NIST SPECIAL PUBLICATION 1800-34

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

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/building-blocks/supply-chain-assurance

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

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

NIST SPECIAL PUBLICATION 1800-34

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

NIST SPECIAL PUBLICATION 1800-34A

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

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/building-blocks/supply-chain-assurance

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

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

Executive 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 project will demonstrate how organizations can verify that the internal components and system firmware of the computing devices they acquire are genuine and have not been unexpectedly altered during manufacturing, distribution, or operational use.

CHALLENGE

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

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

SOLUTION

To address these challenges, the NCCoE is collaborating with technology vendors to develop a prototype implementation in harmony with the National Initiative for Improving Cybersecurity in Supply Chains (NIICS), which emphasizes tools, technologies, and guidance focused on the developers and providers of technology. NIICS' mission is to help organizations build, evaluate, and assess the cybersecurity of products and services in their supply chains. This project aligns with that mission by demonstrating how organizations can verify that the internal components of the computing devices they acquire are genuine and have not been tampered with. This prototype 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. By doing this, organizations can reduce the risk of compromise to products within their supply chains.

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

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

This project will address several processes, including:

- how to create verifiable descriptions of components and platforms, which may be done by original equipment manufacturers (OEMs), platform integrators, and even information technology (IT) departments;
- how to verify devices and components within the single transaction between an OEM and a customer; and
- how to verify devices and components at subsequent stages in the system lifecycle in the operational environment.

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

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

hp	Manufacturer, Platform Integrity Validation System
Hewlett Packard Enterprise	Manufacturer, Platform Integrity Validation System
IBM	Security Information and Event Management
intel	Manufacturer, Platform Integrity Validation System
National Security Agency, United States of America	Certificate Authority, Platform Integrity Validation System
SEAGATE GOVERNMENT SOLUTIONS	Manufacturer, Platform Integrity Validation System

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

HOW TO USE THIS GUIDE

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

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

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

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

SHARE YOUR FEEDBACK

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

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

COLLABORATORS

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

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.

NIST SPECIAL PUBLICATION 1800-34B

Validating the Integrity of Computing Devices

Volume B:
Approach, Architecture, and Security Characteristics

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*

June 2022

DRAFT

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/building-blocks/supply-chain-assurance

DISCLAIMER

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

National Institute of Standards and Technology Special Publication 1800-34B, Natl. Inst. Stand. Technol. Spec. Publ. 1800-34B, 72 pages, (June 2022), CODEN: NSPUE2

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.

KEYWORDS

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

ACKNOWLEDGMENTS

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

Name	Organization
Themistocles Chronis	Archer
Dan Carayiannis	Archer
Charles Robison	Dell Technologies
Mukund Khatri	Dell Technologies
Rick Martinez	Dell Technologies
Daniel Carroll	Dell Technologies
Jason Young	Dell Technologies
Travis Raines	Eclypsium
John Loucaides	Eclypsium
Jason Cohen	Hewlett Packard Enterprise
CJ Coppersmith	Hewlett Packard Enterprise
Ludovic Jacquin	Hewlett Packard Enterprise
Boris Balacheff	HP Inc.
Jeff Jeansonne	HP Inc.
Joshua Schiffman	HP Inc.
Harmeet Singh	IBM
Tom Dodson	Intel

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

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

Technology Partner/Collaborator	Build Involvement
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

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

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

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

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

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

Contents

List of Figures

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.

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

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

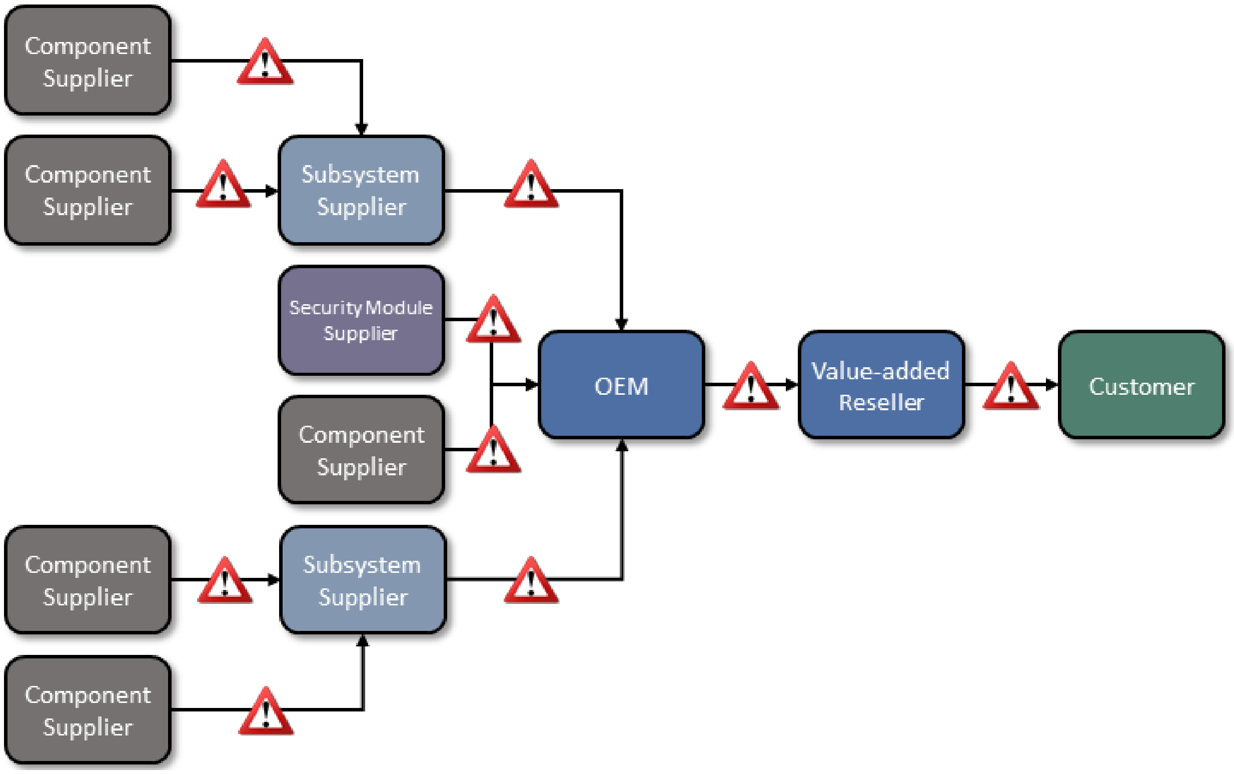
Section 6, Future Build Considerations, conveys the technical characteristics we plan to incorporate as we continue to prototype with our collaborators.

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

1.1 Challenge

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

Figure 1-1 Supply Chain Risk



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

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

1.2 Solution

To address these challenges, the NCCoE is collaborating with technology vendors to develop a prototype implementation. Once completed, this project [1] will demonstrate how organizations can verify that the internal components of the computing devices they acquire are genuine and have not been tampered with. This solution relies on device vendors storing information within each device, and implementers using a combination of commercial off-the-shelf and open-source tools that work together to validate the stored information. By doing this, organizations can reduce the risk of compromise to products within their supply chains.

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

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

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

This project addresses several processes, including:

- how to create verifiable descriptions of components and platforms, which may be done by original equipment manufacturers (OEMs), platform integrators, and even IT departments;
- how to verify the integrity and provenance of computing devices and components within the single transaction between an OEM and a customer; and
- how to continuously monitor the integrity of computing devices and components at subsequent stages in the system lifecycle in the operational environment.

1.3 Benefits

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

- avoid using compromised technology components in your products
- enable customers to readily verify that OEM products are genuine and trustworthy

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

2 How to Use This Guide

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

When completed, this NIST Cybersecurity Practice Guide will demonstrate a standards-based reference design for verifying that the internal components of the computing devices organizations acquire are genuine and have not been tampered with and provide readers with the information they need to replicate the reference design. It 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 **(you are here)**
- NIST SP 1800-34C: *How-To Guides*—instructions for building the example solution

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

Business decision makers, including chief security and technology officers, will be interested in the *Executive Summary, NIST SP 1800-34A*, 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 this part of the guide, *NIST SP 1800-34B*, which describes what we did and why. The following sections will be of particular interest:

- Section 3.4, Risk Assessment, provides a description of 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

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

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

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

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

2.1 Typographic Conventions

The following table presents typographic conventions used in this volume.

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

3 Approach

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

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

3.1 Audience

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

3.2 Scope

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

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

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

3.2.1 Scenario 1: Creation of Verifiable Platform Artifacts

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

3.2.2 Scenario 2: Verification of Components During Acceptance Testing

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

3.2.3 Scenario 3: Verification of Components During Use

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

3.3 Assumptions

This project is guided by the following assumptions:

- The scenario activities above will augment, not replace, the capabilities of existing acceptance testing tools, asset management systems, and configuration management systems.
- Hardware roots of trust represent one technique that can thwart the above types of attacks to the supply chain. However, OEMs may use different approaches to implement a hardware root of trust solution because of hardware constraints or other business reasons.
- Organizational computing devices lifecycle phases for technology include the following activities defined in NIST SP 800-161 Revision 1, Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations [3]: integration (referred to as acceptance testing in this demonstration), operations, and disposal.

3.4 Risk Assessment

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

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

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

3.4.1 Threats

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

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.

