trust model is built, forming the basis in hardware for
287 providing one or more security-specific functions for the system. By leveraging hardware roots of trust
288 as a computing device traverses the supply chain, we can maintain trust in the computing device
289 throughout its operational lifecycle.

290 Platform firmware and its associated configuration data is critical to the trustworthiness of a computing
291 system [2]. Because of the highly privileged position platform firmware has with hardware, in this
292 prototype we also leverage a system firmware integrity detection component that includes mechanisms
293 for detecting when platform firmware code and critical data have been corrupted. These mechanisms
294 complement the hardware authenticity process described above.

295 This project addresses several processes, including:

296 ▪ how to create verifiable descriptions of components and platforms, which may be done by
297 original equipment manufacturers (OEMs), platform integrators, and even IT departments;

298 ▪ how to verify the integrity and provenance of computing devices and components within the
299 single transaction between an OEM and a customer; and

300 ▪ how to continuously monitor the integrity of computing devices and components at subsequent
301 stages in the system lifecycle in the operational environment.

1.3 Benefits

303 This practice guide can help organizations, including but not limited to OEMs and third-party component
304 suppliers, to:

305 ▪ avoid using compromised technology components in your products

306 ▪ enable customers to readily verify that OEM products are genuine and trustworthy

307 ▪ prevent compromises of your organization's information and systems caused by acquiring and
308   using compromised technology products

2   How to Use This Guide

310 This is an initial public comment draft of Volume B of a NIST Cybersecurity Practice Guide.
311 Implementation of the prototype implementation at the NCCoE is ongoing. The NCCoE is providing this
312 draft to gather valuable feedback and inform stakeholders of the progress of the project. Organizations
313 should not attempt to implement this draft.

314 When completed, this NIST Cybersecurity Practice Guide will demonstrate a standards-based reference
315 design for verifying that the internal components of the computing devices organizations acquire are
316 genuine and have not been tampered with and provide readers with the information they need to
317 replicate the reference design. It is modular and can be deployed in whole or in part.

318 This guide contains three volumes:

319 ▪ NIST Special Publication (SP) 1800-34A: *Executive Summary*

320 ▪ NIST SP 1800-34B: *Approach, Architecture, and Security Characteristics*—what we built and why
321   **(you are here)**

322 ▪ NIST SP 1800-34C: *How-To Guides*—instructions for building the example solution

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

324 **Business decision makers, including chief security and technology officers,** will be interested in the
325 *Executive Summary, NIST SP 1800-34A*, which describes the following topics:

326 ▪ challenges that enterprises face in decreasing the risk of a compromise to products in their
327   supply chain

328 ▪ example solution built at the NCCoE

329 ▪ benefits of adopting the example solution

330 **Technology or security program managers** who are concerned with how to identify, understand, assess,
331 and mitigate risk will be interested in this part of the guide, *NIST SP 1800-34B*, which describes what we
332 did and why. The following sections will be of particular interest:

333 ▪ Section 3.4, Risk Assessment, provides a description of the risk analysis we performed

334 ▪ Section 3.5, Security Control Map, maps the security characteristics of this example solution to
335   cybersecurity standards and best practices

336 You might share the *Executive Summary, NIST SP 1800-34A*, with your leadership team members to help
337 them understand the importance of adopting a standards-based method for verifying that the internal
338 components of the computing devices they acquire are genuine and have not been tampered with.

339 **IT professionals** who want to implement an approach like this will find the whole practice guide useful.
340 Once the how-to portion of the guide, *NIST SP 1800-34C*, is complete, you will be able to use it to
341 replicate all or parts of the build created in our lab. The how-to portion of the guide provides specific
342 product installation, configuration, and integration instructions for implementing the example solution.
343 We will not re-create the product manufacturers' documentation, which is generally widely available.
344 Rather, we will show how we incorporated the products together in our environment to create an
345 example solution.

346 This guide assumes that IT professionals have experience implementing security products within the
347 enterprise. While we have used a suite of commercial and open-source products to address this
348 challenge, this guide does not endorse these particular products. Your organization can adopt this
349 solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point
350 for tailoring and implementing parts of a prototype implementation for verifying that the internal
351 components of the computing devices your organization acquires are genuine and have not been
352 tampered with. Your organization's security experts should identify the products that will best integrate
353 with your existing tools and IT system infrastructure. We hope that you will seek products that are
354 congruent with applicable standards and best practices. Section 3.6, Technologies, lists the products we
355 used and maps them to the cybersecurity controls provided by this reference solution.

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

2.1 Typographic Conventions

361 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.

362 3 Approach

363 Organizations currently lack the ability to readily distinguish trustworthy products from others. To
364 address this challenge, the NCCoE proposes an adaptable prototype implementation that organizations
365 can use to verify that the internal components of the computing devices they acquire are genuine and
366 have not been tampered with. The NCCoE leveraged the existing ongoing initiatives by the NIST C-SCRM
367 program, including workshop research findings and use case studies, that sought input from technology
368 and cybersecurity vendors, C-SCRM subject matter experts from academia, and government to define
369 the project scope and reference architecture.

370 This guide describes a proof-of-concept implementation of the approach—a prototype—that is intended
371 to be a blueprint or template for the general security community. It is important to note that the
372 prototype implementation presented in this publication is only one possible way to solve the security
373 challenges. It is not intended to preclude the use of other products, services, techniques, etc., that can
374 also solve the problem adequately, nor is it intended to preclude the use of any products or services not
375 specifically mentioned in this publication.

376 3.1 Audience

377 This guide is intended for organizations and individuals who are responsible for the acquisition,
378 provisioning, and configuration control of computing devices. Examples include IT
379 administrators/system administrators, incident response team members, and Security Operations
380 Center (SOC) staff. OEMs, value-added resellers (VARs), and component suppliers may also benefit from
381 the prototype and lessons-learned at the conclusion of this project.

382 3.2 Scope

383 The scope of the project is limited to manufacturing and OEM processes that protect against
384 counterfeits, tampering, and undocumented changes to firmware and hardware, and the corresponding
385 customer processes that verify that client and server computing devices and components have not been
386 tampered with or otherwise modified. Protection against undocumented changes to the operating
387 system (OS) is considered out of scope for this project. Manufacturing processes that cannot be verified
388 by the customer are also explicitly out of scope.

389 Further, this project is not intended to cover the entire supply chain risk management process; it will
390 focus on the acceptance testing portion of a more holistic defense-in-depth/defense-in breadth supply
391 chain risk management strategy. The project enables verification of the identity of computing devices
392 (including replacement parts and updates or upgrades) once they have been acquired but before they
393 are implemented or installed.

394 Finally, this draft volume documents our experiences with laptop (client) computing devices in a
395 Windows 10 environment and servers that use Linux operationally in the prototype. From this
396 perspective, we have defined the following three project scenarios which outline the prototype scope.

3.2.1 Scenario 1: Creation of Verifiable Platform Artifacts

398 An OEM, VAR, or other authoritative source creates a verifiable artifact that binds reference platform
399 attributes to the identity of the computing device. The platform attributes in this artifact (e.g., serial
400 number, embedded components, firmware and software information, platform configuration) are used
401 by the purchasing organization during acceptance and provisioning of the computing device. Customers
402 may also create their own platform artifacts to establish a baseline that could be used to validate
403 devices in the field.

3.2.2 Scenario 2: Verification of Components During Acceptance Testing

405 In this scenario, an IT administrator receives a computing device through non-verifiable channels
406 (e.g., off the shelf at a retailer) and wishes to confirm its provenance and authenticity as part of
407 acceptance testing to establish an authoritative asset inventory as part of an asset management
408 program.

3.2.3 Scenario 3: Verification of Components During Use

410 In this scenario, the computing device has been accepted by the organization (Scenario 2) and has been
411 provisioned for the end user. The computing device components are verified against the attributes and
412 measurements declared by the manufacturer or purchasing organization during operational usage.

3.3 Assumptions

414 This project is guided by the following assumptions:

415 ▪ The scenario activities above will augment, not replace, the capabilities of existing acceptance
416   testing tools, asset management systems, and configuration management systems.

417 ▪ Hardware roots of trust represent one technique that can thwart the above types of attacks to
418   the supply chain. However, OEMs may use different approaches to implement a hardware root
419   of trust solution because of hardware constraints or other business reasons.

420 ▪ Organizational computing devices lifecycle phases for technology include the following activities
421   defined in NIST SP 800-161 Revision 1, Cybersecurity Supply Chain Risk Management Practices
422   for Systems and Organizations [3]: integration (referred to as acceptance testing in this
423   demonstration), operations, and disposal.

3.4 Risk Assessment

424

425 NIST SP 800-30 Revision 1, Guide for Conducting Risk Assessments [4], states that risk is "a measure of
426 the extent to which an entity is threatened by a potential circumstance or event, and typically a function
427 of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of
428 occurrence." The guide further defines risk assessment as "the process of identifying, estimating, and
429 prioritizing risks to organizational operations (including mission, functions, image, reputation),
430 organizational assets, individuals, other organizations, and the Nation, resulting from the operation of
431 an information system. Part of risk management incorporates threat and vulnerability analyses, and
432 considers mitigations provided by security controls planned or in place."

433 The NCCoE recommends that any discussion of supply chain risk management should begin with a
434 comprehensive review of NIST SP 800-161 Revision 1, Cybersecurity Supply Chain Risk Management
435 Practices for Systems and Organizations [3] —publicly available material. While SP 800-161 is targeted to
436 U.S. federal agencies, much of the guidance is beneficial to private organizations interested in reducing
437 Information and Communications Technology (ICT) supply chain risk. NIST SP 800-161 defines an *ICT*
438 *supply chain compromise* as an occurrence within the ICT supply chain whereby an adversary jeopardizes
439 the confidentiality, integrity, or availability of a system or the information the system processes, stores,
440 or transmits. An ICT supply chain compromise can occur anywhere within the system development life
441 cycle of the product or service.

442 In addition, NIST SP 800-37 Revision 2, Risk Management Framework for Information Systems and
443 Organizations [5] provides Risk Management Framework guidance that gives a baseline for assessing
444 risks to information system assets, including threats to the IT system supply chain.

3.4.1 Threats

445

446 NIST SP 800-161 provides a framework of ICT supply chain threats including insertion of counterfeits,
447 unauthorized production, tampering, theft, and insertion of malicious software and hardware, as well as
448 poor manufacturing and development practices in the ICT supply chain. These threats are associated
449 with an organization's decreased visibility into, and understanding of, how the technology that it
450 acquires is developed, integrated, and deployed, as well as the processes, procedures, and practices
451 used to assure the integrity, security, resilience, and quality of the products and services. Exploits
452 created by malicious actors (individuals, organizations, or nation states) are often especially
453 sophisticated and difficult to detect, and thus are a significant risk to organizations. This prototype
454 implementation does not defend against all ICT threats, but Table 3-1 captures threats from NIST SP
455 800-161 that are relevant to this project.

456   **Table 3-1 NIST SP 800-161 Threat Events**

Threat Events	Description
Craft attacks specifically based on deployed IT environment.	Adversary develops attacks (e.g., crafts targeted malware) that take advantage of knowledge of the organizational IT environment.
Create counterfeit/spoof website.	Adversary creates duplicates of legitimate websites; when users visit a counterfeit site, the site can gather information or download malware.
Craft counterfeit certificates.	Adversary counterfeits or compromises a certificate authority (CA) so that malware or connections will appear legitimate.
Create and operate false front organizations to inject malicious components into the supply chain.	Adversary creates false front organizations with the appearance of legitimate suppliers in the critical life cycle path that then inject corrupted/malicious information system components into the organizational supply chain.
Insert counterfeit or tampered hardware into the supply chain.	Adversary intercepts hardware from legitimate suppliers. Adversary modifies the hardware or replaces it with faulty or otherwise modified hardware.
Insert tampered critical components into organizational systems.	Adversary replaces, through supply chain, subverted insider, or some combination thereof, critical information system components with modified or corrupted components.
Compromise design, manufacture, and/or distribution of information system components (including hardware, software, and firmware).	Adversary compromises the design, manufacture, and/or distribution of critical information system components at selected suppliers.
Conduct supply chain attacks targeting and exploiting critical hardware, software, or firmware.	Adversary targets and compromises the operation of software (e.g., through malware injections), firmware, or hardware that performs critical functions for organizations. This is largely accomplished as supply chain attacks on both commercial off-the-shelf and custom information systems and components.
Obtain unauthorized access.	Adversary with authorized access to organizational information systems gains access to resources that exceeds authorization.
Inadvertently introduce vulnerabilities into software products.	Due to inherent weaknesses in programming languages and software development environments, errors and vulnerabilities are introduced into commonly used software products.

3.4.2 Vulnerabilities

This document is guided by NIST SP 800-161 [3], which describes an ICT supply chain vulnerability as the following:

> "A vulnerability is a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source [FIPS 200], [NIST SP 800-34 Rev. 1], [NIST SP 800-53 Rev 4], [NIST SP 800-53A Rev. 4], [NIST SP 800-115]. Within the ICT SCRM context, it is any weakness in the system/component design, development, manufacturing, production, shipping and receiving, delivery, operation, and component end-of life that can be exploited by a threat agent. This definition applies to both the systems/components being developed and integrated (i.e., within the SDLC) and to the ICT supply chain infrastructure, including any security mitigations and techniques, such as identity management or access control systems. ICT supply chain vulnerabilities may be found in:
>
>   - The systems/components within the SDLC (i.e., being developed and integrated);
>   - The development and operational environment directly impacting the SDLC; and
>   - The logistics/delivery environment that transports ICT systems and components (logically or physically)."

In the context of this project, ICT products (including libraries, frameworks, and toolkits) or services originating anywhere (domestically or abroad) might contain vulnerabilities that can present opportunities for ICT supply chain compromises. For example, an adversary may have the power to insert a malicious component into a product. While it is important to consider all ICT vulnerabilities, in practice it is impossible to completely eliminate all of them. Therefore, organizations should prioritize vulnerabilities that may have a greater impact on their environment if exploited by an adversary.

Additionally, a goal of this prototype implementation is to document a capability that enables organizations to detect the exploitation of vulnerabilities that may exist in firmware over-the-air processes that would allow an attacker to gain a privileged position on the computing device. In this project, we introduce a continuous monitoring component within system firmware that organizations can incorporate into their continuous monitoring programs.

3.4.3 Risk

SP 800-161 Revision 1 [3] provides an analysis framework for organizations to assess supply chain risk by creating a *threat scenario*—a summary of potential consequences of the successful exploitation of a specific vulnerability or vulnerabilities by a threat agent. By performing this exercise, organizations can identify areas requiring increased controls. Here, we walk through a truncated example scenario that may be similar to a threat scenario faced by organizations who implement some or all parts of this prototype demonstration. Readers are encouraged to develop their own threat scenario assessment for their organization as part of a larger risk management program.

492   ### 3.4.3.1 Threat Scenario

493   A company purchases life cycle replacement server computing devices from a third-party VAR with
494   whom it has done business in the past. The business side of the company is pressuring the IT Operations
495   staff to rapidly replace the servers during off-hours to avoid downtime during regular business hours.
496   The IT department responds by accelerating its deployment schedule to nights and weekends, using
497   existing staff augmented with VAR technicians.

498   Following deployment of the new hardware, the IT department observes that computing performance is
499   actually slower in the subnets where the equipment has been installed. Two weeks of load tests are
500   conducted to validate the performance issues, culminating with a report that the new hardware is
501   actually 25% slower than the previous hardware.

502   At the same time, the company's Information Security department notices unusual traffic coming from
503   the new servers in the upgraded subnets. Their investigation finds that these servers in the affected
504   subnets are beaconing out to international IP addresses where the company has no business presence
505   or need. The servers generating the suspicious traffic are taken offline for further investigation.

506   The VAR is called, and their technicians perform a separate analysis, confirming the reduction in
507   computing performance. The VAR launches an investigation into the source of the servers that they sold
508   to the company and finds some of the components in the equipment in question, as well as a portion of
509   their existing stock of components, are counterfeit. The VAR sends a representative server to a security
510   company for analysis. The security company finds that in addition to counterfeit and substandard
511   components, embedded malware has been installed, enabling attackers to take control of the servers
512   and to deliver second-stage malware that enabled them to move laterally through the affected subnets
513   and compromise computers of interest. This also gave the attackers a persistent foothold inside the
514   company.

515   An internal audit finds multiple failures on the part of the purchasing department, the IT department,
516   and the Information Security group to have in place measures to ensure the provenance of the
517   equipment and the secure deployment of devices on the network.

518   As a result of the supply chain breach leading to the installation of compromised hardware, the
519   company suffered several adverse effects, including:

520   ▪ loss of intellectual property through data exfiltration

521   ▪ loss of employee productivity as a result of computers and network equipment being taken
522   offline

523   ▪ additional costs to the IT department for replacement computers and network equipment

524   ▪ loss of confidence with the company's client base

525   ▪ potential loss of revenue due to clients severing their relationship with the company

526 Consequently, the organization develops three mitigation strategies to address the identified risks, in
527 which two are chosen as shown in Table 3-2. One of the chosen strategies, *Increase provenance and*
528 *information requirements*, can be at least partially addressed by the final implementation of this project.
529 Table 3-2 presents a summary of an example threat scenario analysis framework that an organization
530 may use to determine the controls to implement that would cause the estimated residual risk of
531 counterfeit hardware to drop to an acceptable level.

532 **Table 3-2 C-SCRM Example Threat Scenario**

Threat Scenario	Threat Source:	Industrial espionage/cyber criminals
	Vulnerability:	Internal: Loss of intellectual property following system compromise
	Threat Event Description:	Counterfeit hardware with embedded malware introduced into company's network
	Existing Practices:	Hardware system test prior to deployment; network scanning
	Outcome:	Data exfiltration, system degradation, loss of productivity, loss of revenue
Risk	Impact:	30% chance of successful targeting and infiltration
	Likelihood:	40% chance of undetected compromise
	Risk Score (Impact x Likelihood):	High
	Acceptable Level of Risk:	Low (under 25%)
Mitigation	Potential Mitigating Strategies/ SCRM Controls:	1) Improve traceability capabilities 2) Increase provenance and information requirements 3) Choose another supplier
	Estimated Cost of Mitigating Strategies:	1) Cost 20% increase, impact 10% decrease 2) Cost 20% increase, impact 20% decrease 3) Cost 40% increase, impact 80% decrease
	New Risk Score:	Low
	Selected Strategies:	2) Increase provenance and information requirements 3) Choose another supplier
	Estimated Residual Risk:	10%

3.5 Security Control Map

534 The following tables map the security characteristics defined in our project description (Table 3-3) to the
535 applicable NIST Cybersecurity Framework [6] Functions, Categories, and Subcategories (Table 3-4) to

536 assist organizations better manage and reduce C-SCRM risk. We have also included a mapping to specific
537 SP 800-53 r5 security controls [7] and indicated (in bold) if the control is part of the SP 800-161 Revision
538 1 [3] baseline security controls to assist organizations interested in alignment with NIST C-SCRM best
539 practices.

540 **Table 3-3 Security Characteristics**

Identifier	Security Characteristic
1	Establish a strong device identity to support binding artifacts to a specific device.
2	Cryptographically bind platform attributes and other manufacturing information to a given computer system.
3	Establish assurance for multi-supplier production in which components are embedded at various stages.
4	Provide an acceptance test capability that validates source and integrity of assembled components for the recipient organization of the computer system.
5	Detect unexpected component (firmware) swaps or tampering during the life cycle of the computing device in an operational environment.

541 **Table 3-4 Security Characteristics and Controls Mapping**

Cybersecurity Framework v1.1			SP 800-53 R5	Security Characteristics Addressed
Function	Category	Subcategory		
Identify (ID)	Supply Chain Risk Management (ID.SC)	ID.SC-4: Suppliers and third-party partners are routinely assessed using audits, test results, or other forms of evaluations to confirm they are meeting their contractual obligations.	**AU-6**	5
	Asset Management (ID.AM)	ID.AM-1: Physical devices and systems within the organization are inventoried.	**CM-8**	4
Protect (PR)	Identity Management, Authentication and Access Control (PR.AC)	PR.AC-6: Identities are proofed and bound to credentials and asserted in interactions.	**IA-4**	1
	Data Security (PR.DS)	PR.DS-6: Integrity checking mechanisms are used to verify software, firmware, and information integrity.	SI-7	4, 5
		PR.DS-8: Integrity checking mechanisms are used to verify hardware integrity.	SA-10	4, 5

Cybersecurity Framework v1.1			SP 800-53 R5	Security Char-acteristics Ad-dressed
Function	Category	Subcategory		
	Protective Technology (PR.PT)	PR.PT-1: Audit/log records are determined, documented, implemented, and reviewed in accordance with policy	**AU-2**	5
Detect (DE)	Security Continuous Monitoring (DE.CM)	DE.CM-7: Monitoring for unauthorized personnel, connections, devices, and software is performed.	PE-20	5
	Detection Processes (DE.DP)	DE.DP-2: Detection activities comply with all applicable requirements	SR-9	1
NA	NA	NA	**SR-10**	5
NA	NA	NA	**SR-11**	1,3
NA	NA	NA	AU-10	4

3.6 Technologies

Table 3-5 lists all of the technologies used in this project and provides a mapping among the generic
component term, the specific product or technology used, the function or capability it provides, and the
Cybersecurity Framework Subcategories that the product helps support. Refer to Table 3-4 for an
explanation of the NIST Cybersecurity Framework Subcategory codes. While Archer is presented as an
Integrated Risk Management (IRM) platform in Table 3-5, we are only leveraging a subset of capabilities
of the platform in the project to manage risk by providing visibility, reporting, and alerting for the
managed assets at the firmware level.

Table 3-5 Products and Technologies

Component	Product/Technology	Function/Capability	Cybersecurity Framework Subcategories
Component or Subsystem Manufacturer	Intel Transparent Supply Chain	Tools and processes to ensure supply chain security from the manufacturer to the purchasing organization	ID.SC-4, PR.DS-6
	Seagate EXOS X18 18 Terabyte Hard Drive	Secure device authentication, firmware attestation	ID.SC-4, PR.AC-6, PR.DS-6, PR.DS-8
OEM or VAR	Dell Technologies		ID.SC-4

Component	Product/Technology	Function/Capability	Cybersecurity Framework Subcategories
	Hewlett Packard Enterprise	Manufactures computing devices and binds them to verifiable artifacts	
	HP Inc.		
	Intel		
Computing Device	Dell PowerEdge R650 Server	A client device (laptop) or server purchased by an organization to execute tasks by end users	ID.SC-4, PR.AC-6
	Dell Latitude 5420/5520		
	HPE ProLiant DL360		
	HP Inc. Elitebook 360 830 G5		
	HP Inc. 840 G7/Zbook Firefly 14 G7		
	Intel Server Board S2600WTT		
	Lenovo ThinkPad T480		
Integrated Risk Management Platform	Archer IRM Platform	Ensures computing devices and associated components are tracked, uniquely identified, and managed through integrations with Asset Discovery tools. Provides visibility and workflows for addressing security incidents imported from SIEM tools.	ID.AM-1, DE.CM-7
Configuration Management System	Microsoft Configuration Manager	Enforces corporate governance and policies through actions such as applying software patches and updates, removing denylisted software, and automatically updating configurations	DE.CM-7
Security Information and Event Management Tool	IBM QRadar	Performs real-time analysis of alerts and notifications generated by organizational information systems	DE.CM-7
Certificate Authority (CA)	Host Integrity at Runtime and Start-up (HIRS) Attestation Certificate Authority (ACA)	Issues an Attestation Identity Credential in accordance with Trusted Computing Group (TCG) specifications	PR.AC-6, PR.DS-8

Component	Product/Technology	Function/Capability	Cybersecurity Framework Subcategories
Platform Integrity Validation System	Eclypsium Analytic Platform	Validates the integrity of firmware installed on computing devices	PR.DS-6
	HIRS ACA	Validates platform components in accordance with TCG specifications	PR.DS-8
	Platform Certificate Verification Tool (PCVT)	Validates platform components in accordance with TCG specifications	PR.DS-8
	Secure Component Verification (SCV)	Validates platform components in accordance with TCG specifications	PR.DS-8
	Platform Manifest Correlation System	Ingests platform manifest data from participating manufacturers	ID.AM-1

3.6.1 Trusted Computing Group

The technology providers for this prototype implement standards from the TCG, a not-for-profit organization formed to develop, define, and promote open, vendor-neutral, global industry standards supportive of hardware-based roots of trust for interoperable trusted computing platforms. TCG developed and maintains the Trusted Platform Module (TPM) 2.0 specification [8], which defines a cryptographic microprocessor designed to secure hardware by integrating cryptographic keys and services. A TPM functions as a root of trust for storage, measurement, and reporting. TPMs are currently included in many computing devices.

This project applies this foundational technology to address the challenge of operational security by verifying the provenance of a delivered system from the time it leaves the manufacturer until it is introduced in the organization's operational environment. The TPM can be leveraged to measure and validate the state of the system, including:

- binding attributes about the computing device to a strong cryptographic device identity held by the TPM, and

- supporting measurement and attestation capabilities that allow an organization to inspect and verify device components and compare them to those found in the platform attribute credential and OEM-provided reference measurements.

4 Architecture

568

569 This project is based on the notional high-level architecture depicted in Figure 4-1 for an organization
570 incorporating C-SCRM technologies into its existing infrastructure. The architecture depicts a
571 manufacturer that creates a hardware-root-of-trust-backed verifiable artifact associated with a
572 computing device. The verifiable artifact is then associated with existing enterprise IT management
573 systems, such as asset and configuration management systems, during the provisioning process. Finally,
574 an inspection component measures and reports on hardware attributes and firmware measurements
575 during acceptance testing and operational use.

576 **Figure 4-1 Notional Architecture**

Component or Subsystem Manufacturers	produce → Components or Subsystems	that generate a	Verifiable Artifact
OEM or Value Added Reseller	manufactures → Systems	and binds them to one or more	Associated Verifiable Artifacts
Enterprise Customer	performs and accepts	Verification Testing	
	maintains and verifies	Enterprise IT Management Systems	
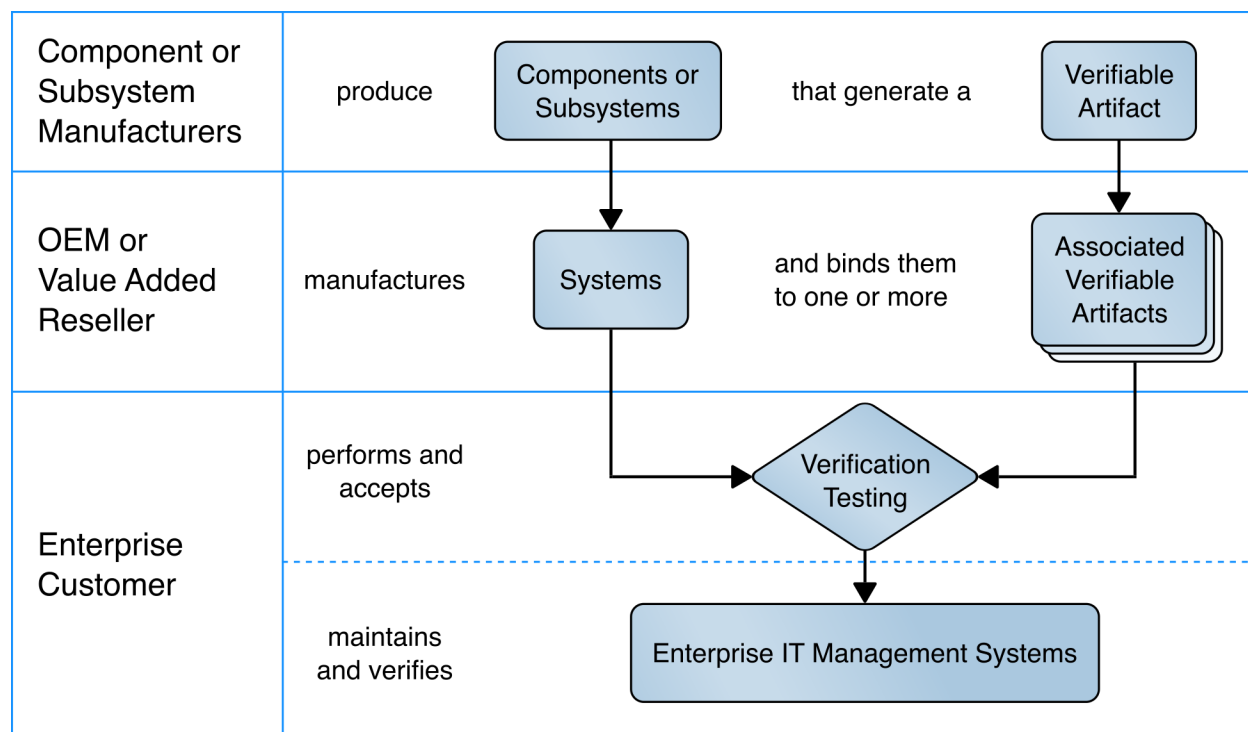
4.1 Architecture Description

577

578 The prototype architecture consists of two focus areas: 1) an implementation of a manufacturer that
579 creates a hardware-root-of-trust-backed verifiable artifact associated with a computing device, and 2)
580 the representational architecture of an organization where end users are issued computing devices that
581 require access to enterprise services for initial acceptance testing of the device and operational
582 validation of the platform.

583   This prototype implementation combines on-premises software, cloud platforms, and end user
584   hardware to demonstrate the security characteristics defined in the project description (Table 3-3).
585   Figure 4-2 presents a component-level view of the current prototype. The remaining sections discuss the
586   existing IT components an organization may have deployed before the prototype has been implemented
587   and how they can be augmented to support a hardware integrity validation capability. They also discuss
588   additional services and platforms that are integrated into the enterprise architecture.

589   **Figure 4-2 Component-Level Architecture**
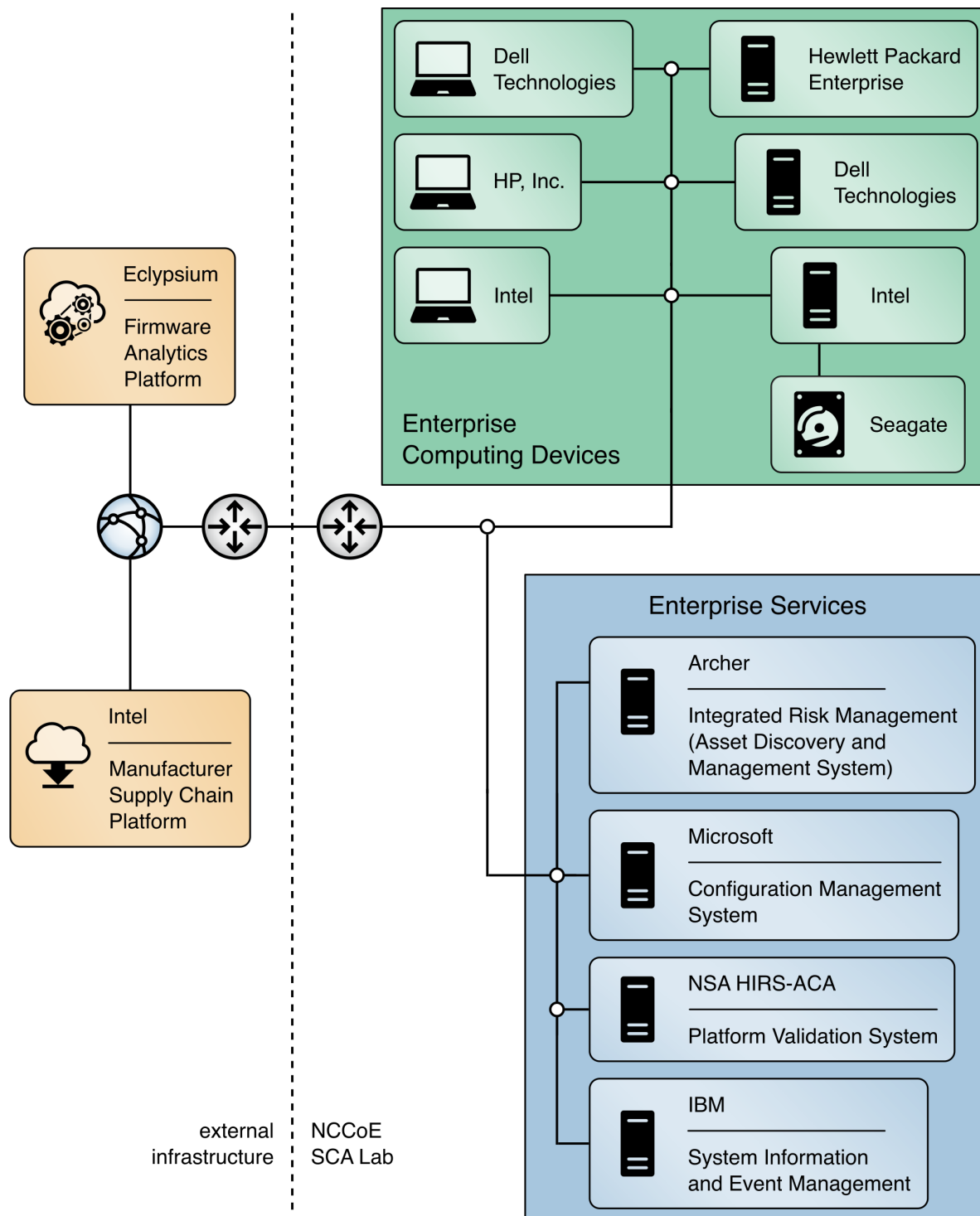
4.2 Existing Enterprise IT Management Systems

This prototype solution aims to augment, not replace, the capabilities of existing acceptance testing tools, asset management systems, configuration management systems, and SIEM systems. The following sections describe each existing capability a typical enterprise may have in operation before deciding to adopt the security characteristics defined in Section 3.5. Each section also describes the specific product that we used to demonstrate each security characteristic.

4.2.1 SIEM Tools

SIEM tools provide real-time analysis of alerts and notifications generated by organizational information systems. They support the Cybersecurity Framework's Detect function to enable the timely discovery of cybersecurity events. A typical use case of SIEM is to consolidate security-related information from organizational client endpoints, where they can be correlated to identify significant events. This demonstration extends this use case to include platform integrity security events collected from agents installed laptops during operational use.

SIEM tools commonly have a dashboard capability as well, which organizations use to present security event data in a human-friendly, unified view, sometimes referred to as "single pane of glass." In this demonstration, we use dashboards to gain better visibility into potential supply chain attacks.

4.2.1.1 IBM QRadar

We demonstrate the capabilities described above with IBM QRadar—a SIEM platform which supports the collection of security events and automated processing of events by way of rules that align with an organization's risk posture. We leverage two of its core capabilities, the log manager and the SIEM. The log manager is the component that collects, analyzes, stores, and reports on security event logs from Dell and HP Inc. laptop endpoints. The SIEM consolidates data gathered by the log manager and executes our custom ruleset which detects potential platform integrity events. This results in identifying *offenses,* events that security operations personnel may need to take remediation action on, which can be consumed by other enterprise systems (such as Dashboards) via the QRadar Representational State Transfer (REST) application programming interface (API).

4.2.2 Asset Discovery and Management System

SP 800-128 [9] states that a *system component* is a discrete identifiable IT asset that represents a building block of a system. An accurate component inventory is essential to record the components that compose the system. The component inventory helps to improve the security of the system by providing a comprehensive view of the components that need to be managed and secured. The organization can determine the granularity of the components, and in the context of this prototype, the *system* is the computing device platform, and the *components* represent the internal hardware such as motherboard, hard drive, and memory.

624 For enabling such an inventory capability, in our project description [1] we described an Asset Discovery
625 and Management System as part of an enterprise architecture which helps organizations ensure that
626 critical assets (systems) are uniquely identified using known identifiers and device attributes. This
627 capability could include discovery tools that identify endpoints and interrogate the platform for device
628 attributes. However, this prototype demonstration uses alternative platforms for these functions that
629 are described in Section 4.2.4.

4.2.2.1 Archer Integrated Risk Management (IRM)Platform

631 To demonstrate this capability, we used the Archer IRM Platform which supports organizational
632 management of governance, risk, and compliance programs. The IRM Platform serves as the foundation
633 for the Archer asset management and Cyber Incident and Breach Response solutions and allows an
634 organization to adapt it to C-SCRM requirements and integrate it with other external data sources. This
635 prototype demonstration incorporates and extends Archer use cases centered on asset management
636 and security operations.

637 Archer is a web-based platform that can be deployed on-premises or via a SaaS model that operates on
638 a Microsoft stack consisting of Windows Server, Internet Information Services, and SQL Server. This
639 prototype demonstration leverages the Archer Data Feed Manager capability that allows consumption
640 of external data via delimited text files, Extensible Markup Language (XML) or JavaScript Object Notation
641 (JSON) data on network locations, File Transfer Protocol (FTP), or Hypertext Transfer Protocol (HTTP) or
642 HTTP Secure (HTTPS) sites. We exercise HTTP(S) data feeds via XML and JSON payloads to import
643 enterprise asset data and platform integrity data, respectively.

644 Additionally, the Archer Platform has a number of built-in applications (repositories) which assist
645 organizations with risk management by way of business processes and workflows. In this prototype
646 demonstration, we extend the Devices application to serve as the central repository for knowledge for
647 platform attributes and other manufacturing information about computing devices within an
648 organization.

649 The default Devices application enables an organization to manage physical IT assets, such as computing
650 devices, to ensure that they are protected, and vulnerabilities are addressed when detected. However,
651 the default Devices application tracked computing device platforms but did not provide the granularity
652 needed to store and track components associated with the computing device. The ability to monitor
653 component changes within the operational use of the computing device is a core capability to ensure
654 computing devices within the organization have not been tampered with or otherwise modified.
655 Therefore, this demonstration extends the Devices application through configuration to fit our use case
656 by creating an additional Archer application named Components that stores component information
657 that is cross-referenced with each computing device.

658 We modeled the structure of the Components application and made configurations to the Devices
659 application via data fields to mimic the structure of the TCG Platform Certificate Profile as a vendor-

660 agnostic method of storing data such as manufacturer, model, and version information. For
661 organizations using the broader Archer IRM platform capabilities, such as their Enterprise and
662 Operational Risk Management or Third-Party Risk Management solutions, records (computing devices)
663 stored in the Devices application can also be associated with other aspects of the enterprise
664 infrastructure [10].

665 Finally, we leveraged Archer's Security Incidents application, part of its Cyber Incident & Breach
666 Response solution, which provides a central location for managing incidents. This demonstration
667 adapted the application to automatically create incident records when a platform security event was
668 detected by our continuous monitoring capability. The platform also allows IT administrators to
669 manually create incident records. In this demonstration we only considered the creation and assignment
670 of security incidents to IT security operations personnel; however, in an operational environment the
671 solution additionally supports escalation, root cause analysis, and the establishment and execution of
672 response procedures.

4.2.3 Configuration Management System

674 The focus of this document is on implementing the information system security aspects of configuration
675 management, and as such the term security-focused configuration management (SecCM) is used to
676 emphasize the concentration on information security. The goal of SecCM activities is to manage and
677 monitor the configurations of information systems to achieve adequate security and minimize
678 organizational risk while supporting the desired business functionality and services [9].

679 As defined in the project description [1], a configuration management system is a component that
680 enforces corporate governance and policies through actions such as applying software patches and
681 updates, removing denylisted software, and automatically updating configurations. These components
682 may also assist in management and remediation of firmware vulnerabilities.

683 SP 800-128 [9] further defines two fundamental concepts that this prototype demonstration references:
684 baseline configuration and configuration monitoring.

685 A *baseline configuration* is a set of specifications for a system, or configuration items within a system,
686 that has been formally reviewed and agreed on at a given point in time, and which can be changed only
687 through change control procedures. The baseline configuration is used as a basis for future builds,
688 releases, and/or changes. In the context of this prototype demonstration, the baseline configuration
689 represents the platform attributes (e.g., serial number, embedded components, firmware and software
690 information, platform configuration) asserted in the OEM's verifiable artifact. The baseline configuration
691 may be updated if a configuration change (e.g., adding hardware components, updating firmware) is
692 approved by an organization's change management process.

693 Configuration monitoring is the process for assessing or testing the level of compliance with the
694 established baseline configuration and mechanisms for reporting on the configuration status of items

695 placed under configuration management. This prototype demonstration uses a combination of
696 monitoring capabilities provided by the configuration management system and OEM platform validation
697 tooling to assess whether the computing device has deviated from the defined baseline configuration.

4.2.3.1 Microsoft Endpoint Configuration Manager

699 Many organizations may already use Microsoft Endpoint Configuration Manager capabilities such as
700 application management, organizational resource access, and OS deployment. This prototype
701 demonstration leverages the existing configuration management activities and extends them to include
702 compliance settings (a set of tools and resources that can help you to assess, track, and remediate the
703 configuration compliance of client devices in the enterprise) and reporting (a set of tools and resources
704 that help you use the advanced reporting capabilities of SQL Server Reporting Services from the
705 Configuration Manager console [11]). These capabilities align to the SP 800-128 best practice of using
706 automation, where possible, to enable interoperability of tools and uniformity of baseline configurations
707 across the computing device.

708 The computing device baseline configuration (defined above) was evaluated using the compliance
709 settings capability. In the Intel laptop use case, we defined a configuration item which deployed a
710 custom PowerShell script to each Intel computing device. The script executed the `TSCVerifyUtil` tool
711 that is part of the Intel Transparent Supply Chain platform to perform two tests:

- a comparison of scanned components to the OEM-generated platform manifest, and

- validation of the Platform Certificate bound to the computing device.

714 If either of the tests fail, an error code is returned to Configuration Manager, where an IT administrator
715 could take remediation action.

716 Similarly, we created a device baseline configuration for the Dell and HP Inc. laptops which evaluated
717 the success or failure of executing a Windows-based version of the HIRS ACA provisioner. When
718 executed, the provisioner scans the laptop and creates a hardware manifest which is compared against
719 the Platform Certificate stored in the HIRS ACA backend during acceptance testing. A failure in the
720 process is detected by Configuration Manager, where remediation action could be taken, such as the
721 creation of a delta Platform Certificate to indicate an authorized platform modification.

4.2.4 Enterprise Dashboards

723 Many organizations leverage informational dashboards that provide security information on a
724 continuing basis to give, as SP 800-53 Revision 5 notes, "organizational officials the ability to make
725 effective and timely risk management decisions, including ongoing authorization decisions." An
726 information management console or *dashboard* in the context of this prototype is a tool that
727 consolidates and communicates platform integrity status relevant to the organizational security posture

728   in near-real-time to security management stakeholders [9]. This demonstration uses an enterprise SIEM
729   dashboard capability to support the continuous monitoring described in Scenario 3.

730   ### 4.2.4.1 *Archer Integrated Risk Management (IRM) Platform*

731   This demonstration leverages the Archer IRM platform to create customized dashboards that alert the
732   appropriate audience of a potential platform integrity issue. Depending on the size of the organization,
733   the targeted audience could be individuals or groups who perform separate roles, such as IT Operations,
734   system administrators, incident response teams, or a SOC. When the appropriate organizational
735   member is alerted by the dashboard of an integrity issue, the Archer platform enables the following
736   actions:

737   1. Act and investigate the computing device by viewing the associated asset management data.

738   2. Review and initiate remediation and recovery capabilities.

739   Our dashboards import platform integrity data from three sources—the Eclypsium Analytic Platform,
740   Microsoft Endpoint Configuration Manager, and IBM QRadar. The monitored integrity data is also
741   correlated with individual computing devices, integrating the asset management capabilities discussed
742   in Section 4.2.2.

743   ## 4.3 Supporting Platform Integrity Validation Systems

744   This section describes supplemental services and systems that support the security characteristics
745   defined in Section 3.5. These systems integrate with existing services that an enterprise may already
746   have fielded, as described in Section 4.2

747   ### 4.3.1 Host Integrity at Runtime and Start-up Attestation Certificate Authority (HIRS
748   ACA)

749   The HIRS ACA [12] is described by the project owners, the National Security Agency, as a proof of
750   concept/prototype intended to spur interest and adoption of Trusted Computing Group standards that
751   leverage the TPM. It is intended for testing and development purposes only, such as this prototype
752   demonstration, and is not intended for production environments. The ACA's functionality supports the
753   provisioning of both the TPM 1.2 and TPM 2.0 with an Attestation Identity Credential (AIC); however, in
754   this prototype we have only exercised TPM 2.0 capabilities.