3.4.3.1 Threat Scenario

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

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

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

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

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

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

- loss of intellectual property through data exfiltration
- loss of employee productivity as a result of computers and network equipment being taken offline
- additional costs to the IT department for replacement computers and network equipment
- loss of confidence with the company's client base
- potential loss of revenue due to clients severing their relationship with the company

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

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

The following tables map the security characteristics defined in our project description (Table 3-3) to the applicable NIST Cybersecurity Framework [6] Functions, Categories, and Subcategories (Table 3-4) to assist organizations better manage and reduce C-SCRM risk. We have also included a mapping to specific SP 800-53 r5 security controls [7] and indicated (in bold) if the control is part of the SP 800-161 Revision 1 [3] baseline security controls to assist organizations interested in alignment with NIST C-SCRM best practices.

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.

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 Characteristics Addressed
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.

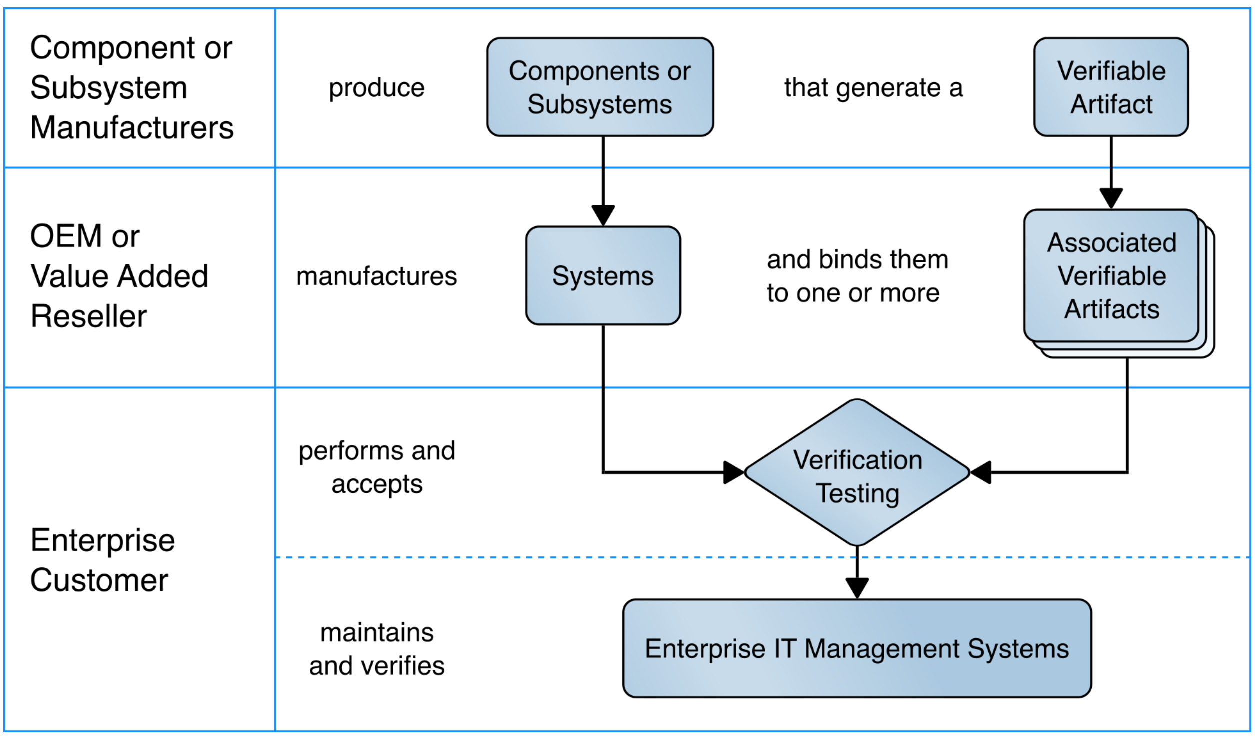
This project applies this foundational technology to address the challenge of operational security by verifying the provenance of a delivered system from the time it leaves the manufacturer until it is introduced in the organization's operational environment. The TPM can be leveraged to measure and validate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

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

Figure 4-1 Notional Architecture

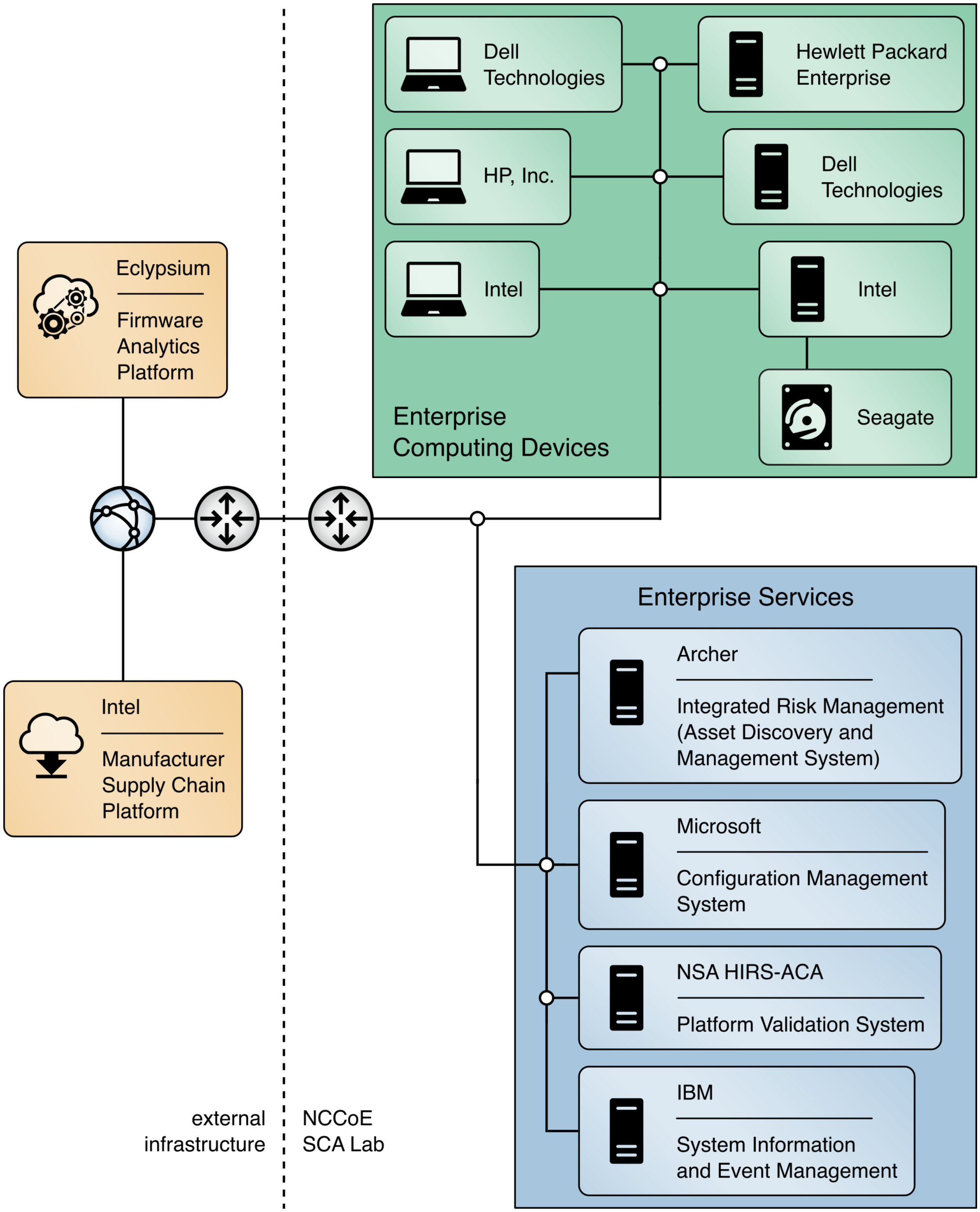


4.1 Architecture Description

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

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

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

4.2.2.1 Archer Integrated Risk Management (IRM)Platform

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

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

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

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

We modeled the structure of the Components application and made configurations to the Devices application via data fields to mimic the structure of the TCG Platform Certificate Profile as a vendor-agnostic method of storing data such as manufacturer, model, and version information. For organizations using the broader Archer IRM platform capabilities, such as their Enterprise and Operational Risk Management or Third-Party Risk Management solutions, records (computing devices) stored in the Devices application can also be associated with other aspects of the enterprise infrastructure [10].

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

4.2.3 Configuration Management System

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

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

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

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

Configuration monitoring is the process for assessing or testing the level of compliance with the established baseline configuration and mechanisms for reporting on the configuration status of items placed under configuration management. This prototype demonstration uses a combination of monitoring capabilities provided by the configuration management system and OEM platform validation tooling to assess whether the computing device has deviated from the defined baseline configuration.

4.2.3.1 Microsoft Endpoint Configuration Manager

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

The computing device baseline configuration (defined above) was evaluated using the compliance settings capability. In the Intel laptop use case, we defined a configuration item which deployed a custom PowerShell script to each Intel computing device. The script executed the `TSCVerifyUtil` tool 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.

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

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

4.2.4 Enterprise Dashboards

Many organizations leverage informational dashboards that provide security information on a continuing basis to give, as SP 800-53 Revision 5 notes, “organizational officials the ability to make effective and timely risk management decisions, including ongoing authorization decisions.” An information management console or *dashboard* in the context of this prototype is a tool that consolidates and communicates platform integrity status relevant to the organizational security posture in near-real-time to security management stakeholders [9]. This demonstration uses an enterprise SIEM dashboard capability to support the continuous monitoring described in Scenario 3.

4.2.4.1 Archer Integrated Risk Management (IRM) Platform

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

1. Act and investigate the computing device by viewing the associated asset management data.
2. Review and initiate remediation and recovery capabilities.

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

4.3 Supporting Platform Integrity Validation Systems

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

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

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

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

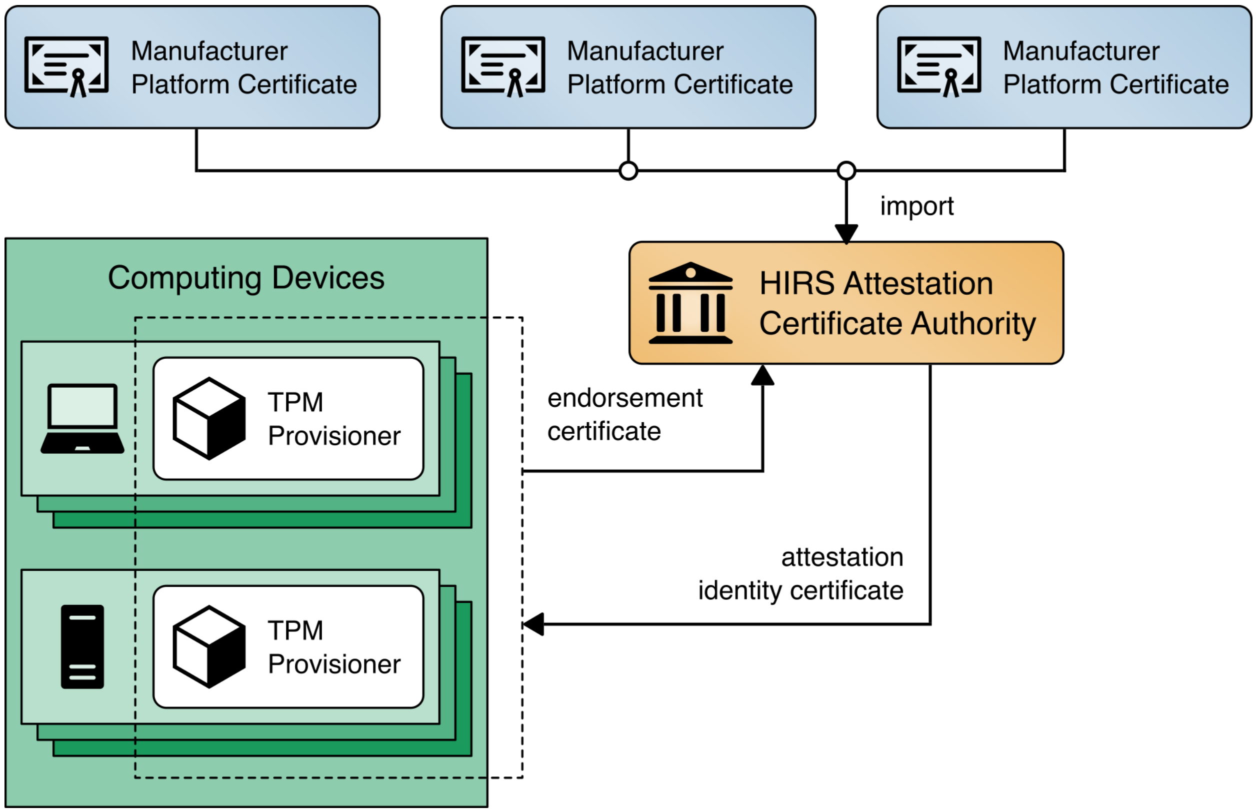
The HIRS ACA project is comprised of multiple components and services that are utilized in this prototype demonstration. The first component, named the TPM Provisioner, is a software utility executed on the target computing device. It takes control of the TPM if it is not already owned and requests an AIC for the TPM from the Attestation Certificate Authority (ACA, described below). The Provisioner communicates with the ACA through a REST API interface to complete the transaction. As part of the transaction, the TPM Provisioner reads the Endorsement Key credentials from the TPM's non-volatile random-access memory (NVRAM) and interrogates the computing device's hardware, network, firmware, and OS info for platform validation. The previous version of this publication documented the TPM Provisioner as applied to acceptance testing of the computing devices. In this revision, we demonstrate the use of a pre-release version of a Windows-based version of the TPM Provisioner for continuous monitoring-based scenarios.

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

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

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

Figure 4-3 HIRS ACA Platform

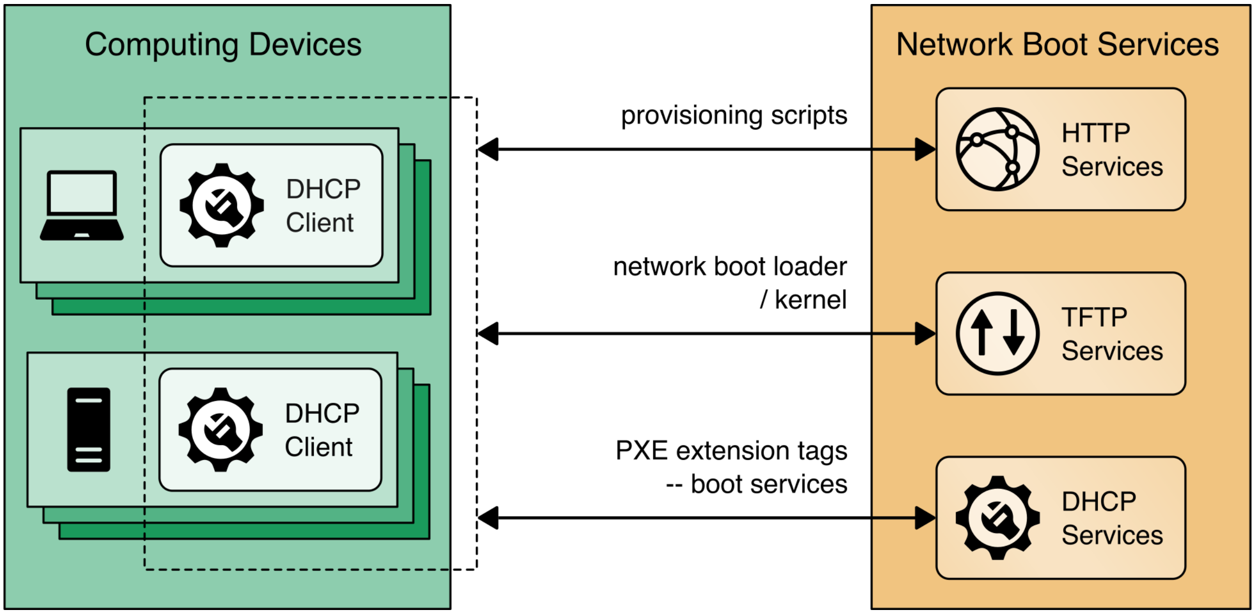


4.3.2 Network Boot Services

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

1. Computing devices are configured to boot over the network via a network interface card (NIC). The DHCP server presents the boot options to the IT administrator. Once the OS is chosen, the DHCP server directs the DHCP client to the Trivial File Transfer Protocol (TFTP) server.
2. The DHCP client downloads and executes boot loaders and kernels associated with the target OS.
3. The IT administrator downloads the latest provisioning script from a centralized repository.

Figure 4-4 Network Boot Services Environment



4.3.3 Platform Manifest Correlation System

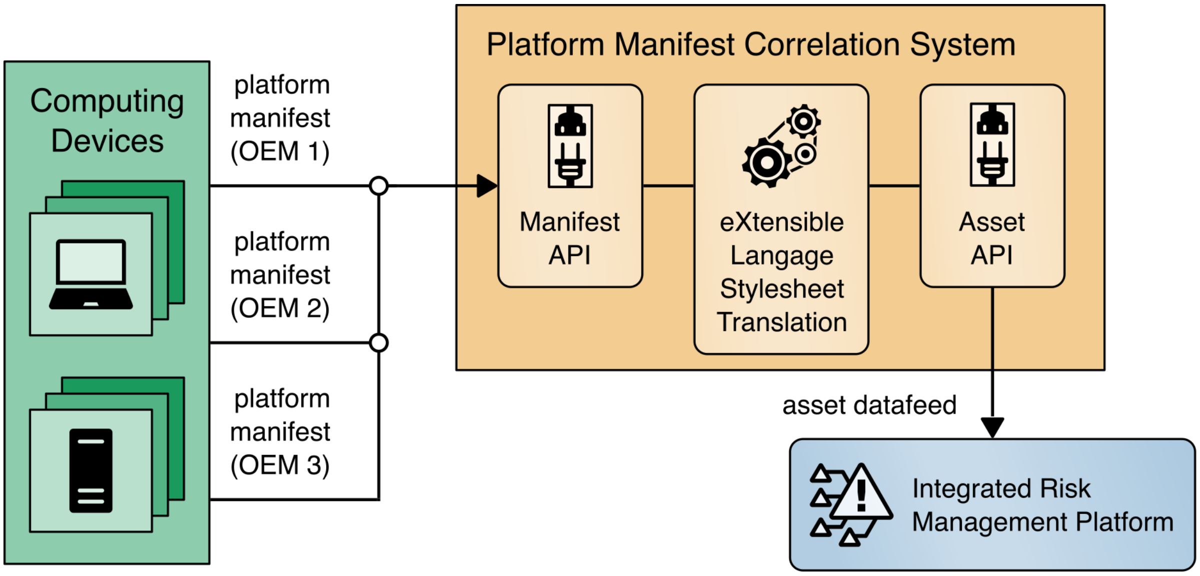
This system assists in providing computing device manifest attributes to the asset management system. The system was built specifically for this demonstration and was built on open-source projects to include the node.js server platform. The requirements of this system were defined as:

1. Provide a web interface for the IT administrator to upload platform manifests.
2. Provide a REST API for scripts to upload platform manifests.
3. Provide a REST API for the asset management system to periodically poll for new computing devices to import in the repository.