755   The HIRS ACA includes a flexible validation policy configuration capability, and in this demonstration's
756   defined scenarios, is configured to enforce the Validation of Endorsement and Platform Credentials to
757   illustrate a supply chain validation capability.

758   The HIRS ACA project is comprised of multiple components and services that are utilized in this
759   prototype demonstration. The first component, named the TPM Provisioner, is a software utility

760   executed on the target computing device. It takes control of the TPM if it is not already owned and
761   requests an AIC for the TPM from the Attestation Certificate Authority (ACA, described below). The
762   Provisioner communicates with the ACA through a REST API interface to complete the transaction. As
763   part of the transaction, the TPM Provisioner reads the Endorsement Key credentials from the TPM's
764   non-volatile random-access memory (NVRAM) and interrogates the computing device's hardware,
765   network, firmware, and OS info for platform validation. The previous version of this publication
766   documented the TPM Provisioner as applied to acceptance testing of the computing devices. In this
767   revision, we demonstrate the use of a pre-release version of a Windows-based version of the TPM
768   Provisioner for continuous monitoring-based scenarios.

769   The ACA is the server component that issues AICs to validated devices holding a TPM. It performs TCG-
770   based supply chain validation of connecting clients by validating endorsement and Platform Credentials.
771   The ACA is in alignment with the TCG EK Credential Profile For TPM Family 2.0 specification to ensure
772   the endorsement key used by the TPM was placed there by the manufacturer. It also aligns with TCG
773   Platform Attribute Credential Profile Specification Version 1.1 Revision 15 [13] while processing platform
774   credentials to verify the provenance of the system's hardware components, such as the motherboard
775   and chassis, by comparing measured component information against the manufacturers, models, and
776   serial numbers listed in the Platform Credential.

777   Finally, the ACA Dashboard is the Endorsement and Platform Credential policy configuration front end,
778   enabling the IT administrator to view all validation reports, credentials, and trust chains. IT
779   administrators also use this interface to upload, and if necessary, remove certificate trust chains and
780   endorsement and platform credentials.

781   Figure 4-3 presents a high-level view of how the HIRS system integrates with our prototype
782   demonstration.

783   **Figure 4-3 HIRS ACA Platform**
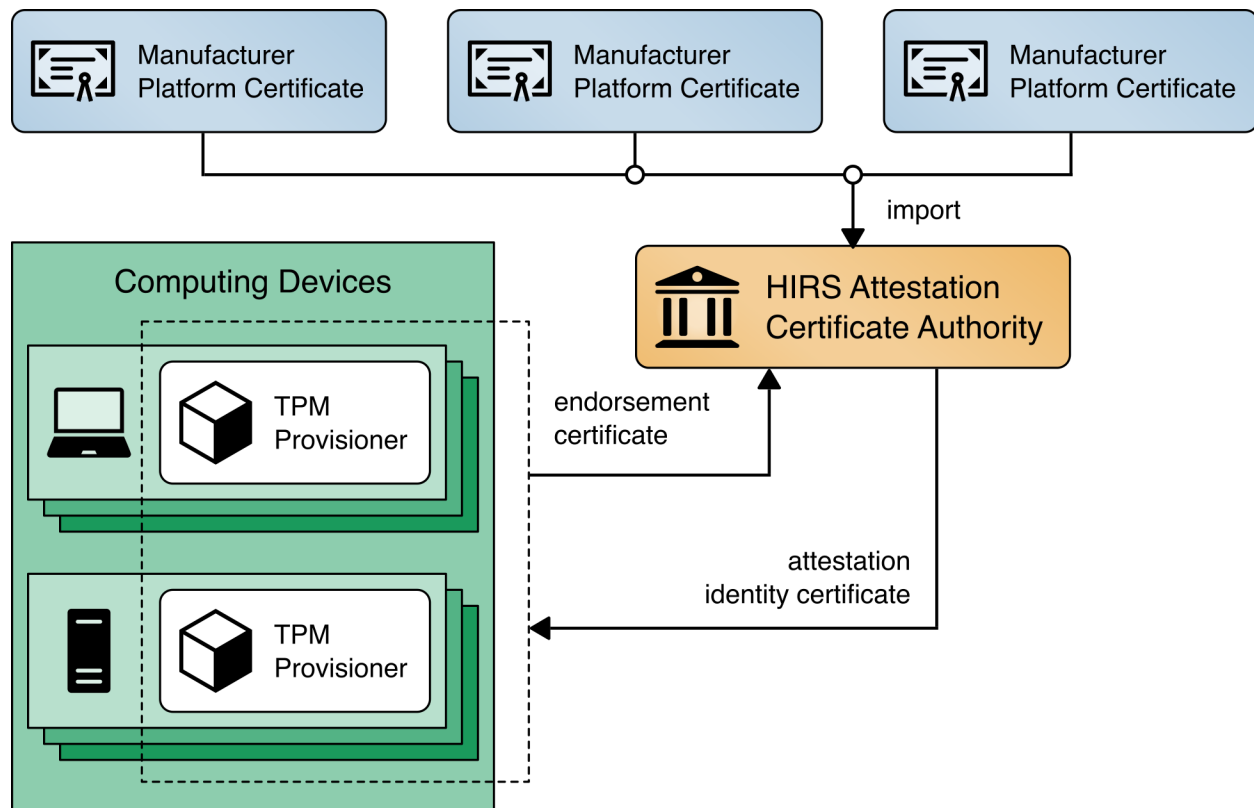
784   ## 4.3.2 Network Boot Services

785   The computing devices in this prototype demonstration support a Dynamic Host Client Protocol (DHCP)
786   based Preboot Execution Environment (PXE), which enables an IT administrator to boot the device over
787   the network. In our environment, the IT administrator can boot into either a customized CentOS7 or a
788   WinPE OS, depending on the platform validation tools that are needed. The CentOS7 environment
789   supports the TPM Provisioner component of the HIRS ACA Platform, the Eclypsium Portable Scanner,
790   and automation scripts. Figure 4-4 details the flow of the boot environment:

791   1. Computing devices are configured to boot over the network via a network interface card (NIC).
792   The DHCP server presents the boot options to the IT administrator. Once the OS is chosen, the
793   DHCP server directs the DHCP client to the Trivial File Transfer Protocol (TFTP) server.

794   2. The DHCP client downloads and executes boot loaders and kernels associated with the target
795   OS.

796   3. The IT administrator downloads the latest provisioning script from a centralized repository.

797 **Figure 4-4 Network Boot Services Environment**
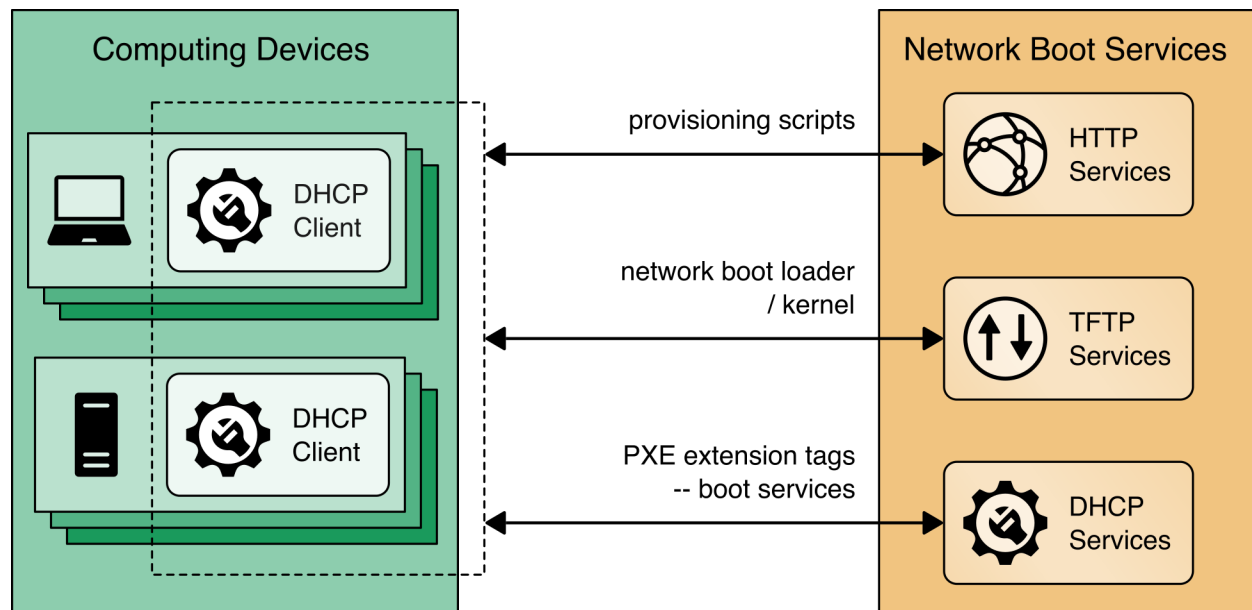
798 ## 4.3.3 Platform Manifest Correlation System

799 This system assists in providing computing device manifest attributes to the asset management system.
800 The system was built specifically for this demonstration and was built on open-source projects to include
801 the node.js server platform. The requirements of this system were defined as:

802   1. Provide a web interface for the IT administrator to upload platform manifests.

803   2. Provide a REST API for scripts to upload platform manifests.

804   3. Provide a REST API for the asset management system to periodically poll for new computing de-
805   vices to import in the repository.

806 Once the platform manifest is uploaded, it is converted to a common XML format that has been defined
807 within the Archer platform console via eXtensible Stylesheet Language Translation (XSLT). XSLTs have
808 been defined that support manifests from the HIRS ACA Provisioner, Intel's TSC applications, HPE's PCVT
809 tool, Dell's SCV tool, and HP Inc. custom scripts.

810 Figure 4-5 presents how it is integrated into the larger architecture.

811   **Figure 4-5 Platform Manifest Correlation System**
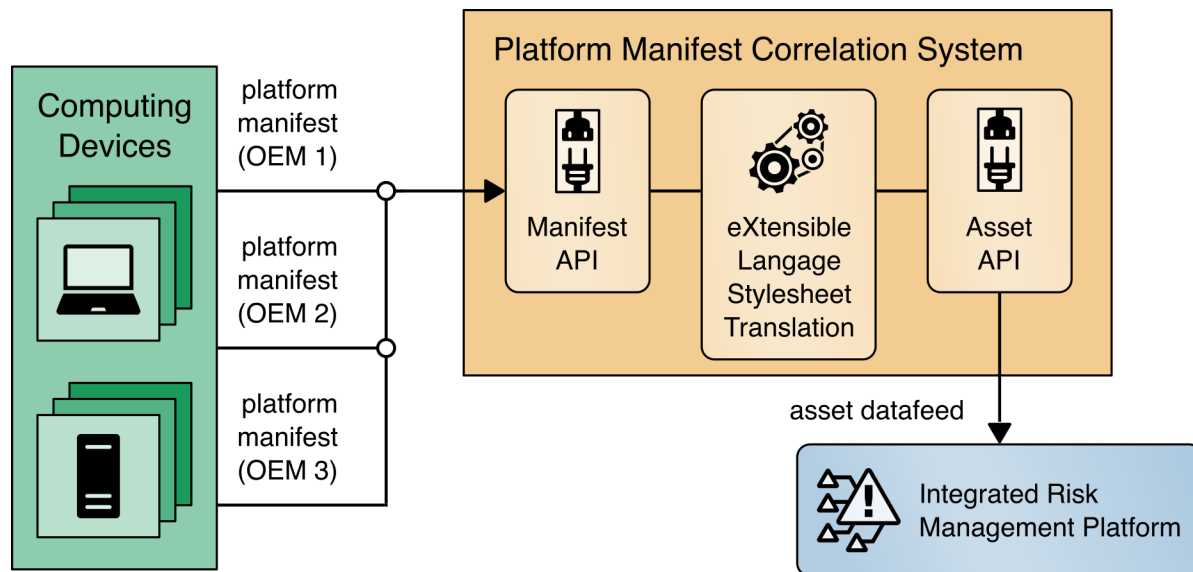
812 ## 4.3.4 Eclypsium Analytic Platform

813   The Eclypsium Analytic Platform is a security solution that focuses on vulnerabilities and threats below
814   the OS layer, to include firmware and component hardware. The platform consists of an endpoint agent,
815   which can be deployed from an enterprise systems configuration manager on each computing device,
816   the analysis backend (either cloud or on-premises), and the device reputation cloud service. The
817   platform continuously updates a profile for each device and collects telemetry about each computing
818   device into the analysis backend. The device reputation cloud provides a database of collected
819   vulnerabilities that could potentially affect computing device components within an organization.

820   The initial endpoint agent scan of the computing device forms a baseline profile, which is used for later
821   comparisons against the original profile stored in the Analysis Backend. Any deviations from the profile
822   are detected and can be communicated to an organization's IT Security department as an integrity issue
823   in multiple ways according to organization policy. For example, the IT Security department can be
824   alerted when the system firmware version has changed from the baseline, which could indicate an
825   unexpected firmware swap or tampering with the computing device in the operational environment.
826   This prototype demonstration leverages a combination of Eclypsium's REST API (Scenario 3—operational
827   monitoring) and web-based dashboard captured in Figure 4-6 (Scenario 2 —provisioning of the
828   computing device).

829   **Figure 4-6 Eclypsium Management Console**
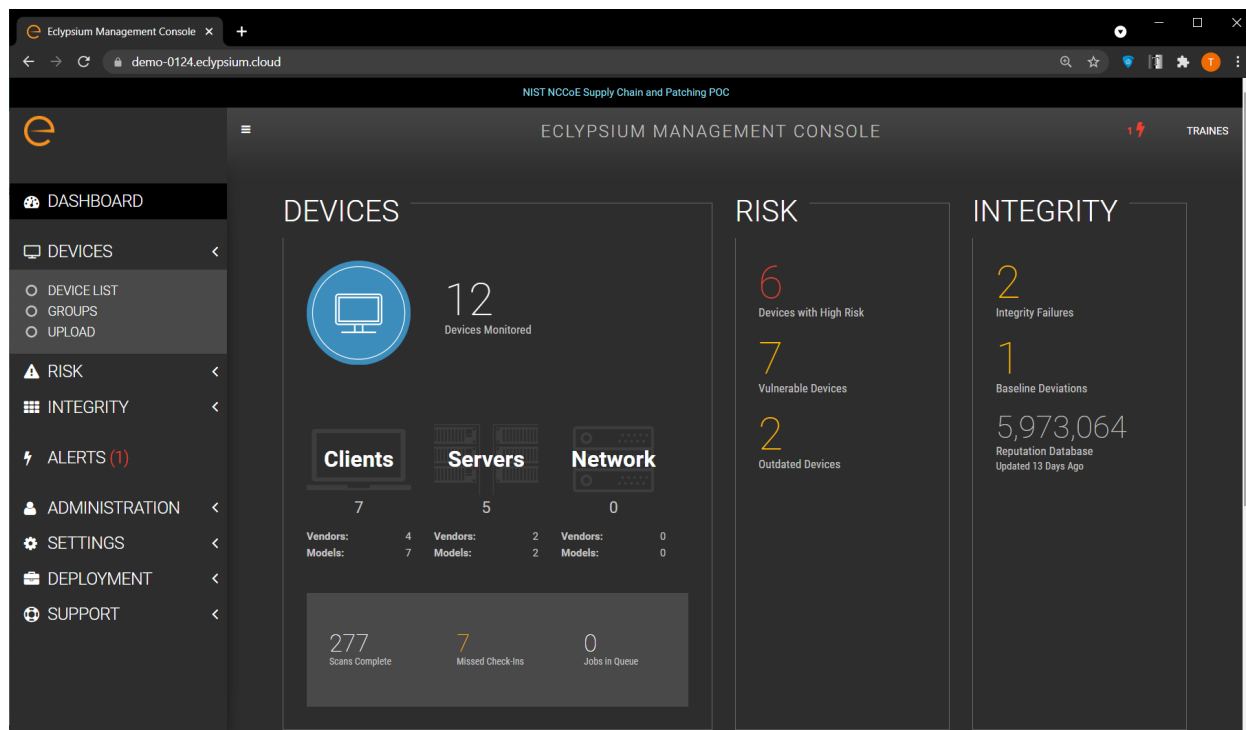
830   In Scenario 2, this demonstration uses a portable version of the Eclypsium agent, as opposed to the
831   installer-based version used in Scenario 3. This is to support an ephemeral environment for the IT
832   administrator where computing device acceptance testing is performed. We have integrated this
833   portable version of the agent into the CentOS7 discussed in Section 4.3.2.

834   The Eclypsium Analytic Platform also supports a disconnected deployment, where the computing
835   devices that are continuously monitored by the Eclypsium agent communicate directly with an on-
836   premises analytics backend. This type of deployment is useful for environments where a computing
837   device, such as a datacenter server, has restricted network access due to an organization's security
838   posture. We demonstrate this use case using the servers contributed to the project (Sections 4.4.3 and
839   4.4.4), and it is represented in Figure 4-7.

840 **Figure 4-7 Eclypsium Analytic Platform Server Implementation**
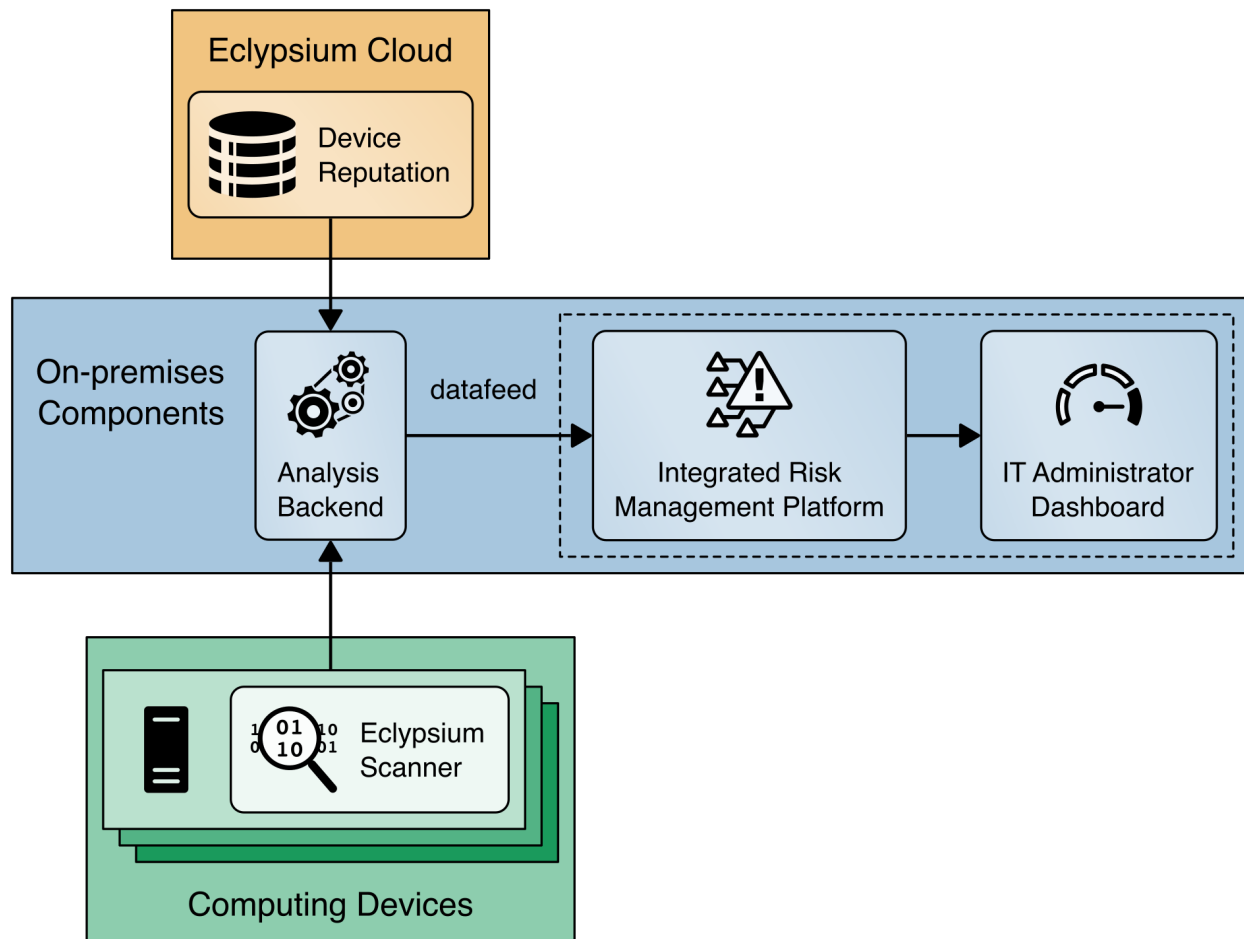
841 Figure 4-8 presents how this project integrates Eclypsium's cloud services into the demonstration
842 architecture for laptops.

843   **Figure 4-8 Eclypsium Analytic Platform Laptop Implementation**
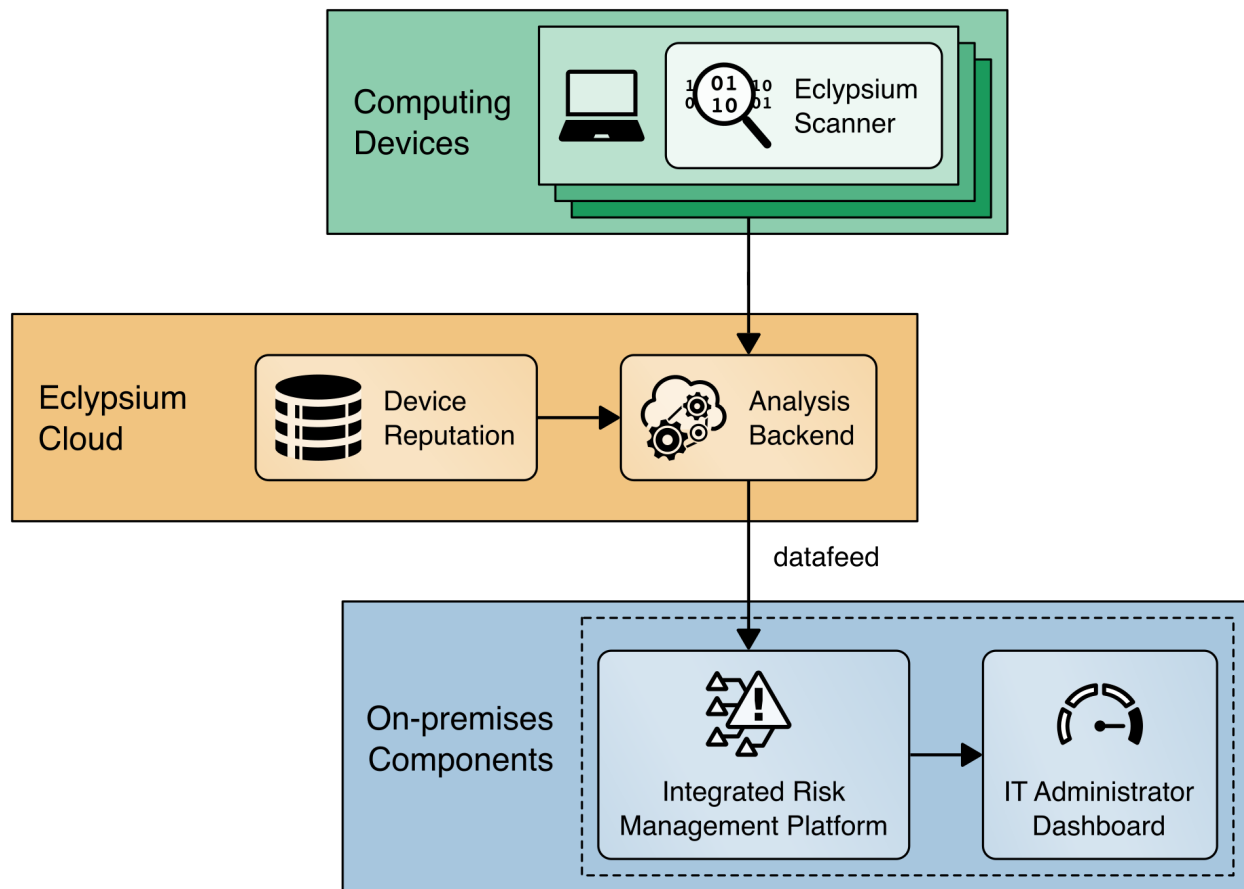
4.4   Computing Devices
844

845   In this prototype demonstration we define a computing device as client and server devices associated
846   with verifiable artifacts. These devices may contain several integrated platform components or
847   subsystems from multiple manufacturers. Our manufacturing partners, HP Inc., Dell Technologies,
848   Hewlett Packard Enterprise, Seagate, and Intel have contributed hardware to the project.

4.4.1   HP Inc.
849

850   HP Inc. functions as an OEM within this prototype demonstration and contributed two HP Inc. Elitebook
851   360 830 G5 laptops. Each laptop has a TCG-Certified TPM v2.0 with embedded Endorsement Key (EK)
852   Certificate.

853   In the preliminary draft of this publication, in support of Scenario 1 the NCCoE lab utilized the HIRS
854   Platform Attribute Certificate Creator (PACCOR) project to generate a representative Platform

855 Certificate bound to the device identity. The Platform Certificate was signed by HP Inc.'s internal test CA.
856 Since that publication, the NCCoE has worked with the HP Inc. technical team to have a demonstration
857 laptop with a Platform Certificate embedded on the device, resulting in a process that aligns with the
858 desired outcome of Scenario 1—a manufacturer-created verifiable artifact.

859 In support of Scenario 2, acceptance testing of the HP Inc. laptops is performed via the HIRS ACA TPM
860 Provisioner described in Section 4.3.1.

861 In support of Scenario 3, the demonstration is utilizing Microsoft Endpoint Configuration Manager
862 integrated with the HP Client Management Script Library (CMSL) PowerShell scripting library for
863 enterprise manageability of platform hardware and firmware security capabilities (e.g., firmware
864 integrity breach detection and physical tampering detection). As described in Section 4.2.1, this
865 demonstration makes use of HP Inc.'s CMSL PowerShell modules. Specifically, the BIOS and Device
866 module provides basic querying of device attributes and secure manipulation of HP Basic Input/Output
867 System (BIOS) settings and managing the HP BIOS, while the Firmware module provides functionality for
868 interfacing with the HP BIOS firmware, such as gathering security-related events from the HP Endpoint
869 Security Controller hardware.

870 Finally, this demonstration utilizes HP Inc. capabilities that augment tooling used to verify the integrity
871 of computing device components during use. These capabilities are intended to be provisioned during
872 the computing device acceptance testing process before issuance to the end user for operational use
873 and can optionally be provisioned in manufacturing and included in the device acceptance testing
874 process.

875 ▪ **HP Sure Admin** enforces a certificate-based authorization model that enables firmware setting
876   security management by an IT administrator. The model is composed of two keys, an
877   Endorsement Key and a Signing Key (note: the Endorsement Key in this context is not related to
878   the TPM Endorsement Key). The Endorsement Key's primary purpose is to protect against
879   unauthorized changes to the Signing Key. The Signing Key is used by the platform to authorize
880   commands sent to the firmware (BIOS) [14] [15].

881 ▪ **HP Sure Start** is a built-in hardware security system that protects platform firmware code and
882   data (including HP BIOS, HP Endpoint Security Controller firmware, and Intel Management
883   Engine firmware) from accidental or malicious corruption by (1) detecting corruption and then
884   (2) automatically restoring the firmware to its last installed HP-certified version and the data
885   (settings) to the last authorized state. The capability also stores events related to firmware
886   integrity that can provide visibility into attempted firmware integrity breaches [16].

887 ▪ **HP Sure Recover** is an OS recovery mechanism that is completely self-contained within the
888   hardware and firmware to allow secure OS recovery from the network or from a local OS
889   recovery copy stored in dedicated flash on the system board. It includes settings that control
890   when, how, and from where BIOS installs the OS recovery image, and which public keys are used
891   by BIOS to validate the integrity of the recovery image. It can also record events due to OS
892   recovery image integrity failures [16].

DRAFT

- **HP TamperLock** provides a general protection mechanism against classes of physical attacks that involve removal of the system cover to obtain access to the system board. This is achieved by providing a cover removal sensor to detect and lock down a system that has been disassembled, along with fully manageable policy controls to configure what action to take in the event a cover removal is detected. Cover removal events and history are stored in platform hardware and can be queried via CMSL PowerShell commands [17].

- **The HP Endpoint Security Controller** is HP's hardware root of trust that enables all the features above and provides isolated/dedicated non-volatile storage on the system board that (1) enables recovery of firmware code and data, policies, and OS images, as well as (2) provides secure hardware-based storage for tampering-related events associated with each of the capabilities described above.

4.4.2 Dell Technologies

Dell contributed hardware and supporting software as part of a pilot program that are aligned with the defined security characteristics of this prototype demonstration.

4.4.2.1 Laptops

The demonstration uses four Dell Latitude laptops as the client computing devices that are evaluated through an enterprise acceptance testing process. These computing devices are equipped with a TPM that is compatible with the TCG's 2.0 specification as discussed in Section 3.6.1. In alignment with the TCG specifications, the TPM endorsement keys were generated by Nuvoton, a supplier of TPMs to OEMs.

In support of Scenario 1, Dell supplied the NCCoE with the infrastructure and tooling to support TCG Platform Certificate generation during Dell computing device manufacturing. Once executed, the tooling collected the computing devices component data and created a Platform Certificate. The Platform Certificate was bound to the device identity (TPM) and digitally signed by a Dell factory Hardware Security Module. The Platform Certificate was stored within the Extensible Firmware Interface (EFI) system partition, where it was later extracted for use in supporting platform integrity validation systems.

In support of Scenario 2, the validation of component authenticity during acceptance testing of the Dell laptops was performed via the HIRS ACA TPM Provisioner described in Section 4.3.1.

Dell contributed the Dell Trusted Device (DTD) platform to the project in support of Scenario 3. Among other capabilities, DTD can detect indicators of hardware attack, which can alert a security operator that a remediation action is required. The DTD platform uses an agent which is installed on the client laptop and a cloud analysis engine hosted by Dell Technologies.

4.4.2.2 Servers

Dell also contributed an R650 PowerEdge server to the demonstration. The R650 along with the PowerEdge portfolio of servers can be shipped with the Secured Component Verification (SCV) feature, which is used to ensure that the server was delivered exactly as it was built at the factory. As part of this capability, an organization can place an order for a customized server, where it is built to their specification. After assembly the server's component data is collected and the Dell Remote Access Controller (iDRAC) is leveraged to create cryptographic keys which are protected by the iDRAC Hardware Root of Trust, to create the x509 Certificate that is then signed by the Dell Manufacturing Certificate Authority. The x509 Certificate (SCV Certificate) that is stored in iDRAC is validated prior to shipment from factory.

SCV provides a strong cryptographic platform identity that is not only bound to the platform's unique hardware but also to Dell's possession of that hardware during assembly due to the creation process requiring the unique hardware to cryptographically sign the Certificate Signing Request (CSR). At the core of the SCV platform is the SCV command-line verification application, which performs the following functions without internet or intranet connectivity:

1. Downloads SCV Certificate that is stored in the iDRAC via SCV Validation Tool.

   a. Validates the SCV Certificate signature is valid and has not been tampered with

   b. Verifies the SCV Certificate Chain of Trust to ensure it chains back to the Dell SCV Root Certificate Authority

   c. Cryptographically challenges iDRAC for possession of the platform-unique SCV private key to ensure the platform matches the SCV Certificate

2. Any error in SCV Certificate signature verification, chain of trust verification, or proof of possession will result in a Fail output before component data is compared or trusted.

3. Interrogates the system to obtain the current inventory and iDRAC Hardware ID Certificate, and collects the TPM Endorsement Key Certificate Serial Number.

   a. Compares current system inventory against the manifest in the Platform Certificate, including the cryptographic identities for the iDRAC Hardware ID Certificate and the TPM Endorsement Key Serial Number

4. Any swapping or removal of the components that are captured in the certificate will be identified as a Mismatch in the SCV application output. An additional detailed log is created describing all the components which were expected (present in factory) versus what has been detected (currently present in platform).

958 The Trusted Platform Module (TPM) Endorsement Key (EK) and iDRAC Hardware ID Certificate as
959 represented in the signed SCV Certificate can then be used as permanent cryptographic identities for the
960 life of the PowerEdge platform in addition to the SCV Certificate.

961 ### 4.4.3 Intel

962 Intel contributed hardware, supporting software, and cloud services that are aligned with the defined
963 security characteristics of this prototype demonstration through its Transparent Supply Chain (TSC)
964 platform [18]. TSC enables organizations to verify the authenticity and firmware version of systems and
965 their components. The remainder of this section summarizes the TSC components used within this
966 prototype demonstration; however, it is not an exhaustive description of the complete platform. Refer
967 to Intel's TSC website for complete documentation.

968 The TSC process starts at the OEM, where an Intel-provided tool called `TSCMFGUtil` enables the creation
969 of a Platform Certificate data file that is compliant with the TCG Platform Certificate Profile Specification
970 Version 1.1. The `TSCMFGUtil` also generates the Direct Platform Data (DPD) file capturing the Platform
971 Snapshot before shipping the platform out to the customer. The Platform Certificate data file contains
972 TPM information such as the Platform Configuration Registers (PCRs), the TPM Serial Number, and the
973 TPM Endorsement Key. The DPD file contains information about the components within the computing
974 device such as component manufacturer part number, batch number, and serial and lot number, as well
975 as sourcing information. The OEM then uploads these files to Intel's Secure File Transport Protocol
976 (SFTP) site where they are processed and digitally signed.

977 Next, after the computing device is purchased by an organization's IT department, an administrator
978 downloads the DPD file and Platform Certificate from the Transparent Supply Chain Web Portal as part
979 of the computing device acceptance testing process. The aforementioned files are processed by Intel
980 software intended for the end customer, the `AutoVerifyTool`. In this prototype demonstration, we use
981 the `AutoVerifyTool` with our demonstration laptops to enable the following capabilities for the IT
982 administrator:

983   1. The `ScanSystem` function initiates the scanning of the system components and the TPM infor-
984   mation. The scanning operation will perform the following operations:

985   a. Read the following platform components: BIOS, system, motherboard, chassis, proces-
986   sor, dual in-line memory modules (DIMMs), batteries, Intel Active Management Tech-
987   nology firmware version, power supplies

988   b. Read the TPM PCRs, public Endorsement Key, and the Endorsement Key serial number

989   c. Read the internal drive information

990   d. Read the Windows Management Instrumentation (WMI) Information for internal key-
991   board, pointer, and network devices

2. After the system has been scanned, the IT administrator executes the `Read Direct Platform Data File` function which opens and displays the DPD associated with the platform.

3. The IT administrator executes the `Compare` function, which compares the current system component value information that was captured by `ScanSystem` operation to the component value information that was read in from the DPD file.

4. The IT administrator executes the `Platform Certificate Verify` function, which validates the Platform Certificate issued for the platform using the TPM as the hardware root of trust. The `Platform Certificate Verify` will check that the TPM Endorsement Key serial number matches the Endorsement Key serial number in the Platform Certificate. The function will also check that the manufacturer, version, and serial number match the values in the Platform Certificate.

In addition to the AutoVerifyTool, Intel provided a similar utility named TSCVerifyUtil that has the same capabilities but is intended to be executed from the command line on Windows and Linux systems. The TSCVerifyUtil is well-suited for automated scripts that run continuously without administrator intervention. We have used TSCVerifyUtil to demonstrate acceptance testing on server platforms and continuous monitoring for laptops.

To demonstrate the TSC platform, Intel contributed laptop computing devices from OEMs Lenovo and HP Inc. (T490 Thinkpad and HP EliteBook x360 830 G5, respectively) and a server based on an Intel S2600WT family server board. Intel also provisioned accounts for the NCCoE project team to use the TSC Web Portal for demonstrating computing device acceptance testing described in Scenario 2.

4.4.4 Hewlett Packard Enterprise (HPE)

HPE contributed hardware and supporting software that are aligned with the defined security characteristics of this prototype demonstration through its HPE Trusted Supply Chain program. The HPE demonstration server's platform integrity is validated using the HPE-developed open-source Platform Certificate Verification Tool (PCVT) [19], leveraging a hardware root of trust (TPM) via TCG Platform Certificate specifications. Our demonstration used an HPE Proliant DL360; however, an implementer of this guide should consult the HPE website for the current roster of servers that support the capabilities described below.

In our demonstration server, the HPE Platform Certificate was provisioned during the manufacturing process in secure storage, digitally signed by an HPE demonstration CA. This enables an offline or "air-gapped" use case for server platform integrity verification. In addition to Platform Certificates, the HPE demonstration implements system Device Identity (IDevID) certificates as a TCG-defined method for platform identity cryptographic attestation via the TPM.

The PCVT enables an organization to ensure that the shipped server configuration matches the configuration from the factory using the following tests:

1027   1.   Ensures the validity of the trust chain and signature of the factory installed initial DevID signing
1028   key and initial Attestation Key (IAK) created by HPE. The initial DevID is a unique, permanent
1029   cryptographically protected identifier for the HPE server. The IDevID certificate is TCG and IEEE
1030   802.1 AR compliant. The IAK is a restricted signing key that is used when performing remote at-
1031   testation of the HPE server using its TPM.

1032   2.   Performs TCG certificate trust chain verification, verifying the chain from the signed certificate
1033   to the HPE Root CA certificate. This step verifies the certificate signature against the intermedi-
1034   ate certificate that signed the Platform Certificate, system IDevID certificate, and associated sys-
1035   tem IAK Certificate.

1036   3.   Verifies the demonstration server's hardware manifest against the Platform Certificate that HPE
1037   issued at its manufacturing facility.

1038 The PCVT is available via the HPE GitHub repository as a bootable optical disc image (ISO) that an
1039 administrator can run via HPE server management tools, which is documented in PCVT's User Guide.
1040 However, in our demonstration we created a customized acceptance testing environment based on
1041 CentOS 8. This environment incorporated a compiled version of the PCVT with additional scripts that
1042 provision the server into the enterprise asset management and discovery system upon successful
1043 execution of the PCVT.

1044 ## 4.4.5   Seagate

1045 Seagate contributed three Exos 18 Terabyte Hard Drives delivered in a 2U12 enclosure. We
1046 demonstrated how an organization could verify the drives are genuine Seagate products through two
1047 capabilities—Secure Device Authentication and Firmware Attestation. Both capabilities are facilitated
1048 via the TCG Storage API (GitHub repository), which we utilized in an integration with Intel TSC platform
1049 integrity tools. Secure Device Authentication (SDA) and Firmware Attestation in conjunction provide a
1050 cryptographically assured method to trace the drive and firmware to the manufacturer (Seagate). Both
1051 features are certificate-driven and verifiable by way of Seagate's root certificate from its internal CA.

1052 As noted above, both capabilities are available via API, and Seagate has published a command-line utility
1053 via GitHub to demonstrate interacting with the drive. The command-line utility provides a roadmap that
1054 organizations can use to strengthen and expand platform integrity verification use cases. To illustrate a
1055 use case in this demonstration, we connected the Seagate enclosure to our Intel-contributed server. An
1056 enterprise may use a server-connected drive enclosure to increase the storage capacity of critical
1057 applications hosted in a datacenter. This organization prioritizes the integrity of the data, and by
1058 extension the integrity of the drive itself. Therefore, the validation of the server platform integrity—to
1059 include measurements from the attached drives—mitigates the risk of an integrity-related breach to an
1060 acceptable level.

1061   With the scenario described above in mind, Seagate, in collaboration with Intel developers, integrated
1062   Transparent Supply Chain validation utilities with the Seagate drive APIs. As a result, this integration
1063   enables an implementing organization to simultaneously derive the benefits of TSC tooling described in
1064   Section 4.4.3 and verify drive integrity measurements with one command. The process of Secure Device
1065   Authentication (SDA) and Firmware Attestation is illustrated below.

1066   **Figure 4-9 Seagate Secure Drive Authentication Integration**
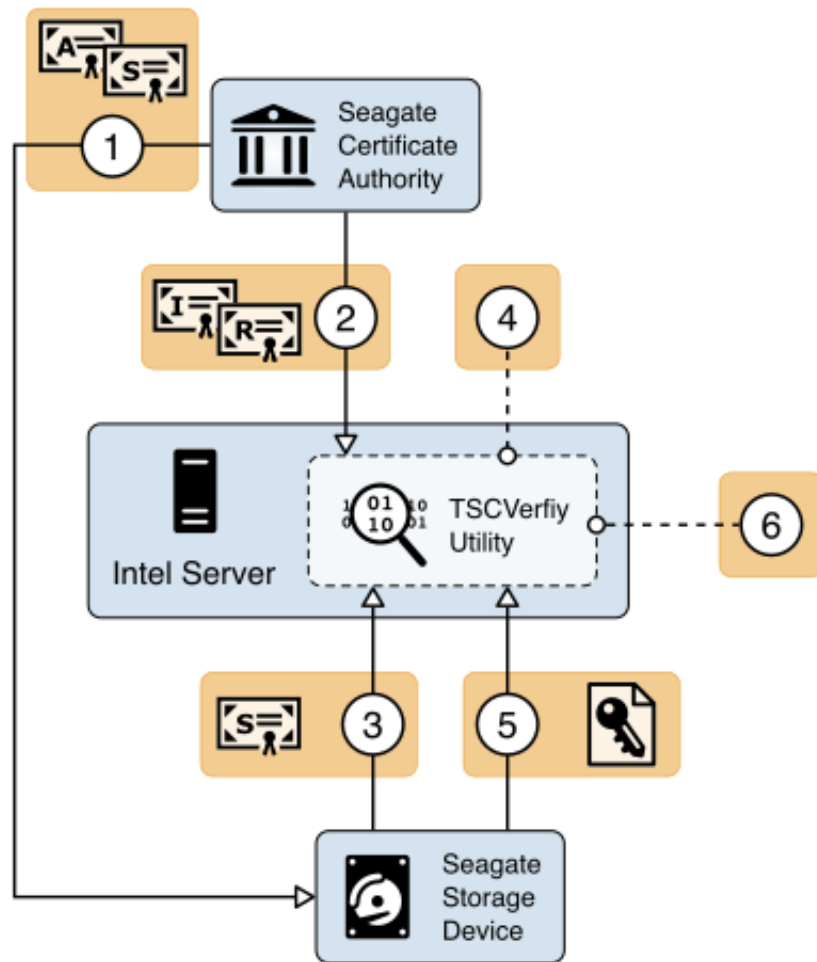
1067   1. During the manufacturing process, Seagate creates a Trusted Peripheral signing certificate (tper-
1068   Sign Certificate) and Attestation Certificate (tperAttestation Certificate) that are signed by the
1069   Seagate Intermediate CA. The tperSign Certificate and tperAttestation Certificate are stored in
1070   the drive's firmware. The drive is now capable of responding to challenges from host computing
1071   devices.

1072 2. The host, in this case the Intel server, stores the Seagate Root and Intermediate CA certificates
1073   in the TSCVerifyUtil application binary. They are used later in the validation process.

1074 3. The Security Operator executes the TSCVerifyUtil application and directs it to initiate the SDA
1075   verification. The drive's certificate is returned in the initial invocation of SDA.

1076 4. The drive's signing certificate is returned to TSCVerify where it is validated against the Seagate
1077   Root and Intermediate CA certificates. If validation succeeds, the process continues.

1078 5. TSCVerifyUtil generates a challenge (timestamp) that is transmitted to the drive. The drive re-
1079   turns a cryptographically signed response based on the challenge.

1080 6. TSCVerifyUtil verifies the digital signature on the response with the drive's public key retrieved
1081   in Step 3.

1082 Upon the successful completion of the SDA process, Seagate's Firmware Attestation capability is
1083 exercised. The Firmware Attestation process is illustrated below.