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

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

811 **Figure 4-5 Platform Manifest Correlation System**

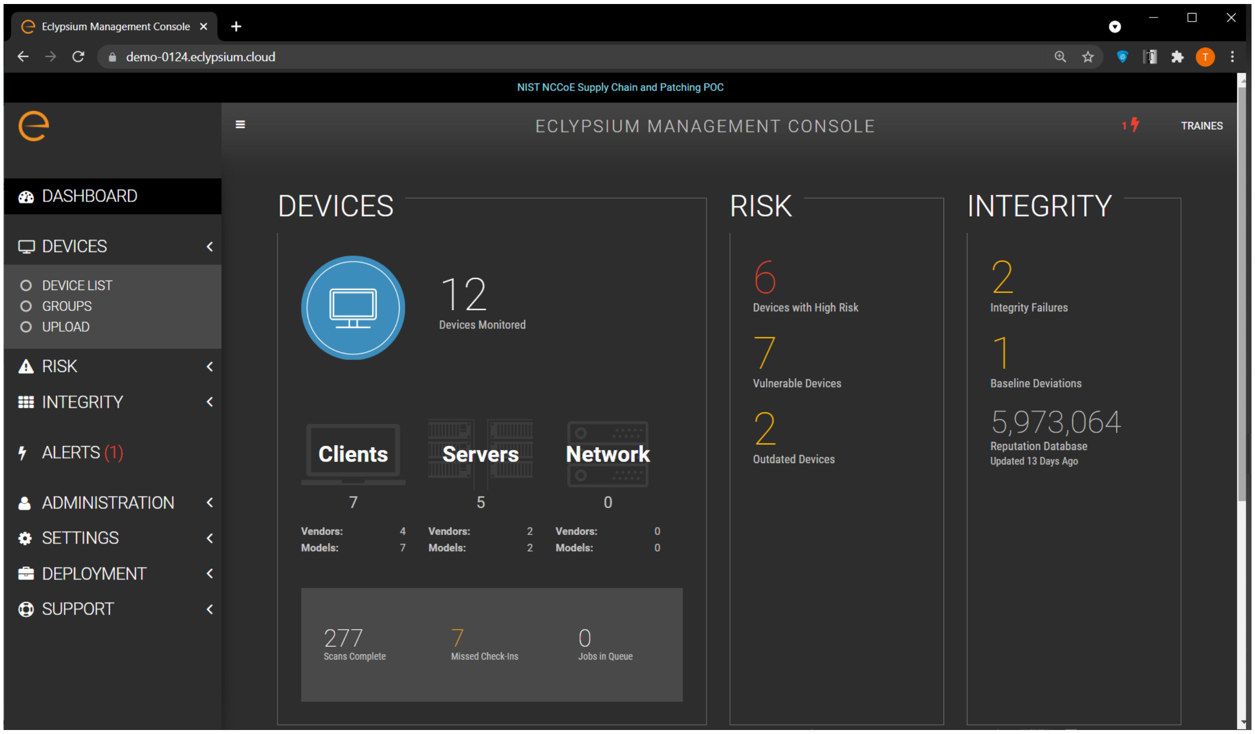


812 ### 4.3.4 Eclypsium Analytic Platform

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).

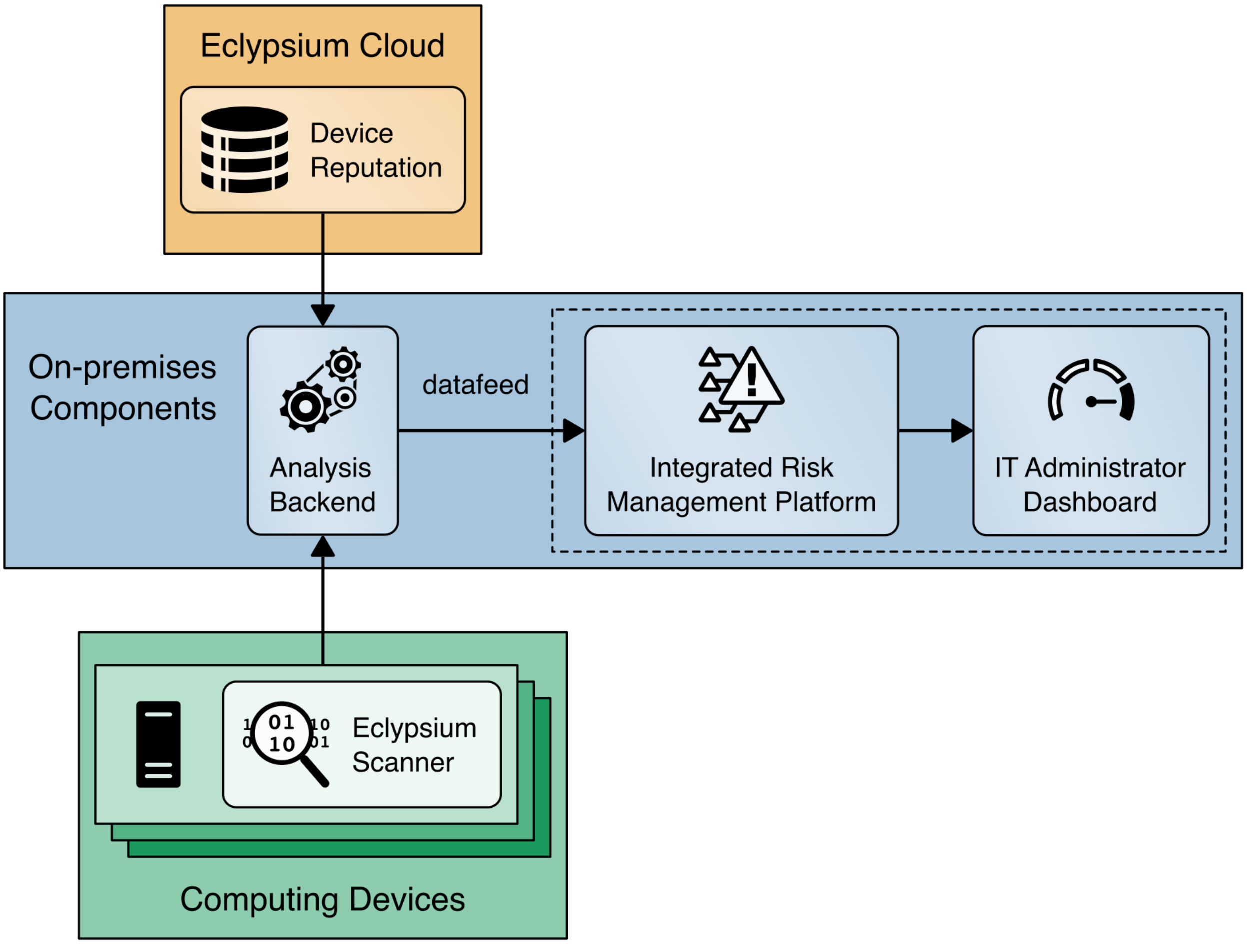
Figure 4-6 Eclypsium Management Console



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

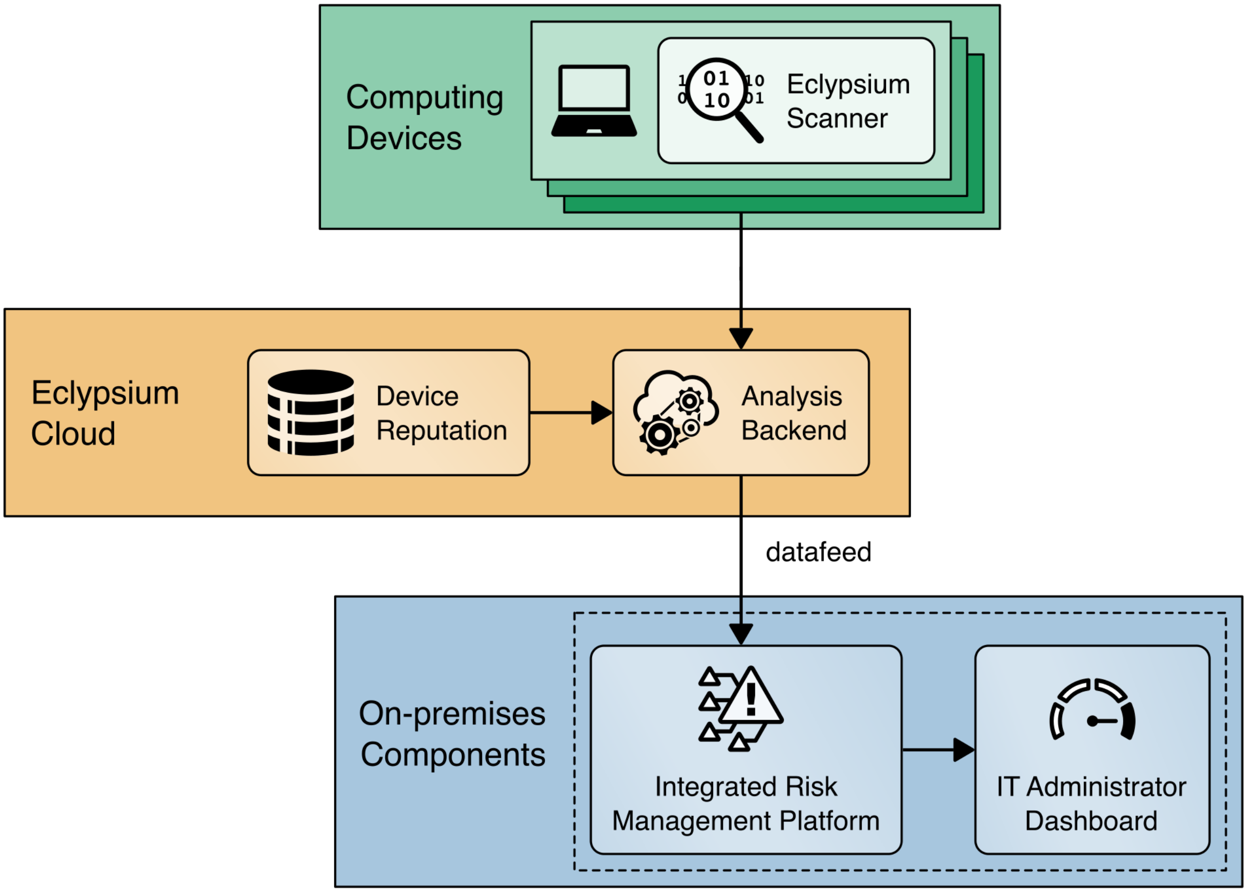
The Eclypsium Analytic Platform also supports a disconnected deployment, where the computing devices that are continuously monitored by the Eclypsium agent communicate directly with an on-premises analytics backend. This type of deployment is useful for environments where a computing device, such as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

Figure 4-8 Eclypsium Analytic Platform Laptop Implementation



4.4 Computing Devices

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

4.4.1 HP Inc.

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

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

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

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

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

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

- **HP Sure Admin** enforces a certificate-based authorization model that enables firmware setting security management by an IT administrator. The model is composed of two keys, an Endorsement Key and a Signing Key (note: the Endorsement Key in this context is not related to the TPM Endorsement Key). The Endorsement Key's primary purpose is to protect against unauthorized changes to the Signing Key. The Signing Key is used by the platform to authorize commands sent to the firmware (BIOS) [14] [15].
- **HP Sure Start** is a built-in hardware security system that protects platform firmware code and data (including HP BIOS, HP Endpoint Security Controller firmware, and Intel Management Engine firmware) from accidental or malicious corruption by (1) detecting corruption and then (2) automatically restoring the firmware to its last installed HP-certified version and the data (settings) to the last authorized state. The capability also stores events related to firmware integrity that can provide visibility into attempted firmware integrity breaches [16].
- **HP Sure Recover** is an OS recovery mechanism that is completely self-contained within the hardware and firmware to allow secure OS recovery from the network or from a local OS recovery copy stored in dedicated flash on the system board. It includes settings that control when, how, and from where BIOS installs the OS recovery image, and which public keys are used by BIOS to validate the integrity of the recovery image. It can also record events due to OS recovery image integrity failures [16].

- **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).

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

4.4.3 Intel

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

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

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

1. The `ScanSystem` function initiates the scanning of the system components and the TPM information. The scanning operation will perform the following operations:
    a. Read the following platform components: BIOS, system, motherboard, chassis, processor, dual in-line memory modules (DIMMs), batteries, Intel Active Management Technology firmware version, power supplies
    b. Read the TPM PCRs, public Endorsement Key, and the Endorsement Key serial number
    c. Read the internal drive information
    d. Read the Windows Management Instrumentation (WMI) Information for internal keyboard, 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:

1. Ensures the validity of the trust chain and signature of the factory installed initial DevID signing key and initial Attestation Key (IAK) created by HPE. The initial DevID is a unique, permanent cryptographically protected identifier for the HPE server. The IDevID certificate is TCG and IEEE 802.1 AR compliant. The IAK is a restricted signing key that is used when performing remote attestation of the HPE server using its TPM.
2. Performs TCG certificate trust chain verification, verifying the chain from the signed certificate to the HPE Root CA certificate. This step verifies the certificate signature against the intermediate certificate that signed the Platform Certificate, system IDevID certificate, and associated system IAK Certificate.
3. Verifies the demonstration server's hardware manifest against the Platform Certificate that HPE issued at its manufacturing facility.

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

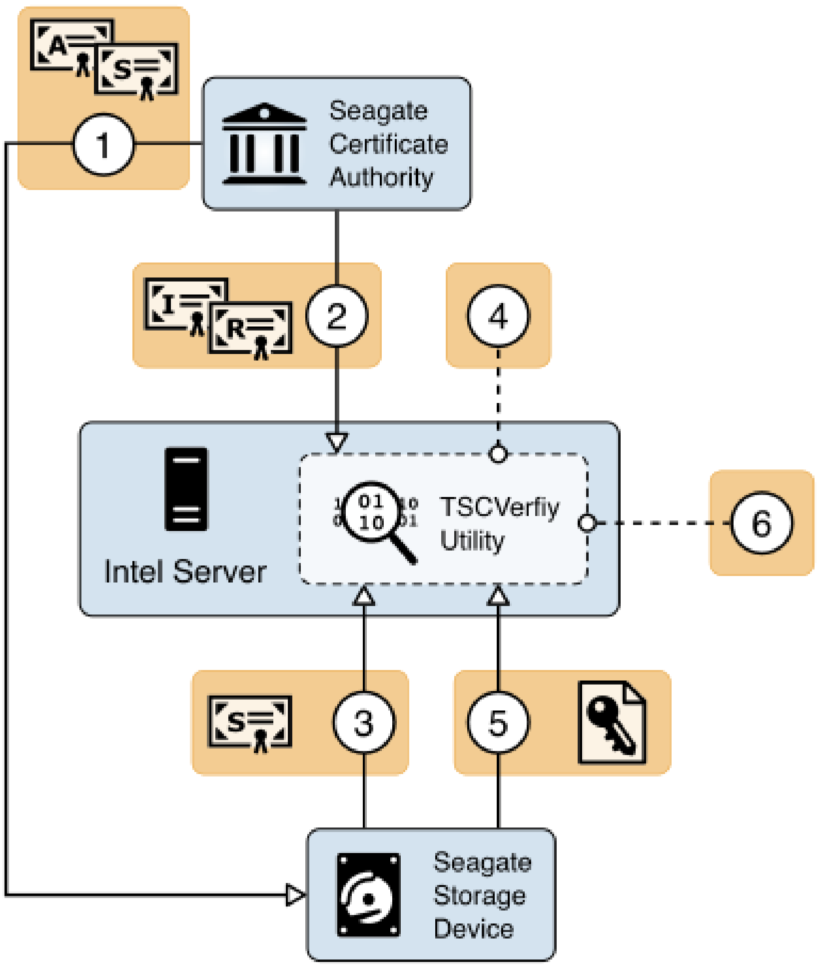
4.4.5 Seagate

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

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

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

Figure 4-9 Seagate Secure Drive Authentication Integration

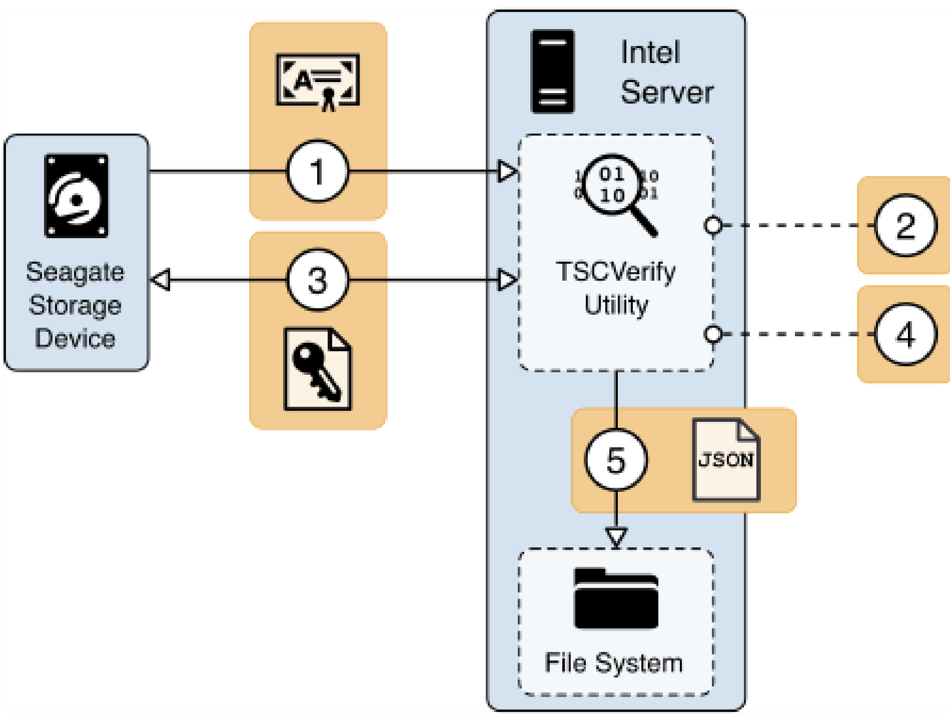


1. During the manufacturing process, Seagate creates a Trusted Peripheral signing certificate (tperSign Certificate) and Attestation Certificate (tperAttestation Certificate) that are signed by the Seagate Intermediate CA. The tperSign Certificate and tperAttestation Certificate are stored in the drive’s firmware. The drive is now capable of responding to challenges from host computing devices.