1084 **Figure 4-10 Seagate Firmware Attestation Integration**
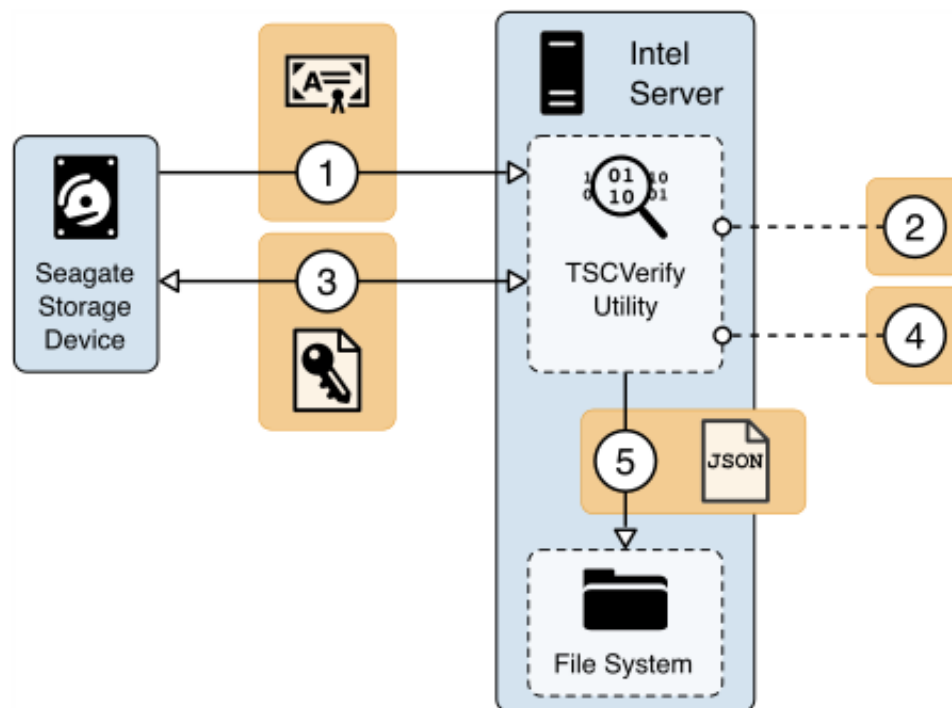
1085 1. TSCVerifyUtil requests the tperAttestation Certificate from the drive. The certificate path is vali-
1086   dated against the Seagate Intermediate and Root CAs.

1087   2.   TSCVerifyUtil generates an Assessor Identifier and a nonce. The Assessor Identifier is a static
1088   host server identifier (such as the hostname) and the nonce is a randomly generated set of 16
1089   bytes for each invocation of the firmware attestation method. These values, in addition to the
1090   common name of the tperAttestation Certificate, are stored for the next step.

1091   3.   The values from Step 2 are transmitted to the drive via the Get Signed Firmware Message com-
1092   mand and the response is returned.

1093   4.   The digital signature on the response is verified using the drive's public key from the tperAttes-
1094   tation Certificate retrieved in step 1.

1095   5.   If Step 4 succeeds, the associated firmware hashes are exported from TSCVerifyUtil as a JSON-
1096   formatted file.

1097   The firmware attestation outputs multiple integrity measurement values, which in isolation give the
1098   verifier information about the current running version of the drive firmware. Ideally, measurements are
1099   compared against a baseline set of integrity measurements for the drive which are known by the verifier
1100   before the attestation is produced. For the purposes of this demonstration, the measurements
1101   produced by the firmware attestation capability were validated against values that were communicated
1102   to the project team and incorporated into the TSCVerifyUtil.

5   Security Characteristic Analysis

1104   The purpose of the security characteristic analysis is to understand the extent to which the project
1105   meets its objective of creating a prototype that demonstrates how organizations can verify that the
1106   components of their acquired computing devices are genuine and have not been tampered with or
1107   otherwise modified throughout the devices' life cycles. In addition, it seeks to understand the security
1108   benefits and drawbacks of the prototype solution.

5.1   Assumptions and Limitations

1110   The security characteristic analysis has the following limitations:

1111   ▪   It is neither a comprehensive test of all security components nor a red-team exercise.

1112   ▪   It cannot identify all weaknesses.

1113   ▪   It does not include the lab infrastructure. It is assumed that devices are hardened. Testing these
1114   devices would reveal only weaknesses in implementation that would not be relevant to those
1115   adopting this reference architecture.

1116   ▪   It will evolve and expand as the project as collaborators are integrated into the final architecture
1117   in the next publication of this document.

5.2 Build Testing

1118

1119 This section addresses how this prototype demonstration addresses each scenario and identifies gaps
1120 that will be addressed as the project progresses.

5.2.1 Scenario 1

1121

1122 The desired outcome of Scenario 1 is the creation of verifiable platform artifacts, either by the
1123 manufacturer or the customer in the field. In the case of Intel laptops, this demonstration uses a
1124 manufacturer-created platform artifacts by way of Intel's Transparent Supply Chain platform (Section
1125 4.4.3).

1126 In the preliminary draft version of this guide, we emulated a customer-created platform artifact using
1127 the HIRS ACA project's PACCOR software for Dell and HP Inc. laptops. In this revision, Dell and HP Inc.
1128 contributed laptops with pre-installed verifiable artifacts created at the factory, where they are signed
1129 by manufacturer-specific certificate authorities as opposed to NCCoE-generated authorities.
1130 Additionally, Dell has made their root certificate publicly available to those customers who participate in
1131 this pilot program.

1132 The Platform Certificates are subsequently stored in the laptop's EFI partition where they are accessible
1133 to the customer for validation, in alignment with the TCG's PC Client Platform Firmware Integrity
1134 Measurement specification which defines the Platform Certificate format, naming convention, and
1135 common directory location when stored locally on the laptop. In this demonstration, we simulate the
1136 process of an IT administrator taking delivery of the laptops by accessing and uploading the Dell and HP
1137 Inc. verifiable artifacts to the HIRS ACA validation system for use in Scenarios 2 and 3.

1138 The server contributed by Intel uses the same TSC platform as the laptops to deliver platform artifacts to
1139 the customer. HPE servers that support platform artifacts are generated at the factory (Section 4.4.4)
1140 and are available to the customer via the Integrated Lights-Out API. Dell server platform artifacts are
1141 generated at the factory through the Secure Component Validation program (Section 4.4.2).

1142 In all cases, the platform artifact is instantiated as a Platform Attribute Certificate defined in the TCG
1143 Platform Attribute Credential Profile Specification version 1.0. The profile defines structures that extend
1144 the X.509 certificate definitions to achieve interoperability between platform validation systems that
1145 ingest artifacts. Figure 5-1 shows the relationship between the Platform Certificate and the TPM
1146 Endorsement Credential, based on a graphic from the *TCG Credential Profiles for TPM* [20].

1147   **Figure 5-1 Platform Certificate Binding to Endorsement Credential**
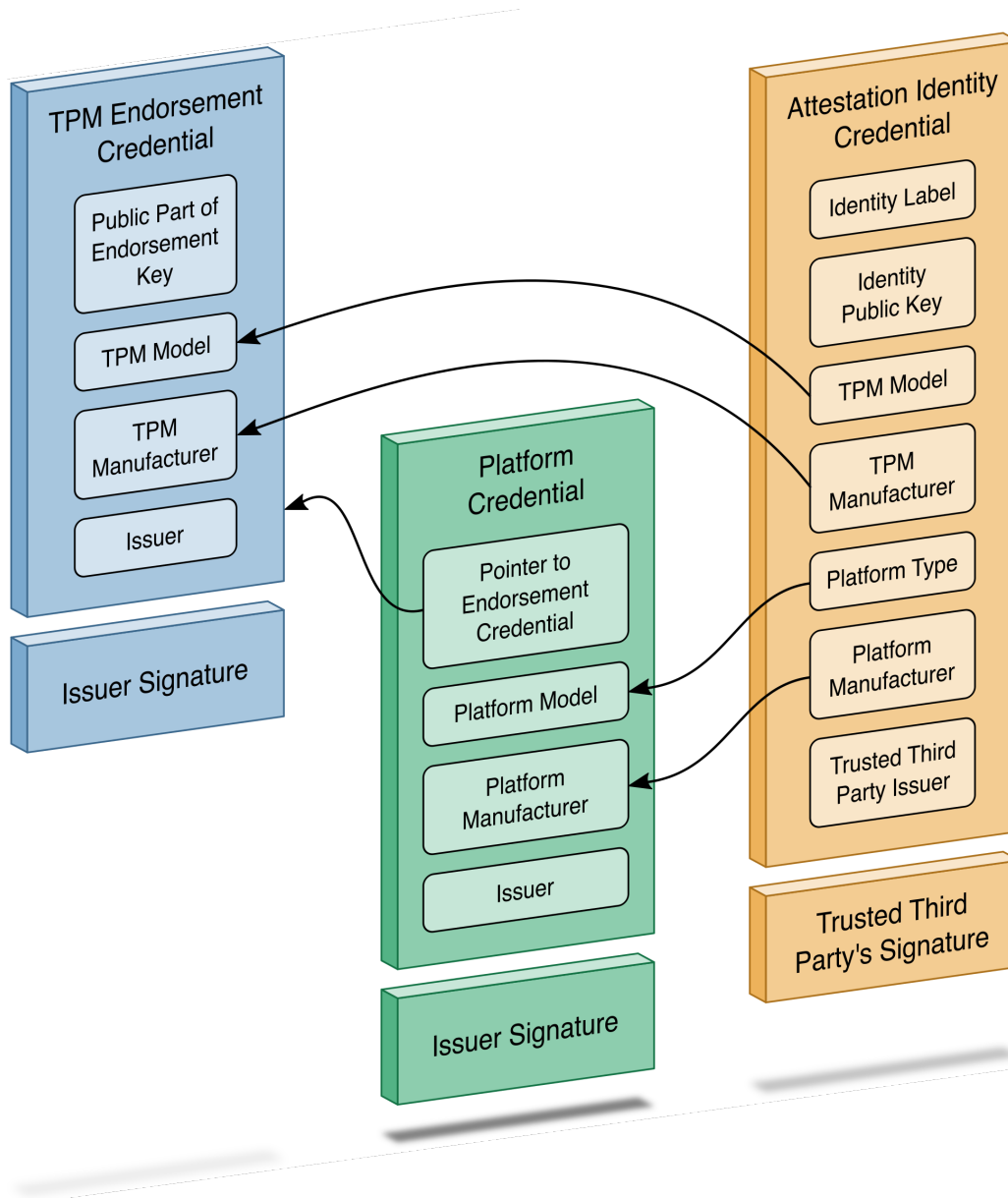
1148   Below, we use an open-source tool (openssl) to parse one of our demonstration platform artifacts to
1149   validate alignment with the TCG specification. Note that the current profile allows the manufacturer to
1150   choose between Attribute Certificate or Public Key Certificate format. The example in Table 5-1 uses the
1151   Attribute Certificate format and is not an exhaustive comparison of all requirements within the profile. It
1152   is intended to highlight the binding of authoritative attributes (Attribute Extension) to the hardware
1153   itself (Holder).

1154   **Table 5-1 Demonstration Verifiable Artifact**

Platform Certificate Assertion	Field Name	Field Description
C=US, ST=California, L=Palo Alto, O=HP Inc., OU=HP Labs Pilot, CN=HP Inc. NCCOE-Test	Issuer	Distinguished name of the Platform Certificate issuer
C=DE, O=Infineon Technologies AG, OU=OPTIGA(TM), CN=Infineon OPTIGA(TM) TPM 2.0 RSA CA 042	Holder	Identity of the associated TPM EK Certificate
2.23.133.18.3.1	Component Class Registry	Example Component Identifier
00020001	Component Class Value (Chassis)	
HP	Component Manufacturer	
10	Component Model	

1155   In addition to a Platform Certificate, a manufacturer may implement IDevID and IAK certificates as
1156   complementary capabilities. This is demonstrated by our HPE server with the PCVT described in Section
1157   4.4.4. As noted above, Platform Certificates are defined as attribute certificates without a key. IDevID
1158   certificates are defined by TCG's TPM 2.0 Keys for Device Identity and Attestation [21], and its purpose is
1159   to bind a key to a device's TPM using carefully constructed protocols that align with TCG specifications.
1160   TCG IDevID certificates provide evidence that a key belongs to a specific computing device by binding
1161   that key to the device's TPM. Further, the private key associated with the IDevID certificate is created
1162   such that it cannot be exported from the TPM. Applications, such as network onboarding, can leverage
1163   the IDevID certificate for automated provisioning.

1164   This prototype demonstrates only the validation of IDevID certificates via HPE's Platform Certificate
1165   Validation Tool. Interested readers should follow the progress NCCoE's Trusted Internet of Things (IoT)
1166   Device Network-Layer Onboarding and Lifecycle Management project and/or review the Trusted
1167   Internet of Things (IoT) Device Network-Layer Onboarding and Lifecycle Management (Draft) White
1168   Paper [22] for an in-depth discussion of device identity use cases.

1169   Finally, the Trusted Peripheral (TPer) signing certificates that are embedded in the Seagate drive
1170   firmware serve as verifiable artifacts in this demonstration. These certificates support the Secure Device

1171 Authentication and Firmware Attestation capabilities, and attributes in the certificates are used to
1172 uniquely identify the drive. Table 5-2 identifies these attributes.

1173 **Table 5-2 Seagate Drive Verifiable Artifacts**

Seagate Drive Certificate Assertion	Field Name	Field Description
CN=ZR5056HD, OU=DriveTrust, O=Seagate Technology, C=US	Subject	Distinguished name of the Seagate drive device certificate
SN=ZR5056HD	Subject Alternative Name	Alternative name of the Seagate drive device certificate
C=US, O=Seagate Technology LLC, OU=Seagate Technology TDCI, CN=Seagate Technology TPer Attestation [022300085000C500CAD93EA3]	Subject	Distinguished name of the Seagate firmware attestation certificate

5.2.2 Scenario 2

1175 The desired outcome of Scenario 2 is to verify the provenance and authenticity of a computing device
1176 that has been received through non-verifiable channels. The project description defined four notional
1177 steps that an IT administrator might perform to augment, not replace, an existing asset management
1178 acceptance testing process. The remainder of this section discusses the status of each step, with
1179 supplemental sequence diagrams available in Appendix C.

1180 ***Step 1:*** *As part of the acceptance testing process, the IT administrator uses tools to extract or obtain the*
1181 *verifiable platform artifact associated with the computing device.*

1182 Using the Intel Transparent Supply Chain platform, an IT administrator obtains the verifiable artifact for
1183 compatible laptops and servers from the download portal in two ways—manually via the web interface,
1184 and programmatically through the download portal API, depending on the organizational use case. In
1185 our lab, we demonstrated a manual process where an IT administrator uses a web browser to access the
1186 Intel download portal, input the computing device serial number, and download the associated
1187 verifiable artifacts. The download portal API may be useful for organizations that have an automated
1188 computing device acceptance testing process. The download portal screenshot in Figure 5-2 provides a
1189 visual of the interface viewed from the IT administrator's perspective.

1190   **Figure 5-2 Intel Transparent Supply Chain Download Portal**

1191   In this prototype demonstration for the Dell and HP Inc. laptop platforms, the IT administrator obtains
1192   the platform verifiable artifact from the EFI system partition storage (ESP). The ESP provides a
1193   convenient storage mechanism because it is available by all manufacturers that support Unified
1194   Extensible Firmware Interface (UEFI) and is OS-independent. Therefore, it is accessible either through
1195   our Linux network boot environment or the native OS (Windows 10). Alternatively, the verifiable artifact
1196   can be delivered to the IT administrator through an out-of-band process or stored directly on the TPM, if
1197   available on the computing device.

1198   For the Dell and HPE server platforms, the verifiable artifact is extracted using via the SCV and PCVT
1199   tools, respectively.

1200   ***Step 2:*** *The IT administrator verifies the provenance of the device's hardware components by validating*
1201   *the source and authenticity of the artifact.*

1202   ***Step 3:*** *The IT administrator validates the verifiable artifact by interrogating the device to obtain*
1203   *platform attributes that can be compared against those listed in the artifact.*

1204   For simplicity, we have combined discussion of steps 2 and 3 because they are performed in tandem
1205   using platform validation tools.

1206	In the Intel TSC platform, we execute the `AutoVerifyTool` described in Section 4.4.2 to verify the
1207	provenance of the device's hardware components in the native Windows 10 environment using the
1208	verifiable artifact retrieved from Step 1. The tool is preconfigured with trusted manufacturer signing
1209	certificates that are used in the validation process. Second, the IT administrator scans the machine using
1210	the `AutoVerifyTool`, where the results are compared against those listed in the artifact. The tool
1211	subsequently gives the IT administrator a visual indicator of whether or not the validation process was
1212	successful. The tool can be accessible to the IT administrator in a number of ways, depending on the
1213	existing acceptance testing process. For this prototype, the tool is available to the IT administrator via a
1214	network share accessible to IT staff with sufficient privileges.

1206 In the Intel TSC platform, we execute the `AutoVerifyTool` described in Section 4.4.2 to verify the
1207 provenance of the device's hardware components in the native Windows 10 environment using the
1208 verifiable artifact retrieved from Step 1. The tool is preconfigured with trusted manufacturer signing
1209 certificates that are used in the validation process. Second, the IT administrator scans the machine using
1210 the `AutoVerifyTool`, where the results are compared against those listed in the artifact. The tool
1211 subsequently gives the IT administrator a visual indicator of whether or not the validation process was
1212 successful. The tool can be accessible to the IT administrator in a number of ways, depending on the
1213 existing acceptance testing process. For this prototype, the tool is available to the IT administrator via a
1214 network share accessible to IT staff with sufficient privileges.

1215 In this prototype demonstration for the Dell and HP Inc. platforms, prior to the acceptance testing
1216 process, the IT administrator supplies the verifiable artifact's (Platform Certificate's) root (and
1217 potentially intermediate) CA certificates to the HIRS ACA portal to form a chain used later in the
1218 validation process. This process is repeated for the endorsement credential issuing certificates. We
1219 recommend that readers of this guide contact their specific manufacturer to retrieve the correct
1220 certificate chain to reduce the risk of false-negative validation failures.

1221 Next, the IT administrator boots the target computing device into the ephemeral Linux CentOS7
1222 environment described in Section 4.3.2 where the HIRS ACA Provisioner component is installed. Here,
1223 the IT administrator runs a script where the Provisioner is invoked, and the provenance of the device's
1224 hardware components is verified by the HIRS ACA backend component. The IT administrator confirms
1225 validation of the verifiable artifact by observing the output of the script and optionally accessing the
1226 HIRS ACA portal web interface, as shown in Figure 5-3. The checkmark in the Result column indicates the
1227 verifiable artifact has been validated and the assertions made by the artifact have been validated
1228 against the interrogation process.

1229 **Figure 5-3 HIRS ACA Validation Dashboard**

1230 Finally, in addition to the platform validation steps described above, this prototype demonstration
1231 interrogates and analyzes the target computing device across all participating manufacturers using the
1232 Eclypsium platform described in Section 4.3.4. This analysis gives the IT administrator immediate
1233 feedback on any firmware integrity issues, such as an unexpected or outdated firmware version, so they
1234 can be corrected before being fielded to the end user.

1235 Dell and HPE servers follow a similar process. Dell servers are network booted into a custom WinPE
1236 environment where the SCV tool and project-specific automation scripts are available. The IT
1237 administrator runs the script which executes the SCV tool described in Section 4.4.2 and collects the
1238 validation status from the SCV tool exit code. HPE servers are network booted into a custom CentOS8
1239 environment where the PCVT and project-specific automation scripts are available and collect the
1240 validation status from the PCVT exit code.

1241 ***Step 4:*** *The computing device is provisioned into the Asset Discovery and Management System and is*
1242 *associated with a unique enterprise identifier. If the administrator updates the configuration of the*
1243 *platform (e.g., adding hardware components, updating firmware), then the administrator might create*
1244 *new platform artifacts to establish a new baseline.*

1245 Following the successful platform validation of the target computing device, it is provisioned into the
1246 Asset Discovery and Management System described in Section 4.2.1. This demonstration associates the
1247 system's Universally Unique Identifier (UUID), available via the System Management BIOS (SMBIOS),
1248 with the computing device in the asset management system. The SMBIOS is a standard for delivering
1249 management information via system firmware developed by the DMTF (formerly known as the
1250 Distributed Management Task Force). The standard presentation format of the SMBIOS provides a
1251 benefit to this prototype in that it is available in an OS-independent manner, and therefore available
1252 using any of our network boot environments. We also associate the system UUID with each computing
1253 device that has been provisioned into the Eclypsium platform. This enables the Asset Discovery and
1254 Management System to correlate device data from the Eclypsium cloud to existing assets. Organizations
1255 that adopt the UUID model described here can extend it to other data sources that store device
1256 platform data, provided that the Asset Discovery and Management System is configured to update
1257 existing records based on the UUID, and the platform data is mapped to the appropriate data fields in
1258 the Asset Discovery and Management System.

1259 The provisioning process for computing devices in this prototype demonstration that are included in the
1260 Intel TSC platform uses `TSCVerifyUtil` (Section 4.4.3) to export a platform manifest that is uploaded to
1261 the Platform Manifest Correlation System's web-based interface (Section 4.3.3) by the IT administrator.

1262 For Dell and HP Inc. laptops which use the HIRS ACA platform, we opted to use a script-based approach
1263 to automatically upload the platform manifest to the Platform Manifest Correlation System's REST API.
1264 Similarly, for HPE and Dell server platforms, the manifests produced by each manufacturer's validation
1265 tool is uploaded via the REST API. The use of a web interface or REST API demonstrates flexibility in the

1266 architecture that can assist organizations with a heterogeneous manufacturer environment or use cases
1267 where automation is not feasible.

1268 Once the platform manifests across manufacturers are uploaded, a JavaScript based Data Feed within
1269 the Archer IRM platform continuously polls the Platform Manifest Correlation System database API for
1270 new computing devices (Section 4.3.3). A DataFeed can be thought of as a scheduled task that
1271 aggregates data within the Archer Platform.

5.2.2.1 Provisioning Example

1273 Figure 5-4 presents a representative example for an individual computing device that has been
1274 provisioned into the Asset Inventory component of the Archer Platform using the Intel TSC platform. The
1275 screenshot shows the baseline data available across all demonstration computing devices including
1276 manufacturer, device model, and serial number.

1277 **Figure 5-4 Asset Inventory and Discovery Example 1**

▼ GENERAL INFORMATION	
Enterprise Unique Identifier: 00787415-1181-e411-906e-0012795d96dd	**Serial Number:** 4734A10C
Platform Model: S2600WTT	**Manufacturer:** Intel Corporation
Continuous Monitoring Platform Integrity Status: No Data from Configuration Management System	

1278 Figure 5-5 below shows a partial listing of the components associated with the server in Figure 5-4. Note
1279 that in this case, the three demonstration Seagate drives (Section 4.4.5) are also associated with the
1280 platform.

1281 **Figure 5-5 Asset Inventory and Discovery Example 2**

▼ ASSOCIATED COMPONENTS   View Less

This section displays the computing device declared components.

Tracking ID	Class	Manufacturer	Model	Serial
277286	Baseboard	Intel Corporation	S2600WTT	BQWL51650568
277287	CPU	Intel(R) Corporation	Central Processor	F2060300FFFBEBBF
277288	Memory	Micron	DDR4	0F663371
277290	Storage Drive	SEAGATE	ST18000NM005J	ZR5056HD0000C107GP5G
277291	Storage Drive	SEAGATE	ST18000NM005J	ZR5056GS0000C105D6S3
277292	Storage Drive	SEAGATE	ST18000NM005J	ZR504Z6W0000C105972J
277293	Trusted Platform Module	IFX	SLB9665	4734A10C

1282   Once the Archer's JavaScript DataFeed that retrieves data from the Eclypsium Analytic Backend (cloud or
1283   on-premises) executes, the asset record is updated accordingly with system firmware data, as Figure 5-6
1284   shows.

1285   **Figure 5-6 Asset Inventory and Discovery Example 3**

1286

▼ ECLYPSIUM FIRMWARE ANALYTICS

Integrity data from the Eclypsium platform.

Last System Scan Date:	1/19/2022	**System Firmware Date:**	9/2/2020
Eclypsium Integrity Scan Status:	Integrity Issue Detected - Action Recommended	**System Firmware Version:**	SE5C610.86B.01.01.1029.090220201031
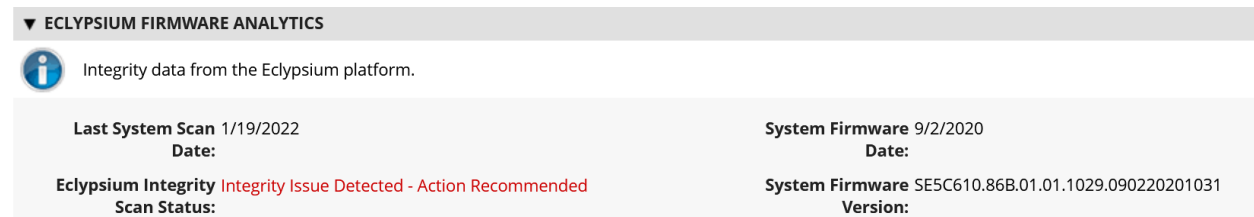
1287   ***Step 4b:*** *If the administrator updates the configuration of the platform (e.g., adding hardware*
1288   *components, updating firmware), then the administrator might create new platform artifacts to*
1289   *establish a new baseline.*

1290   A common use case for IT organizations is the replacement of a component in a fielded computing
1291   device. For example, an end user may request additional memory or the replacement of a broken
1292   component. This will cause future platform validation errors because the fielded computing device
1293   manifest will be updated to reflect the changed components and will differ from the as-built manifest.
1294   Below, we discuss three examples of updating the configuration of the platform that were
1295   demonstrated during the project.

1296   In the preliminary draft of this publication, for laptop systems that leveraged the HIRS ACA platform, the
1297   verifiable artifact (Platform Certificate) is re-generated and uploaded to the HIRS ACA backend, and the
1298   device is re-provisioned by the IT administrator. In this revision, we have utilized delta certificates, which
1299   are defined as part of the TCG Platform Certificate Profile Specification 1.1. The specification defines a
1300   "base" Platform Certificate (Section 5.2.1) and a "delta" which attests to specific changes made to the
1301   platform that are not reflected in the original Platform Certificate. Generally, the Delta Platform
1302   Certificate is issued by the organizational owner of the computing device, as opposed to the base
1303   Platform Certificate, which is issued by the manufacturer. Once the HIRS-ACA has been updated with a
1304   new Delta Platform Certificate, it is able to track changes to the platform, forming a "chain" of Delta
1305   Platform Certificates which reference the Base Platform Certificate.

1306   For systems that use Intel's TSC platform, the IT administrator uploads the new computing device
1307   configuration to the TSC Web Portal using Intel's software tools. The Intel TSC platform subsequently
1308   regenerates the verifiable artifacts, and the IT administrator makes them available for download when
1309   the provisioning process is restarted. We were able to exercise this process successfully using Intel-
1310   contributed laptops.

1311   Finally, Dell server manifests are updated in the field by manufacturer technicians using specialized
1312   tools. The tooling generates a new manifest for the server, which is delivered to Dell's environment and

1313 re-signed by Dell's high-assurance certificate issuing authority that previously signed the original
1314 verifiable artifact embedded from the factory.

5.2.3 Scenario 3

1316 The desired outcome of Scenario 3 is to ensure computing device components are verified against the
1317 attributes and measurements declared by the manufacturer or purchasing organization during
1318 operational usage. This scenario is primarily enabled by the Configuration Management System (Section
1319 4.2.3), Eclypsium Analytic Platform (Section 4.3.4), and manufacturer-specific integrity monitoring tools.
1320 Supplemental sequence diagrams are available in Appendix C.

1321 To support build testing of Intel TSC platforms in this scenario, we implemented a negative test case to
1322 simulate a platform integrity issue, such as a component swap. The scenario used the DPD intended for
1323 another system in place of the correct DPD to ensure the Intel platform validation would fail. We
1324 repeated this test with an incorrect Platform Certificate, which also failed validation as expected. The
1325 failed validation was subsequently detected by the configuration management system, which monitored
1326 the validation status of the Intel TSC tools as described in Section 4.2.3.

1327 Similarly, we performed build testing of laptops that were continuously monitored by the HIRS-ACA
1328 Windows agent. In this test case we used a virtual machine to perform initial acceptance testing with
1329 the network-booted TPM Provisioner. The Windows-based TPM Provisioner was subsequently installed
1330 and monitored by the Configuration Management System. We then updated the virtual hardware to
1331 produce an integrity error (component swap) which was detected by the Configuration Management
1332 System.

1333 HP Inc. supplied additional integrity event continuous monitoring scenarios and remediations that were
1334 demonstrated in our lab environment. In the first, we simulated an attempt by a locally present user to
1335 gain access to the firmware configuration user interface, and the system was rebooted to block a brute
1336 force attack. This event may be an indication of a malicious, locally present actor attempting to modify
1337 firmware settings. In the second demonstration, we simulated an event that indicated there was a
1338 repeated programmatic attempt made to modify a firmware (BIOS) setting without the proper
1339 authorization and that interface has been disabled until the next reboot. A reboot is required to re-
1340 enable the WMI interfaces that can be used to modify BIOS setting with proper authorization. This event
1341 may be an indication of malicious software present on the target device attempting to modify firmware
1342 settings. The two previous events may cause an action by the IT administrator, such as removing access
1343 to network enterprise resources. Finally, we ran a scenario in which the physical cover was removed
1344 from the laptop. This is indicative of potential physical tampering by an unauthorized party and the
1345 laptop is disabled. The remediation in this case is for the IT administrator to unlock the laptop.

1346 The final use case we examined across all manufacturers is when system firmware is updated on the
1347 fielded laptop. This may be initiated by the end user who is guided by a helpdesk or by the IT
1348 administrator. In either case, the Eclypsium scanner that is installed during Scenario 2 detects this

change and reflects it in the Eclypsium Analytic Backend. The Archer JavaScript Transporter Data Feed
subsequently ingests the change, and it is reflected in the asset repository. Similarly, the Eclypsium
Analytic Backend will detect out-of-date firmware versions and other potential platform integrity issues
from laptops and servers that are monitored by the Eclypsium Analytic Platform. The demonstration
observed this behavior through the normal lifecycle of manufacturer-provided firmware updates that
include modifications to address vulnerabilities and active threats.

Similarly, firmware measurements produced by the Seagate Firmware Attestation capability are tracked
for changes, and those changes are associated with the Intel server that the drives are connected to in
this demonstration. A firmware measurement change in this case could be indicative of a non-malicious
act, such as a firmware update. However, it could also represent an attack on the drive firmware that
requires a recovery mechanism by the Security Operator.

With the platform and monitoring data collected from Scenario 3, we created a dashboard that enables
an organization to achieve better visibility into supply chain attacks and detect advanced persistent
threats and other advanced attacks. Depending on the size of the organization, the targeted audience
may all be the same person. In the *Validating the Integrity of Computing Devices* project description of
an IT administrator, it is possible that for some organizations, one person performs all those functions.
In other organizations, functions might be addressed by separate teams within a SOC.

5.2.3.1 Continuous Monitoring Example

A snippet of the demonstration enterprise dashboard is provided in Figure 5-7. There are two security
event panels shown, which enable the IT administrator to quickly identify enterprise computing devices
that are out of compliance and may require a remediation action. *Enterprise Computing Devices with
Out of Compliance Platform Manifests* refers to the number of inventoried computing devices that have
failed a compliance rule in the Configuration Management System. *Enterprise Computing Devices with
Out of Compliance Platform Integrity* refers to the number of inventoried computing devices that the
Eclypsium Analytic Platform (either on-premises or cloud) has identified as having an integrity issue.
When either panel is clicked, a list of computing devices is presented, and the systems security engineer
can make a risk management decision on the individual computing device.

Figure 5-7 Scenario 3 Dashboard
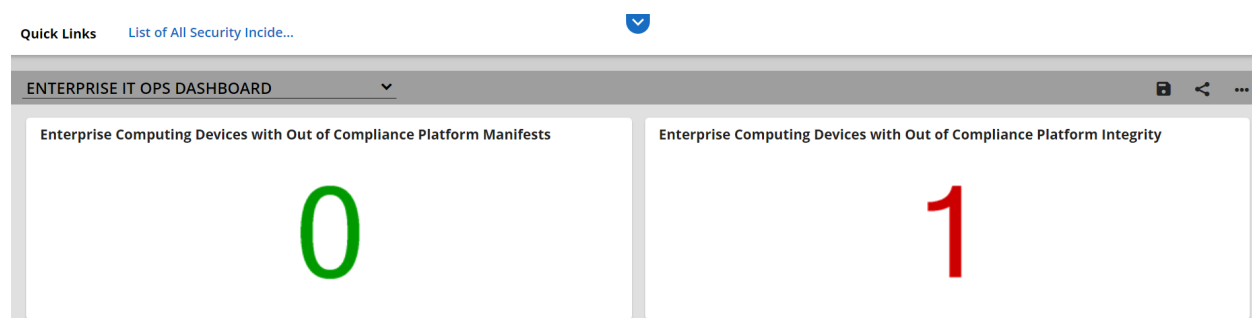
1377 In addition to the dashboard described above, we demonstrated the capability to automatically create
1378 an incident tracking record when our SIEM detects a platform integrity security event for a SOC's
1379 incident response team. The record is associated with the computing device as shown in Figure 5-8. In
1380 this example incident, Archer has imported a security event (offense) from the SIEM involving a
1381 continuously monitored HP Inc. laptop.

1382 **Figure 5-8 Scenario 3 Security Event**

	Incident ID ▲	SCA Computing Device	Incident Summary	Days Open	Incident Status
⊞	INC-277233	3206d7fa-d7d3-b406-daf5-62d4c47d6d79	HP_Sure_Start Integrity violation	0 Day(s)	New

Page 1 of 1 (1 records)

1383 Clicking on the Incident ID reveals more details about the incident for the personnel assigned to
1384 investigate the incident for additional context. This is pictured in Figure 5-9.

1385 **Figure 5-9 Scenario 3 Security Event Summary**

▼ INCIDENT SUMMARY

Incident ID: INC-277233   Source: IBM Qradar

Title: HP_Sure_Start Integrity violation

Incident Summary: HP_Sure_Start Integrity violation

Incident Details: This indicates that HP Sure Start has detected that the main drive partition table has been altered, and HP Sure Start has returned the partition table to the desired state.   This event could be indicative of an attack on the device in the event the change to the drive partition tables was made by an unauthorized party.

SCA Computing Device
Enterprise Unique Identifier
3206d7fa-d7d3-b406-daf5-62d4c47d6d79

1386 Finally, the Incident summary can provide a set of remediation actions for the security personnel. In the
1387 example (Figure 5-10), an analyst has recommended that the incident response personnel remove the
1388 computing device in question from the environment. Other remediation actions related to platform
1389 integrity security events could include replacing a system component, updating or changing the
1390 firmware configuration, or executing manufacturer-specific platform recovery capabilities that are
1391 aligned with NIST SP 800-193, Platform Firmware Resiliency Guidelines.

1392   **Figure 5-10 Scenario 3 Security Event Remediation**
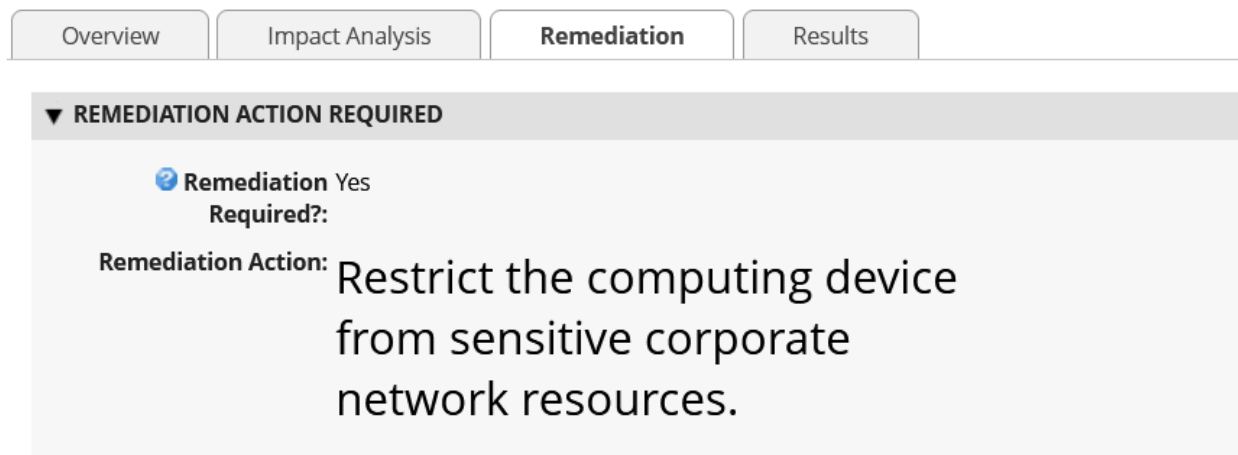
1393   ## 5.3 Scenarios and Findings

1394   One aspect of our security evaluation involved assessing how well the reference design addresses the
1395   security characteristics that it was intended to support. The Cybersecurity Framework Subcategories
1396   were used to provide structure to the security assessment by consulting the specific sections of each
1397   standard that are cited in reference to a Subcategory. The cited sections provide validation points that
1398   the example solution would be expected to exhibit. Using the Cybersecurity Framework Subcategories
1399   as a basis for organizing our analysis allowed us to systematically consider how well the reference design
1400   supports the intended security characteristics.

1401   ### 5.3.1 Supply Chain Risk Management (ID.SC)

1402   #### 5.3.1.1 ID.SC-4: Suppliers and third-party partners are routinely assessed using audits, test
1403   results, or other forms of evaluations, to confirm they are meeting their contractual
1404   obligations.

1405   This Cybersecurity Framework Subcategory is supported in the prototype implementation by the
1406   manufacturer-specific validation tools and the HIRS ACA platforms. Specifically, Scenario 2 acceptance
1407   testing acts as an initial evaluation of the manufacturer (supplier) to validate the source and integrity of
1408   assembled components for the recipient organization of the computing device.

1409 5.3.2 Asset Management (ID.AM)

1410 *5.3.2.1 ID.AM-1: Physical devices and systems within the organization are inventoried*

1411 This Cybersecurity Framework Subcategory is supported in the prototype implementation by Archer and
1412 the Platform Manifest Correlation System. When used in conjunction, they form the basis of an Asset
1413 Discovery and Management System that accurately reflects computing devices within an organization,
1414 including all components therein.

1415 5.3.3 Identity Management, Authentication and Access Control (PR.AC)

1416 *5.3.3.1 PR.AC-6: Identities are proofed and bound to credentials and asserted in*
1417 *interactions*

1418 This Cybersecurity Framework Subcategory is supported in the prototype implementation by Archer and
1419 all hardware contributors. The manufacturers in this prototype support device-unique identifiers which
1420 are associated with organizational computing devices. Identifiers are prevented from being re-used
1421 through Archer data integrity (primary key) constraints.

1422 5.3.4 Data Security (PR.DS)

1423 *5.3.4.1 PR.DS-6: Integrity-checking mechanisms are used to verify software, firmware, and*
1424 *information integrity*

1425 This Cybersecurity Framework Subcategory is supported in the prototype implementation by Archer and
1426 the Eclypsium Analytic Platform. Together, they provide the capability to detect unauthorized changes
1427 to firmware. All participating manufacturers provide capabilities to report firmware version information.

1428 *5.3.4.2 PR.DS-8: Integrity-checking mechanisms are used to verify hardware integrity*

1429 This Cybersecurity Framework Subcategory is supported in the prototype implementation by Archer,
1430 Microsoft Configuration Manager, IBM QRadar, and manufacturer-specific integrity validation tools.
1431 Together, these products provide the capability to document, manage, and control the integrity of
1432 changes to organizational computing devices.

1433 5.3.5 Security Continuous Monitoring (DE.CM)

1434 *5.3.5.1 DE.CM-7: Monitoring for unauthorized personnel, connections, devices, and*
1435 *software is performed*

1436 This Cybersecurity Framework Subcategory is supported in the prototype implementation by Archer,
1437 Microsoft Configuration Manager, IBM QRadar, and the Eclypsium Analytic Platform. Together, these
1438 products form part of an organizational continuous monitoring program. Microsoft Endpoint

1439 Configuration Manager, IBM QRadar, and the Eclypsium platform enable automated monitoring of
1440 computing devices for hardware and firmware integrity issues at an organization-defined frequency.
1441 This security information is made available to organizational officials through an Archer dashboard,
1442 where a risk management decision can be made when a computing device is deemed out of compliance.

1443 # 6 Future Build Considerations

1444 In this updated publication, we have described an architecture that decreases the risk of a compromise
1445 to products in an organization's supply chain, which in turn may reduce risks to customers and end users
1446 that use computing devices operationally. This draft has built on the preliminary demonstration
1447 prototype and has incorporated servers into the architecture, to include hardware contributed by Dell,
1448 Hewlett Packard Enterprise, Intel, and Seagate. Additionally, we have extended the architecture to
1449 include a SIEM contributed by IBM to support continuous monitoring scenarios.

1450 In the future, this project may expand the hardware root of trust capabilities to include platform
1451 components such as internal storage drives, network controllers, and memory modules. As we've
1452 demonstrated in this project, the TPM module provides a basis for a laptop or server's root of trust.
1453 Newer specifications, such as the TCG's Device Identifier Composition Engine (DICE) implementation,
1454 which currently addresses IoT devices, can be extended to platform components where a hardware root
1455 of trust is not feasible. Further, the Security Protocol and Data Model (SPDM) will provide the ability to
1456 securely communicate with the platform components, providing a similar mechanism that exists today
1457 with the Platform Certificates.

1458 Similarly, TCG's Reference Integrity Manifest (RIM) specification could extend our acceptance testing
1459 capability to provide firmware validation. This capability is dependent on manufacturer support in the
1460 form of a digitally signed "bundle" as a reference to the as-shipped firmware measurements.

1461 Further, the concepts we have demonstrated in this project and described in this section could be
1462 integrated into a zero trust architecture. NIST SP 800-207, Zero Trust Architecture addresses this
1463 capability as part of a continuous diagnostics and mitigation (CDM) system. A CDM system is a core
1464 component of a zero trust architecture, which, among other functions, can detect the presence of non-
1465 approved components.

1466 In closing, the NCCoE Supply Chain Assurance project team will continue to monitor the development of
1467 best practices and standards from industry and organizations such as the Trusted Computing Group that
1468 address platform integrity. We invite comments and suggestions from the C-SCRM community of
1469 interest that will enable organizations to operationalize the prototype demonstrations presented in this
1470 publication.

Appendix A   List of Acronyms

ACA	Attestation Certificate Authority
AIC	Attestation Identity Credential
API	Application Programming Interface
BIOS	Basic Input/Output System
C-SCRM	Cyber Supply Chain Risk Management
CA	Certificate Authority
CDM	Continuous Diagnostics and Mitigation
CMSL	(HP) Client Management Script Library
CSR	Certificate Signing Request
DevID	Device Identity
DHCP	Dynamic Host Client Protocol
DICE	Device Identifier Composition Engine
DIMM	Dual In-Line Memory Module
DPD	Direct Platform Data
DTD	Dell Trusted Device
EFI	Extensible Firmware Interface
EK	Endorsement Key
ESP	EFI System Partition Storage
FIPS	Federal Information Processing Standards
FTP	File Transfer Protocol
GIDEP	Government-Industry Data Exchange Program
GRC	Governance, Risk, and Compliance
HIRS	Host Integrity at Runtime and Start-Up
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAK	Initial Attestation Key
ICT	Information and Communications Technology
IDevID	Initial Device Identity
iDRAC	Dell Remote Access Controller
IoT	Internet of Things
IT	Information Technology

JSON	JavaScript Object Notation
NCCoE	National Cybersecurity Center of Excellence
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NvRAM	Non-Volatile Random-Access Memory
OEM	Original Equipment Manufacturer
OS	Operating System
OT	Operational Technology
PACCOR	Platform Attribute Certificate Creator
PCR	Platform Configuration Register
PCVT	Platform Certificate Verification Tool
PXE	Preboot Execution Environment
REST	Representational State Transfer
RIM	Reference Integrity Manifest
SCRM	Supply Chain Risk Management
SCV	Secured Component Verification
SDA	Secure Device Authentication
SDLC	System Development Life Cycle
SecCM	Security-Focused Configuration Management
SFTP	Secure File Transfer Protocol
SIEM	Security Information and Event Management
SMBIOS	System Management BIOS
SOC	Security Operations Center
SP	Special Publication
SPDM	Security Protocol and Data Model
TCG	Trusted Computing Group
TFTP	Trivial File Transfer Protocol
TPer	Trusted Peripheral
TPM	Trusted Platform Module
TSC	(Intel) Transparent Supply Chain
UEFI	Unified Extensible Firmware Interface
UUID	Universally Unique Identifier

VAR	Value-Added Reseller
WMI	Windows Management Instrumentation
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Translation

Appendix B   References

[1]   T. Diamond et al., Validating the Integrity of Computing Devices: Supply Chain Assurance, National Institute of Standards and Technology (NIST), Gaithersburg, Md., March 2020, 14 pp. Available: https://www.nccoe.nist.gov/sites/default/files/library/project-descriptions/tpm-sca-project-description-final.pdf.