2. The host, in this case the Intel server, stores the Seagate Root and Intermediate CA certificates in the TSCVerifyUtil application binary. They are used later in the validation process.
3. The Security Operator executes the TSCVerifyUtil application and directs it to initiate the SDA verification. The drive's certificate is returned in the initial invocation of SDA.
4. The drive's signing certificate is returned to TSCVerify where it is validated against the Seagate Root and Intermediate CA certificates. If validation succeeds, the process continues.
5. TSCVerifyUtil generates a challenge (timestamp) that is transmitted to the drive. The drive returns a cryptographically signed response based on the challenge.
6. TSCVerifyUtil verifies the digital signature on the response with the drive's public key retrieved in Step 3.

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

Figure 4-10 Seagate Firmware Attestation Integration



1. TSCVerifyUtil requests the tperAttestation Certificate from the drive. The certificate path is validated against the Seagate Intermediate and Root CAs.

2. TSCVerifyUtil generates an Assessor Identifier and a nonce. The Assessor Identifier is a static host server identifier (such as the hostname) and the nonce is a randomly generated set of 16 bytes for each invocation of the firmware attestation method. These values, in addition to the common name of the tperAttestation Certificate, are stored for the next step.
3. The values from Step 2 are transmitted to the drive via the Get Signed Firmware Message command and the response is returned.
4. The digital signature on the response is verified using the drive's public key from the tperAttestation Certificate retrieved in step 1.
5. If Step 4 succeeds, the associated firmware hashes are exported from TSCVerifyUtil as a JSON-formatted file.

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

5 Security Characteristic Analysis

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

5.1 Assumptions and Limitations

The security characteristic analysis has the following limitations:

- It is neither a comprehensive test of all security components nor a red-team exercise.
- It cannot identify all weaknesses.
- It does not include the lab infrastructure. It is assumed that devices are hardened. Testing these devices would reveal only weaknesses in implementation that would not be relevant to those adopting this reference architecture.
- It will evolve and expand as the project as collaborators are integrated into the final architecture in the next publication of this document.

5.2 Build Testing

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

5.2.1 Scenario 1

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

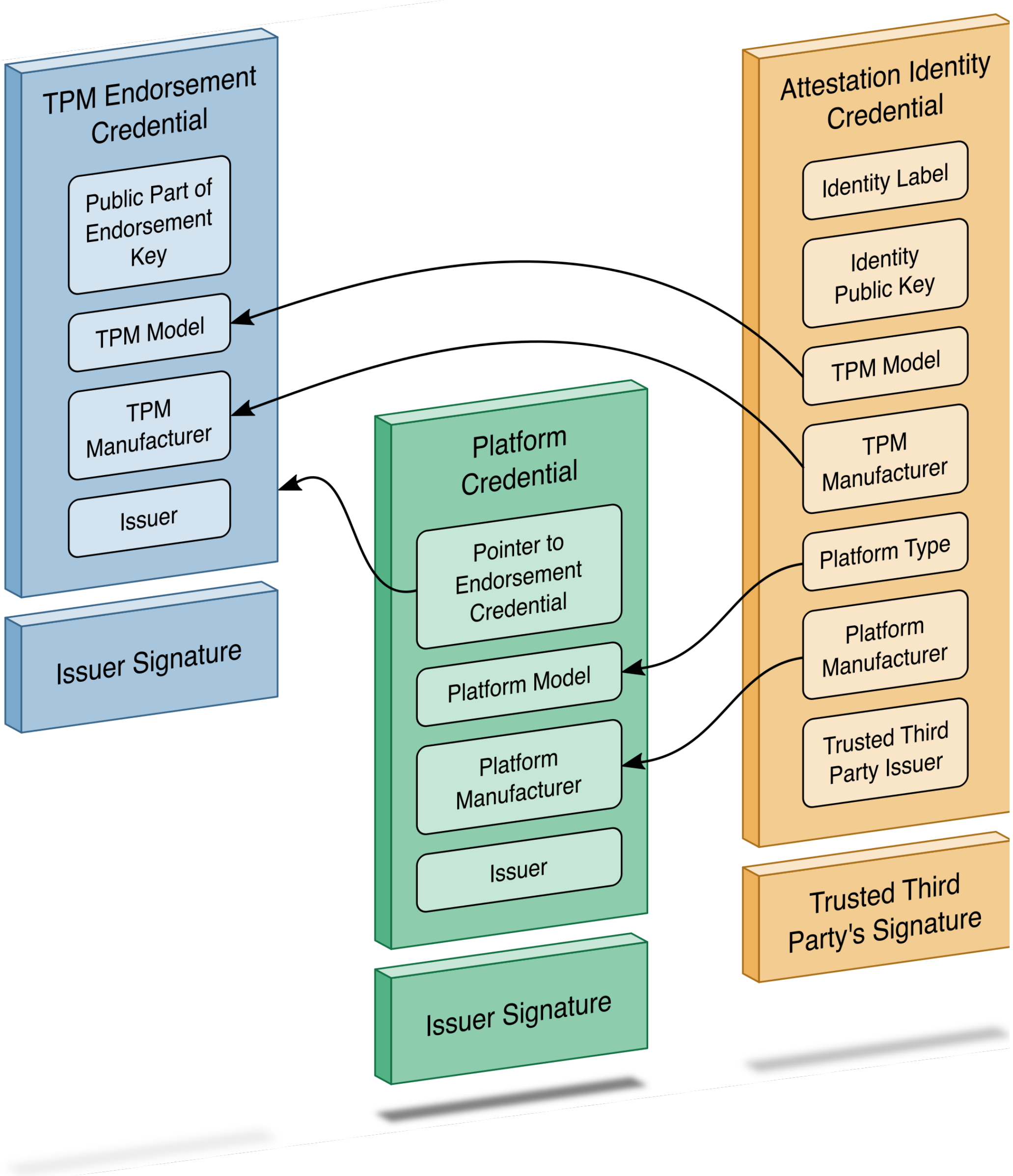
In the preliminary draft version of this guide, we emulated a customer-created platform artifact using the HIRS ACA project's PACCOR software for Dell and HP Inc. laptops. In this revision, Dell and HP Inc. contributed laptops with pre-installed verifiable artifacts created at the factory, where they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