[2]   A. Regenscheid, *Platform Firmware Resiliency Guidelines,* NIST Special Publication (SP) 800-193, Gaithersburg, Md., May 2018, 45 pp. Available: https://doi.org/10.6028/NIST.SP.800-193.

[3]   J. Boyens et al., *Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations,* NIST SP 800-161 Revision 1, Gaithersburg, Md., May 2022, 326 pp. Available: https://doi.org/10.6028/NIST.SP.800-161r1.

[4]   Joint Task Force, *Guide for Conducting Risk Assessments,* NIST SP 800-30 Revision 1, Gaithersburg, Md., September 2012, 95 pp. Available: https://doi.org/10.6028/NIST.SP.800-30r1.

[5]   Joint Task Force, *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy,* NIST SP 800-37 Revision 2, Gaithersburg, Md., December 2018, 183 pp. Available: https://doi.org/10.6028/NIST.SP.800-37r2.

[6]   *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*, NIST, Gaithersburg, Md., April 2018, 55 pp. Available: https://doi.org/10.6028/NIST.CSWP.04162018.

[7]   Joint Task Force, *Security and Privacy Controls for Information Systems and Organizations,* NIST SP 800-53 Revision 5, Gaithersburg, Md., September 2020, 492 pp. Available: https://doi.org/10.6028/NIST.SP.800-53r5.

[8]   *Trusted Platform Module Library Specification, Family "2.0," Level 00, Revision 01.59,* Trusted Computing Group, November 2019. Available: https://trustedcomputinggroup.org/resource/tpm-library-specification/.

[9]   A. Johnson et al., *Guide for Security-Focused Configuration Management of Information Systems,* NIST SP 800-128, Gaithersburg, Md., August 2011, 99 pp. Available: https://doi.org/10.6028/NIST.SP.800-128.

[10]   *Archer Platform Documentation,* RSA. Available: https://community.rsa.com/t5/archer-platform-documentation/data-governance-design/ta-p/556139.

[11]   *Introduction to Configuration Manager*, Microsoft, June 2015. Available: https://help.archerirm.cloud/archer_suite_help/en-us/Content/portal_home.htm.

[12]   *Host Integrity at Runtime and Start-up (HIRS): Attestation Certificate Authority (ACA) and TPM Provisioning with Trusted Computing-based Supply Chain Validation,* 2020. Available: https://github.com/nsacyber/HIRS/.

[13]   *TCG Platform Attribute Credential Profile, Specification Version 1.1, Revision 15,* Trusted Computing Group (TCG), 2019, 61 pp. Available: https://trustedcomputinggroup.org/wp-content/uploads/IWG_Platform_Certificate_Profile_v1p1_r15_pubrev.pdf.

[14]   *HP Secure Platform Management with the HP Client Management Script Library,* HP Inc. Available: https://developers.hp.com/hp-client-management/blog/hp-secure-platform-management-hp-client-management-script-library.

[15]   *Secure BIOS with HP Sure Admin and CMSL,* HP Inc. Available: https://developers.hp.com/hp-client-management/blog/secure-bios-hp-sure-admin-and-cmsl-upd-292021.

[16]   *HP Sure Start Whitepaper: Firmware Security and Resilience,* HP Inc., 2021, 24 pp. Available: https://www8.hp.com/h20195/v2/getpdf.aspx/4AA7-6645ENW.pdf.

[17]   *HP TamperLock: Protecting Devices from Physical Attacks,* HP Inc., 2021, 6 pp. Available: https://www8.hp.com/h20195/v2/GetPDF.aspx/4AA7-8167ENW.pdf.

[18]   *Transparent Supply Chain,* Intel. Available: https://www.intel.com/content/www/us/en/products/docs/servers/transparent-supply-chain.html.

[19]   *Platform Certificate Verification Tool*, HPE. Available: https://github.com/HewlettPackard/PCVT.

[20]   *TCG Credential Profiles For TPM Family 1.2; Level 2*, Specification Version 1.2, Revision 8, Trusted Computing Group (TCG), 2013, 64 pp. Available: https://trustedcomputinggroup.org/wp-content/uploads/Credential_Profiles_V1.2_Level2_Revision8.pdf.

[21]   TPM 2.0 *Keys for Device Identity and Attestation*, Version 1.00, Revision 12, Trusted Computing Group, 2021, 71 pp. Available: https://trustedcomputinggroup.org/wp-content/uploads/TPM-2p0-Keys-for-Device-Identity-and-Attestation_v1_r12_pub10082021.pdf.

[22]   S. Symington et al., *Trusted Internet of Things (IoT) Device Network-Layer Onboarding and Lifecycle Management*, NIST Cybersecurity White Paper (Draft), Gaithersburg, Md., September 2020, 88 pp. Available: https://doi.org/10.6028/NIST.CSWP.09082020-draft.

Appendix C   Project Scenario Sequence Diagrams

The figures in this appendix detail the flow of scenario interactions between a demonstration computing device and the supporting software/services. Note that not all scenarios were supported by every manufacturer. We have represented the software that is installed on the computing device and the platform integrity/provisioning services as blue boxes across the top. Steps that are part of a larger process are bounded by black boxes.

Figure C-1 Dell and HP Inc. Laptop Scenario 2 Part 1
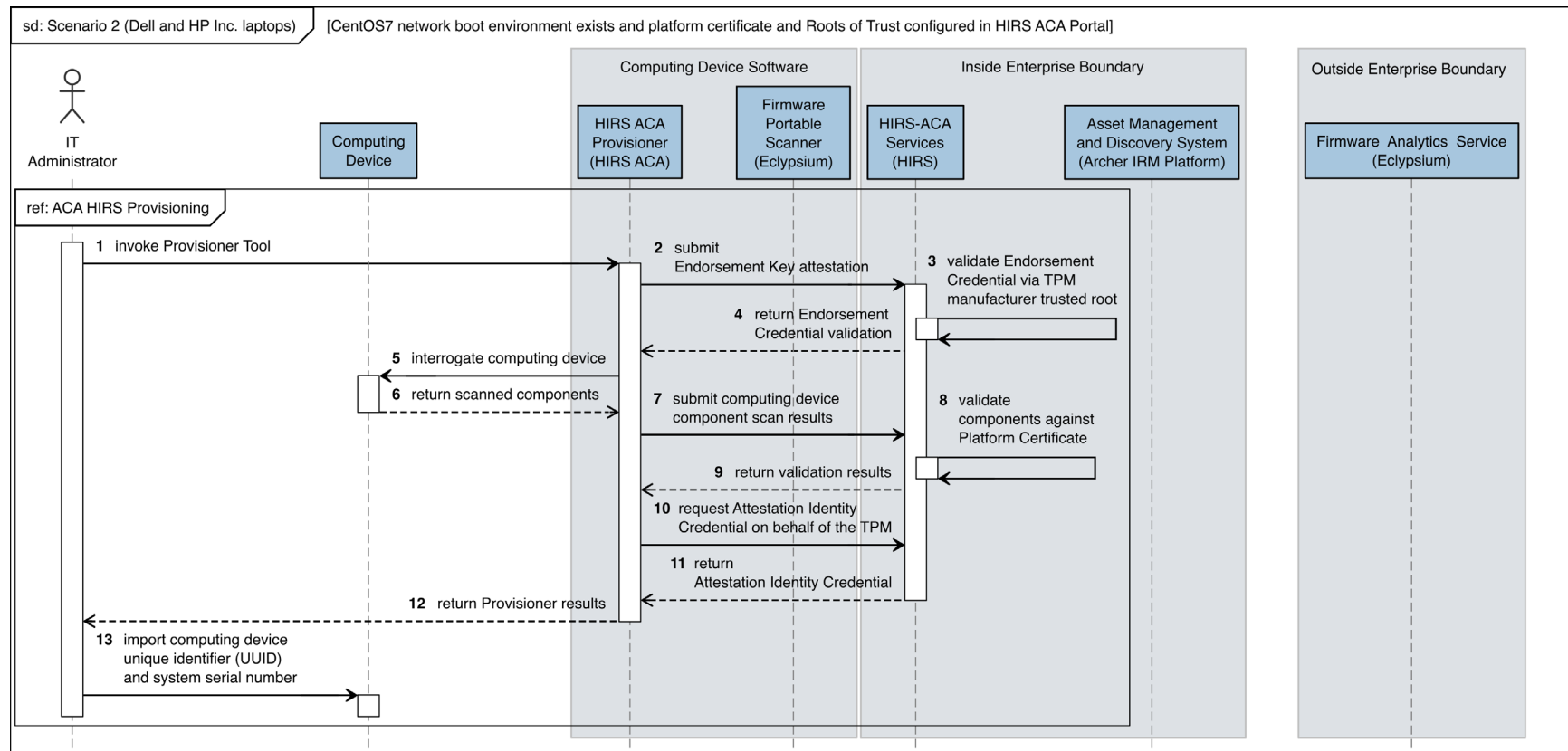
Figure C-2 Dell and HP Inc. Laptop Scenario 2 Part 2
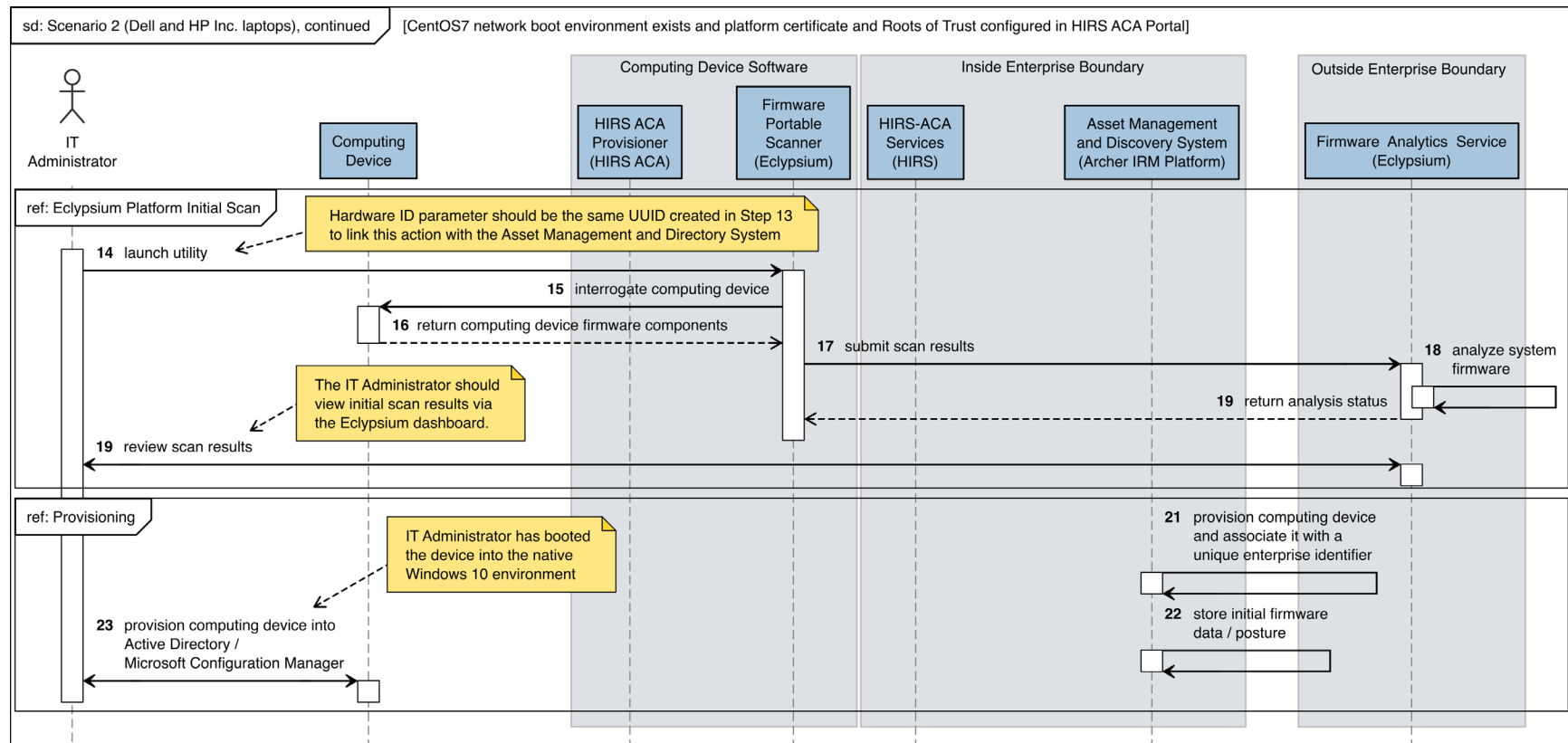
Figure C-3 Intel Laptop Scenario 2 Part 1
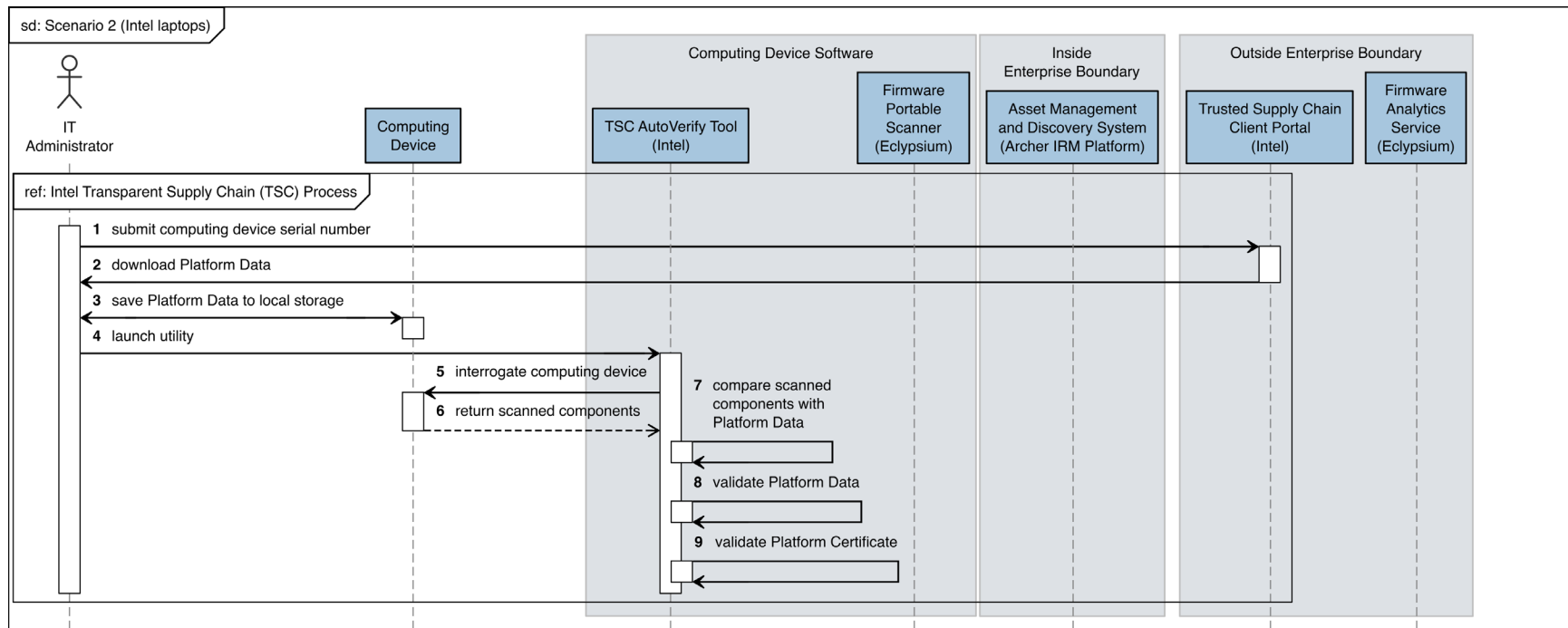
Figure C-4 Intel Laptop Scenario 2 Part 2
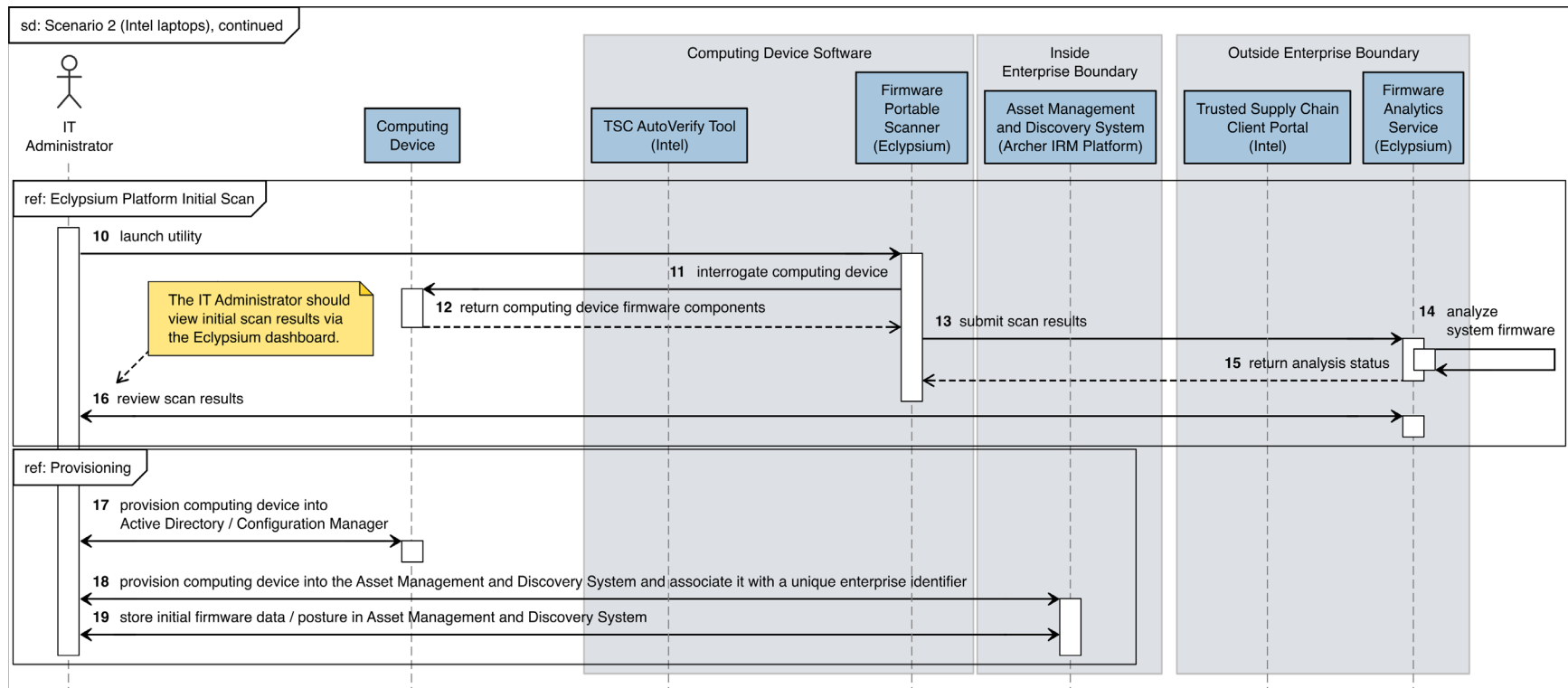
Figure C-5 Intel Server Scenario 2 Part 1

DRAFT
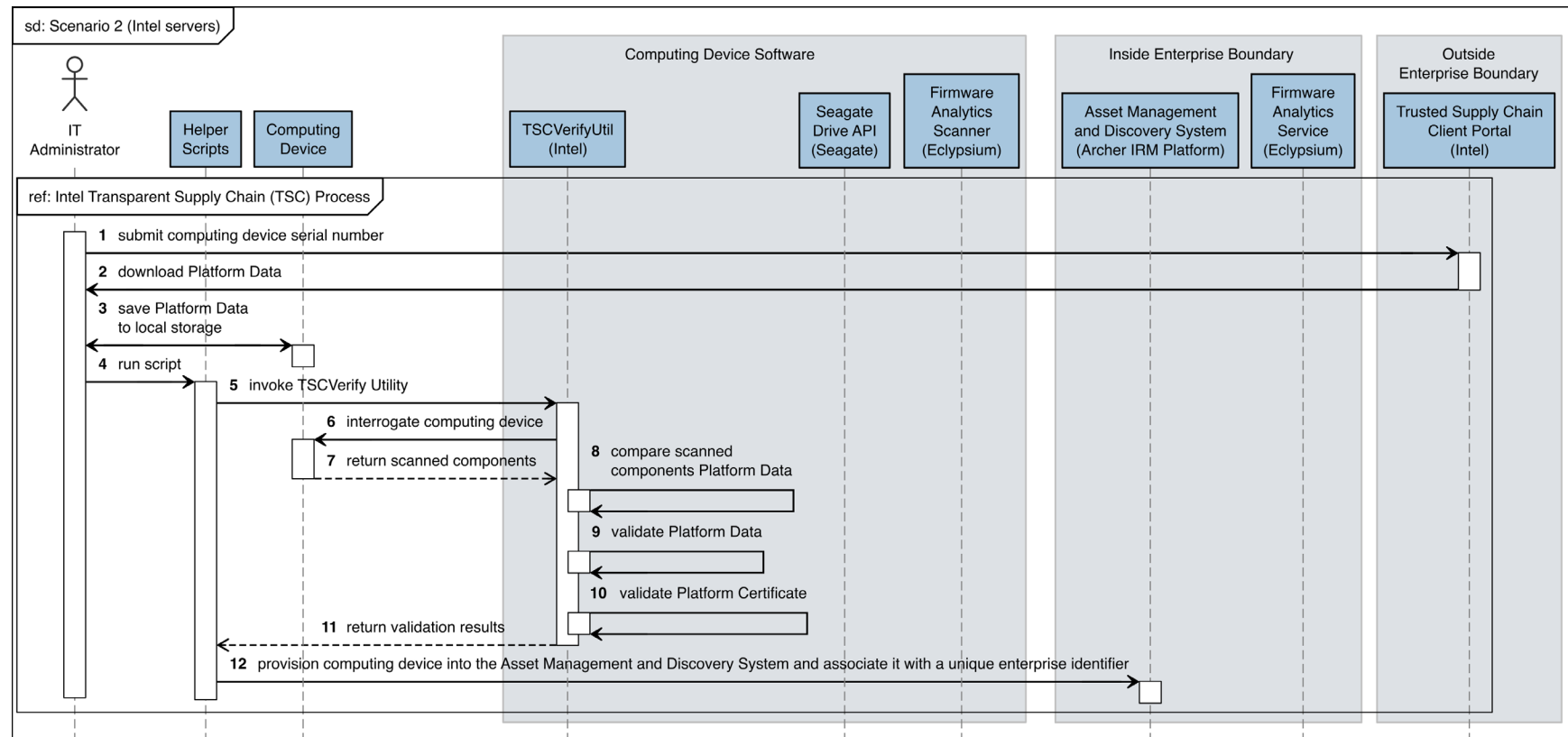
Figure C-6 Intel Server Scenario 2 Part 2
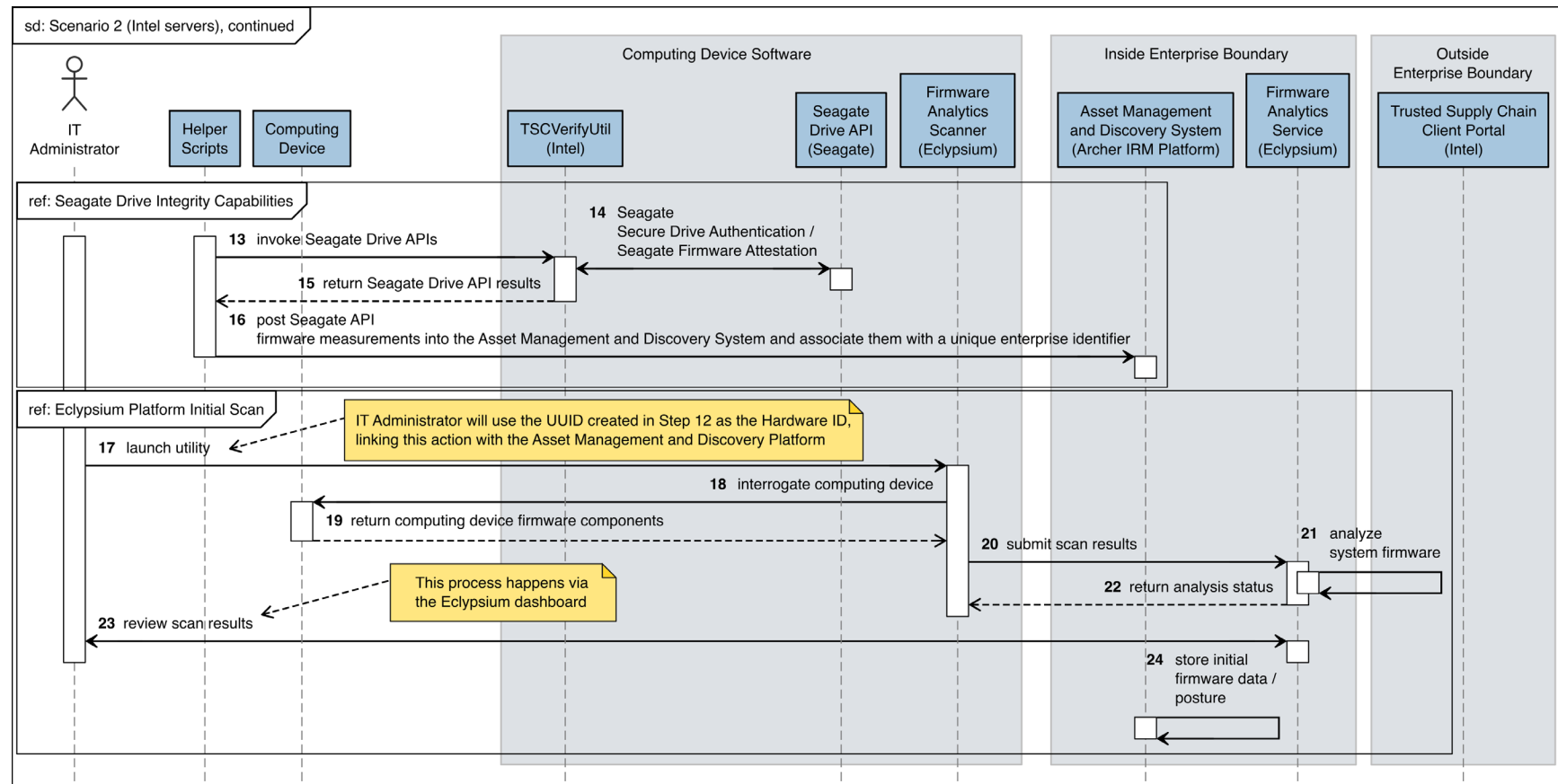
Figure C-7 Dell Server Scenario 2
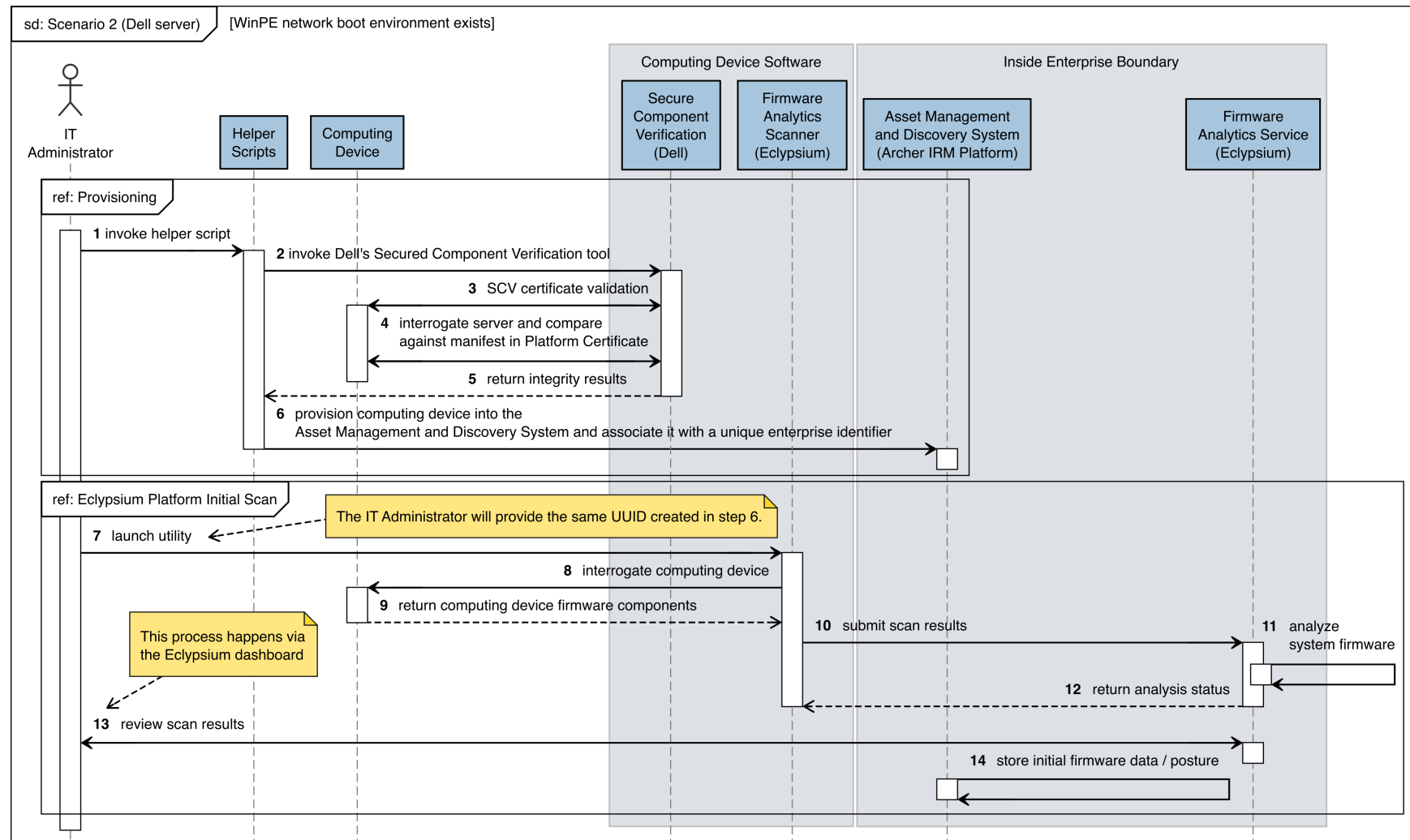
Figure C-8 HPE Server Scenario 2
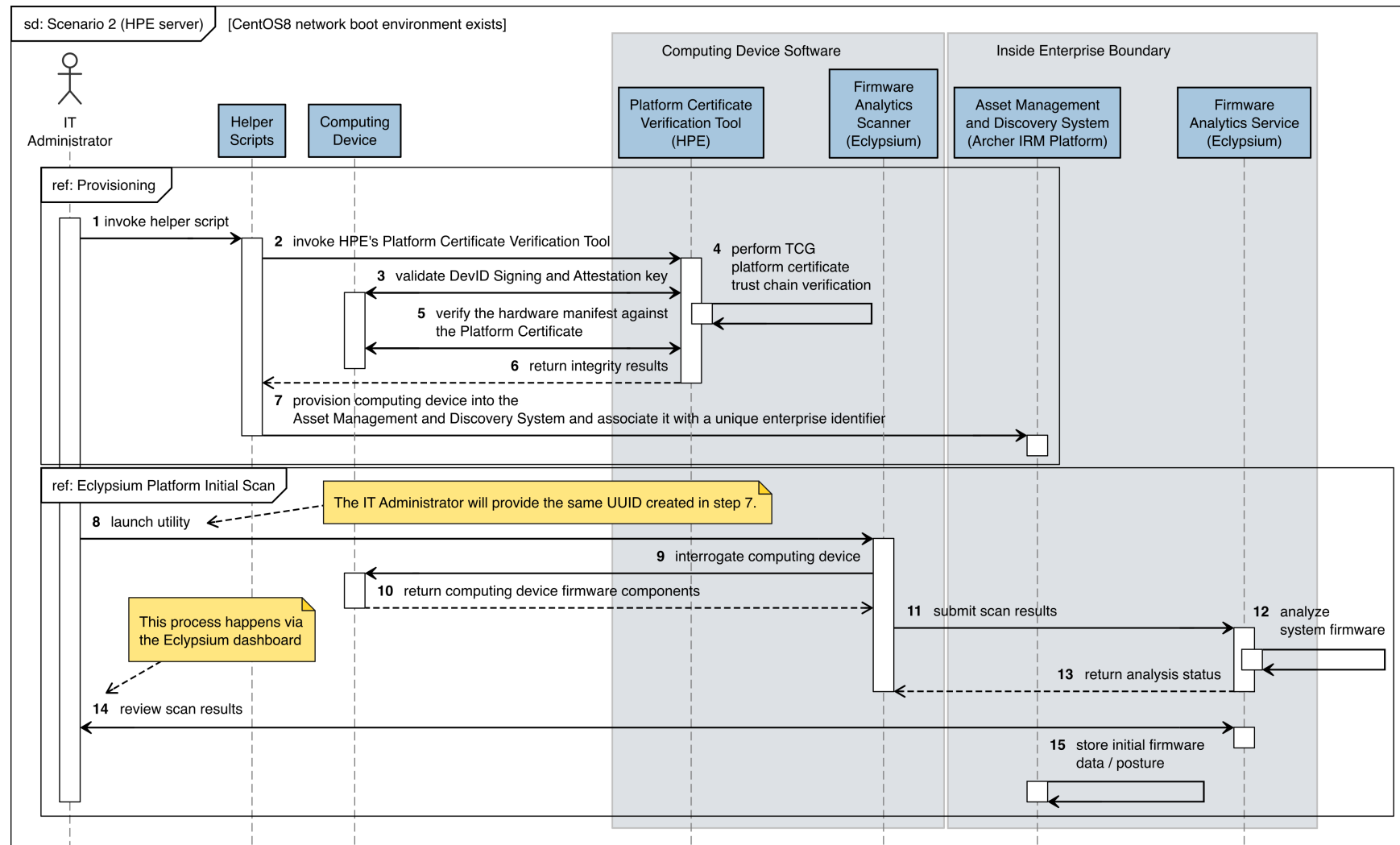
Figure C-9 Intel Laptop Scenario 3
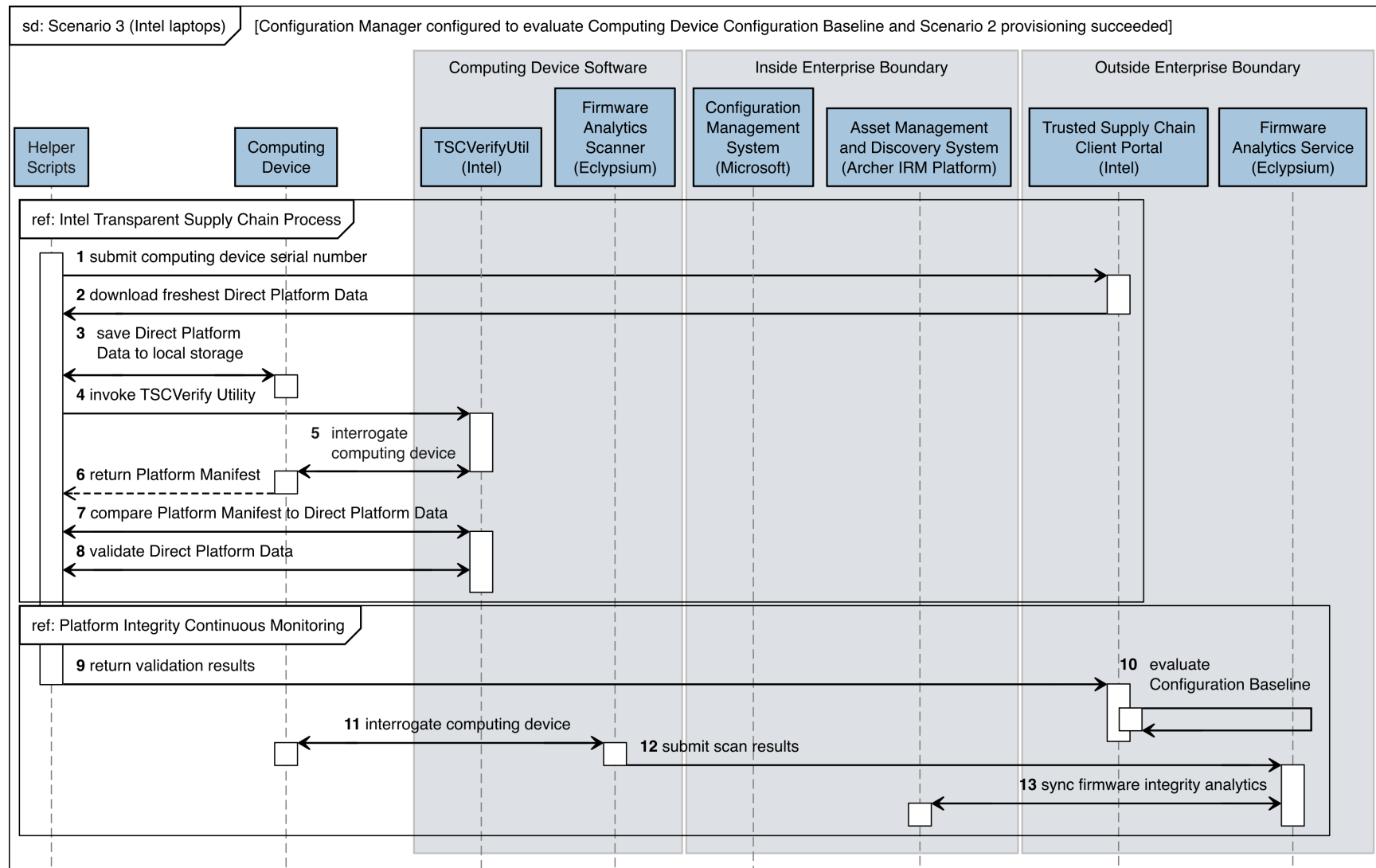
Figure C-10 Dell Laptops Scenario 3
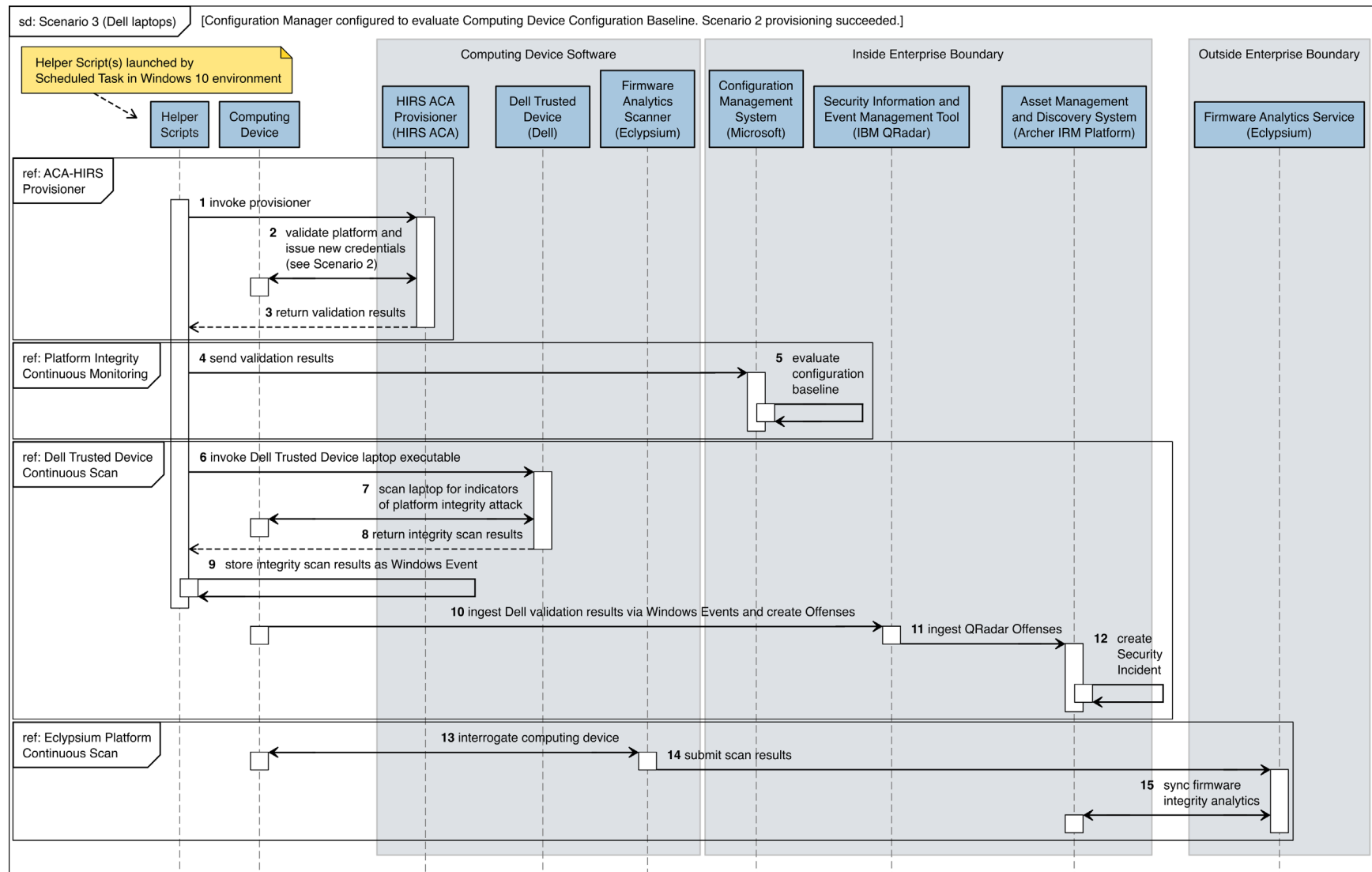
Figure C-11 HP Inc. Laptops Scenario 3

NIST SPECIAL PUBLICATION 1800-34C

Validating the Integrity of Computing Devices

Tyler Diamond*
Nakia Grayson
William T. Polk
Andrew Regenscheid
Murugiah Souppaya
National Institute of Standards and Technology
Information Technology Laboratory

Christopher Brown
Chelsea Deane
The MITRE Corporation
McLean, Virginia

Karen Scarfone
Scarfone Cybersecurity
Clifton, Virginia

Former employee; all work for this publication was done while at employer

June 2022

DRAFT

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.

While NIST and the NCCoE address goals of improving management of cybersecurity and privacy risk through outreach and application of standards and best practices, it is the stakeholder's responsibility to fully perform a risk assessment to include the current threat, vulnerabilities, likelihood of a compromise, and the impact should the threat be realized before adopting cybersecurity measures such as this recommendation.

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to: supplychain-nccoe@nist.gov.

Public comment period: June 23, 2022 through July 27, 2022

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 at supplychain-nccoe@nist.gov.

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

31 ## NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

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

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

47 ## NIST CYBERSECURITY PRACTICE GUIDES

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

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

57 ## ABSTRACT

58 Organizations are increasingly at risk of cyber supply chain compromise, whether intentional or
59 unintentional. Cyber supply chain risks include counterfeiting, unauthorized production, tampering,
60 theft, and insertion of unexpected software and hardware. Managing these risks requires ensuring the
61 integrity of the cyber supply chain and its products and services. This project will demonstrate how
62 organizations can verify that the internal components of the computing devices they acquire, whether
63 laptops or servers, are genuine and have not been tampered with. This solution relies on device vendors
64 storing information within each device, and organizations using a combination of commercial off-the-
65 shelf and open-source tools that work together to validate the stored information. This NIST
66 Cybersecurity Practice Guide provides a draft describing the work performed so far to build and test the
67 full solution.

68 KEYWORDS

69 *computing devices; cyber supply chain; cyber supply chain risk management (C-SCRM); hardware root of*
70 *trust; integrity; provenance; supply chain; tampering.*

71 ACKNOWLEDGMENTS

Name	Organization
Spike E. Dog	The MITRE Corporation
Joe Sain	The MITRE Corporation
Thomas Walters	The MITRE Corporation
Andrew Medak	National Security Agency (NSA)
Lawrence Reinert	NSA
Themistocles Chronis	Archer
Dan Carayiannis	Archer
Manuel Offenberg	Seagate
David Kaiser	Seagate
Paul Gatten	Seagate
Simon Phatigaraphong	Seagate
Bill Downer	Seagate Government Solutions
Jack Fabian	Seagate Government Solutions

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

Technology Partner/Collaborator	Build Involvement
Archer	Archer Suite 6.9
Dell Technologies	PowerEdge R650, Secured Component Verification tool; Precision 3530, CSG Secured Component Verification tool
Eclypsium	Eclypsium Analytics Service, Eclypsium Device Scanner

Technology Partner/Collaborator	Build Involvement
HP Inc.	(2) Elitebook 840 G7, HP Sure Start, HP Sure Recover, Sure Admin, HP Client Management Script Library (CMSL), HP Tamperlock
Hewlett Packard Enterprise	Proliant DL360 Gen 10, Platform Certificate Verification Tool (PCVT)
IBM	QRadar SIEM
Intel	HP Inc. Elitebook 360 830 G5, Lenovo ThinkPad T480, Transparent Supply Chain Tools, Key Generation Facility, Cloud Based Storage, TSCVerify and AutoVerify software tools
National Security Agency (NSA)	Host Integrity at Runtime and Start-Up (HIRS), Subject Matter Expertise
Seagate Government Solutions	(3) 18TB Exos X18 hard drives, 2U12 Enclosure, Firmware Attestation API, Secure Device Authentication API

DOCUMENT CONVENTIONS

The terms "shall" and "shall not" indicate requirements to be followed strictly to conform to the publication and from which no deviation is permitted. The terms "should" and "should not" indicate that among several possibilities, one is recommended as particularly suitable without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. The terms "may" and "need not" indicate a course of action permissible within the limits of the publication. The terms "can" and "cannot" indicate a possibility and capability, whether material, physical, or causal.

CALL FOR PATENT CLAIMS

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

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

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

96 b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring
97 to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft
98 publication either:

99   1. under reasonable terms and conditions that are demonstrably free of any unfair discrimination;
100   or

101   2. without compensation and under reasonable terms and conditions that are demonstrably free
102   of any unfair discrimination.

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

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

110 Such statements should be addressed to: supplychain-nccoe@nist.gov.

Contents

List of Figures

171 # List of Tables

191

1   Introduction

The following volumes of this guide show information technology (IT) professionals and security engineers how we implemented this example solution. We cover all of the products employed in this reference design. We do not re-create the product manufacturers' documentation, which is presumed to be widely available. Rather, these volumes show how we incorporated the products together in our environment.

Note: These are not comprehensive tutorials. There are many possible service and security configurations for these products that are out of scope for this reference design.

1.1   How to Use This Guide

This National Institute of Standards and Technology (NIST) Cybersecurity Practice Guide demonstrates a standards-based reference design and provides users with the information they need to replicate verifying that the internal components of the computing devices they acquire are genuine and have not been tampered with. This reference design is modular and can be deployed in whole or in part.

This guide contains three volumes:

- NIST Special Publication (SP) 1800-34A: *Executive Summary*
- NIST SP 1800-34B*: Approach, Architecture, and Security Characteristics* – what we built and why
- NIST SP 1800-34C: *How-To Guides* – instructions for building the example solution **(you are here)**

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 1800-34A*, which describes the following topics:

- challenges that enterprises face in decreasing the risk of a compromise to products in their supply chain
- 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 *NIST SP 1800-34B*, which describes what we did and why. The following sections will be of particular interest:

- Section 3.4, Risk, describes the risk analysis we performed.
- Section 3.5, Security Control Map, maps the security characteristics of this example solution to cybersecurity standards and best practices.

223 You might share the *Executive Summary, NIST SP 1800-34A*, with your leadership team members to help
224 them understand the importance of adopting a standards-based solution for verifying that the internal
225 components of the computing devices they acquire are genuine and have not been tampered with.

226 **IT professionals** who want to implement an approach like this will find this whole practice guide useful.
227 You can use this How-To portion of the guide, *NIST SP 1800-34C*, to replicate all or parts of the build
228 created in our lab. This How-To portion of the guide provides specific product installation, configuration,
229 and integration instructions for implementing the example solution.

230 This guide assumes that IT professionals have experience implementing security products within the
231 enterprise. While we have used a suite of commercial products to address this challenge, this guide does
232 not endorse these particular products. Your organization can adopt this solution or one that adheres to
233 these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing
234 parts of verifying that the internal components of the computing devices they acquire are genuine and
235 have not been tampered with. Your organization's security experts should identify the products that will
236 best integrate with your existing tools and IT system infrastructure. We hope that you will seek products
237 that are congruent with applicable standards and best practices. Section 3.6, Technologies, of *NIST SP*
238 *1800-34B* lists the products that we used and maps them to the cybersecurity controls provided by this
239 reference solution.

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

1.1.1 Supplemental Material

245 Throughout this draft there are references to code, scripts, and/or configuration files. Due to the size of
246 some of the files, and to provide a more efficient method of access, we have made these assets
247 available via a NIST GitHub repository. This will also enable quicker updates of published code to those
248 interested in replicating parts or all of our demonstration.

1.2 Build Overview

250 In a previous draft of Volume C, we described the steps necessary to set up an environment that focuses
251 on laptop (sometimes referred to by industry as *client*) computing devices. It also provided guidance on
252 the operational usage of manufacturers' tools that may be useful to your IT personnel who verify that
253 the computing device is acceptable to receive into the acquiring organization. In this draft of Volume C,
254 we incorporate validating the integrity of servers and include additional enterprise services as required
255 to support this capability.

256 ## 1.3   Typographic Conventions

257 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.

258 ## 1.4   Logical Architecture Summary

259 Figure 1-1 depicts the architecture for the prototype demonstration environment used within the
260 NCCoE network boundaries. The environment uses a combination of physical and virtual systems to
261 emulate an enterprise architecture. We recommend the reader start with Volume B, Section 4 of this
262 publication for a component-level view of the completed architecture before implementing the systems
263 in this section.

264 Common enterprise services, such as Active Directory (AD) and Domain Name System (DNS), are
265 provided by NCCoE's Trusted Enterprise Infrastructure (TEI). TEI provides common services that labs can
266 use. Previously each lab would spend time and resources to set up common services at the beginning of
267 each project and tear them down after the end of the project. To provide efficiency and consistency
268 across projects, and to represent a true enterprise infrastructure, NCCoE has initiated the TEI effort,
269 which offers common services such as core services and shared security services for those labs who
270 would like to use them.

271   **Figure 1-1 Demonstration Network Architecture**
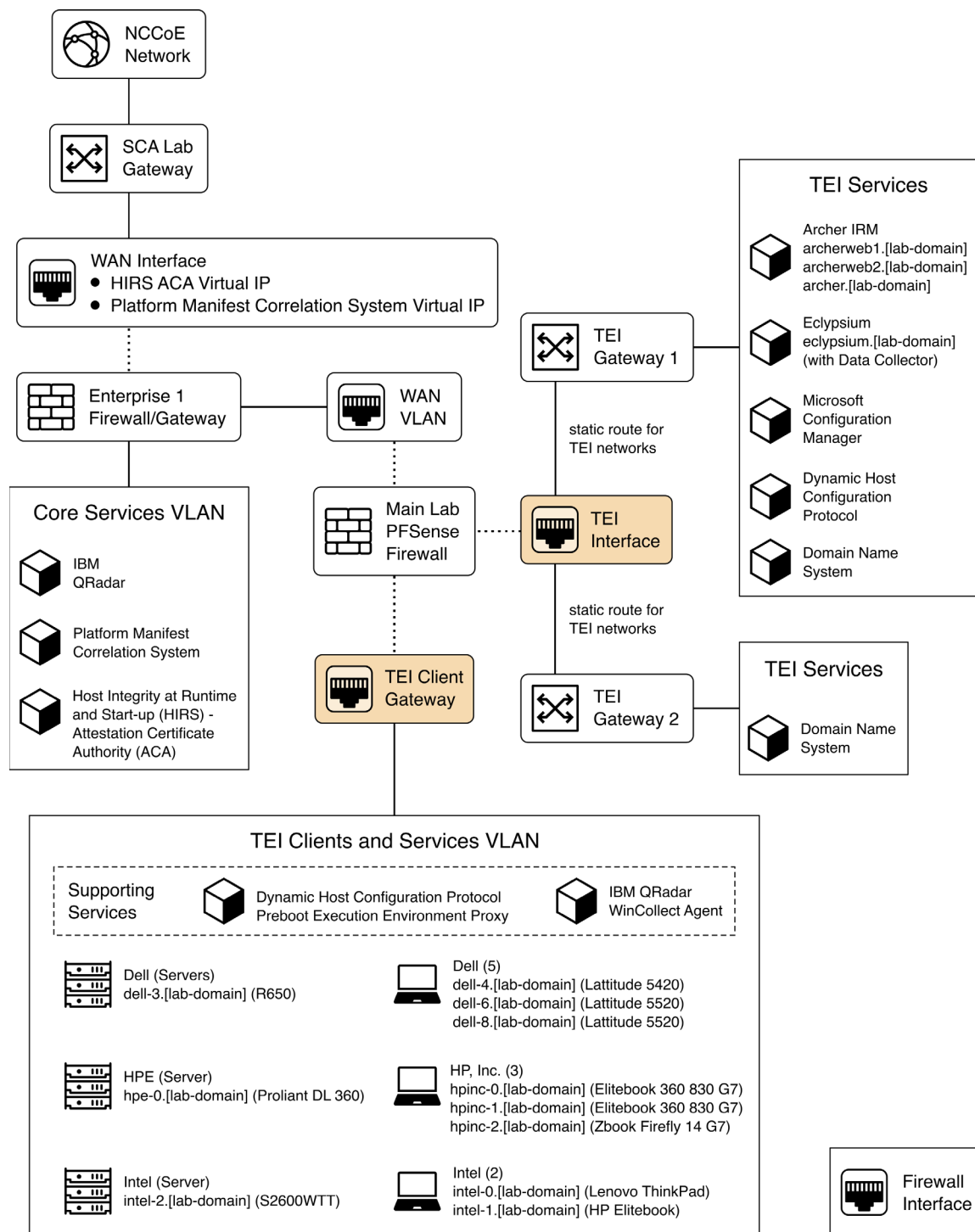
272 Services specific to the capabilities of this prototype demonstration are instantiated on the Core Services
273 virtual network. This virtual network represents the integration of supply chain risk management (SCRM)
274 requirements into an enterprise architecture to support the SCRM controls, as described in the Risk
275 Assessment section of Volume B.

2 Product Installation Guides

277 This section of the practice guide contains detailed instructions for installing and configuring all of the
278 products used to build an instance of the example solution.

2.1 Supporting Systems and Infrastructure

280 This section describes the supporting infrastructure required to execute the acceptance testing and
281 continuous monitoring capabilities provided by our collaborators.

2.1.1 Network Boot Services

283 The following procedures will create an environment that will enable the acceptance testing of
284 computing devices into an enterprise. First, we create CentOS 7, CentOS 8, and WinPE images that will
285 be booted on computing devices via a Preboot Execution Environment (PXE). We then configure the PXE
286 environment to boot the images.

2.1.1.1 Linux-Based Acceptance Testing Image Creation

288 On a development CentOS 7 system, install the latest version of the Host Integrity at Runtime and Start-
289 Up (HIRS) Trusted Platform Module (TPM) Provisioner. We'll use the system as a basis to create the
290 network booted image. Note that there are a number of dependencies that you'll need to satisfy before
291 installing the HIRS TPM Provisioner package. One of those dependencies, PACCOR, is maintained by the
292 HIRS project. In our prototype demonstration, we used version 1.1.4 revision 5 but recommend using
293 the latest version available. Note that any version prior to revision 5 will not successfully complete the
294 provisioning process with the laptop computing devices used in this demonstration.

2.1.1.1.1 HIRS TPM Provisioner Configuration
296 The HIRS TPM provisioner is the core application in the computing device acceptance testing process.
297 The system running the provisioner must be configured for your local environment before use.

298   1. Use a text editor to configure the HIRS TPM Provisioner for your local environment.

299   ```
        $ [your favorite editor] /etc/hirs/hirs-site.config
        ```

300     2.  Change the variables noted below and save the file.

301     
302     
303
```
#***
#* HIRS site configuration properties file
#***
```

```
304
305        # Client configuration
306        CLIENT_HOSTNAME=localhost
307        TPM_ENABLED=true
308        IMA_ENABLED=false
309
310        # Site-specific configuration
311        ATTESTATION_CA_FQDN=hirs-server.yourdomain.test
312        ATTESTATION_CA_PORT=8443
313        BROKER_FQDN=hirs-server.yourdomain.test
314        # Change this port number to your local configuration
315        BROKER_PORT=61616
316        PORTAL_FQDN=hirs-server.yourdomain.test
317        # Change this port number to your local configuration
318        PORTAL_PORT=8443
```

319   3. If using a network boot environment, use the configuration file (step 2) in the kickstart file that
320   creates the CentOS 7 provisioner image in the `%post` section.

2.1.1.1.2   Eclypsium Agent Configuration

322   On the same CentOS 7 system described in Section 2.1.1.1.1, install the Eclypsium Linux agent using the
323   following procedures.

324   1. Navigate to the **Eclypsium Management Console** in a web browser.

325   2. Select **Deployment > Download.**

3. Download the Linux (RPM) Portable Scanner. The filename will have the format `eclypsium_agent_builder-x.x.x.run`.

4. Install the prerequisites for the builder script.

```
# yum groupinstall "Development Tools"
```

```
# yum install kernel-devel
```

5. Run the builder script downloaded above as a user with root privileges. This will build the Eclypsium Portable Scanner drivers, extract the application binaries, and place them into a directory named `eclypsium_agent`.

```
# ./eclypsium_agent_builder-X.X.X.run –out [PATH]
```

6. Confirm the previous step was successful by listing the `eclypsium_agent` directory and ensuring the portable scanner was created with the name `EclypsiumAppPortable`. This executable is referenced by our customized acceptance testing script.

2.1.1.1.3   CentOS 7 Image Creation

The CentOS 7 image we created enables quick revisions and simultaneous measurements on our devices. The image runs the required kernel, configures the system for reaching our infrastructure, and includes vendor tools to perform platform measurements. In order to generate the CentOS 7 image, the livecd-creator tool is utilized on a separate CentOS 7-based system. This tool uses Anaconda, Kickstart, and Lorax to generate the image. The following steps are performed:

7. Install the latest *livecd-tools* package, preferably built directly from the project GitHub repository.

8. Create your own kickstart file or use the kickstart that will be provided by this project as a basis for your own. In our kickstart, we will insert commands to install required dependencies of our vendor products. Your environment will require further configuration to include networking, host file modification, and user management. You will also need to adjust hostnames and IP addresses to fit your environment.

9. Some tools, such as required drivers, were installed into a local repository (repo) on the image generating system using the `createrepo` command. This repo can be accessed by kickstart during the image generation. Copy *HIRS_Provisioner_TPM_2_0-X.X.X.x86_64.rpm* and *paccor-X.X.X-X.noarch.rpm* into the newly created repository.

```
$ createrepo -u file:///sca-packages sca-packages
```

10. Generate the ISO image from the kickstart file.

```
$ livecd-creator –-config=kickstart-filename.ks
```

358 11. The ISO file will be created in the local directory with a filename indicating the time of
359   generation. Once this is done, the *pxeboot* directory can be generated:

360   ```
         $ livecd-iso-to-pxeboot imagename.iso
         ```

361 12. The *pxeboot* directory will be created, containing the required *vmlinuz* and *initrd0.img* files. It
362   will also create a directory named *pxelinux.cfg* which contains a file named *default*. *default*
363   contains the kernel flags necessary to boot the image. Use these files in the PXE environment
364   detailed in Section 2.1.1.3.

365 ### 2.1.1.1.4 CentOS 8 Image Creation
366 Before continuing with CentOS 8 image creation, create the prerequisite files in Section 2.6. This set of
367 procedures creates an acceptance testing environment similar to what is described in Section 2.1.1.1.3
368 with the following deviations:

369 13. In Step 2, retrieve the CentOS 8 kickstart file *(Integration-Scripts\Acceptance Testing*
370   *Environment Build Scripts\HPE PCVT - Centos8\HPE - Centos8.ks)* from the project repository.

371 14. In Step 3, retrieve the latest version of the Java 11 Java Development Kit (JDK). This
372   demonstration uses Azul Zulu build, but other builds may also work. Additionally, create a folder
373   named `HPE Tooling` in your working directory. Copy the provisioning scripts *(Integration-*
374   *Scripts\Manufacturer-specific Scripts and Tools\HPE Tooling)* from our repository into the
375   directory as well as the HPE Platform Certificate Verification Tool (PCVT) binaries built in Section
376   2.6.

377 15. Complete the remaining steps as documented.

378 ## 2.1.1.2 Windows-Based Acceptance Testing Image Creation
379 The following procedures will produce a Windows Preinstallation Environment (WinPE) bootable image
380 that can be used in computing device acceptance testing. You will need to have a Windows Server (2016
381 or above) environment available to complete the following steps.

382 ### 2.1.1.2.1 Build WinPE
383 1. Download and install the Windows Assessment and Deployment Kit (ADK) and WinPE add-on.

384 2. Download the Dell EMC iDRAC Tools for Microsoft WinPE (R), v10.1.0.0 software package.

385 3. Run the self-extractor and choose all defaults.

386 4. Launch cmd.exe as an administrator and change directory to the extracted folder, then run our
387   modified batch file (WinPE10.x_driverinst - ps1.bat).

388

389   5.   If successful, the preceding batch script will create a folder in the same directory with a name
390   similar to *WINPE10.x-%timestamp%* or *WINPE5.x-%timestamp%.*

391

2.1.1.3 Preboot Execution Environment (PXE)

392

2.1.1.3.1   Dynamic Host Configuration Protocol (DHCP) Proxy
393

394   In this prototype demonstration, we use a combination of DNSMasq and the iPXE project to deliver the
395   acceptance testing capabilities to computing devices. DNSMasq provides network boot services via
396   DHCP on a network that already has other DHCP services present, such as assigning IP addresses to
397   hosts. Since our network used DHCP services that could not easily be modified for network boot, we
398   made the design decision to use DNSMasq as a proxy. However, for your setup you may want to include
399   network boot services directly into the DHCP product that is used in your environment.

400   The iPXE project provides open-source network boot firmware. Using iPXE enabled a script-based boot
401   process from an HTTP server. We also chainload the iPXE boot process from a Trivial File Transfer
402   Protocol (TFTP) server, avoiding the need to replace the network card firmware with an iPXE client.

403   The system specification and procedures follow below. Note that this project uses computing devices
404   that support Unified Extensible Firmware Interface (UEFI) booting and does not support legacy personal
405   computer (PC) Basic Input/Output System (BIOS) booting. Table 2-1 shows the system information used
406   in our prototype demonstration.

407 **Table 2-1 DHCP Proxy System Information**

Operating System	Version	Platform
Ubuntu Server	Release 20.04	Virtual Machine

408 6. Install DNSMasq, the TFTP server, and the HTTP server using the software package manager of
409   your chosen operating system (OS). On Ubuntu, use the following command.

410
```
$ apt install dnsmasq tftpd-hpa apache2
```

411 7. Create a custom iPXE bootloader that directs iPXE to boot from a fixed URL.

412   a. Create a file named *embed.ipxe* with the following contents.

413
```
#!ipxe
```
414
```
dhcp
```
415
```
chain http://<IP or Hostname>/ipxe/boot.ipxe || shell
```

416   b. [Download](#) and extract the iPXE source files. Install all software dependencies noted on
417   the download page.

418   c. Change directory to *ipxe/src* and run the following command.

419
```
$ make bin-x86_64-efi/ipxe.efi EMBED=/path/to/embed.ipxe
```

420 8. Copy the newly built iPXE efi boot file to */var/lib/tftpboot*.

421 9. Edit the DNSMasq configuration file to suit your environment.

422   a. `$ [your favorite editor] /etc/dnsmasq.conf`

423   b. Ensure the following configuration variables are set in the configuration file:

424
```
pxe-service=x86-64_efi,"Network Boot EFI",ipxe.efi
```
425
```
enable-tftp
```
426
```
tftp-root=/var/lib/tftpboot
```

427 10. Restart DNSMasq.

428
```
$ systemctl restart dnsmasq
```

429 11. Copy the WinPE, CentOS 7, and CentOS 8 images to the HTTP server.

430   a. In the root of your HTTP server, create two directories to store the images.

431
```
$ mkdir -p images/winpe images/centos7
```

432   b.   Copy the */media* directory created in Section 2.1.1.2.1 to *images/winpe*.

433   c.   Copy *initrd.img* and *vmlinuz* created in Section 2.1.1.1.2 to *images/centos7*.

434   d.   Copy *initrd.img* and *vmlinuz* created in Section 2.1.1.1.4 to *images/centos8*.

435   e.   Download the latest wimboot binary from the iPXE repository and store it in the *images*
436   directory.

437 12. Create a directory named *ipxe* in the HTTP server root, and copy the *boot.ipxe* file supplied by
438   this project's repository to this location. Consider our configuration file as a starting point and
439   ensure the contents of this file match your environment. Errors may result in a non-functioning
440   network boot service.

2.1.2 Platform Manifest Correlation System (PMCS)

441

442 The PMCS is custom software that allows original equipment manufacturer (OEM) platform manifests
443 (post-acceptance testing) to be translated into a format that is suitable for the Asset Discovery and
444 Repository System (Archer Integrated Risk Management [IRM]). The system provides a web user
445 interface (UI) for the IT administrator, and representational state transfer (REST) application
446 programming interfaces (APIs) are provided for programmatic access. The following steps will set up the
447 environment.

448 13. The system is based on Node.js, an open-source JavaScript runtime built on Chrome's V8
449   JavaScript engine designed to build scalable network applications. Download and install Node.js
450   on a system best suited for your environment. This demonstration uses an Ubuntu 20.04.2 LTS
451   virtual machine.

452 14. Install the node package manager (npm).

453 15. Install git on the platform chosen in Step 1. Git provides source code management capabilities
454   used in later steps.

455 16. Install Process Manager 2 (PM2). This package will manage the Node.js processes that run the
456   PMCS codebase.

457
```
$ npm install pm2 -g
```

458 17. Start the application using *pm2* from the cloned copy of the project repository:

459
```
$ cd platform-manifest-collation-system
```

460
```
$ pm2 start index.js
```

461 The PMCS should now be running as a background process. Consider using a startup script to keep your
462 process list intact across expected or unexpected machine restarts.

2.2   Dell

2.2.1   Laptops

The following section describes how to prepare Dell laptops for acceptance testing and continuous monitoring scenarios. Note that the Dell Trusted Device agent requires access to the Dell cloud. Consult the Dell website to determine the ports and IP addresses. Additionally, download the custom scripts for the scheduled tasks from our repository and store them on each target Dell laptop. In this demonstration, we chose *c:\Dell\HIRS* and *c:\Dell\TrustedDevice*.

2.2.1.1   Extract the Platform Certificate

Perform the following preparatory steps to create an acceptance testing environment suitable for Dell laptops. Contact your Dell representative to ensure the target laptop has been provisioned with a Platform Certificate from the factory.

18. Boot the target Dell laptop to the Windows 10 environment.

19. Start cmd.exe as an Administrator and run the following command:

    ```
    mountvol o: /s
    ```

20. Copy *o:\EFI\tcg\cert\platform\Dell.[Line of Business].[Servicetag].ver2.Base.cer* to a system with a text editor available. Note that *Line of Business* and *Servicetag* will be specific to your laptop.

21. Separate the Platform Certificate from the signing certificate:

    a. Cut the signing certificate out of the file and save the Platform Certificate.

        ```
        -----BEGIN CERTIFICATE-----<cert content>  -----END CERTIFICATE-----
        ```

        ```
        {Ctrl} + X
        ```

        ```
        {Ctrl} + S
        ```

    b. Create a new file and save it as the signing certificate.

        ```
        {Ctrl} + N
        ```

        ```
        {Ctrl} + V
        ```

        ```
        {Ctrl} + S
        ```

    c. Name the signing certificate.

        ```
        <HSM-Signing-Certificate.cer>
        ```

22. Create a dedicated CentOS 7 host for running the HIRS Attestation Certificate Authority (ACA) portal that is accessible to the computing device undergoing acceptance testing. This step is detailed in Section 2.4.

DRAFT

493   23. Create a network bootable CentOS 7 image. This step is detailed in <u>Section 2.1.1</u>.

494   Note that to perform acceptance testing with Dell laptops, two settings in the BIOS are modified:

495   24. Power-on the laptop and boot to the BIOS setup by pressing the Function 2 (F2) key.

496   25. Clear the TPM to remove Windows ownership of the device. Navigate to *Security > TPM 2.0*
497   *Security > Clear* in the main menu. Click the *Clear* radio box and select **Yes** in the dialog box.

498   26. Turn off *Secure Boot*. Navigate *Secure Boot > Secure Boot Enable* in the main menu. Click the
499   *Clear* radio box and select **Yes** in the dialog box.

500   27. Reboot the laptop by clicking **Apply** and **Yes** in the dialog box followed by **Exit.**

501   ## 2.2.1.2 Install the Dell Trusted Device Agent

502   General installation instructions are posted on the Dell website. Below, we use the interactive graphical
503   installation wizard, but other **deployment options** are also available.

504   28. Download the latest version of the Dell Trusted Agent from the Dell <u>website</u>.

505   29. Open a command prompt as an Administrator. Install the agent with the following command:

506   ```
msiexec.exe /i Trusted-Device-<version>\Win64R\TrustedDevice-64bit.msi
```

507    30. An installation wizard will launch. Click **Next** and then the **Install** button. The installation
508        package will warn that the laptop will require a reboot. Accept the warning.

509    31. Follow the prompt to reboot the laptop. After the reboot, check the installation by manually
510        launching the agent. If successful, a browser window will launch with a message similar to the
511        following.

NIST SP 1800-34C: Validating the Integrity of Computing Devices                                    13

512 ### 2.2.1.3  Create the Scheduled Tasks

513 These procedures will create two tasks that periodically execute our custom scripts, which silently
514 launch the Dell Trusted Device (DTD) agent/HIRS Provisioner Agent and detect platform integrity issues.

515     32. Open the Task Scheduler as an Administrator on the target laptop.

516     33. Select **Action > Create New Task.**

517     34. In the **General** tab, enter a name for the task in the **Name** field. Click the **Change User or Group**
518         button and select the *System* account. Select *Windows 10* from the **Configure for** pull-down
519         menu.

520    35. In the Triggers tab, click the **New**… button. Select a scheduled time appropriate for your
521        environment. Once per day is shown in the example below.

522  36. In the Action tab, click the **New…** button. Enter *powershell.exe* in the Program/script field. Enter
523      *-file "C:\Dell\HIRS\hirs_script.ps1"* in the **Add arguments (optional)** field. Adjust this value if
524      needed if the custom script is installed in a different location. Click the **OK** button.

525  37. Click the **OK** button to save the new scheduled task.

526  Repeat this section to create a scheduled task that will periodically execute the Dell Trusted Device
527  agent using the custom script.

## 2.2.2 Servers

529  The Dell R650 used in this demonstration does not require any preparatory activities for acceptance
530  testing. All platform validation tools are included in the network-booted acceptance testing
531  environment. Continue with creating the WinPE acceptance testing environment as described in
.

## 2.3 Eclypsium

534  Eclypsium is a firmware security solution with cloud-based and on-premises deployment options. It
535  secures firmware in servers, endpoints, and network devices by:

536    ▪    identifying devices that contain firmware and creating detailed profiles of each component;

537    ▪    verifying these profiles are free of vulnerabilities, have maintained their integrity, and are
538         properly configured; and

539    ▪    fortifying device firmware through a combination of configuration hardening, automated
540         updates, and packaged guidance.

541    For this demonstration, Eclypsium is leveraged in the acceptance testing and continuous monitoring
542    scenarios. The procedures below will install the Eclypsium agent and continuously monitor Windows-
543    based laptops and Linux-based servers. In the server use case, we configured the agent to communicate
544    with the on-premises deployment of the Eclypsium analytic backend. Refer to Section 3 in NIST SP 1800-
545    31C for installation procedures.

## 2.3.1  Download Eclypsium Agent

547    1.  Navigate to the **Eclypsium Management Console** in a web browser.



548    2.  Select **Deployment > Download.**

549    3.  Download the installer for the appropriate OS (Windows, macOS, Linux (Deb), or Linux (RPM)).

## 2.3.2  Install Eclypsium Agent for Windows

551    4.  Start the Eclypsium bundled installer, *Eclypsium-<version>.exe*.

552      5.   Select **Next**.

553      6.   Ensure **Register with Eclypsium Analytics Service** and **Enable Service for Monitoring** are
554          selected. Enter the **Domain** and Registration **Token** that can be found on the Download page of
555          the **Eclypsium Management Console**, then select **Next.**



556      7.   Select **Install** to start the Eclypsium installation**.**

557      8.   When prompted, select **Finish.**

558      9.   The Eclypsium agent has successfully installed once the page depicted below is reached. Select
559          **Close.**

Installation Successfully Completed

560 When the system scan completes on a newly installed system, the Eclypsium console will identify supply
561 chain integrity concerns and recommend a resolution.

### 2.3.3 Install Eclypsium Agent for Linux

563 1. Ensure the *App* and *Driver* installation packages that are appropriate for your distribution are
564    available on the host server system. The example below is an Ubuntu distribution.

565 2. Install the packages with the following command with root privileges. Note that there may be
566    prerequisite packages that are required before installing the Eclypsium packages.

567
```
dpkg -i eclypsiumapp-2.8.1.deb eclypsiumdriver-2.8.1.deb
```

568 3. Register the Eclypsium agent with the on-premises backend with the following command with
569    root privileges.

570
```
EclypsiumApp -s2 <Eclypsium-backend-hostname> reg_<token>
```

571 If successful, the server is registered and an initial scan is performed. The output should be similar to the
572 following.

573
574
```
Scan data dumped to '/home/<user>/<hostname>-21ee761e90f38bb0-2022-05-
09T12_26_27Z.tar.gz'
```

575
576
```
Basic info updated successfully. Check the device at https://<backend-
hostname>/resolve-job/6279087374e1ae0726c3d68f
```

577
```
Successful registration.
```

578
```
[Dumping system firmware through SPI] \ 16777KB
```

579
```
[Dumping system firmware through MMIO] / 16777KB
```

580

```
[Uploaded 100%] [############################] 12999KB/12999KB
```

```
Scan data dumped to '/home/<user>/<hostname>-21ee761e90f38bb0-2022-05-
09T12_26_27Z.tar.gz'
```

```
Scan data updated successfully. Check the device at <backend-hostname>/resolve-
job/627908e374e1ae3a06c3d800
```

2.4 Host Integrity at Runtime and Start-Up (HIRS) Attestation Certificate Authority (ACA)

This section describes the installation and configuration of the HIRS-ACA backend components used in the acceptance testing scenario. HIRS-ACA is an open-source tool with three components that are used in this demonstration – the Attestation Certificate Authority, dashboard, and provisioner. The ACA issues identity credentials to devices that have a TPM 2.0 security module; these credentials are requested by the provisioner software. The HIRS-ACA dashboard is available to administrators to view and configure validation reports, credentials, and certificate trust chains. Table 2-2 shows the system information used in our prototype demonstration.

Table 2-2 HIRS-ACA System Information

Operating System	Version	Platform
Centos	7	Virtual Machine

2.4.1 Installing the HIRS-ACA

4. Before installing the required packages, ensure the target system has a fully qualified distinguished hostname. Modify the */etc/hosts, /etc/hostname,* and */etc/resolv.conf* system configuration files as appropriate.

600 5. Install the HIRS-ACA dependencies using the following command. This will install
601   MySQL/MariaDB, OpenSSL, Tomcat, Java, RPM Dev Tools, GNU Core Utilities, and other Linux
602   commands (initscripts, chkconfig, sed, grep, firewalld, and policycoreutils).

```
603     # sudo yum install mariadb-server openssl tomcat java-1.8.0 rpmdevtools
604     coreutils initscripts chkconfig sed grep firewalld policycoreutils
```

605 6. Download the latest version of HIRS ACA from the Release page on GitHub and execute the
606   following command to install the HIRS ACA.

```
607     # sudo yum install HIRS_AttestationCA*.rpm
```

608 Ensure the installation was successful by navigating to the dashboard using the fully qualified domain
609 name (FQDN) configured above. It should look like the screenshot below.

2.5 HP Inc.

611 The following steps install the HP Client Management Script Library (CMSL) and execute prerequisite
612 provisioning for HP Inc. laptops. The CMSL installs several PowerShell commands on the laptop that will
613 assist in platform validation. Once CMSL is installed, an administrator configures the HP Inc. specific
614 device security feature. In this prototype demonstration, the target computing devices were an HP Inc.
615 Elitebook 840 G7 and Zbook Firefly 14 G7.

2.5.1.1 Install the HP CMSL

617 7. Download the latest CSML from the HP Developers website onto the target HP Inc. laptop.

DRAFT

618   8. Launch the executable file and proceed through the wizard. Accept the agreement and click
619   **Next.**

620   9. Select **Install into PowerShell** path and click **Next.**

621   10. Click **Install.**

622   11. Click **Finish.**

623   12. Test the installation by opening PowerShell as an administrator and executing a CMSL command
624   such as `Get-HPBIOSVersion`.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
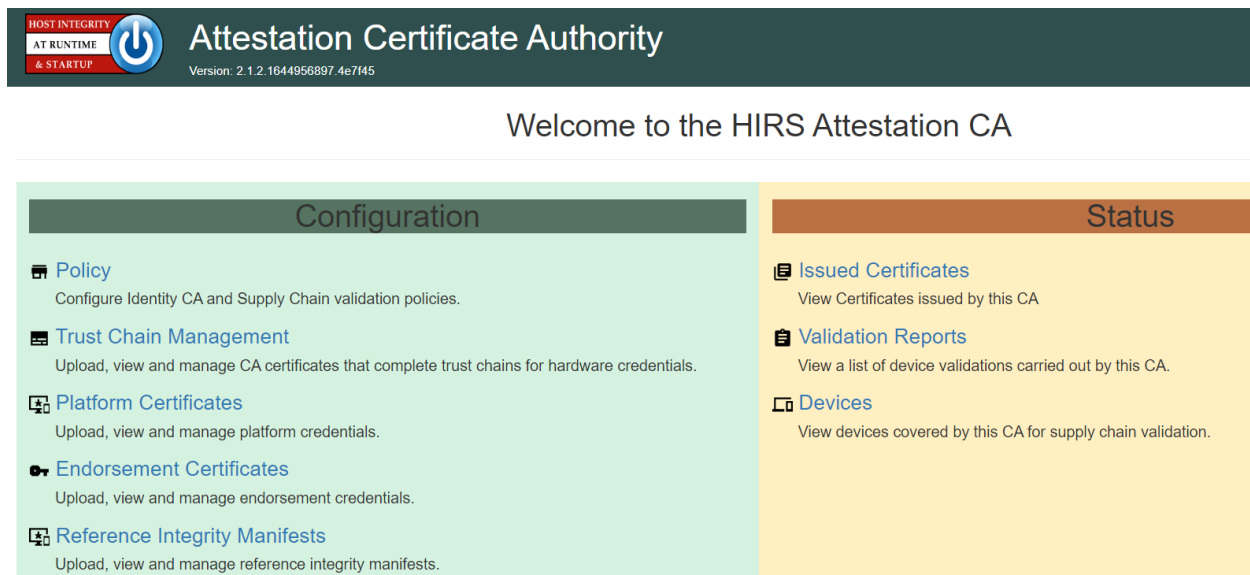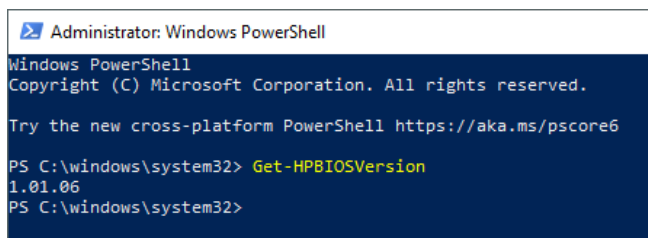
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\windows\system32> Get-HPBIOSVersion
1.01.06
PS C:\windows\system32>
```

2.5.1.2 Execute Provisioning Steps

626 The next steps are used to provision the HP Inc. specific firmware and device security features, HP Sure
627 Start, HP Sure Admin, HP Tamperlock, and HP Sure Recover. Implementers may also want to consult the
628 HP Inc. Developers Blog for [more](#) [information](#) on how these payloads were created. Using the example
629 provisioning payloads available from our project repository, use the CMSL to apply the six provisioning
630 payloads as shown below:

631   13. Open PowerShell as an administrative user. Execute the following commands.

632   `Set-HPSecurePlatformPayload -PayloadFile EKProvisionPayload.dat`

633   `Set-HPSecurePlatformPayload -PayloadFile SKProvisionPayload.dat`

634   14. Reboot the laptop. A local administrator must accept the *Physical Presence Prompt* to complete
635   provisioning of the Endorsement and Signing Key.

636   15. Execute the following commands from PowerShell as an administrator.

637   `Set-HPSecurePlatformPayload -PayloadFile EnableEBAMPayload.dat`

638   `Set-HPSecurePlatformPayload -PayloadFile LAKProvisionPayload.dat`

639   16. Reboot the laptop. This will expose settings that require a BIOS administrator be configured
640   before the next step can be completed.

641   17. Execute the following commands from PowerShell as an administrator.

642   `Set-HPSecurePlatformPayload -PayloadFile BIOSsettingsPayloadFile.dat`

643   `Set-HPSecurePlatformPayload -PayloadFile SureRecoverProvision.dat`

2.6 Hewlett Packard Enterprise (HPE)

645 We demonstrate HPE's Platform Certificate Verification Tool (PCVT) in this project by creating a network
646 bootable acceptance testing environment which has PCVT tools and dependencies pre-installed on the
647 image. This image also includes a bash script which executes the PCVT command and, if successful,
648 uploads the hardware manifest to the PMCS.

649 First, compile the PCVT tools on a separate CentOS 8 system. The general procedures are on the HPE
650 GitHub site and our specific commands follow.

651   18. Download and extract the source code from the HPE repository.

652   19. Install the software prerequisites onto the system.

653   `yum -y install systemd-devel golang maven java-11-openjdk java-11-openjdk-devel`

654   20. Change directory into the PCVT source code. Run the following command:

```
655    mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
656    pcvt_v1.0.0/lib/HIRS_Utils-1.1.1.jar -DgroupId=HIRS_Utils -
657    DartifactId=HIRS_Utils -Dversion=1.1.1 -Dpackaging=jar -
658    DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
659    mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
660    pcvt_v1.0.0/lib/HIRS_Structs-1.1.1.jar -DgroupId=HIRS_Structs -
661    DartifactId=HIRS_Structs -Dversion=1.1.1 -Dpackaging=jar -
662    DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
663    mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
664    pcvt_v1.0.0/lib/paccor-1.1.3-2.jar -DgroupId=paccor -DartifactId=paccor -
665    Dversion=1.1.3-2 -Dpackaging=jar -
666    DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
```

667   21. Build the PCVT.

668   `mvn clean compile assembly:single`

669   22. Change to the **diskScan** directory.

670   23. Set the **GOPATH** to a local directory and set **GO11Module** to **off**.

671   `export GOPATH=$HOME/<local_path>/gowork`

672   `go env -w GO111MODULE=off`

673   24. Execute the build script in the **build** directory.

674   `./build/create_install_bundle.sh`

675 Ensure two files named **pcvt-mvn-0.0.1-jar-with-dependencies.jar** and **libdiskscan.so** are generated.
676 Next, the acceptance testing environment is built. Continue with the procedures documented in Section
677 2.1.1.1.4.

2.7 Intel

679 The Intel Transparent Supply Chain (TSC) requires two client applications to support acceptance testing
680 and continuous monitoring scenarios: **TSCVerifyUtil** and **AutoVerifyTool**. Contact your Intel
681 representative to download the installation packages for both utilities.

2.7.1 Laptops

683 Once the binaries have been retrieved, follow these procedures on the target laptop. Table 2-3 lists the
684 laptops used within this demonstration.

685 **Table 2-3 Intel-Contributed Laptops**

Machine Name	Operating System	Manufacturer	Model
intel-0	Windows 10	HP Inc.	Elitebook 360 830 G5
intel-1	Windows 10	Lenovo	ThinkPad T480

686 1. Download and install the latest Microsoft Visual C++ Redistributable for Visual Studio.

687 2. Launch the AutoVerifyTool installation wizard. Click **Next**.

688 3. Accept the license and client **Next**.

689   4. Enter your Name and Organization. Click **Next**.

690   5. Select the **Typical** installation. Click **Next**.

691   6. Click **Install**.

2.7.2 Servers

693   The server contributed by Intel requires the installation of the TSCVerifyUtil application. Contact your
694   Intel representative to determine the best method in your use case. In this prototype implementation,
695   we opted to execute TSCVerifyUtil from a directory created at */opt/intel/tsc*. Table 2-4 lists the server
696   contributed by Intel for this demonstration.

697   **Table 2-4 Intel-Contributed Server**

Machine Name	Operating System	Manufacturer	Model
intel-2	CentOS 8	Intel	S2600WTT Server Board

698   Additionally, to complete the implementation we connected the Seagate enclosure to this server board.
699   Refer to Section 2.9 for a description of this process.

2.8   Archer Integrated Risk Management (IRM)

This section describes the installation of the Archer IRM system for this demonstration. Our instantiation of Archer IRM is viable for a lab environment, but the reader is encouraged to refer to the architecture planning guide on the Archer IRM website for specific guidance for your environment. We elected to install the Archer IRM system across two virtual machines—one hosting a Microsoft SQL database and the other hosting the remainder of the Archer IRM services. Note that the screenshots below are from our original installation of Archer IRM 6.9. During the course of the project, we updated our Archer IRM instance to version 6.10. As a result, some screenshots may differ in your implementation from what is presented in this document.

Table 2-5 shows the system information used in this prototype demonstration for Archer IRM.

Table 2-5 Archer IRM System Information

Machine Name	Machine Type	Operating System
Archer Database Server	Virtual	Windows 2019 Server
Archer Services	Virtual	Windows 2019 Server

2.8.1   Prerequisites

Before installing Archer IRM services, several prerequisites must be fulfilled. In this section, we describe those prerequisites involving the database server and Microsoft's Internet Information Services (IIS) web server.

2.8.1.1   Install SQL Server on Database Server

1. Download SQL Server 2019 from https://www.microsoft.com/en-us/sql-server/sql-server-downloads onto the database server.

2. Run the SQL Server 2019 executable.

3. Select the **Custom** installation type.

720   4. Specify the download location and select **Install.**

721   5. Allow the installer to download the SQL Server 2019 package.

722   6. The SQL Server Installation Center should automatically open. From the left menu panel, select
723   **Installation.** Select the option **New SQL Server stand-alone installation or add features to an**
724   **existing installation**.

725   7. Enter the product key or select a free edition of the software. Then select **Next.**

726   8. Read and accept the License Terms. Then select **Next**.