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

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

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

Figure 5-1 Platform Certificate Binding to Endorsement Credential



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

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	

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

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

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

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

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

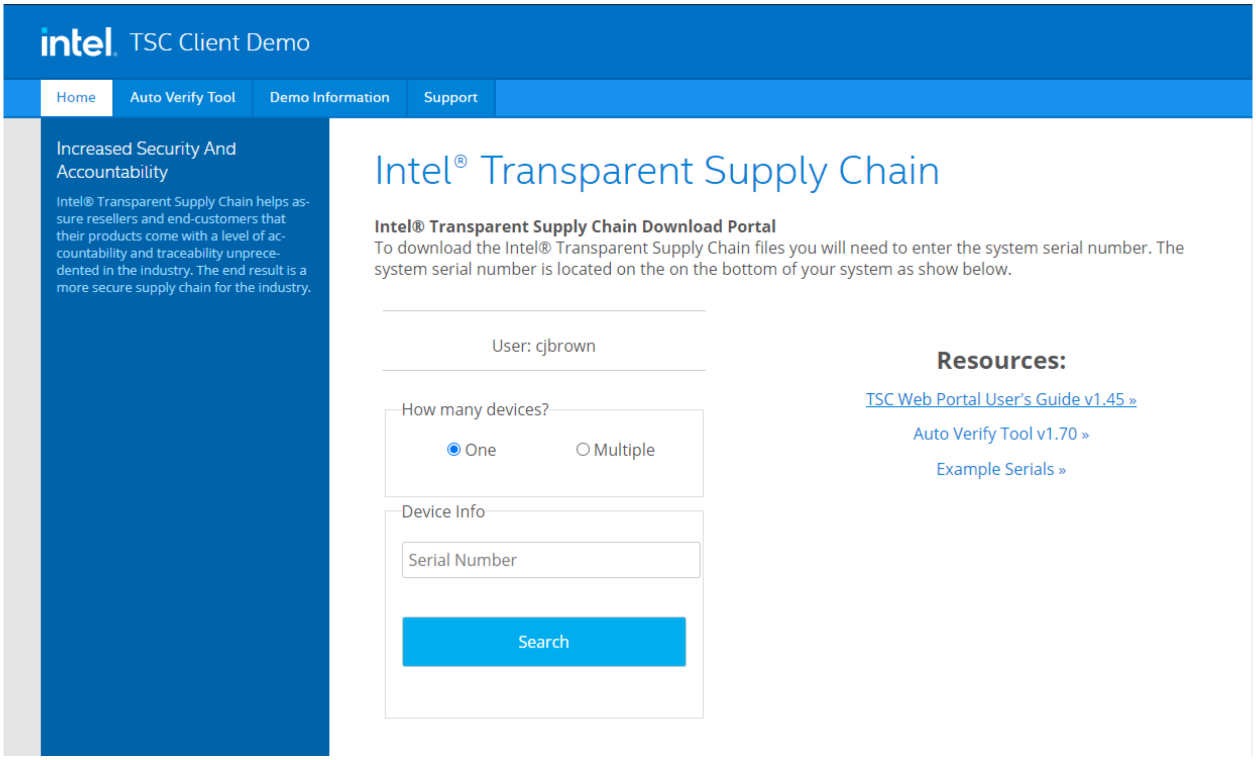
5.2.2 Scenario 2

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

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

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

Figure 5-2 Intel Transparent Supply Chain Download Portal



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

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

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

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

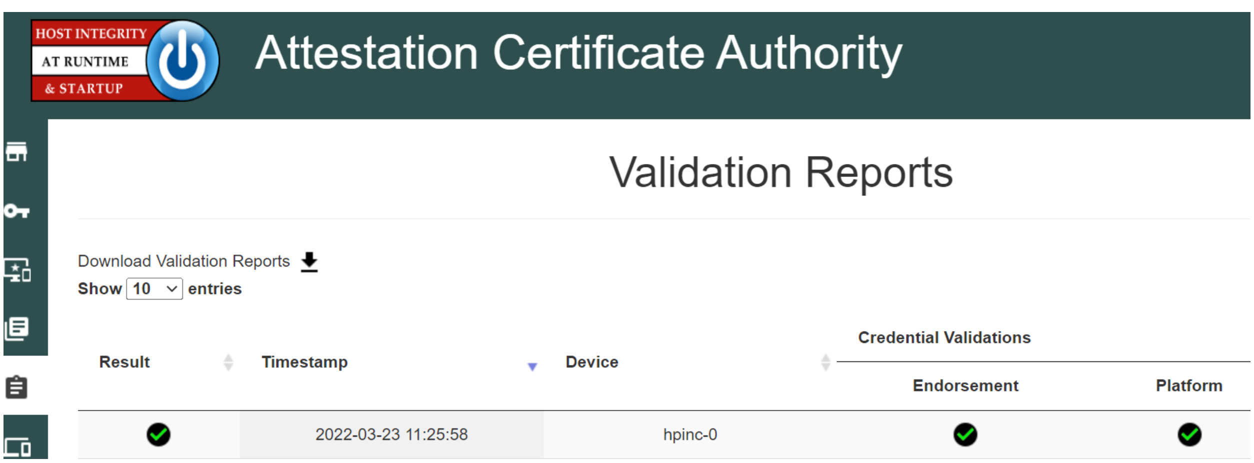
For simplicity, we have combined discussion of steps 2 and 3 because they are performed in tandem using platform validation tools.

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

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

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

Figure 5-3 HIRS ACA Validation Dashboard



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

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

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

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

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

For Dell and HP Inc. laptops which use the HIRS ACA platform, we opted to use a script-based approach to automatically upload the platform manifest to the Platform Manifest Correlation System's REST API. Similarly, for HPE and Dell server platforms, the manifests produced by each manufacturer's validation tool is uploaded via the REST API. The use of a web interface or REST API demonstrates flexibility in the architecture that can assist organizations with a heterogeneous manufacturer environment or use cases where automation is not feasible.

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

5.2.2.1 Provisioning Example

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

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

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

Figure 5-5 Asset Inventory and Discovery Example 2

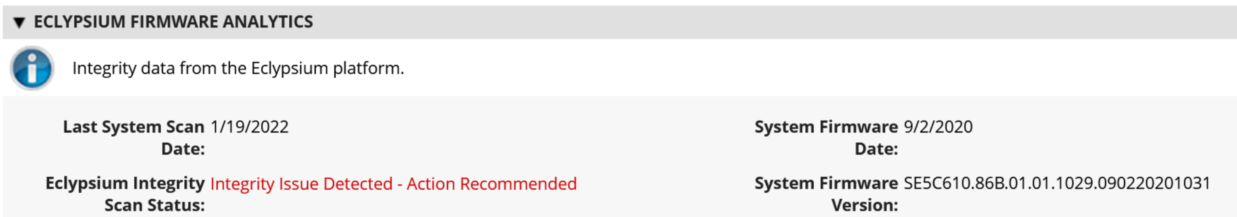
▼ **ASSOCIATED COMPONENTS** View Less

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

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

Figure 5-6 Asset Inventory and Discovery Example 3



Step 4b: *If the administrator updates the configuration of the platform (e.g., adding hardware components, updating firmware), then the administrator might create new platform artifacts to establish a new baseline.*

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

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

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

Finally, Dell server manifests are updated in the field by manufacturer technicians using specialized tools. The tooling generates a new manifest for the server, which is delivered to Dell's environment and re-signed by Dell's high-assurance certificate issuing authority that previously signed the original verifiable artifact embedded from the factory.

5.2.3 Scenario 3

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

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

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

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

The final use case we examined across all manufacturers is when system firmware is updated on the fielded laptop. This may be initiated by the end user who is guided by a helpdesk or by the IT 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.

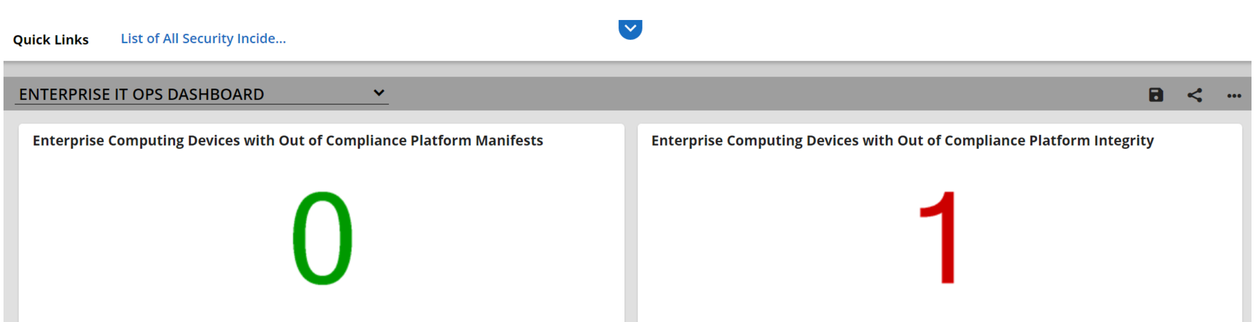
Similarly, firmware measurements produced by the Seagate Firmware Attestation capability are tracked for changes, and those changes are associated with the Intel server that the drives are connected to in this demonstration. A firmware measurement change in this case could be indicative of a non-malicious 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



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

Figure 5-8 Scenario 3 Security Event

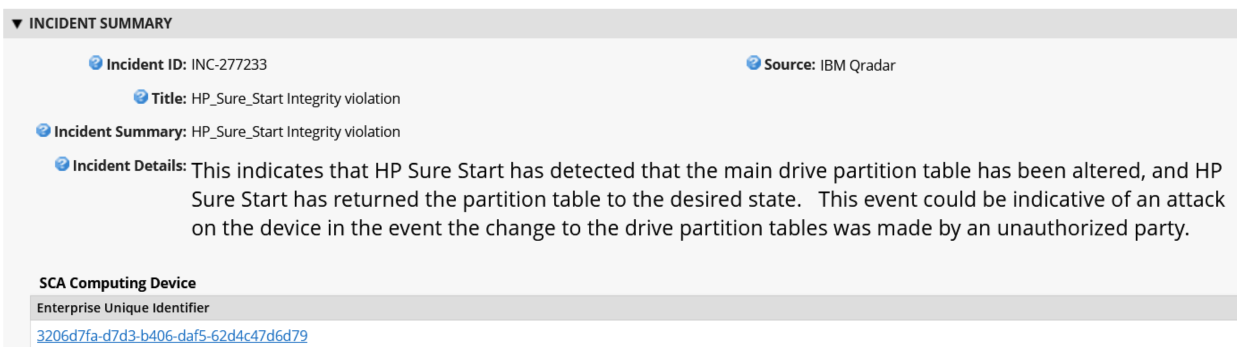
Drag a column name here to group the items by the values within that column.