727   9. Ensure that all the **Global Rules** have passed. Then select **Next**.

728   10. To use Microsoft Update to automatically deliver updates, check the box **Use Microsoft Update**
729   **to check for updates (recommended).** Then select **Next.**

730   11. Ensure that all the **Install Rules** have passed. Then select **Next.**

731   12. Select the desired features to install. Then select **Next.** Complete the sections for the selected
732   features.

733   13. In the **Instance Configuration** section, select the **Named instance** radio button and choose a
734   name for the database server, or select the **Default instance** radio button to use the default
735   name. Then select **Next.**

736   14. In the **Database Engine Configuration** section, select the desired Authentication Mode. Select
737   **Add Current User** to add the current user as a SQL Server administrator and select **Next.**

738   15. Ensure that all the **Feature Configuration Rules** have passed and select **Next.**

739   16. Confirm the selected settings are desired and select **Install.**

740   17. Once the installation completes, select **Close.**

2.8.1.2 Create the Archer IRM Databases

742   1. Download SQL Server Management Studio (SSMS) from https://aka.ms/ssmsfullsetup. Follow
743   the installation steps.

744   2. Once installed, open SSMS.

745   3. Expand the ARCHERSQLSERVER tree. Right-click on **Databases** and select **New Database.** Create
746   three databases: *ArcherInstanceDB, ArcherConfigurationDB,* and *ArcherLoggingDB.*

747   4. Next, create a local Administrator user. Right-click **Security** and select **New Login.**

748   5. Under the **General** tab, input the **Login Name** and select the **SQL Server Authentication** radio
749   button. Create a password for this user. These credentials will be used during the Archer IRM
750   installation.

751   6. Navigate to the **User Mapping** tab. Ensure all the databases have the **Default Schema** set to
752   **dbo.** Also, ensure that **db_owner** is selected for each database under the **Database role**
753   **membership** section. **Select OK.**

754 *2.8.1.3 Install Internet Information Services on the Web Server*

755   1. On the web server, open **Server Manager.**

756   2.   Under **Manage**, select **Add Roles and Features.**

757   3.   Select **Next.**

758   4.   Select the **Role-based or feature-based installation** radio button. Select **Next.**

759   5.   Select the **Web Server (IIS)** server role. Then select **Next.**

760   6.   In the pop-up window, select **Add Features.**

761   7.   Select **Next.**

DRAFT

762   8.   Select **Next.**

763   9. Ensure that the **Role Services** shown below are selected. Then select **Next.**

764   10. Confirm that the selected options are correct and select **Install.**

765   11. Once the installation completes, select **Close.**

766   12. Restart the computer.

767   *2.8.1.4 Configure IIS*

768   1. Open the IIS application.

769   2. Click on the web server in the left pane. **Select Authentication.**

770   3. Ensure that **Anonymous Authentication** is enabled and **ASP.NET Impersonation** and **Forms**
771   **Authentication** are disabled for the **Default Web Site.**

772   4. Expand the web server tree and select **Application Pools.** In the far-right pane, select **Add**
773   **Application Pool.**

774   5. Add a name to the **Name** input field. Ensure that **Managed pipeline mode** is set to **Integrated**
775   and that **Start application pool immediately** is selected. Then, select **OK.**

776   6. Right-click on the newly created application pool and select **Advanced Settings.** Under **Process**
777   **Model,** select the ellipsis button that is next to the **Identity** field.

778   7. Select **Custom account,** select **Set,** and enter the appropriate information. Then select **OK.**

779   8. Click on the web server. In the far-right pane, select **Restart.**

780   9. Open a browser and navigate to localhost. If the screen below is shown, then the web server is
781   running properly, and Archer IRM can now be installed.

DRAFT

2.8.2 Archer IRM Installation

1. Before installing Archer IRM, .NET Framework version 4.7.2 must be installed. It can be downloaded at https://dotnet.microsoft.com/download/dotnet-framework/net472.

2. Extract the zip file that was downloaded from the Archer IRM download page.

3. Open the folder and run the executable **ArcherInstall.**

4. Accept the License Agreement and select **Next.**

5. Select **Next.**

6. For the web server, make sure the components **Web Application, Services,** and **Instance Database** are selected, then select **Next.**

791   7. Select **Create a certificate** from the dropdown menu and select **Next.**

792 8. Select the database server that was previously created. Enter the credentials that were created
793   in SSMS. Then select the configuration database from the dropdown menu and click **Next.**

794   9. Select the preferred language from the dropdown menu and select **Next.**

795   10. Repeat step 8 and select the instance database from the dropdown menu. Then select **Next.**

796   11. Select the time zone and select **Next.**

797   12. Select **Default Web Site** as the website location and choose the **Install an IIS application** radio
798   button. Select **RSAarcher** from the dropdown menu. Then select **Next.**

799   13. To add an Instrumentation Database, repeat step 8 and use the **ArcherLogging** database that
800   was created in SSMS. Otherwise, select **Not using Archer IRM Instrumentation service.** Select
801   **Next.**

802   14. Specify the account to run the services. Then select **Next.**

803   15. Confirm or edit the installation paths for the services and application files. Select the **Create**
804   **Archer IRM program group for all users** radio button. Then select **Next.**

805   16. Confirm or edit the path for installation logs. Then select **Next.**

806   17. Select **Install** and wait for the installation to complete. Once completed, select **Finish.**

807   *2.8.2.1 Configure Options in the Control Panel*

808   1. Open the RSA Control Panel.

809   2. In the left pane, select **Add New Instance.**

810   3. Enter a name for the instance in the **Instance Name** field. Select **Go.**

811   4. Double-click on the new instance. Input the required information in the **General**, **Web**, and
812   **Database** tabs. When completed, click **Save** in the top left corner.

813   *2.8.2.2   Add New Application to Application Pool*

814   1.   Navigate back to IIS. Expand the web server directory, expand the **Sites** directory, and expand
815   the **Default Web Site** directory.

816   2.   Select the RSAarcher site. Click on **Authentication** and ensure that **Anonymous Authentication**
817   is the only thing that is enabled.

818   3.   Right-click on the RSAarcher site and select **Manage Application > Advanced Settings.**

819   4.   Click on **Application Pool** and select the ellipsis button. You will see a screen similar to the
820   following:

821   5.   Select the application pool that was previously created and select **OK.**

822   6.   Select **OK.** You should see something similar to the screenshot below:

823   7.   Restart the Archer IRM site.

824   8.   Open a browser and navigate to the URL that was set in the RSA Control Panel application. If the
825   following page displays, then Archer IRM installed successfully.

2.9 Seagate

827 Seagate contributed three hard drives (Table 2-6) stored within a 2U12 enclosure. As described in
828 Section 2.7.2, the enclosure is connected to our demonstration Intel server via a Serial Attached SCSI
829 (SAS) interface. The demonstration server did not have the required SAS interface, so we purchased a
830 Broadcom 9500-8e Tri-Mode Storage Adapter to complete the connection.

831 **Table 2-6 Seagate Hardware Contribution**

Machine Name	Operating System	Manufacturer	Model
N/A	N/A	Seagate	Exos 18TB Self Encrypting Hard Disk Drive x 3
N/A	N/A	Seagate	Exos E 2U12 Rackmount Enclosure

832 Once the enclosure is connected to the server, power on the server into the native Linux environment.
833 Execute the **lshw** command which prints detailed hardware information about the server. The output
834 should resemble the following for one of the Seagate drives. Note that because these are SAS drives
835 there are two paths to the drive. As a result, you will notice two */dev/sdx* devices pointing to the same
836 physical drive.

```
837     *-disk:0
838                     description: SCSI Disk
839                     product: ST18000NM005J
840                     vendor: SEAGATE
841                     physical id: 0.0.0
842                     bus info: scsi@0:0.0.0
843                     logical name: /dev/sdb
844                     version: ET02
```

```
845                              serial: ZR5056HD0000C107GP5G
846                              size: 16TiB (18TB)
847                              capacity: 45TiB (50TB)
848                              capabilities: 7200rpm
849                              configuration: ansiversion=7 logicalsectorsize=512
850            sectorsize=4096
```

851 Additionally, we recommend using Seagate's command line interface tool that communicates with the

852 drives via the Trusted Computing Group (TCG) Storage API to confirm successful integration. Use the

853 following command to print drive information:

854
```
python3 sed_cli.py --device=/dev/sdb --operation=printdriveinfo
```

2.10 IBM QRadar

855

856 This section describes the installation of the IBM QRadar system for this demonstration. Our

857 instantiation of IBM QRadar is viable for a lab environment, but the reader is encouraged to refer to the

858 architecture planning guide on the IBM website for specific guidance for your environment.

859 We opted to install the full IBM QRadar suite onto a single virtual machine via an ISO provided by the

860 IBM engineering team. Note that Red Hat Enterprise Linux Server V7.6 (or binary equivalent) must be

861 deployed on the virtual machine before the QRadar installation. Once this prerequisite is met, boot the

862 virtual machine using the ISO provided by IBM. This process will be unique to your environment. Next,

863 follow the instructions provided by the IBM documentation website. The remainder of this section

864 includes example screenshots from the installation wizard we used in our environment.

865   1. Select the **Software Install** option for the appliance type.

866   2. For the functionality, select **"All-In-One" Console.**

867   3. Select **Normal Setup (default)** as the type of setup.

868   4. Either manually adjust the date and time, or add the name or IP address of a Network Time
869   Protocol (NTP) server to automatically update the date and time.

870   5. Select the appropriate time zone.

871   6. Select the appropriate network adapter that will allow communication with the installed system.

872   7. Enter the network information for this installation. Note that only static addresses are
873   supported.

874   8. Set the Admin user password.

875   9. Set the Root password for console access.

2.10.1 WinCollect Agent

877   On a separate Windows Server system, configure and install the WinCollect agent. This component polls
878   the remote hosts (laptops), and then sends event information to QRadar.

879   1. Install the WinCollect application on the QRadar system if not already present or upgrade to the
880   latest version. This process is documented on the IBM website.

881   2. Create an authentication token so that the managed WinCollect agents can exchange data with
882   QRadar appliances. This process is documented on the IBM website. Note that you will not be
883   able to retrieve the token from QRadar after it has been created.

Authorized Service Created Successfully

The authorized service has been created successfully.

| ********_****_****_****_************ | 👁 📋 |

The authorized service token cannot be made visible after you close this dialog.
Copy the token to a secure location for storage before you close this dialog.

Close

884   3. Configure a forwarding destination host for the log source data. This process is documented on
885   the IBM website. Enter the appropriate values for your environment.

WinCollect Forwarding Destination Properties

All fields are required

Destination Details

Name	qradar
Host Name	qradar
Port	514
Protocol	UDP
Throttle (events per second)	5,000

Store and Forward Options

Schedule Mode	Forward Events
Schedule(s)	New...

Save   Cancel

886   4. Install the managed WinCollect agent on the Windows Server host. This process is documented
887   on the IBM [website](). If successful, the agent will appear in the QRadar console under **Admin >**
888   **Data Sources > WinCollect > Agents.**

889 **2.11 Integrations**

890 This section describes the steps we took to configure and integrate the products described earlier in this
891 volume. The integrations are generally network-based and require connectivity both between the
892 systems and to Internet-based cloud services.

893 **2.11.1 Microsoft Endpoint Configuration Manager and Platform Validation Tools**

894 For the Intel laptops, a command-line version of the AutoVerify tool named TSCVerifyUtil periodically
895 monitors the changes to laptop components. A custom PowerShell script installed on each laptop and
896 run every hour via task scheduler captures the result of TSCVerifyUtil execution and stores it in the
897 Windows registry. This section describes how to configure Microsoft Endpoint Configuration Manager to
898 run a configuration baseline which monitors the results of the customized PowerShell script. This data is
899 reflected in the Archer IRM dashboard.

900 Similarly for HP Inc. and Dell laptops, the HIRS-ACA Windows-based Provisioner periodically monitors
901 the changes to laptop components. We chose to use the same monitoring approach for consistency –
902 the Windows task scheduler captures the result of the Provisioner execution and stores it in the
903 Windows registry. Repeat this section to configure Microsoft Endpoint Configuration Manager with the
904 HIRS Provisioner, changing input where noted.

905 *2.11.1.1 Set Up Configuration Item*

906   1. In the Microsoft Endpoint Configuration Manager console, under **Assets and Compliance >**
907   **Overview**, select **Compliance Settings**.

908   2. Next, select **Configuration Items**.

909   3. From the **Home** panel at the top, select **Create Configuration Item**.

910   4. Enter a name and description for the configuration item in the **Name** and **Description** fields.
911   Ensure that **Windows Desktops and Servers (custom)** is selected. Then select **Next.**

912   5.   Ensure that all versions are selected and click **Next.**

913   6.   On the **Settings** tab, select **New**.

914   7.   On the **General** tab, enter a name and description in the **Name** and **Description** fields. For
915   **Setting type**, select **Registry value** from the dropdown. For **Data type**, selection **String** from the
916   dropdown. To specify the registry value, select the appropriate **Hive Name** and enter the **Key**
917   **Name** and **Value Name** in their respective fields (Note: When configuring the HIRS Provisioner,
918   use `SOFTWARE\HIRS\provisioner` as the **Key Name**). Next, switch to the **Compliance Rules** tab.

919   8.   Select **New.**

920   9.   Specify the name and description for the rule in the **Name** and **Description** fields. For **Rule type**,
921   select **Value** from the dropdown. Under **The setting must comply with the following rule**, select
922   **Registry Value** and **Equals**, and enter 0 (zero) in **the following values:** field. Ensure that **Report**
923   **noncompliance if this setting instance is not found** is selected. Choose the **Noncompliance**
924   **severity for reports** appropriate for your environment. Then select **OK**.

925   10. Select **Apply**. Then select **OK**.

926   11. Review the configurations on the Summary page. After confirming that the configurations are
927   correct, select **Next**.

928   12. After the wizard completes, select **Close.**

2.11.1.2 Set Up Configuration Baseline

930   1. In the Microsoft Endpoint Configuration Manager console, under **Assets and Compliance >**
931   **Overview**, select **Compliance Settings**.

932   2. Next, select **Configuration Baselines**.

933   3. From the **Home** panel at the top, select **Create Configuration Baseline**.

934   4. Provide a name and description for the configuration baseline in the **Name** and **Description**
935   fields. Next, select **Add** and choose **Configuration Items**.

936   5. Select the previously created configuration item from the list and select **Add**.

937   6. Select **OK**.

938　　　7. Select **OK**.

939   *2.11.1.3 Set Up Registry Entry on Intel Devices*

940   1. On the Windows 10 laptop, go to **Start**, search for the **Registry Editor,** and open that program.

941   2. Find the Intel folder located in **HKEY_LOCAL_MACHINE\SOFTWARE**. Right click and select **New >**
942   **Key**. Name the key **TSCVerify**.

943   3. Select the **TSCVerify** key, right-click and select **New > String Value**.

944   4. Enter *Return Value* in the **Name** field.

2.11.1.4 Run Script Via Task Manager

945

946   1.   Place the script onto the local machine (snippet shown below). A copy of this script can be
947   obtained from our repository.

```
948    # Run Scan and capture exit code.
949    # 0=No components have changed and platform certificate validation passed
950    # 1=At least one component has changed OR platform certificate validation
951    failed
952    # 2=At least one component has changed AND Platform Certificate validation
953    failed
954
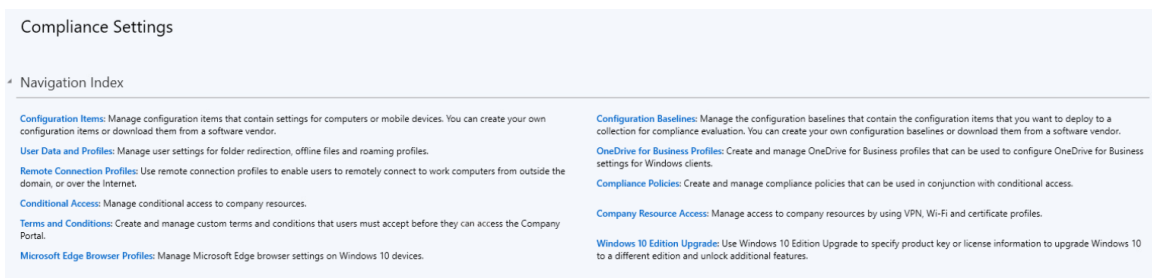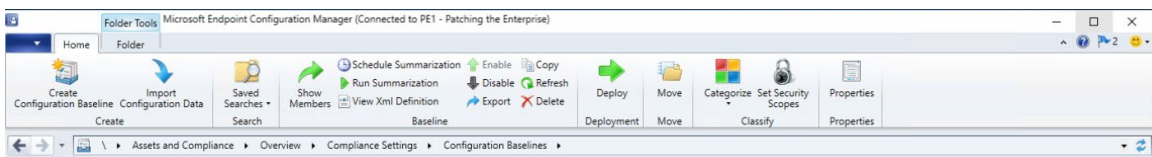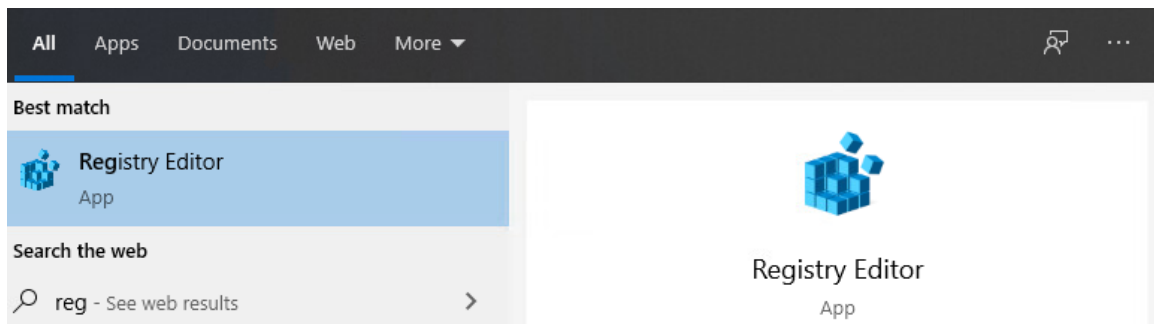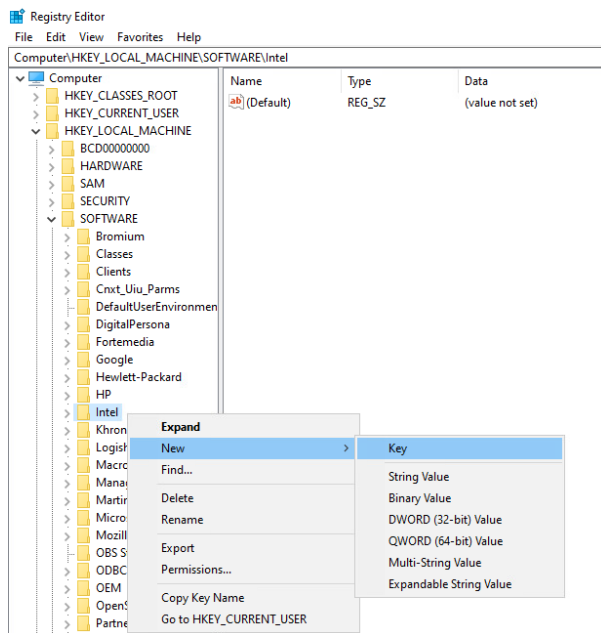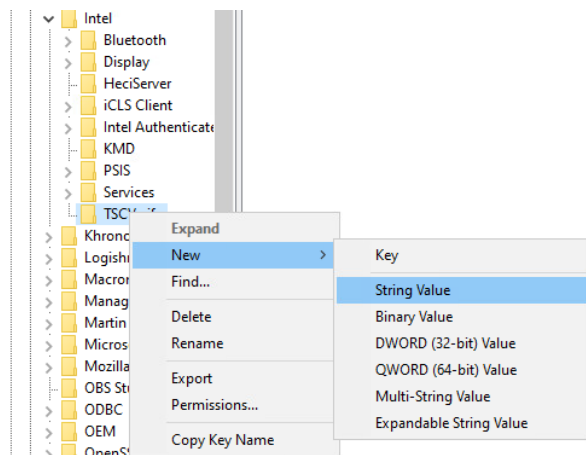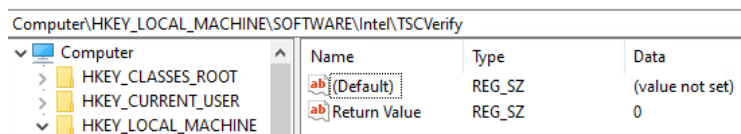
955    # Write-Output "Starting DPD file scan and compare..."
956    $tscpinfo = New-Object System.Diagnostics.ProcessStartInfo
957    $tscpinfo.FileName = "TSCVerifyTool_3.40.exe"
958    $tscpinfo.WorkingDirectory = $artifactdirectory
959    $tscpinfo.RedirectStandardError = $true
960    $tscpinfo.RedirectStandardOutput = $true
961    $tscpinfo.UseShellExecute = $false
962    $tscpinfo.Arguments = "SCANREADCOMP -in $dpdfile"
963    $dpdprocess = New-Object System.Diagnostics.Process
964    $dpdprocess.StartInfo = $tscpinfo
965    $dpdprocess.Start() | Out-Null
966    $stdout = $dpdprocess.StandardOutput.ReadToEnd()
967    $dpdprocess.WaitForExit()
968

969    # Write-Output "Starting Platform Certificate validation ..."
970    $tscpinfo.Arguments = "PFORMCRTCOMP -in $platformcertificatefile"
971    $platformcertprocess = New-Object System.Diagnostics.Process
972    $platformcertprocess.StartInfo = $tscpinfo
973    $platformcertprocess.Start() | Out-Null
974    $stdout = $platformcertprocess.StandardOutput.ReadToEnd()
975    $platformcertprocess.WaitForExit()
976

977    # If the return value is nonzero, then the computer is not compliant
978    $retValue = $dpdprocess.ExitCode + $platformcertprocess.ExitCode
979    Write-Output $retValue
980

981    # Add retValue to registry location
982    $regPath = "HKLM:\SOFTWARE\Intel\TSCVerify"
983    Set-ItemProperty -Path $regPath -Name "Return Value" -Value $retValue
```

984   2.   From the **Start Menu**, search for **Task Scheduler** and open the program.

985   3.   Under the **Actions** panel, select **Create Basic Task**.

DRAFT

986   4. Fill in the **Name** and **Description** fields. Then select **Next**.

987   5. Select the frequency for this task to run. Then select **Next**.

988   6. Select the start date and time for the task. Then select **Next**.

989   7. Select the action **Start a program**. Then select **Next**.

990   8. In the **Start a program** section, type the following in the **Program/script** field: *powershell.exe*.
991   Next, add the following to the add arguments (optional) field: *-file "<Location of script>"*. Then
992   select **Next**.

993   9. Confirm the settings are correct and select **Finish**.

994   10. On the main page of Task Scheduler, select the newly created task, right-click it, and select
995   **Properties**.

996   11. On the **General** tab, under **Security Options**, change the user to **SYSTEM**. Next, ensure that the
997   option **Run with highest privileges** is checked.

998   12. Navigate to the **Triggers** tab. Select the existing trigger and select **Edit**.

999   13. Under the **Advanced Settings** section, ensure that **Repeat task every 1 hour for a duration of**
1000   **Indefinitely** is checked, as well as **Enabled**. Select **OK**.

1001   14. Select **OK**.

1002	15.	Navigate to the **Settings** Tab and ensure the following are checked, then select **OK**.
1003		a. Allow task to be run on demand
1004		b. Run task as soon as possible after a scheduled start is missed
1005		c. If the running task does not end when requested, force it to stop
1006		d. Select other options to suit your environment.

2.11.2 Archer IRM DataFeed Integrations

Archer IRM serves a dual role in the prototype demonstration - the Asset Management and Discovery System and the IT Administrator Dashboard. This section will detail the steps necessary to integrate Archer IRM with the PMCS, the Eclypsium Firmware Analytics Platform, and Microsoft Configuration Manager, which will form the basis of the Asset Management and Discovery System. From there, we will describe how to create a dashboard using the data gathered from the preceding integrations.

2.11.2.1 Create the Devices Application

Before platform and firmware data can be stored in the in the Asset Management and Discovery System, the Archer IRM application must be created. For this task, we leverage the default *Devices* application described as *the central repository of knowledge about your business-critical devices.*

We use the Devices application as a starting point for our customizations that are described in the section. Your organization may have additional requirements that can also be integrated into this solution. As a user with administrative privileges, ensure your installation has the *IT Asset Catalog* solution included before starting the following procedures.

1. In the administration menu, navigate to **Application Builder** > **Solutions**. Select **Add New**.

1022   2.   Select **Copy an existing Solution** and the **IT Asset Catalog**. Click **OK**.

1023   3.   Enter an identifier for the catalog in the **Name** field. Click **SAVE AND CLOSE**.

1024   2.11.2.1.1 Create Supporting Applications

1025   Next, create custom applications that will augment the default *Devices* application. Refer to Appendix B
1026   as you work through creating the supporting application. The application in the following steps, named
1027   *Components*, will store the components associated with each computing device that satisfies acceptance
1028   testing.

1029   1.   In the administration menu, navigate to **Application Builder > Applications.** Select **Add New.**

1030   2. Select Create a new **Application from scratch** and click **OK.**

1031   3. Create an identifier in the **Name** field and select the solution created earlier. Click **OK.**

1032   4. Click **Save.**

1033   In the next series of steps, we will add several Data Fields to the newly created application. These are
1034   like table columns you might define in a relational database. Note that we will only walk through one

DRAFT

1035 example, but the steps can be repeated for the remaining data fields. Before starting these steps,
1036 download and open the Components application schema from our repository. Some data fields, such as
1037 **Tracking ID**, **First Published**, and **Last Updated** are automatically created with each new application and
1038 do not need to be repeated.

1039   5. Open the target Components application from the Administration menu under **Application**
1040   **Builder > Applications.**

1041   6. Click the **Fields** tab.

1042   7. Click **Add New.** Match the Field Type from Appendix B to the **Field Type** field in Archer IRM.
1043   Click **OK.**

NIST SP 1800-34C: Validating the Integrity of Computing Devices   78

1044   8.   Match the **Field Name** from Appendix B to the **Field Name** field in Archer IRM. Click **Save.**

1045   9.   Repeat this process for all remaining data fields in Appendix B. Refer to the online
1046   documentation for other data types that might require additional configuration.

1047 At this point, you have created the first supporting application for the Asset Discovery and Inventory
1048 system. Repeat these procedures to create the *HP UEFI Configuration Variables, Seagate Firmware*
1049 *Attestation, and Seagate Firmware Hash* applications. These applications support the demonstration's
1050 dashboard capability that continuously monitors HP Inc.'s laptop platform security configurations and
1051 Seagate measurement values respectively. Make note of these applications as they are also referenced
1052 in the integration procedures (Section 2.11.2.2).

1053 **2.11.2.1.2   Modify Default *Devices* Application**
1054 In the next series of steps, modify the *Devices* with custom data fields that support the capabilities of
1055 this demonstration. You will also link this application to the supporting applications created in Section
1056 2.11.2.1.1.

1057   1.   Using the Devices table in Appendix B, add the custom data fields using the same method as
1058   described in Section 2.11.2.1.1. Note that cross-referenced data fields are links that will
1059   automatically create a new data field in the associated application.

1060   2.   Modify the layout of the Devices application to include data field customizations created in this
1061   section. The layout will be used to display detailed information about a computing device that
1062   has completed the acceptance testing process. Of note, we have added three sections—*General*
1063   *Information, Eclypsium Firmware Analytics,* and *Associated Components*. Use the screenshots
1064   below as a starting point for customizations that fit into your organization's workflow. More
1065   information regarding layouts can be found on RSA's website.

About
About

General Information
Enterprise Unique Identifier
Serial Number
Make
Manufacturer
Operational Use Validation Status

Eclypsium Firmware Analytics
Last System Scan Date
System Firmware Date
Eclypsium Integrity Scan Status
System Firmware Version

Associated Components
Manufacturer Specific Attributes
Intel | HP, Inc | Seagate | Dell Technologies | Hewlett Packard Enterprise | HP Inc. Security Events | HP Inc UEFI Variables | New

Direct Platform Data
Original Equipment Manufacturer
Product Name
Original Design Manufacturer
SKU
Model
Family

Default Tab Set
Business Continuity | Issues Management | Vulnerability Management | Privacy Management | New
Technology Profile | Business Context | Risk Management | Compliance Management

Operating System Details
Operating System

Network Details
Additional IPs Discovered On Asset
Subnet Mask
Default Gateway
DHCP Server
WINS Server
Domain Name
Placeholder
Network Role
MAC Address
Network Name
Secondary DNS Servers

Server Details
Drive Type
Processors
Server Drives
Total Storage Capacity
Hardware Specification
Rack Identifier
Rack Location
Physical/Virtual
Installation Date
Location

1066 2.11.2.1.3 Modify Default Security Incidents Application
1067 Modify the *Security Incidents* application with custom data fields that support the capabilities of this
1068 demonstration. Using Table 2-7, add the custom data fields using the same method as described in
1069 Section 2.11.2.1.1. Note that cross-referenced data fields are links that will automatically create a new
1070 data field in the associated application.

1071   **Table 2-7 Security Incidents Application Custom Data Fields**

Data Field Name	Data Field Type	Notes
Date/Time QRadar LastUpdate	Date	Stores the date from each *QRadar Offense*
Incident ID (QRadar)	Text	Stores the *QRadar Offense* unique identifier
SCA Computing Device	Cross-Reference	Links to the *Devices* application computing device unique identifier

1072   *2.11.2.2   Create Data Feed Integrations*

1073   In this section, the implementer will create *data feeds* in Archer IRM that will complete the integration
1074   with the PMCS, Microsoft Configuration Manager, IBM QRadar, and Eclypsium. The data feeds will
1075   periodically pull data from the three data sources and map it to the *Devices* application created in the
1076   preceding section.

1077   2.11.2.2.1   Create Eclypsium Data Feeds
1078   1.   In the Administration menu, navigate to **Integration > Data Feeds**. Click **Add New**.

1079   2.   Select **Create a new Data Feed from scratch**. Click **OK**.

1080   3.   Create an identifier in the **Name** field. Select the **Devices** application created in Section 2.11.2.1
1081   in the **Target** field.

DRAFT

1082   4. Click the **Transport** tab. Select **JavaScript Transporter**.

1083   5. Click **Upload** in the **Transport Configuration** section.

1084   6. Click **Add New**.

1085   7. In the file selection modal, select the Eclypsium JavaScript data feed file from the repository.
1086   Click **OK**.

1087   8. Enter "scenario" in the **Key** field and "2" in the **Value** field.

DRAFT

1088　9.　Click the **Navigation** tab. Ensure **XML File Iterator** is selected in the **Navigation Method**
1089　　　dropdown menu.

1090　10.　Click the **Source Definition** tab. In the **Source Data** sub-tab, select **Load Fields**. Select the
1091　　　Eclypsium example XML file. The configuration in Archer should populate the **Source Fields** as
1092　　　follows.

1093　11.　Click the **Data Map** and tab which will default to the **Field Map** sub-tab. Drag and drop the
1094　　　source fields onto the application data fields. Due to the large amount of data fields in the
1095　　　Devices application, below we present a truncated view of the mapping.

1096 12. Click the **Key Field Definitions** tab. Select **Enterprise Unique Identifier** in the Field Name
1097   column.

1098 13. Click the **Update / Archive** tab. Ensure only the **Update** option is selected. Choose **None** for the
1099   **Archive Options**.

1100 14. Click the **Schedule** tab. Select a cadence appropriate for your organization. In this example,
1101   we've chosen to run the data feed on a daily frequency at 12:00AM.

1102 At this point, the data feed for Eclypsium (Scenario 2) is configured. Scenario 3 is configured with the
1103 same process, except a "3" is used in the Value field in Step 8. Click the **Start** button to confirm that the
1104 data feed has been properly configured. Archer IRM will report any errors that are useful for debugging.

1105 2.11.2.2.2 Create Microsoft Configuration Manager Data Feed
1106 Repeat the preceding steps to add the Microsoft Configuration Manager Data Feed with the following
1107 modifications:

1108 15. In the **Transport** tab, select **Database Query Transporter**. Insert the following values in the
1109   form:

Provider	Odbc Data Provider

Connection String	`Driver=ODBC Driver 17 for SQL Server;server=PEMSQL2019;database=CM_PE1;PWD=[SQL USER PASSWORD];UID=[SQL USER]`
Query	`select dbo.vSMS_R_System.Name0, dbo.vSMS_R_System.SMBIOS_GUID0 from dbo.vSMS_R_System inner join dbo.v_CIComplianceStatusDetail on dbo.v_CIComplianceStatusDetail.Netbios_Name0 = dbo.vSMS_R_System.Netbios_Name0 where dbo.v_CIComplianceStatusDetail.CurrentValue = '2' and dbo.v_CIComplianceStatusDetail.ConfigurationItemName = 'TSCVerify - Registry'`

1110   16. In the **Navigation** tab, select **Database Query Iterator**.

1111   17. In the **Source Definition** tab, add a new **Source Field** named Compliance.

1112   18. Edit the new **Source Field** with the static text "Out of Policy".

1113   19. In the **Field Map** sub-tab in the **Data Map** tab, drag and drop the **Source Fields** onto the **Target**
1114   **Fields** as shown in the images below.

1115   20. In the **Key Field Definitions** sub-tab in the **Data Map** tab, select **Enterprise Unique Identifier**.

1116   21. In the **Update / Archive** sub-tab in the **Data Map** tab, ensure only **Update** is selected.

1117   At this point, the Data Feed for the Microsoft Configuration Manager is configured. Click the **Start**
1118   button to confirm that the Data Feed has been properly configured. Archer will report any errors that
1119   are useful for debugging.

DRAFT

2.11.2.2.3 Create the PMCS Data Feed

Repeat the initial steps to add the Data Feed for the PMCS with the following modifications:

22. In the **Transport** tab, upload the custom JavaScript from the project repository. In the Custom Parameters fields, add **filter** and **url** keys as shown below. The value for **filter** may be blank or set to a specific manufacturer (refer to comments in the script for the specific values we used). Set **url** to the location of the PMCS in your environment.

23. In the **Source Definition** tab, upload the example XML file from the project repository. The **Source Fields** should resemble the following screenshot.

24. Map the **Source Fields** to the **Target Fields** and the **Field Map** sub-tab in the **Data Map** tab. Use Table 2-8 for reference.

Table 2-8 PMCS Data Feed Source Field to Destination Field Mapping

Source Field	Destination Field
/Component/Addresses/Address	Associated Components/Addresses/Address

Source Field	Destination Field
/Component/Class	Associated Components/Class
/Component/Field_Replaceable	Associated Components/Field Replaceable
/Component/Manufacturer	Associated Components/Manufacturer
/Component/Model	Associated Components/Model
/Component/Platform_Certificate	Associated Components/Platform Certificate
/Component/Platform_Certificate_URI	Associated Components/Platform Certificate URI
/Component/Revision	Associated Components/Revision
/Component/Serial	Associated Components/Serial
/Component/Version	Associated Components/Version
UUID	Enterprise Unique Identifier
Family	Family
Make_and_Model	Make
Manufacturer	Manufacturer/Value
Original_Design_Manufacturer	Original Design Manufacturer
Original_Equipment_Manufacturer	Original Equipment Manufacturer
Product_Name	Product Name
Serial_Number	Serial Number
SKU	SKU

1131   25. In the **Key Field Definitions** sub-tab in the **Data Map** tab, choose Enterprise Unique Identifier as
1132   the **Key Field** definition.

1133   The Data Feed for the PMCS is configured. Click the **Start** button to confirm that the Data Feed has been
1134   properly configured. Archer will report any errors that are useful for debugging.

1135 2.11.2.2.4 Create IBM QRadar Offenses Data Feed

1136 Repeat the steps from Section 2.11.2.2.1 to add the Data Feed for IBM QRadar with the following
1137 modifications:

1138 26. In the **Transport Settings** section of **Source Settings,** choose the IBM QRadar script *(Integration-*
1139 *Scripts\Archer Integrated Risk Management Data Feed Integrations\IBM QRadar\app.js)* from
1140 the project repository.

FILE NAME	SIZE	UPLOAD DATE
qradar_data_feed.js	12.36 KB	4/22/2022, 10:33:09 AM

▸ TRANSPORT CONFIGURATION ⓘ

1141 27. In the Custom Parameters section of the Source Connection tab, enter the hostname of the
1142 QRadar system and the API key created in Section 2.11.3.2.4. Ensure that the QRadarAPIKey is of
1143 type Protected.

▸ CUSTOM PARAMETERS ⓘ

KEY	TYPE	VALUE
QRadarHostname	Plain Text ⌄	qradar.lab.nccoe.org
QRadarAPIKey	Protected ⌄	••••••••••••••••••••••••••••••••••

1144 28. In the **Source Data** section of the **Source Definition** tab, upload the example XML QRadar
1145 response file.

1146   29. Map the **Source Fields** to the **Target Fields** in the **Field Map** sub-tab in the **Data Map** tab. Use
1147   Table 2-10 for reference.

1148   **Table 2-9 QRadar Data Feed Source Field to Destination Field Mapping**

Source Field	Destination Field
UUID	/SCA Computing Device/Enterprise Unique Identifier
lastUpdate	Date/Time QRadar LastUpdate
description	Incident Summary
event	Title
id	Incident ID (QRadar)

1149   30. In the **Key Field Definition** sub-tab in the **Data Map** tab, choose **Incident ID (QRadar)** as the Key
1150   Field Definition. Additionally, choose **Enterprise Unique Identifier** as the **Key Field** definition for
1151   the **SCA Computing Device** reference field.

DRAFT

1152 **2.11.2.2.5 Create Seagate API Data Feeds**

1153 Repeat steps from Section 2.11.2.2.1 to add the Data Feed for Seagate drive firmware attestation and
1154 firmware hash data with the following modifications:

1155 31. Enter *Seagate Attestation Feed* in the **Name** field section of the **General** tab. In the **Feed**
1156   **Information** section of the same tab, select *Seagate Firmware Attestation* from the **Target**
1157   **Application** pull-down menu.

1158 32. In the **Transport Configuration** section of **Source Settings,** choose the Seagate script from the
1159   project repository.

FILE NAME	SIZE	UPLOAD DATE
archer_script.js	9.7 KB	2/10/2022, 3:42:17 PM

1160
1161

33. In the **Custom Parameters** section of **Source Connection** tab, enter the PMCS URL and the **filter** value of *seagate.fw.attestation*.

KEY	TYPE	VALUE
filter	Plain Text ∨	seagate.fw.attestation
url	Plain Text ∨	http://

1162
1163

34. In the **Source Data** section of the **Source Definition** tab, upload the example Seagate Firmware Attestation XML response file.

SOURCE FIELD	FIELD TYPE	SOURCE
▾ SeagateDriveFirmwareAttestation	None	SeagateDriveFirmwareAttestation
device_uuid	Raw Field Data	device_uuid
drive_serial	Raw Field Data	drive_serial
assessor_id	Raw Field Data	assessor_id
root_of_trust_id	Raw Field Data	root_of_trust_id
root_of_trust_nonce	Raw Field Data	root_of_trust_nonce
device_nonce	Raw Field Data	device_nonce
fw_version	Raw Field Data	fw_version
secure_boot_device_state	Raw Field Data	secure_boot_device_state
signing_auth_database	Raw Field Data	signing_auth_database

1164
1165

35. Map the **Source Fields** to the **Target Fields** and the **Field Map** sub-tab in the **Data Map** tab. Use Table 2-10 for reference.

DRAFT

Table 2-10 Seagate Drive Data Feed Field Mapping

1166

Source Field	Destination Field
drive_serial	/Seagate Drive Serial/Serial
assessor_id	Assessor Identifier
root_of_trust_id	Root of Trust Identifier
root_of_trust_nonce	Root of Trust Nonce
device_nonce	Device Nonce
fw_version	Firmware Version
secure_boot_device_state	Secure Boot Device State
signing_auth_database	Signing Auth Database

1167 36. In the **Key Field Definition** tab within the **Data Map** tab, select *Serial* in the pull-down **Field**
1168 **Name** column.

1169 37. Save the new Data Feed.

1170 Repeat the procedures in this section to create a Data Feed that will collect the Seagate drive firmware
1171 hash values. Note that this Data Feed will target the *Seagate Firmware Hash* application.

2.11.2.3 Create the Dashboard

1173 1. Create a new report by clicking **Reports** in the administrative console and **Add New.**

1174 2. Select the Devices application that was created in the preceding steps—in this case, **Enterprise**
1175 **Computing Devices.**

footer
NIST SP 1800-34C: Validating the Integrity of Computing Devices   95

DRAFT

1176   3.   Click the **Statistics Mode** option. In the **Fields to Display** section, select **Operational Use**
1177   **Validation Status** and remove the default selections.

1178   4.   In the **Filters** section, select *Operational Use Validation Status* for **Field to Evaluate**, *Equals* for
1179   **Operator,** and *Policy violation* for **Value(s).**

1180   5.   Select **Display Totals** in the **Display Options** section.

1181   6.   Select **Chart Only** and click **Save** and supply a unique name for the report.

1182   7.   Create a new iView by navigating to **Workspaces and Dashboards > Global iViews** in the
1183   administrative menu. Click **Add New.**

1184   8.   In the **iView Types** section, select **Report** and click **OK.**

1185   9.   In the **General Information** section, supply a name and a folder to store the new iView.

1186   10.   In the **Options** section, choose the report that was created in the preceding steps and save the
1187   iView.

1188   11. Create a new Dashboard by navigating to **Workspaces and Dashboards > Dashboards** in the
1189   administration menu. Click **Add New.**

1190   12. Select **Create a new Dashboard from scratch** and click **OK.**

1191   13. In the **General** tab, supply a name for the Dashboard.

1192   14. In the **Layout** tab, click **Select iViews.** Choose *Select* from **Global iView Library** for the **Creation**
1193   **Method.** Choose the iView created in the preceding steps and click **OK.**

1194   15. The selected iView will appear in the layout. Save the Dashboard.

1195   16. Open the solution workspace by navigating to **Workspaces and Dashboards > Workspaces** in
1196   the administration menu. In the **Dashboards** tab, choose the Dashboard created in the
1197   preceding steps by clicking **Select Dashboards.**

1198 17. Save the workspace. At this point, the new Dashboard will appear as part of the workspace. For
1199 further customization options, refer to the RSA website.

1200 18. Repeat the steps in this section to create a report that tracks platform integrity issues that are
1201 detected from the following sources:

Platform	Archer Application	Archer Data Field
Eclypsium Analytic Platform	Enterprise Computing Devices	Eclypsium Integrity Scan Status
HP Inc	HP UEFI Configuration Variables	HP Inc BIOS Configuration Status
Seagate	Seagate Firmware Hash	Firmware Hash Status

1202 ## 2.11.3 IBM QRadar Integrations

1203 The following sections describe how to integrate Dell and HP Inc. laptops with QRadar so that the
1204 laptops transmit continuous monitoring event logs to the QRadar console.

1205 ### 2.11.3.1 Dell and HP Inc. Laptops

1206 Perform the prerequisite steps in Section 2.2.1.3, then on each target laptop:

1207 1. Ensure Remote Event Log Management is enabled for each laptop.

1208 2. (Optional) In the QRadar console, create a new log source group which may be desirable to help
1209 organize target laptops. In our demonstration, we created a group for each manufacturer.

1210 3. Create a new log source for the WinCollect Agent (see Section 2.10.1). Note that when
1211 configuring the Log Source parameters, a Windows account is required to retrieve the relevant

1212   security event. This demonstration created a domain account with privileges limited to the
1213   scope of this capability (Manage auditing and security log permission enabled).

2.11.3.2   IBM QRadar

1215   The section describes the procedures that will create *Offenses* generated from detected laptop platform
1216   integrity security events. Additionally, it also describes an API key that is used to access the QRadar REST
1217   API. The key is used as input to Section 2.11.2.2.4.

2.11.3.2.1   Create Custom Event Property (UUID)

1219   This property uses a regular expression (regex) to identify universally unique identifiers (UUIDs) that are
1220   embedded in Windows 10 Event Logs that are sent from laptops when a platform integrity issue is
1221   detected.

1222   4.   In the QRadar console, navigate to **Admin** > **Custom Event Properties**. Click **Add** and a new
1223   window pops up. In the **Test Field,** paste in the example event log.

Test Field

```
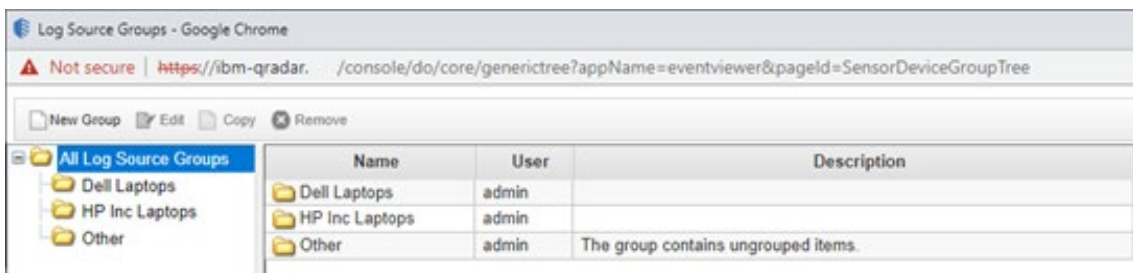<13>Dec 09 12:08:55 dell-6 AgentDevice=WindowsLog
AgentLogFile=Dell      PluginVersion=7.3.1.22
Source=Trusted Device | BIOS Verification
Computer=dell-6 OriginatingComputer=10  User=   Domain=
```

1224   5.   In the **Property Definition** section, select **New Property** and enter *UUID for Supply Chain.* Check
1225   *Enable for use in Rules, Forwarding Profiles and Search Indexing*.

Property Definition

Existing Property:
New Property: UUID for Supply Chain
☑ Enable for use in Rules, Forwarding Profiles and Search Indexing
Field Type: AlphaNumeric
Description:

1226   6.   In the **Property Expression Definition** section, ensure *Enabled* is checked. In **the Log Source**
1227   **Type** pull-down, select *Microsoft Windows Security Event Log* and select *All* in the **Log Source**
1228   pull-down. Select the *Category* radio button. Choose *Any* in both the **High Level Category** and
1229   **Low Level Category** pull-downs. In the **Regex** field, insert the value below.

1230   `([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-`
1231   `F]{12})`

1232   7. Click the **Test** button. If successful, a message will appear that the expression has been
1233   highlighted in the payload. Click the **Save** button.

1234 **2.11.3.2.2 Create Custom Event Properties (Security Events)**
1235 This section describes how to create filters that will identify the individual HP Inc. and Dell platform
1236 integrity events that have been detected and reported to QRadar. Use Table 2-11 as a guide. We used
1237 existing QRadar Categories which group manufacturer security events. These procedures also require an
1238 example of the security event payload that is created on the manufacturer's laptop when a platform
1239 integrity issue is detected. For HP Inc laptops, the payloads are generated by custom PowerShell scripts
1240 which consume the output from the CMSL Get-HPFirmwareAuditLog command. Dell security event
1241 payloads are generated directly by the Dell Trusted Devices platform.

1242   **Table 2-11 QRadar Security Event Mapping**

QRadar Category	Manufacturer Event Category	Manufacturer Event Value
Custom Policy 1	HP_Sure_Start	Integrity violation
Custom Policy 2	HP_Sure_Start	Policy violation
Custom Policy 3	HP_Sure_Start	Recovery
Custom Policy 4	HP_Sure_Start	Revert to default
Custom Policy 5	Sys_Config	Policy violation
Custom Policy 6	HP_Sure_Start	Attack mitigation
Custom Policy 7	HP_Sure_Start	SMM execution halted
Custom Policy 8	Secure_Platform	Management Attack mitigation
Custom Policy 9	HP_Sure_Recover	Recovery initiated
Custom User 1	HP_Sure_Recover	Recovery success
Custom User 2	HP_Sure_Recover	Recovery failure
Custom User 3	HP_Sure_Start	Illegal DMA Blocked
Custom User 4	HP_Sure_Admin	Power off due to failure authentication
Custom User 5	HP_Sure_Admin	WMI blocked due to failed authentication
Custom User 6	HP_Sure_Start	EpSC execution halted
Custom User 7	HP_TamperLock	Cover removed
Custom User 8	HP_TamperLock	TPM cleared based on Policy
Custom User Medium	Dell Laptop DTD BIOS Violation	N/A

1243   1. In the QRadar console, navigate to **Admin** > **Custom Event Properties**. Click **Add** and a new
1244   window pops up. In the **Test Field,** paste in the example event payload. In the screenshots
1245   below, we are using a payload which includes a *HP_Sure_Start Policy violation*.

1246
1247

2. In the **Property Definition**, select *New Property*. Name the new property "[Event Category] [Event Value]". Check *Enable for use in Rules, Forwarding Profiles and Search Indexing*.

1248
1249
1250

3. In the **Property Expression Definition** section, make sure **Enabled** is checked. In **Log Source Type**, select *Microsoft Windows Security Event Log*. In **Log Source** select **All**. Select the *Event Name* radio button.

1251
1252

  a. Click **Browse** and search for *"Application Information Event"* (with quotes) in the **QID/Name** field. Select it and click **OK**.

1253
1254

  b. Select **Extraction using JSON Keypath**. *"HP_Sure_Start Policy violation"* will look like the following as an example:

1255

```
/"data"/"Events"/"HP_Sure_Start"/"Policy violation"[]
```

1256   4. Click the **Test** button. If successful, the security event is found in the **Test Field**. Click **Save**.

1257   Continue the process for all events listed in Table 2-11.

1258   2.11.3.2.3 Create QRadar Rules

1259   5. In the QRadar console, click **Log Activity**. Select **Rules** > **Rules** then **Actions** > **New Rule**.

1260   6. Ensure **Events** is selected, then click **Next.**

1261   7. Enter a name for the rule. We used the following pattern: *"[Event Category] [Event Value] rule".*

1262   8. In the rules editor, search for *"event matches this AQL filter query"*. Click the *"this"* hyperlink to
1263   launch the Ariel Query Language (AQL) filter query. Enter the query below and click **Submit**.

1264   ```
       "Event ID"=3001
       ```

1265   9. Create another criteria by using *"when the event matches this search filter"*. Click *"this search
1266   filter"* and locate the matching **Custom Property**. Select *"is not N/A"* and click **Add**. Click **Submit**.

Rule (Click on an underlined value to edit it)
Invalid tests are highlighted and must be fixed before rule can be saved.

Apply HP_Sure_Start Policy Violation Rule   on events which are detected by the Local ▾ system
 🔴🔵🔘 and when the event matches "Event ID"=3001 AQL filter query
 🔴🔵🔘 and when the event matches HP_Sure_Start Policy violation (custom) is not N/A

1267   10. (Optional) Make the rule part of a group to organize platform integrity offenses. We created a
1268   custom group named *"Supply Chain Security Event"*.

Please select any groups you would like this rule to be a member of:
☐ Response
✔ Supply Chain Security Event
☐ Suspicious
☐ System
☐ Threats

1269   11. Click **Next**. In the **Rule Response** section, select **Dispatch New Event**. Create an **Event Name** and
1270   **Event Description** following the same pattern as above.

1271   12. In the **Event Details** section, select the **High-Level Category** of *"User Defined"* and choose the
1272   **Low-Level Category** noted in Table 2-11.

1273   13. Check *"Ensure the dispatched event is part of an offense"*. Index offense based on *"UUID for
1274   Supply Chain"* in the pull-down menu.

1275   14. In the **Offense Naming** section, select the second option (replace).

1276   15. Click **Finish.** The new rule will appear in the **Offenses** > **Rules** tab.

1277   Repeat this section for every security event listed in Table 2-11.

1278   **2.11.3.2.4 Create an Authorized Service Token**

1279   1. In the administration console, click **Authorized Services**, then **Add New**. Enter an **Authorized**
1280   **Service Label** and appropriate **Security Profile** and **User Role** for your environment. Click **Save**.

1281 2. The QRadar console will display the following dialog. Click the "eye" to reveal the secret token.
1282   Store the token securely.

1283 3 Operational Considerations

1284 This section describes the execution steps of an IT administrator assigned to the acceptance testing or
1285 monitoring of computing devices during their operational lifecycle. Each subsection restates the
1286 scenarios from the project description, but this prototype demonstration does not address each

1287 scenario in totality. This preliminary draft will be updated later with additional guidance for laptops and
1288 servers.

1289 Create an environment as described in Section 2 before attempting to use the proof-of-concept tools
1290 below.

3.1 Scenario 2: Verification of Components During Acceptance Testing

1292 In this scenario, an IT administrator receives a computing device through nonverifiable channels (e.g.,
1293 off the shelf at a retailer) and wishes to confirm its provenance and authenticity to establish an
1294 authoritative asset inventory as part of an asset management program.

1295 The general execution steps are as follows:

1296   1. As part of the acceptance testing process, the IT administrator uses tools to extract or obtain the
1297   verifiable platform artifact associated with the computing device.

1298   2. The IT administrator verifies the provenance of the device's hardware components by validating
1299   the source and authenticity of the artifact.

1300   3. The IT administrator validates the verifiable artifact by interrogating the device to obtain
1301   platform attributes that can be compared against those listed in the artifact.

1302   4. The computing device is provisioned into the physical asset management system and is
1303   associated with a unique enterprise identifier. If the administrator updates the configuration of
1304   the platform (e.g., adding hardware components, updating firmware), then the administrator
1305   might create new platform artifacts to establish a new baseline.

3.1.1 Technology Configurations

3.1.1.1 Configure the HIRS ACA

1308 Before running the acceptance test on Dell and HP Inc. laptops, the HIRS ACA must be configured with
1309 the target laptop's platform attribute certificate and any trust chains associated with the platform
1310 attribute certificate and endorsement credential.

1311   1. On the HIRS ACA web portal, under the **Configuration** panel, select **Policy.**

DRAFT
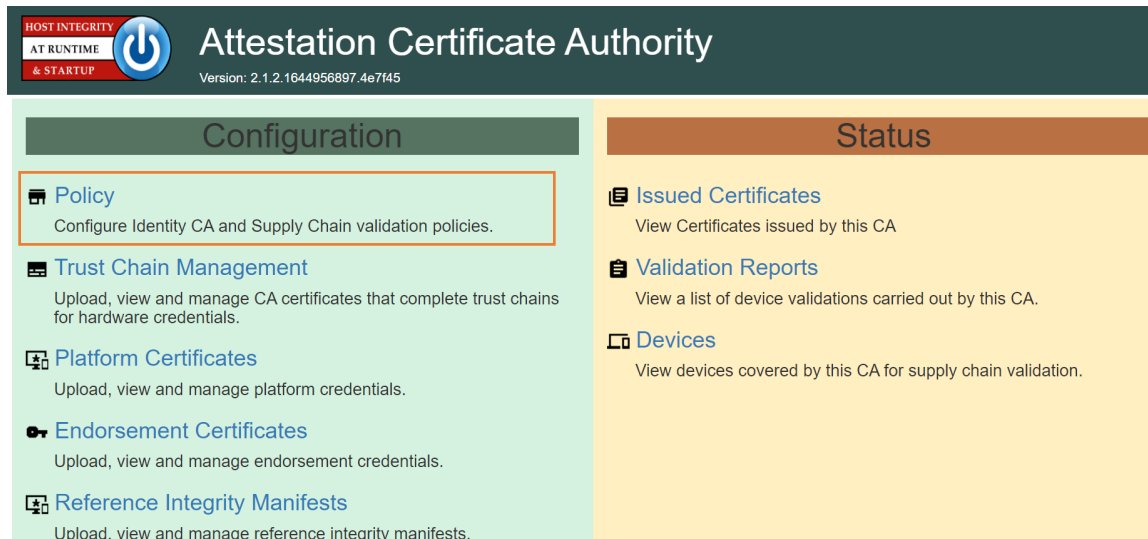
1312
1313

2. For this prototype demonstration, make sure the following policy options are set as listed in the table below.

| Policy Option	Setting
Endorsement Credential Validation	Enabled
Platform Credential Validation	Enabled
Platform Attribute Credential Validation	Enabled
Firmware Validation	Disabled
Ignore IMA PCR Entry	Disabled
Ignore TBOOT PCRs Entry	Disabled
Ignore GPT PCRs Entry	Disabled
Ignore OS Events	Disabled
Generate Attestation Certificate	Enabled
Attestation Certificate Validity period	Disabled
Attestation Certificate Renewal period	Disabled

DRAFT
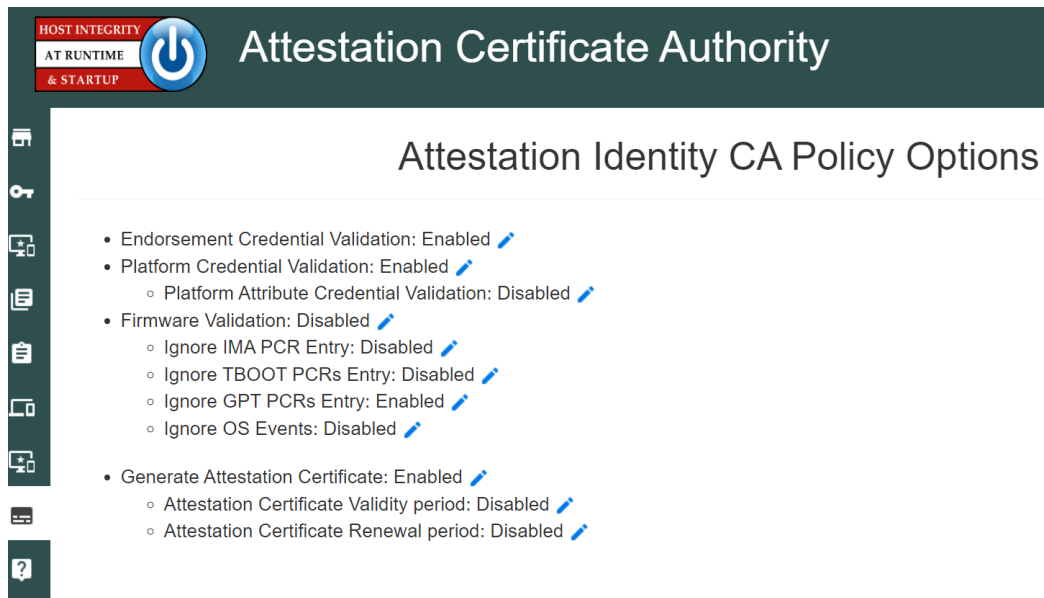
1314   3. Upload the trust chain certificates by navigating to the **Configuration** panel, then selecting **Trust**
1315   **Chain Management**.
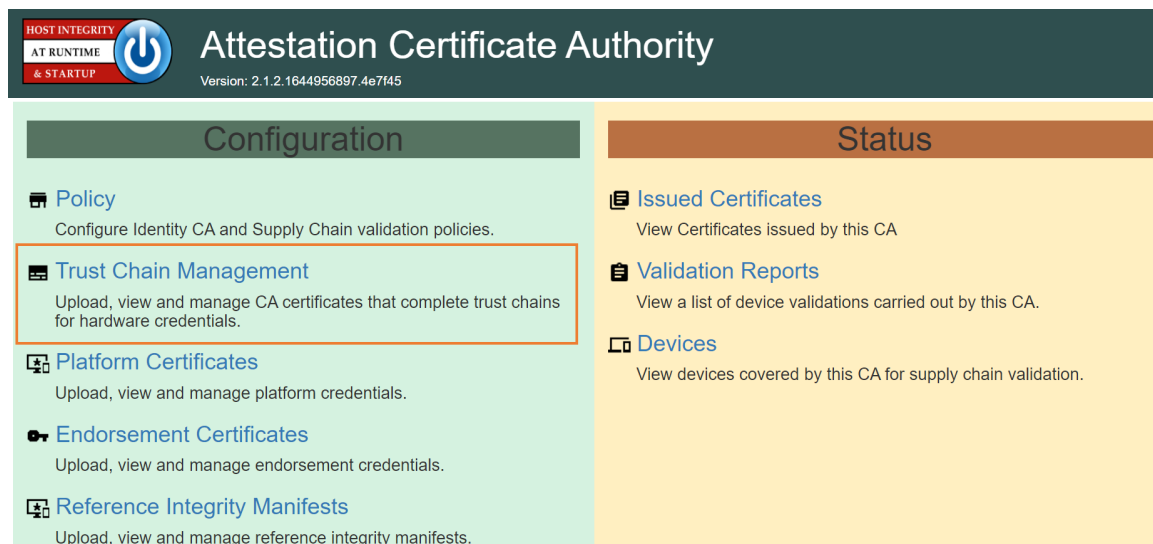
1316   4. Select the icon beside **Import Trust Chain CA Certificates**.
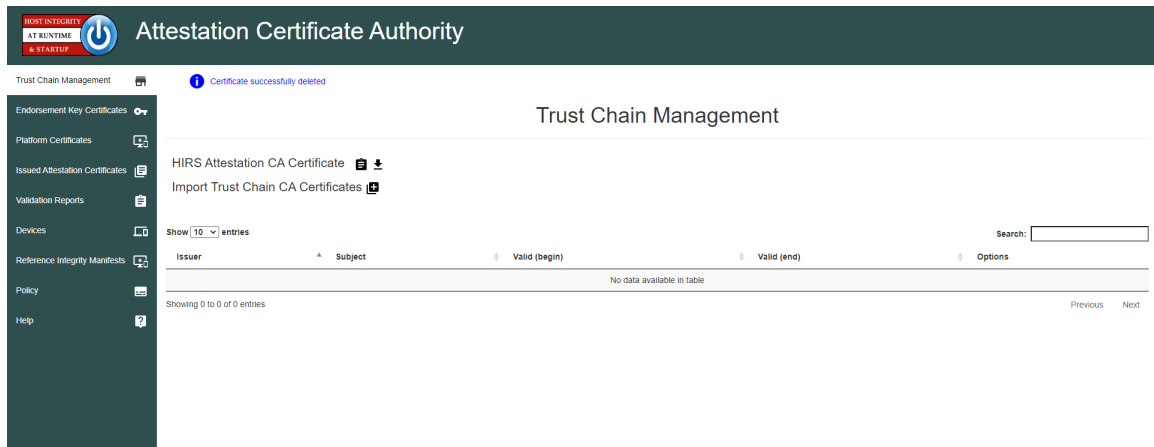
1317   5. Select **Choose Files**.

1318   6. Select the Trust Chain Certificate from the local computer. In the example below, the .crt file is
1319   named *PCTestCA.example.com*. Optionally, select multiple certificates if your implementation
1320   includes computing devices from distinct manufacturers. Click **Open**.
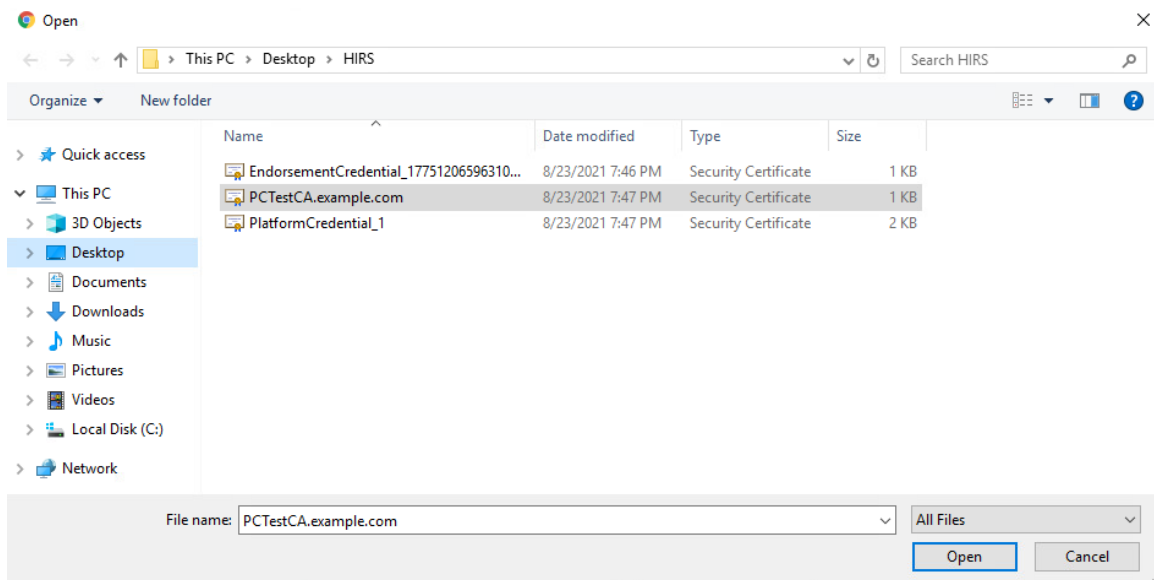
1321   7. Select **Save.**

1322   8. The Trust Chain certificate should appear under the **Trust Chain Management** tab. Repeat this
1323   process for all root and intermediate certificates.
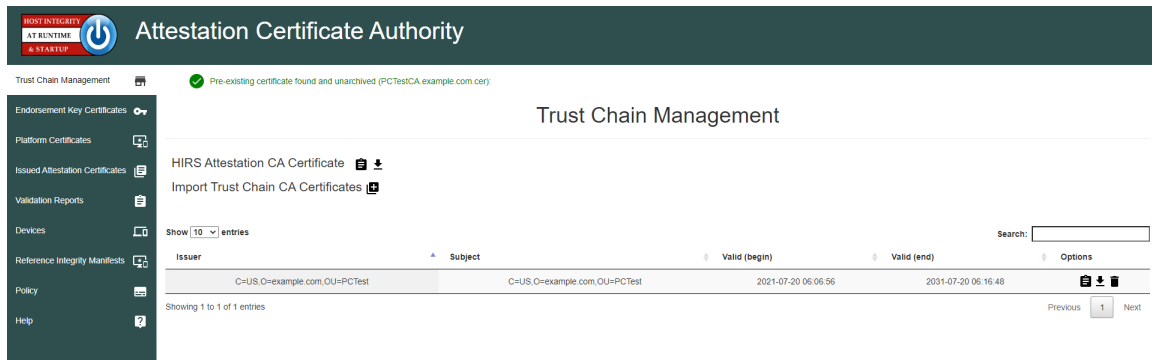
1324   9. Update the Platform Attribute certificates by navigating to the **Configurations** panel, then
1325   selecting **Platform Certificates**.

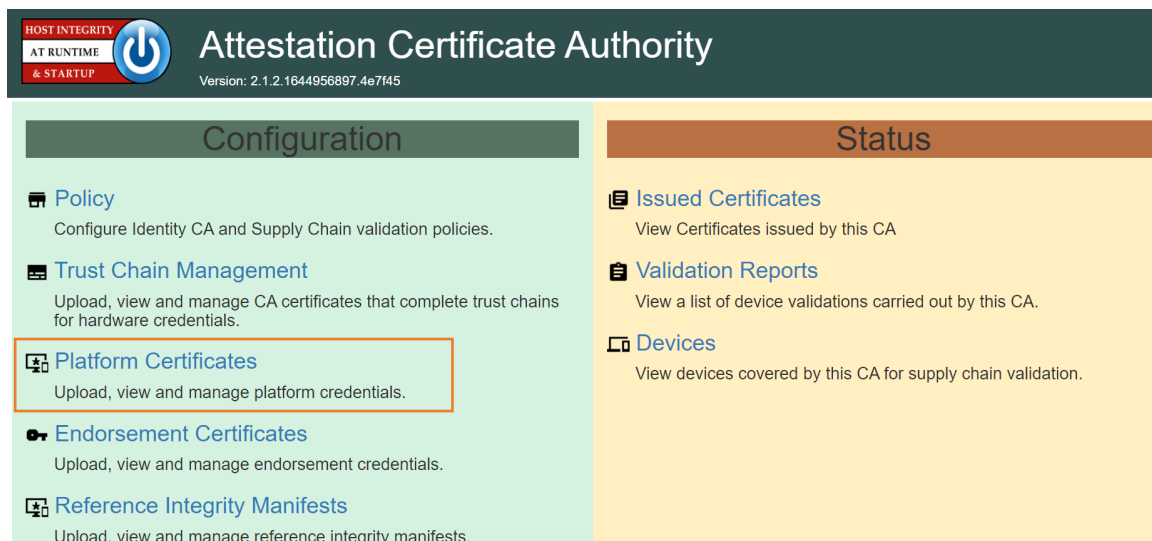1326   10. Select the icon beside **Import Platform Certificates**.
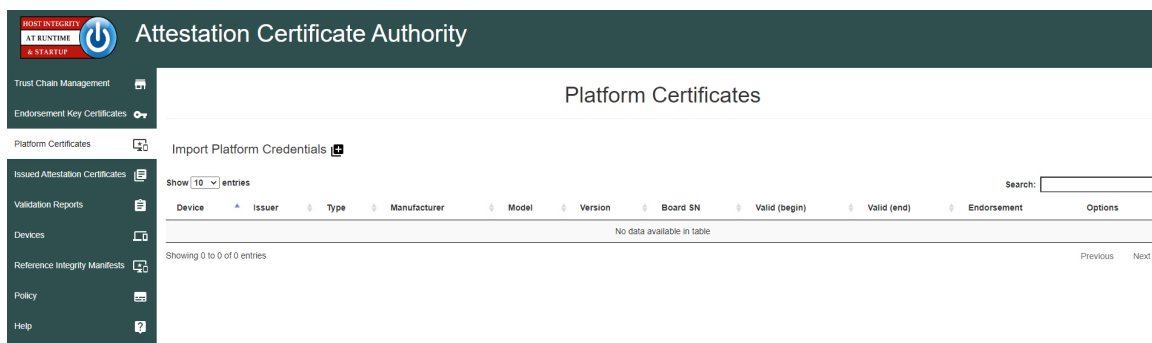
1327   11. Select **Choose Files.**

DRAFT

1328   12. Select the Platform Certificate from the local computer. In the example below, the .crt file is
1329   named **PlatformCredential_1**. Select the file and click **Open**.
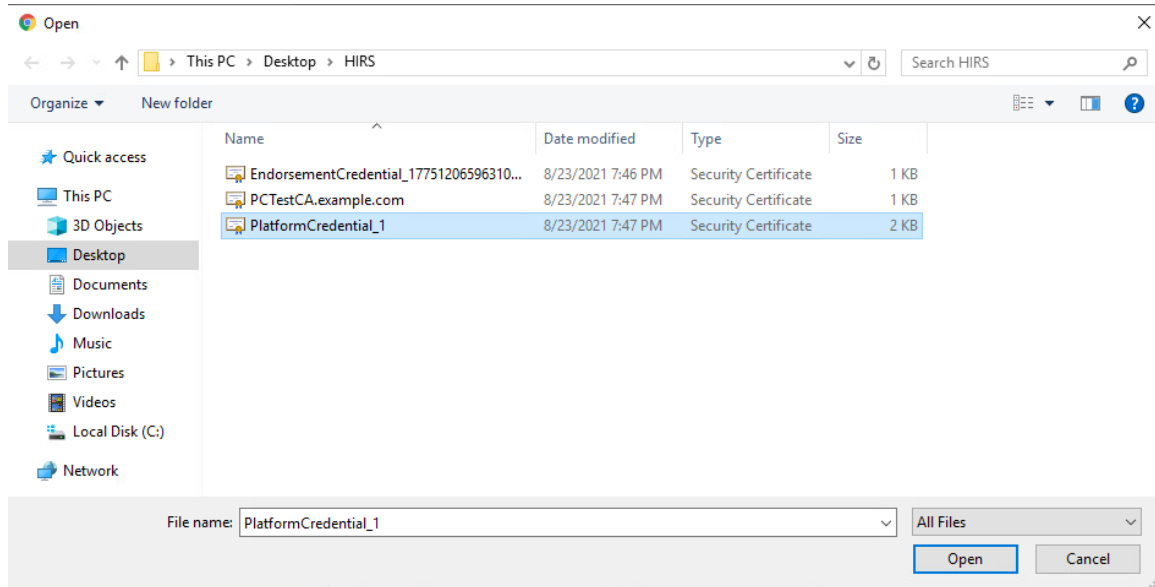
1330   13. Select **Save**.

1331   14. The Platform certificate should appear under the **Platform Certificates** tab.
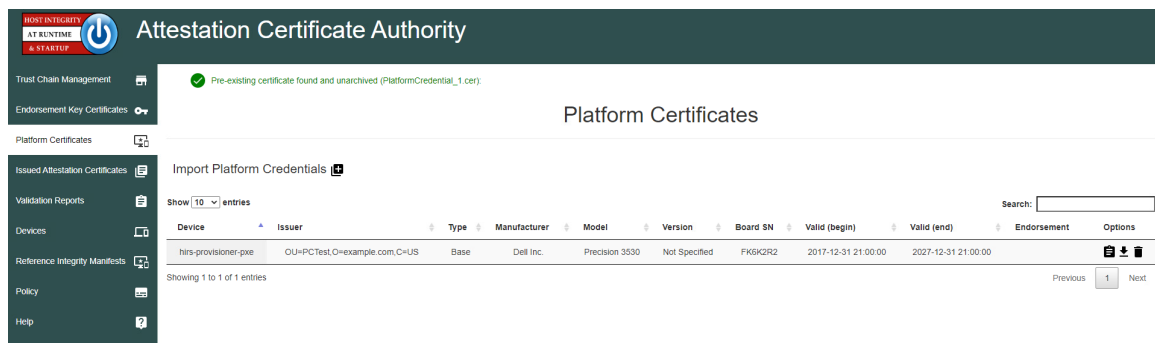
1332   15. Upload the Endorsement Key certificate by navigating to the **Configuration** panel, then selecting
1333   **Endorsement Certificates**.

DRAFT
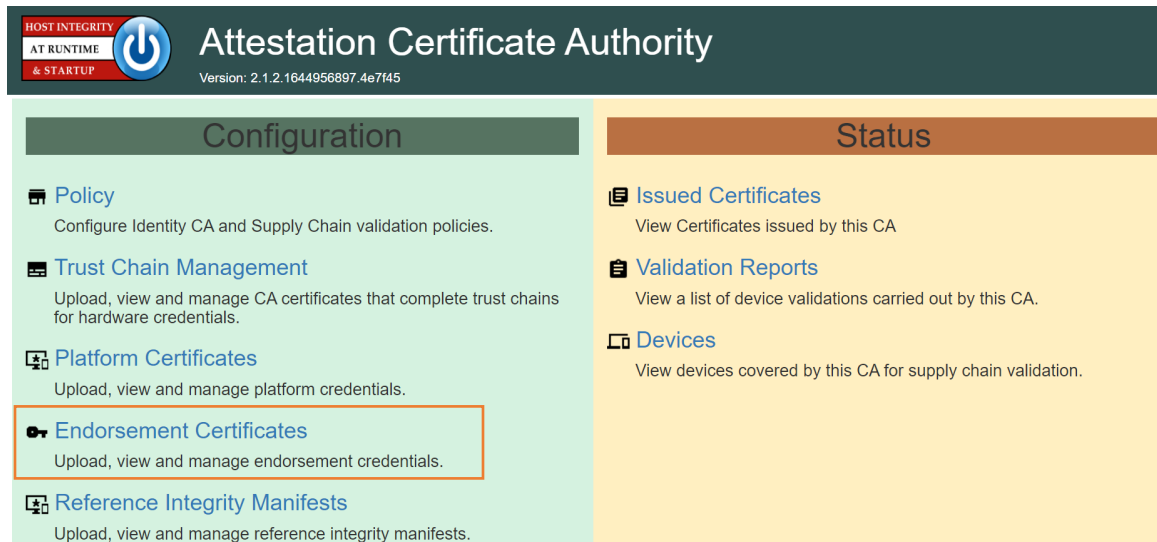
1334   16. Select the icon beside **Import Endorsement Key Certificates**.
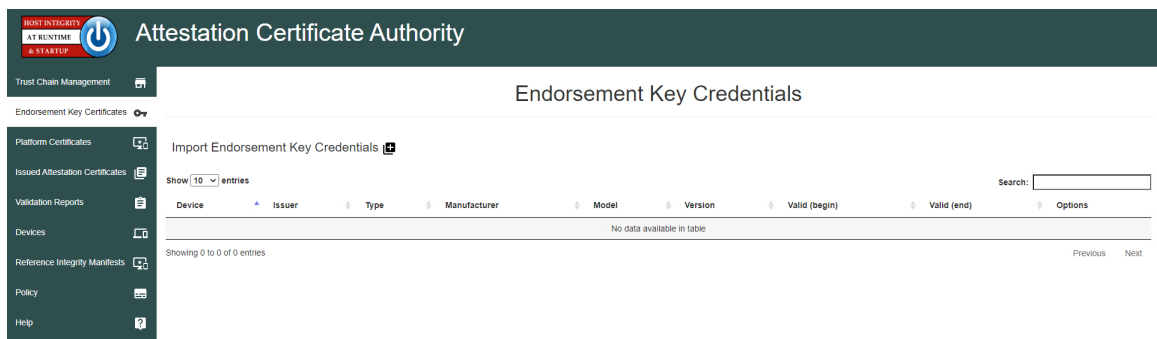
1335   17. Select **Choose Files**.

1336   18. Select the Endorsement Credential from the local computer. For this project, the .crt file is
1337   *EndorsementCredential_17751206596310784982788*. Select the file and click **Open**.
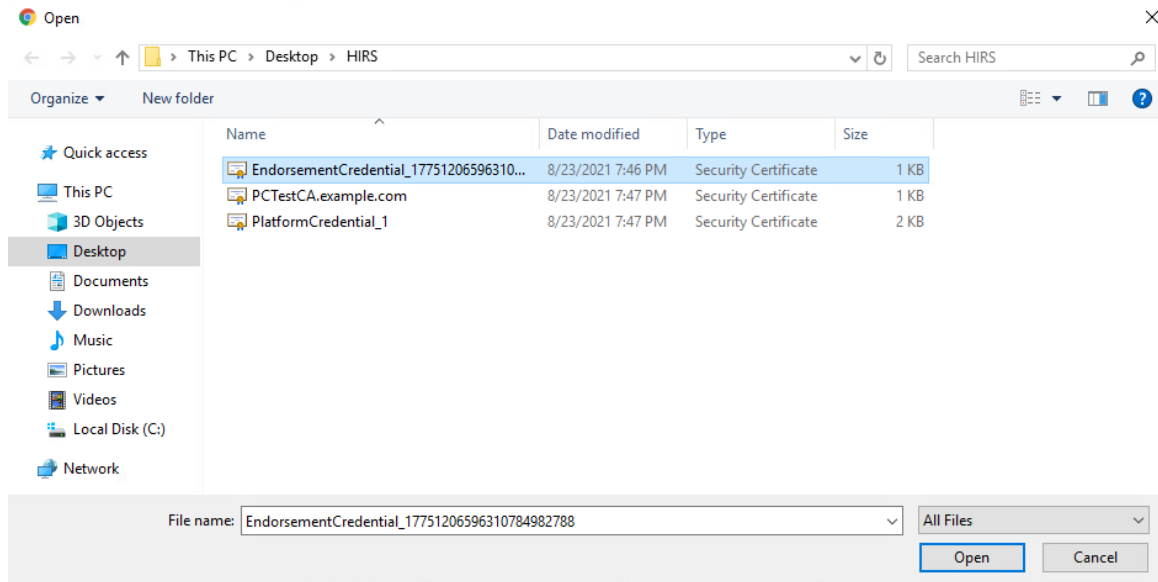
1338   19. Select **Save.**

1339   20. The Endorsement Key certificate should appear under the **Endorsement Key Credentials** tab.
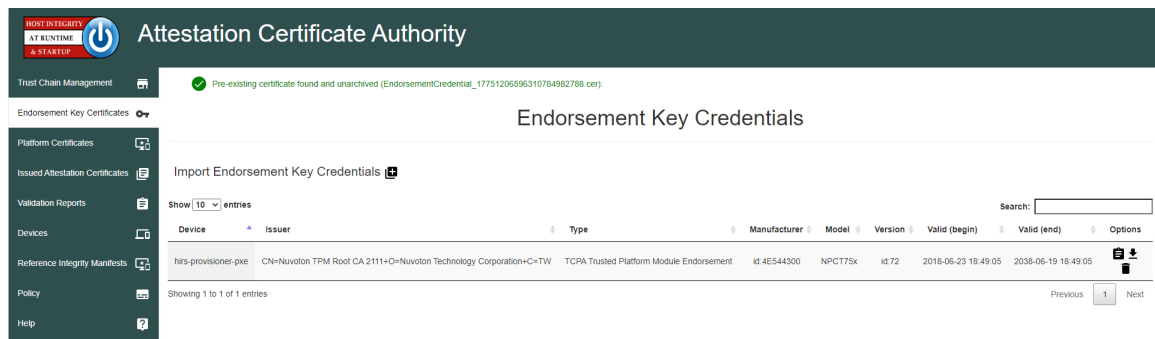
3.1.1.2 Dell and HP Inc. Laptops

1341   1. Boot the target laptop into the CentOS 7 acceptance testing environment via iPXE. This typically
1342   requires a one-time boot execution to prevent the laptop from loading the native OS. Consult
1343   the manufacturer's documentation for the appropriate steps. Choose HIRS Provisioner Live from
1344   the iPXE boot menu.
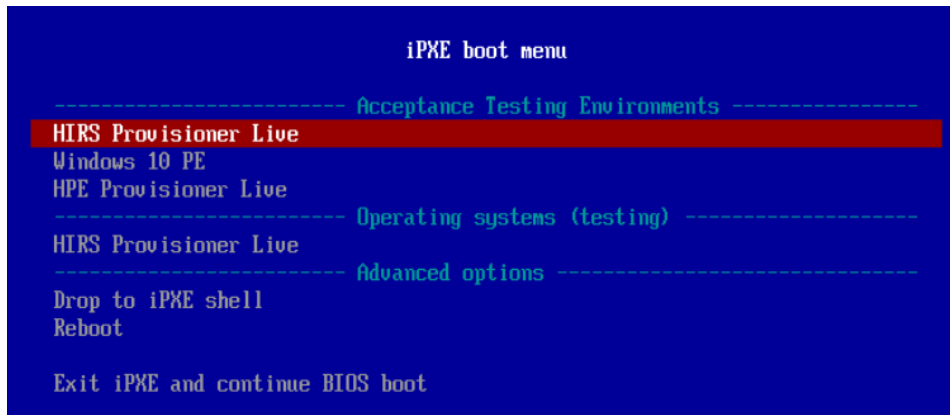
1345   2. Once the live environment has loaded, log in as a user with root privileges. Run the provision.sh
1346   script. The script will attempt to:

1347   a. Change the hostname of the live environment. This assists the administrator in locating
1348   the target machine in the Eclypsium console.

1349   b. Run the Eclypsium scanner and submit results to the Eclypsium Analytic cloud platform.

1350   c. Run the HIRS provisioning script. If successful, post the results to the PMCS.

1351   The script will exit at any point an error is detected. Refer to the comments in the script to set
1352   this up in your own environment. Up-to-date information related to debugging the HIRS
1353   provisioning process can be found on the project site.

3.1.1.3 Intel-Contributed Laptops
1354

1355 The Auto Verify tool is central to scenario 2 acceptance testing. The tool compares the Direct Platform
1356 Data (DPD), allowing the customer to identify certain system changes from the time of manufacturing to
1357 the time of first boot. Install the Auto Verify Tool on the target system before attempting to execute the
1358 steps in this section.

1359 The DPD files and platform certificate files for the target laptop are available from Intel's Transparent
1360 Supply Chain demo page, https://tsc.intel.com/client-demo/. Work with your Intel representative to
1361 obtain credentials for your organization.

3.1.1.3.1 Download DPD File and Platform Certificate
1362
1363   1. Authenticate to the Intel TSC Client Demo portal page.

1364   2. Enter the serial number of the Intel laptop. Select **Search.**

1365   3. Download the zip file containing the DPD files and platform certificate. Save and unzip the file
1366   on the target laptop. These files will be used with the Auto Verify tool to determine if any
1367   components have been changed.

1368   4. Launch the Auto Verify Tool.

1369   5.   Click the **Scan System** button.
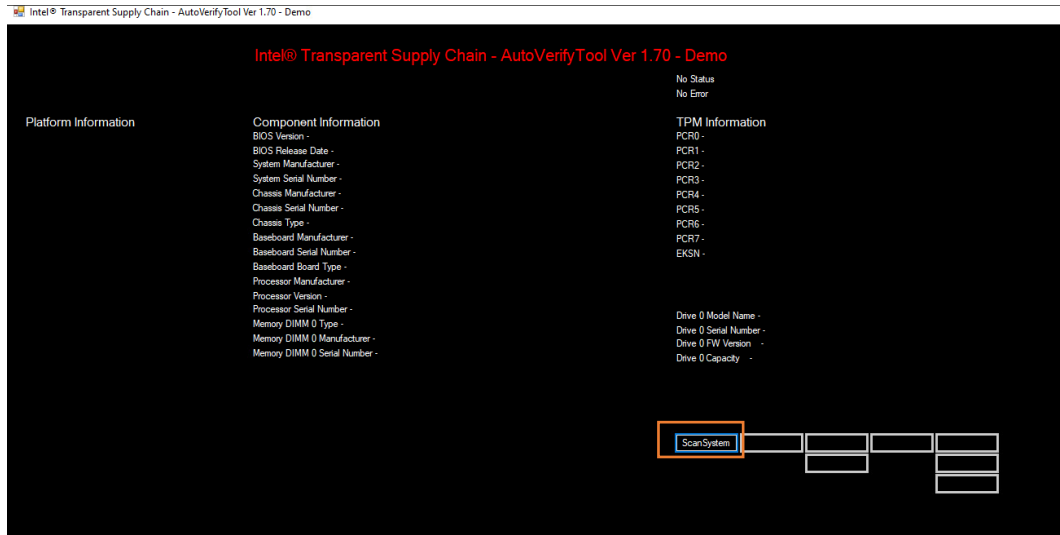
1370   6.   The Auto Verify Tool should populate the Component Information entries with the platform
1371   details of the computer. To compare the data to the DPD file stored on the local computer, click
1372   **ReadFile**.
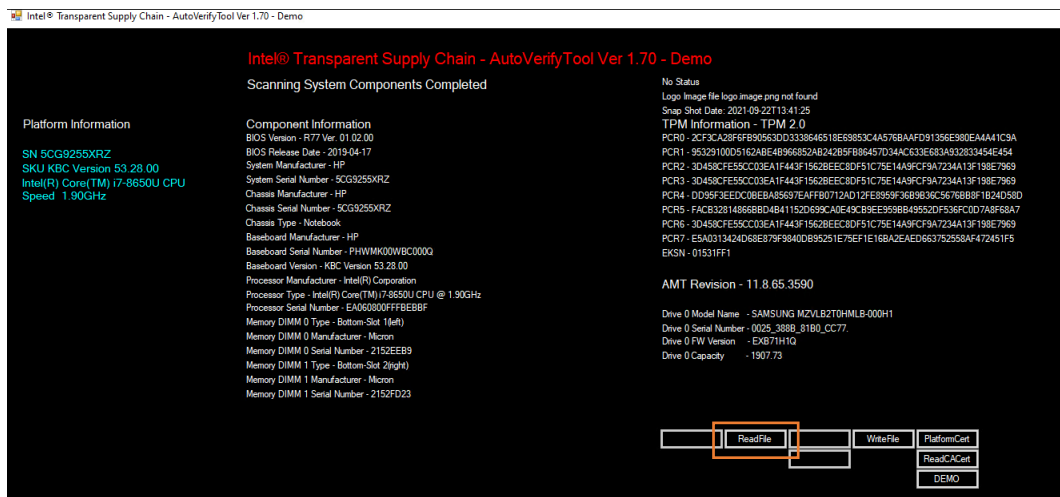
1373   7.   Navigate to the downloaded DPD file and select **Open.**
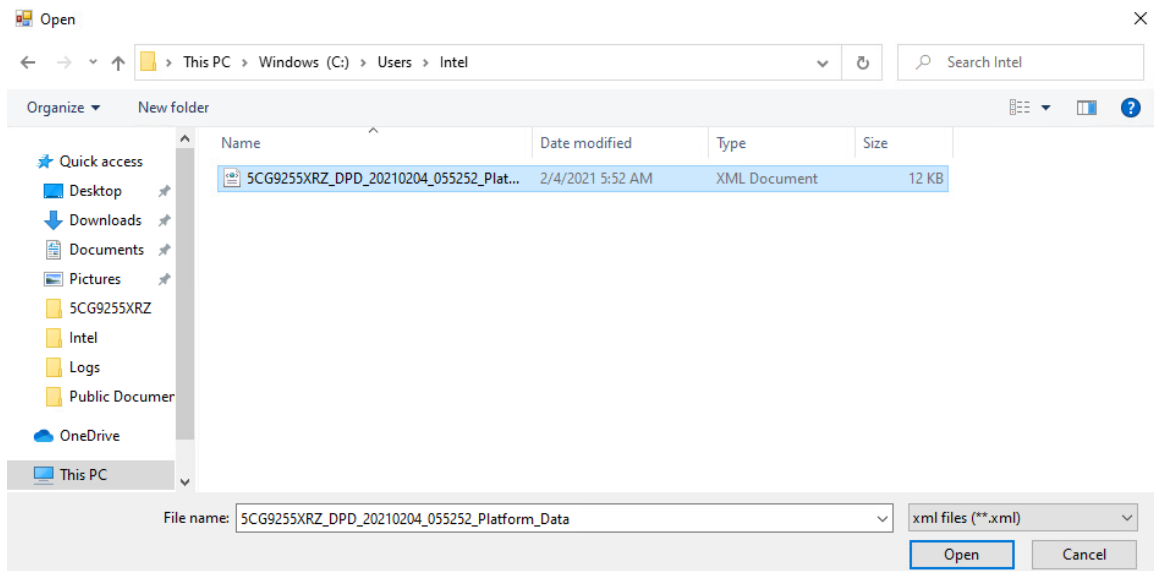
1374   8. Next, click the **Compare** button.
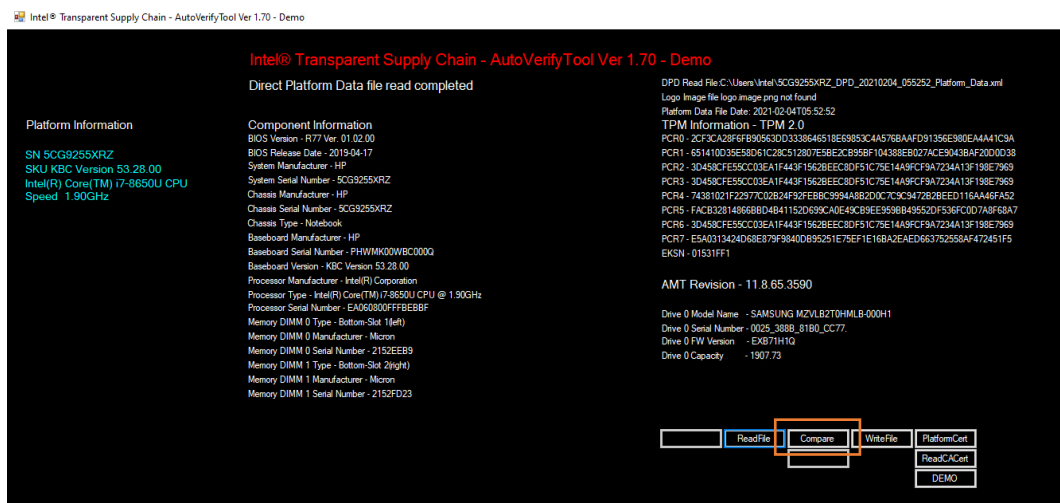
1375   9. If no changes have been made, the Auto Verify tool should output a green message that says,
1376   "**No Component Changes have been detected**." To compare the certificate file, click the
1377   **PlatformCert** button.

DRAFT
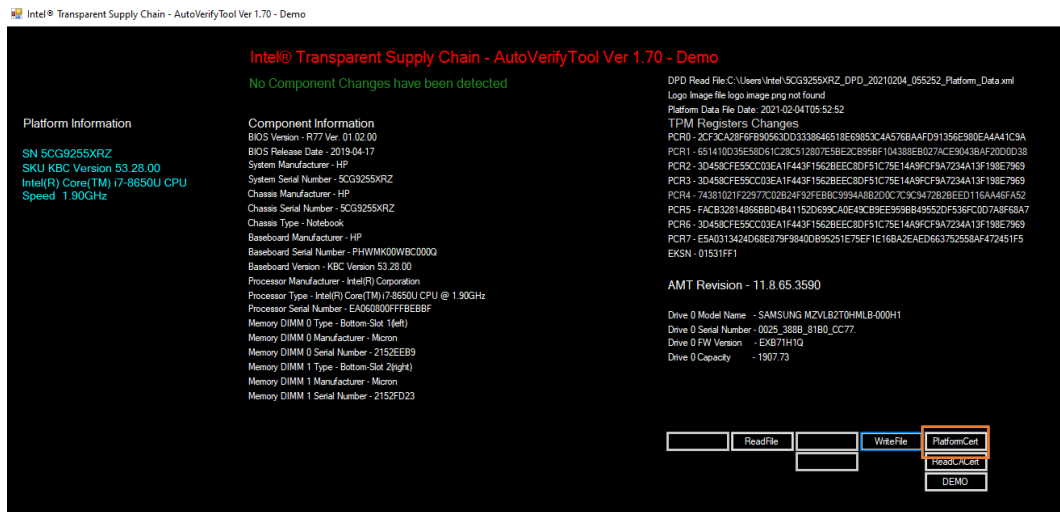
1378   10. Navigate to the location of the platform certificate and select **Open.**
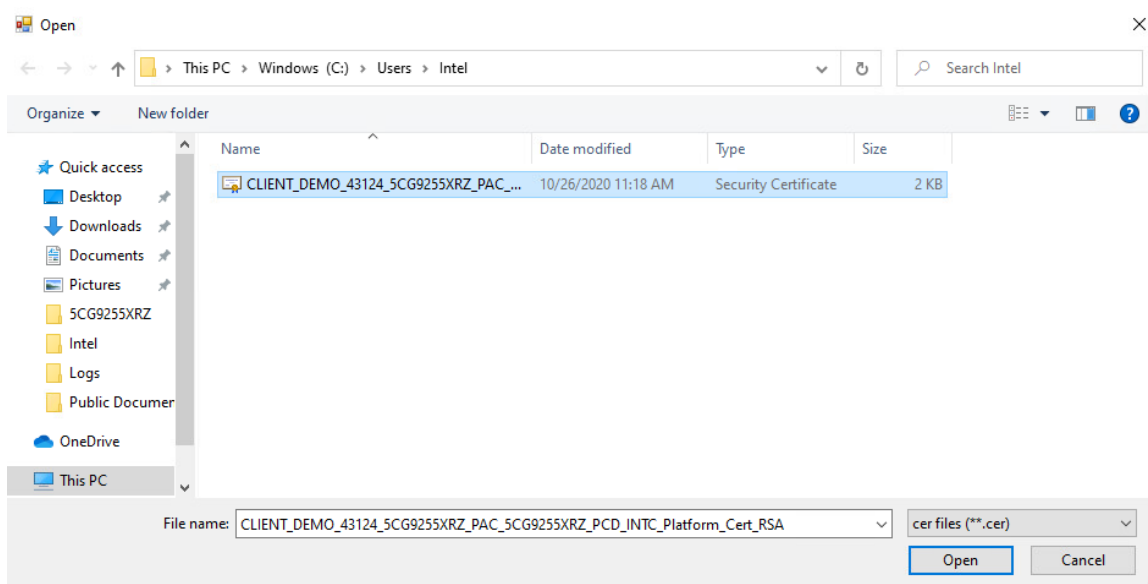
1379   11. If the certificate matches the certificate that the AutoVerify tool detected, the tool will output
1380   another green message that reads "*Platform Certificate Matches*."

3.1.1.4 HPE Servers

1382   1. Boot the target HPE server into the CentOS 8 acceptance testing environment via iPXE. This
1383   requires a one-time boot execution to prevent the server from loading the native OS. Press F11
1384   in the POST screen after a server reboot to access the one-time boot menu and choose the
1385   appropriate network interface card. Then choose *HPE Provisioner Live* from the iPXE boot menu.
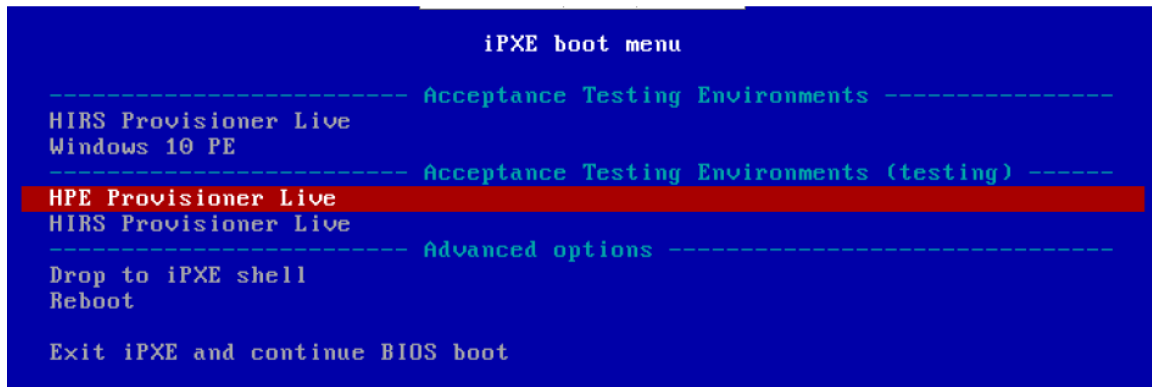
1386   2. Once the live environment has loaded, log in as a user with root privileges. Run the
1387   *hpe_provision.sh script*. The script will attempt to execute the PCVT against the verifiable
1388   artifacts stored in the image. If successful, the script posts the platform manifest to the PMCS.

1389   The script will exit when an error is detected. Refer to the comments in the script to set this up
1390   in your own environment.

3.1.1.5 Dell Servers

1392   1. Boot the target Dell server into the Windows PE acceptance testing environment via iPXE. This
1393   requires a one-time boot execution to prevent the server from loading the native OS. Press F12
1394   in the POST screen after a server reboot to access the one-time PXE boot option and choose the
1395   appropriate network interface card. Then choose *Windows 10 PE* from the iPXE boot menu.
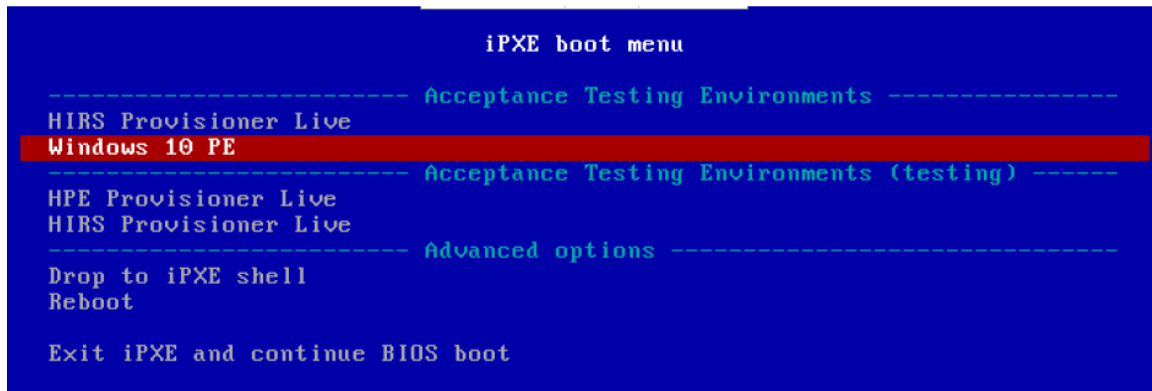
1396   2. Once the live environment has loaded, log in as a user with root privileges. Run the *dell-server-
1397   scv.ps1* script. The script will attempt to execute the Dell Secured Component Verification (SCV)
1398   tool against the verifiable artifacts stored on the server. If successful, the script posts the
1399   platform manifest to the PMCS.

| 1400 | | The script will exit when an error is detected. Refer to the comments in the script to set this up |
| 1401 | | in your own environment. |

3.1.1.6 Intel Server

| 1403 | 3. | Boot the Intel Server into the CentOS 8 host OS environment. Note that for the demonstration |
| 1404 | | Intel server, a network-booted acceptance testing environment was not implemented. |

1405	4.	Once the operating system has completed booting, log in as a user with root privileges. Run the
1406		*provision.sh* script. The script will attempt to execute the *TSCVerifyUtil* against the verifiable
1407		artifacts stored on the server. If successful, the script posts the platform manifest to the PMCS.

1408		The script will run *TSCVerifyUtil* again with different command arguments which directs the
1409		program to access the Seagate drive APIs. If successful, the drive attestation data and
1410		measurements are posted to the PMCS.

| 1411 | | The script will exit when an error is detected. Refer to the comments in the script to set this up |
| 1412 | | in your own environment. |

3.1.2 Asset Inventory and Discovery

1414 Organizational members with access to the enterprise database of computing devices can access a
1415 listing by authenticating to the Archer system. We have configured our instance to display only the
1416 relevant Archer solution menus. In Figure 3-1, the administrator clicks the *SCA Devices* menu link to
1417 retrieve the listing.

1418 **Figure 3-1 Archer Solution Menu**

1419
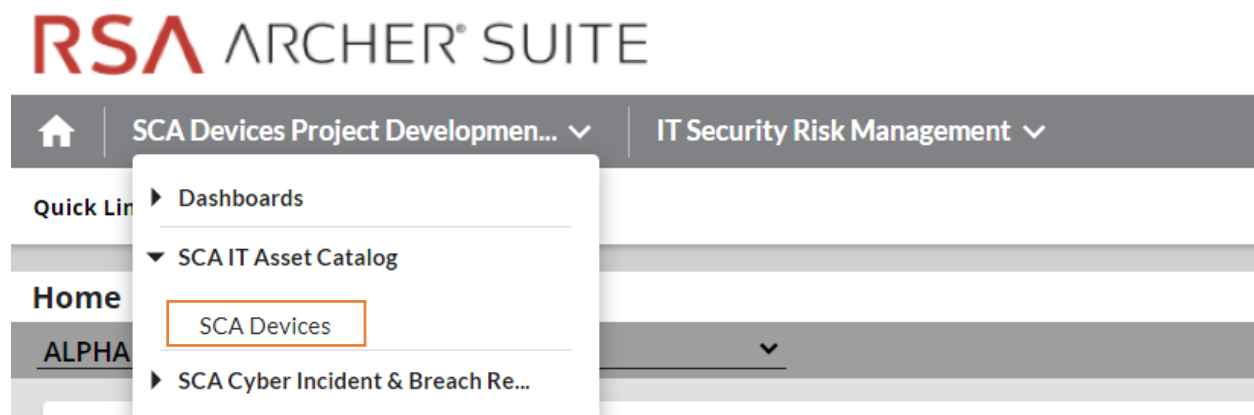
1420 Figure 3-2 shows a listing of all enterprise computing devices that have had their platform validated in
1421 accordance with Scenario 2. The computing device *Enterprise Unique Identifier* is hyperlinked and when
1422 clicked displays additional data, as described below.

1423   **Figure 3-2 Enterprise Computing Devices Listing**

1424

1425   Figure 3-3 shows a representative laptop computing device that has completed the acceptance testing
1426   process by an IT administrator. In the **General Information** section, we have opted to display
1427   characteristics that are common across all the manufacturers in our project such as the serial number
1428   and the make of the computing device. Separately in the **Associated Components** section, we store and
1429   track the components from the initial manufacturer manifest. We will continue to iterate on the asset
1430   inventory user interface to surface meaningful and easily understandable information that is
1431   appropriate for individuals responsible for IT security.

1432   **Figure 3-3 Asset Inventory Screenshot**

1433   For those computing devices that support Eclypsium during acceptance testing, Archer retrieves the
1434   initial firmware data from the Eclypsium backend (cloud or on-premises) and displays it in the Eclypsium
1435   Firmware Analytics section of the record as shown in Figure 3-4.

1436   **Figure 3-4 Eclypsium Acceptance Testing Firmware Data**

1437   *3.1.2.1   Manufacturer-Specific Attributes*

1438   As described in Volume B of this guide, this demonstration also collects manufacturer-specific platform
1439   integrity attributes in addition to the agnostic data described above. For HP Inc. laptops, BIOS
1440   configuration settings, represented as UEFI variables, are associated with the laptop in the asset
1441   inventory when available. From this perspective the security operator is able to view each variable
1442   value, description, and the recommended setting for each value. The operator is also alerted if the
1443   variable value has changed since the initial baseline (column 2), where a remediation action could be
1444   initiated.

1445   Computing devices that use the Intel Transparent Supply Chain platform declare (if present) additional
1446   attributes such as values for the OEM, original design manufacturer (ODM), model, product name, stock
1447   keeping unit (SKU), and product family. The screenshot below is an example from a demonstration
1448   laptop asset inventory record.

1449 Finally, each Seagate drive asset inventory entry displays associated data from its firmware attestation
1450 and measurement capabilities. The security operator can view the currently running version of the
1451 firmware and click on the Tracking ID hyperlink for more details associated with the firmware. In the
1452 lower section, the Firmware Hash Status column informs the operator if measurement values have
1453 changed since the baseline, which may indicate an integrity issue that requires remediation.

▼ SEAGATE FIRMWARE ATTESTATION (SEAGATE DRIVE SERIAL)

First Published	Firmware Version	Tracking ID
5/2/2022 4:26 PM	0x01	277346
5/2/2022 4:26 PM	0x01	277348
5/2/2022 4:26 PM	0x01	277349

▼ SEAGATE FIRMWARE HASH (SEAGATE DRIVE)

Firmware Hash Status	Tracking ID
No Change Detected	277347

3.2 Scenario 3: Verification of Components During Use

1454

1455 In this scenario, the computing device has been accepted by the organization (Scenario 2) and has been
1456 provisioned for the end user. The computing device components are verified against the attributes and
1457 measurements declared by the manufacturer or purchasing organization during operational usage.

1458 The general execution steps are as follows:

1459 1. The end user takes ownership of the computing device from the IT department and uses it to
1460 perform daily work tasks within the scope of normal duties.

1461 2. The computing device creates a report that attests to the platform attributes, such as device
1462 identity, hardware components, and firmware measurements that can be identified by
1463 interrogating the platform.

1464 3. The attestation is consumed and validated by existing configuration management systems used
1465 by the IT organization as part of a continuous monitoring program.

1466 4. The measured state of the device is maintained and updated as the authorized components of
1467 the device are being maintained and associated firmware is updated throughout the device's
1468 operational life cycle.

1469 5. Optionally, the IT administrator takes a remediation action against the computing device if it is
1470 deemed out of compliance. For example, the computing device could be restricted from
1471 accessing certain corporate network resources.

1472 3.2.1 Technology Configurations

1473 *3.2.1.1 Monitoring Using Intel and HIRS-ACA Validation Clients*

1474 This section describes the steps that monitor for unexpected component changes using Intel TSC/HIRS-
1475 ACA tooling and Microsoft Configuration Manager capabilities.

1476 3.2.1.1.1 Deploy Baseline

1477 1. Navigate to the newly created configuration baseline located at **Assets and Compliance >**
1478 **Overview > Compliance Settings > Configuration Baselines.**
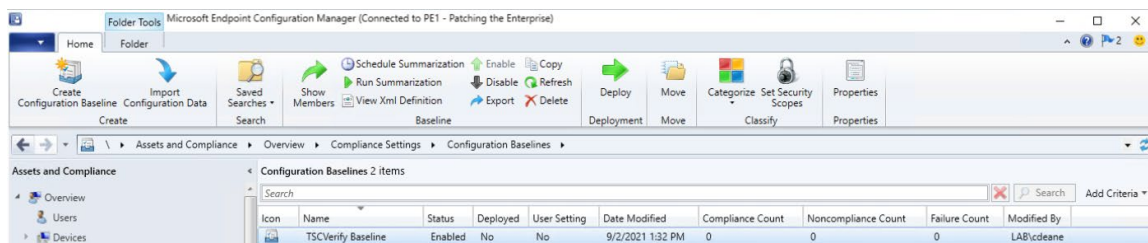
1479 2. Right-click on the configuration baseline and select **Deploy.**
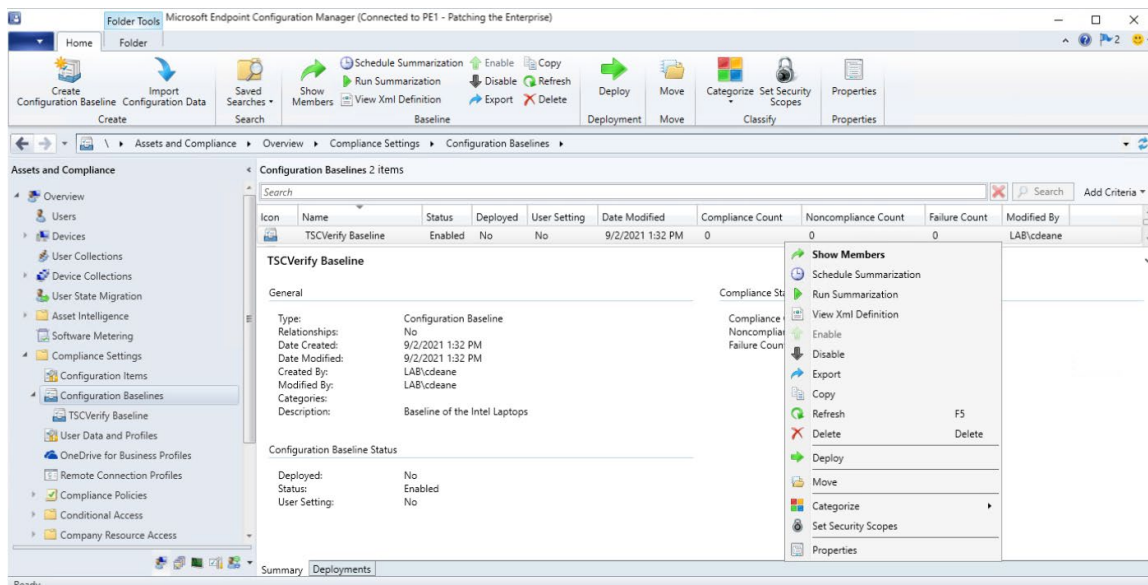
1480 3. Select the device collection for the Intel TSC-supported machines. For this project, the device
1481 collection is named **Intel.** Select **OK.**
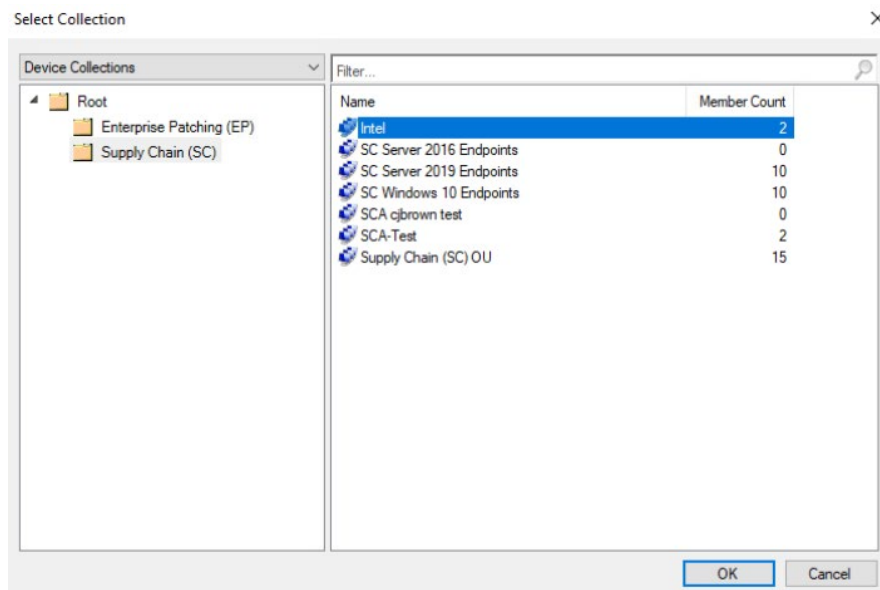
1482   4. Ensure that the baseline is selected and then select the desired frequency of when to run the
1483   baseline. Select **OK.**

1484 This completes the configuration for Intel TSC platform validation tools. Repeat this section to create a
1485 similar baseline for Dell and HP Inc. laptops that leverage HIRS-ACA platform validation tools.

1486 ### 3.2.1.2 Updating the Platform Verifiable Artifact During Operational Use

1487 During the operational use of a computing device, a member of security operations may observe a
1488 warning in a computing device's asset record that it is out of compliance. This could indicate that the
1489 platform has been updated but the change has not been reflected in the verifiable artifact. Archer will
1490 continue to display this warning until the verifiable artifact is updated with the new platform manifest.
1491 Figure 3-5 illustrates this scenario.

1492   **Figure 3-5 Out of Policy Computing Device**

1493   Address the policy warning by using the following procedures to create a Delta Platform Certificate on
1494   HP Inc. and Dell laptops which reflects changes in the target platform components. A Delta Platform
1495   Certificate can be created in Linux or Windows; however, this demonstration only exercises creation on
1496   the Windows platform.

1497   Ensure the following prerequisites are met:

1498   ▪   The administrator has installed PACCOR onto the target laptop.

1499   ▪   A base Platform Certificate has been created and configured in the HIRS ACA. Creation of a Delta
1500   Platform Certificate is dependent on the existence of another base Platform Certificate for the
1501   same laptop.

1502   Next, complete the following procedures to create a Delta Platform Certificate.

1503   5.   Open a command prompt as an Administrator on the target laptop. Change directories to the
1504   following:

1505   `<paccor install folder>\scripts\windows`

1506   6.   Create a directory named *pc_testgen* in the working directory from the previous step if it does
1507   not already exist.

1508   7.   Retrieve the base Platform Certificate from the HIRS ACA portal or other means. Change the
1509   filename of the Platform Certificate to *holder.crt* and place it into the *pc_testgen* directory.

1510   8.   Execute PACCOR's component gathering script and capture the output with the following
1511   command.

1512   `powershell -ExecutionPolicy Bypass ./allcomponents.ps1 components.json`

1513   9.   The component list needs to be manually edited to reflect added, modified, or removed
1514   components of the system. Using a JSON file editor, open the *components.json* file.

1515   a.   In the **COMPONENTS** object, identify the objects that represent components to be
1516   saved in the new Delta Platform Certificate. Add a **STATUS** field at the end of these
1517   components with a value of **ADDED, MODIFIED,** or **REMOVED**. For example, to modify
1518   the chassis serial number, create a **COMPONENTS** entry similar to the following.

```
1519                    {
1520                      "COMPONENTS": [
1521                        {
1522                          "COMPONENTCLASS": {
1523                            "COMPONENTCLASSREGISTRY": "2.23.133.18.3.1",
1524                            "COMPONENTCLASSVALUE": "00020001"
1525                          },
1526                          "MANUFACTURER": "Example Manufacturer",
1527                          "MODEL": "1",
1528                          "SERIAL": "1234",
1529                          "STATUS": "MODIFIED"
1530                        }
1531                      ]
1532                    }
```

1533   b.   Delete all other objects under **COMPONENTS.**

1534   c.   Once finished editing the *components.json* file, move it to the **pc_testgen** folder.

1535   10. Using a text editor, modify the **pc_certgen** script header variables.

1536   a.   Set the **ekcert** variable to point to **holder.crt** from step 3.

1537   b.   Set the **componentlist** variable to point to **components.json** from step 5.

1538   c.   Change the value of **serialnumber** to 0002.

1539   d.   If you have a specific signing key and cert, move those files to **pc_testgen** as well and
1540   update the **sigkey** and **pcsigncert** variables to point to them.

1541   11. Execute the *pc_certgen.ps1* script using the following command:

1542   ```
        powershell -ExecutionPolicy Bypass ./pc_certgen.ps1
        ```

1543    12. The resulting Delta Platform Certificate will be stored in the **pc_testgen** folder.

1544    13. Upload the new Delta Platform Certificate to the HIRS-ACA portal.

1545    Note that laptops that are continuously monitored with the Windows-based HIRS Provisioner will be
1546    evaluated against this new baseline.

## 3.2.2  Dashboards

1548    The dashboard created in Section 2.11.2.3 attempts to consolidate and communicate potential integrity
1549    issues to the IT administrator while computing devices are in operational use. The timeliness of this
1550    information will depend on the cadence that your organization chooses to update the various data feeds
1551    from Microsoft Configuration Manager and the Eclypsium Analytic platform. This demonstration displays
1552    to the administrator if there are detected component swaps from computing devices that can leverage

1553 Intel TSC and HIRS-ACA platforms. Further, it displays any detected firmware platform integrity issues
1554 from the Eclypsium Analytic cloud and on-premises platform across all manufacturers in this prototype.

1555 The Archer IRM dashboard should resemble the screenshots below, where a count of computing devices
1556 with potential integrity issues is displayed (Figure 3-6 and Figure 3-7). Your organization's security
1557 operations team may also want to access the Eclypsium Analytic platform directly to obtain detailed
1558 information, including remediation actions, for computing devices with detected integrity issues.

1559 **Figure 3-6 Dashboard with No Integrity Issues Detected**



1560 **Figure 3-7 Dashboard with Integrity Issues Detected**



1561 The demonstration dashboards are also capable of monitoring manufacturer-specific platform integrity
1562 datapoints. In Figure 3-8, we show a dashboard component that captures the number of UEFI
1563 configuration parameters that have changed from the baseline values (Y-axis) for each HP Inc.
1564 computing device (X-axis).

DRAFT

1565    **Figure 3-8 HP Inc. Laptop Continuous Monitoring**



HP, Inc Laptops with Detected UEFI Configuration Changes

1566    In the final dashboard component, the security operator can display the number of Seagate drives with
1567    firmware hash values that have changed since the initial acceptance testing baseline. In a production
1568    setting, it could be more useful to compare the drive measurements against known values
1569    communicated directly from the manufacturer (Seagate).

### 3.2.3 Platform Integrity Incident Management

1570

1571    The final continuous monitoring scenario we demonstrate is the automated creation of Archer *Incidents*
1572    when the QRadar continuous monitoring data feed (Section 2.11.2.2.4) retrieves a platform integrity
1573    issue. In the asset inventory record shown in Figure 3-9, we have triggered a platform integrity issue in
1574    one of our demonstration HP Inc. laptops, which has automatically created an Archer *Security Incident*.
1575    Note that the Archer platform offers workflow customization options that are not documented here
1576    that can support more complex organizational structures.

1577    **Figure 3-9 New Security Incident**



1578    The security operator can click the hyperlink, which displays more detailed information about the issue.
1579    In the case depicted in Figure 3-10, the *HP Sure Start* capability has flagged a potential issue.

1580    **Figure 3-10 Incident Summary**

▼ INCIDENT SUMMARY

    **Incident ID:** INC-277233        **Source:** IBM Qradar

        **Title:** HP_Sure_Start Integrity violation

    **Incident** HP_Sure_Start Integrity violation
    **Summary:**

      **Incident** This indicates that HP Sure Start has detected that the main drive partition table
      **Details:** has been altered, and HP Sure Start has returned the partition table to the
          desired state.   This event could be indicative of an attack on the device in the
          event the change to the drive partition tables was made by an unauthorized
          party.

1581    In the *Incident Status* section, metadata associated with the incident is displayed, including whether
1582    remediation is requested by the security operator. Figure 3-11 shown an example of this.

1583    **Figure 3-11 Incident Status**

▼ INCIDENT STATUS

    **Date Created:** 2/7/2022 7:42 AM        **Priority:**

      **Date/Time** 3/24/2022 2:13 PM      **Incident** Remediation Requested
      **Modified:**          **Status:**

      **Date/Time**          **Incident** Administrator, ITAdmin
      **Closed:**          **Owner:**

    **Days Open:** 44 Day(s)        **Incident** L1 Incident Handlers
          **Queue:**

      **No. of** 0        **Incident**
    **Aggregated**      **Coordinator:**
      **Alerts:**

          **Members:** Administrator, ITAdmin

1584    If remediation is requested, the security operator clicks the *Remediation* tab within the *Security Incident*
1585    where a suggested action is displayed (see Figure 3-12).

DRAFT

1586    **Figure 3-12 Incident Remediation Action**



| Overview | Impact Analysis | **Remediation** | Results |

▼ REMEDIATION ACTION REQUIRED

**Remediation Required?:** Yes

**Remediation Action:** Restrict the computing device from sensitive corporate network resources.

**Appendix A     List of Acronyms**

| | |
|---|---|
| **ACA** | Attestation Certificate Authority |
| **AD** | Active Directory |
| **ADK** | (Windows) Assessment and Deployment Kit |
| **API** | Application Programming Interface |
| **AQL** | (IBM QRadar) Ariel Query Language |
| **BIOS** | Basic Input/Output System |
| **CMSL** | (HP) Client Management Script Library |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DNS** | Domain Name System |
| **DPD** | Direct Platform Data |
| **DTD** | Dell Trusted Device |
| **FQDN** | Fully Qualified Domain Name |
| **HIRS** | Host Integrity at Runtime and Start-Up |
| **HPE** | Hewlett Packard Enterprise |
| **HTTP** | Hypertext Transfer Protocol |
| **IIS** | (Microsoft) Internet Information Services |
| **IP** | Internet Protocol |
| **IRM** | (Archer) Integrated Risk Management |
| **IT** | Information Technology |
| **JDK** | Java Development Kit |
| **JSON** | JavaScript Object Notation |
| **NCCoE** | National Cybersecurity Center of Excellence |
| **NIST** | National Institute of Standards and Technology |
| **NTP** | Network Time Protocol |
| **ODM** | Original Design Manufacturer |

| | |
|---|---|
| **OEM** | Original Equipment Manufacturer |
| **OS** | Operating System |
| **PC** | Personal Computer |
| **PCVT** | (HPE) Platform Certificate Verification Tool |
| **PM2** | Process Manager 2 |
| **PMCS** | Platform Manifest Correlation System |
| **PXE** | Preboot Execution Environment |
| **REST** | Representational State Transfer |
| **SAS** | Serial Attached SCSI |
| **SCA** | Supply Chain Assurance |
| **SCRM** | Supply Chain Risk Management |
| **SCSI** | Small Computer System Interface |
| **SCV** | (Dell) Secured Component Verification |
| **SKU** | Stock Keeping Unit |
| **SP** | Special Publication |
| **SSMS** | (Microsoft) SQL Server Management Studio |
| **TB** | Terabyte |
| **TCG** | Trusted Computing Group |
| **TEI** | (NCCoE) Trusted Enterprise Infrastructure |
| **TFTP** | Trivial File Transfer Protocol |
| **TPM** | Trusted Platform Module |
| **TSC** | (Intel) Transparent Supply Chain |
| **UEFI** | Unified Extensible Firmware Interface |
| **UI** | User Interface |
| **URL** | Uniform Resource Locator |
| **UUID** | Universally Unique Identifier |

DRAFT

**WinPE**        Windows Preinstallation Environment

**XML**          Extensible Markup Language

# Appendix B    Archer Applications

The following tables detail the data fields in each Archer application for use in . The first column is the name of the data field we used in this demonstration and the second column is the data type. Data fields that are calculated are indexed in the third column and available in the subsequent table. Bolded rows are *Key Fields*, similar to a primary key.

**Table 3-1 Devices Application**

| Data Field Name | Data Field Type | Calculated |
|---|---|---|
| Associated Components | Cross-Reference | |
| Last Event Timestamp | Date | |
| Last System Scan Date | Date | |
| System Firmware Date | Date | |
| Firmware Integrity Aggregation Status | Numeric | |
| Firmware Integrity Check Status | Numeric | |
| Count of Failed Configuration Scan Results | Text | |
| Count of Configuration Scans | Text | |
| **Enterprise Unique Identifier** | **Text** | |
| Family | Text | |
| Platform Model | Text | |
| Model | Text | |
| Original Design Manufacturer | Text | |
| Original Equipment Manufacturer | Text | |
| Product Name | Text | |
| SKU | Text | |
| System Firmware Version | Text | |
| Manufacturer | Values List | |
| Device Scan State | Values List | 1 |
| Eclypsium Integrity Scan Status | Values List | 2 |
| Continuous Monitoring Platform Integrity Status | Values List | 3 |

1595 **Table 3-2 Calculated Fields (Devices)**

| Index | Calculation |
|---|---|
| 1 | `IF (ISEMPTY([Helper Previous Last Scanned Date Calc]), VALUEOF([Device Scan State],"New"),`<br>`IF (DATEDIF([Helper Max Last Scanned Date Calc],[Helper Previous Last Scanned Date Calc])=0, [Device Scan State], VALUEOF([Device Scan State], "Matched")))` |
| 2 | `IF (ISEMPTY([Firmware Integrity Check Status]), VALUEOF([Eclypsium Integrity Scan Status], "No Data"),`<br>`IF ([Firmware Integrity Check Status]=1, VALUEOF([Eclypsium Integrity Scan Status], "No Integrity Issues Detected"),`<br>`IF ([Firmware Integrity Check Status]=0, VALUEOF([Eclypsium Integrity Scan Status], "Integrity Issue Detected - Action Recommended"))))` |
| 3 | `IF (ISEMPTY([Continuous Monitoring Platform Integrity Status]),`<br>`VALUEOF([Continuous Monitoring Platform Integrity Status], "No Data from Configuration Management System"))` |

1596 **Table 3-3 Components Application**

| Data Field Name | Data Field Type |
|---|---|
| Addresses | Text |
| Class | Text |
| Field Replaceable | Text |
| First Published | First Published Date |
| Free Text | Text |
| Last Updated | Last Updated Date |
| Manufacturer | Text |
| Model | Text |
| Platform Certificate | Text |
| Platform Certificate URI | Text |
| Revision | Text |
| SCA Devices (Associated Components) | Related Records |
| Seagate Firmware Attestation (Seagate Drive Serial) | Related Records |
| Seagate Firmware Hash (Seagate Drive) | Related Records |
| Serial | Text |
| **Tracking ID** | **Tracking ID** |
| Version | Text |
| Associated Components | Cross-Reference |

1597    **Table 3-4 HP UEFI Configuration Variables Application**

| Data Field Name | Data Field Type | Calculated |
|---|---|---|
| Associated Computing Device | Cross-Reference | |
| CompositeUUIDVariable | Text | 1 |
| **Computing Device UUID** | **Text** | |
| First Published | First Published Date | |
| HP Inc BIOS Configuration Status | Values List | |
| Last Updated | Last Updated Date | |
| Tracking ID | Tracking ID | |
| UEFI Variable Description | Text | 2 |
| UEFI Variable Friendly Name | Text | |
| UEFI Variable Name | Text | |
| UEFI Variable Possible Values | Text | 3 |
| UEFI Variable Recommended Values | Text | 4 |
| Value | Text | |

1598    **Table 3-5 Calculated Fields (HP UEFI Configuration Variables)**

| Index | Calculation |
|---|---|
| 1 | `CONCATENATE([Computing Device UUID],[UEFI Variable Name])` |
| 2 | `IF ([First Published]<>[Last Updated], "Change Detected",`<br>`IF ([First Published]=[Last Updated], "No Change Detected"))` |
| 3 | `IF ([UEFI Variable Name]="SS_SB_KeyProt", "Provides enhanced protection of the secure boot databases and keys used by BIOS to verify the integrity and authenticity of the OS bootloader before launching it at boot.",`<br>`IF ([UEFI Variable Name]="FW_RIPD", "Utilizes specialized hardware in the platform chipset to prevent, detect, and remediate anomalies in the Runtime HP SMM BIOS.",`<br>`IF ([UEFI Variable Name]="TL_Power_Off", "HP Tamperlock feature: The system immediately turns off if the cover is removed while the system is On or in Sleep state S3 or Modern Standby.",`<br>`IF ([UEFI Variable Name]="TL_Clear_TPM", "TPM is cleared on the next startup after the cover is removed. Be aware that all customer keys in the TPM are cleared. This setting should only be Enabled in a situation where manual recovery is possible using remote backups, or no recovery is desired. In the case of BitLocker being enabled, the BitLocker recovery key is required to decrypt the drive.",` |

| Index | Calculation |
|---|---|
| | IF ([UEFI Variable Name]="SS_GPT_HDD", "HP Sure Start maintains a protected backup copy of the MBR/GPT partition table from the primary drive and compares the backup copy to the primary on each boot. If a difference is detected, the user is prompted and can choose to recover from the backup to the original state, or to update the protected backup copy with the changes.",<br>IF ([UEFI Variable Name]="SS_GPT_Policy", "Defines Sure Start behavior to either Local User Control or Automatic to restore the MBR/GPT to the saved state any time differences are encountered.",<br>IF ([UEFI Variable Name]="DMA_Protection", "BIOS will configure IOMMU hardware for use by operating systems that support DMA protection.",<br>IF ([UEFI Variable Name]="PreBoot_DMA", "IOMMU hardware-based DMA protection is enabled in a BIOS pre-boot environment for Thunderbolt and / or all internal and external PCI-e attached devices.",<br>IF ([UEFI Variable Name]="Cover_Sensor", "Policy defined actions taken when Tamperlock cover removal sensor is triggered. Administrator credential or password requires valid response before continuing to startup after the cover is opened.",<br>IF ([UEFI Variable Name]="", "No Description", "No Description")<br>)))))))) |
| 4 | IF ([UEFI Variable Name]="SS_SB_KeyProt", "[Disable, Enable]",<br>IF ([UEFI Variable Name]="FW_RIPD", "[Disable, Enable]",<br>IF ([UEFI Variable Name]="TL_Power_Off", "[Disable, Enable]",<br>IF ([UEFI Variable Name]="TL_Clear_TPM", "[Disable, Enable]",<br>IF ([UEFI Variable Name]="SS_GPT_HDD", "[Disable, Enable]",<br>IF ([UEFI Variable Name]="SS_GPT_Policy", "[Local user control, Recover in event of corruption]",<br>IF ([UEFI Variable Name]="DMA_Protection", "[Disabled, Enabled]",<br>IF ([UEFI Variable Name]="PreBoot_DMA", "[Thunderbolt Only, All PCI-e Devices]",<br>IF ([UEFI Variable Name]="Cover_Sensor", "[Disable, Notify user, Administrator Credential, Administrator Password]",<br>IF ([UEFI Variable Name]="", "No Possible Values", "No Possible Values")<br>)))))))) |

1599 **Table 3-6 Seagate Firmware Attestation Application**

| Data Field Name | Data Field Type |
|---|---|
| Assessor Identifier | Text |
| Associated Computing Device | Cross-Reference |
| Device Nonce | Text |
| Firmware Version | Text |
| First Published | First Published Date |
| Last Updated | Last Updated Date |
| Root of Trust Identifier | Text |

DRAFT

| Data Field Name | Data Field Type |
|---|---|
| Root of Trust Nonce | Text |
| Seagate Drive Serial | Cross-Reference |
| Secure Boot Device State | Text |
| Signing Auth Database | Text |
| **Tracking ID** | **Tracking ID** |

1600 **Table 3-7 Seagate Firmware Hash Application**

| Data Field Name | Data Field Type | Calculated |
|---|---|---|
| Associated Computing Device | Cross-Reference | |
| BFW IDBA Hash | Text | |
| BFW ITCM Hash | Text | |
| CFW Hash | Text | |
| Drive Serial Number | Text | |
| Firmware Hash Status | Values List | 1 |
| First Published | First Published Date | |
| History | History Log | |
| Last Updated | Last Updated Date | |
| Seagate Drive | Cross-Reference | |
| SEE Firmware Hash | Text | |
| SEE Signing AuthN Key Certificate Hash | Text | |
| SERVO Firmware Hash | Text | |
| Signing AuthN Key Certificate Hash | Text | |
| **Tracking ID** | **Tracking ID** | |

1601 **Table 3-8 Calculated Fields (Seagate Firmware Hash)**

| Index | Calculation |
|---|---|
| 1 | `IF ([First Published]<>[Last Updated], "Change Detected",`<br>`IF ([First Published]=[Last Updated], "No Change Detected"))` |