	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)

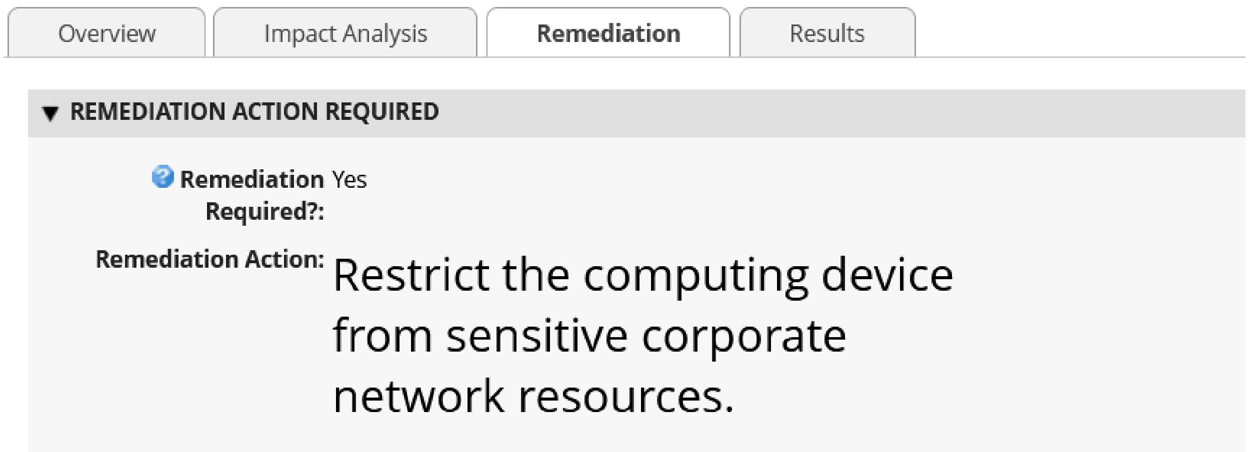
Clicking on the Incident ID reveals more details about the incident for the personnel assigned to investigate the incident for additional context. This is pictured in Figure 5-9.

Figure 5-9 Scenario 3 Security Event Summary



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

Figure 5-10 Scenario 3 Security Event Remediation



5.3 Scenarios and Findings

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

5.3.1 Supply Chain Risk Management (ID.SC)

5.3.1.1 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.

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

5.3.2 Asset Management (ID.AM)

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

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

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

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

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

5.3.4 Data Security (PR.DS)

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

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

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

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

5.3.5 Security Continuous Monitoring (DE.CM)

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

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

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

6 Future Build Considerations

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

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

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

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

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

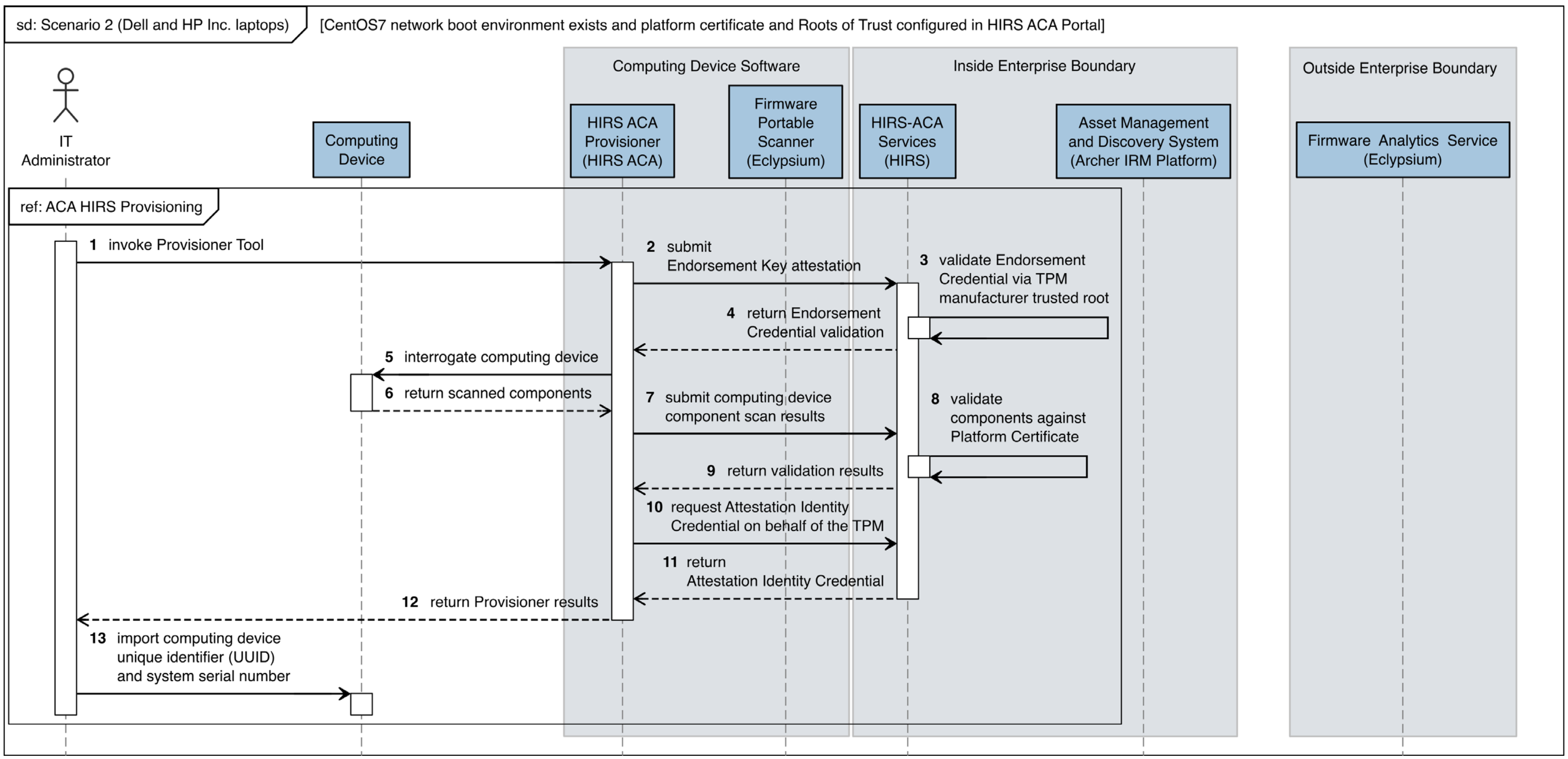


Figure C-2 Dell and HP Inc. Laptop Scenario 2 Part 2

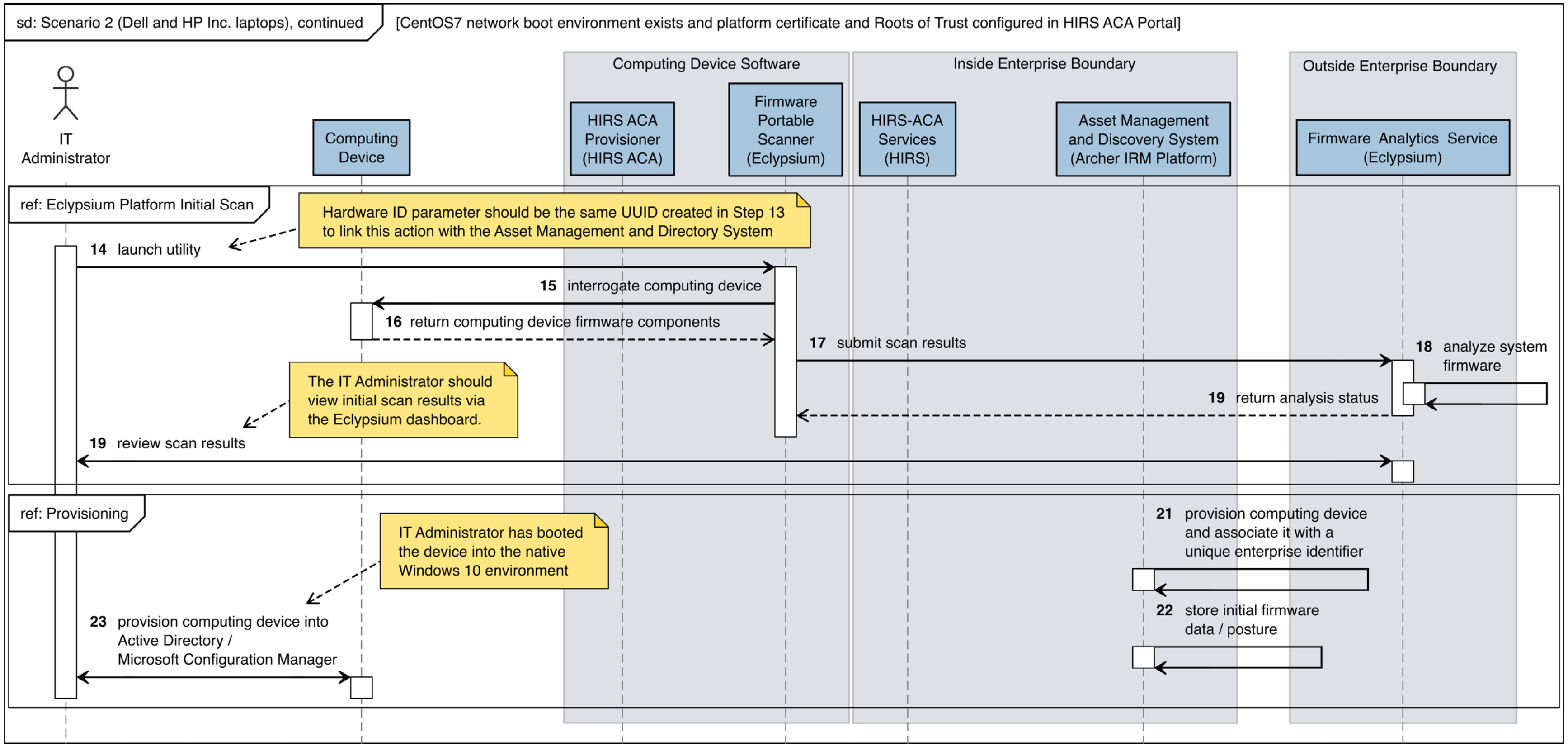


Figure C-3 Intel Laptop Scenario 2 Part 1

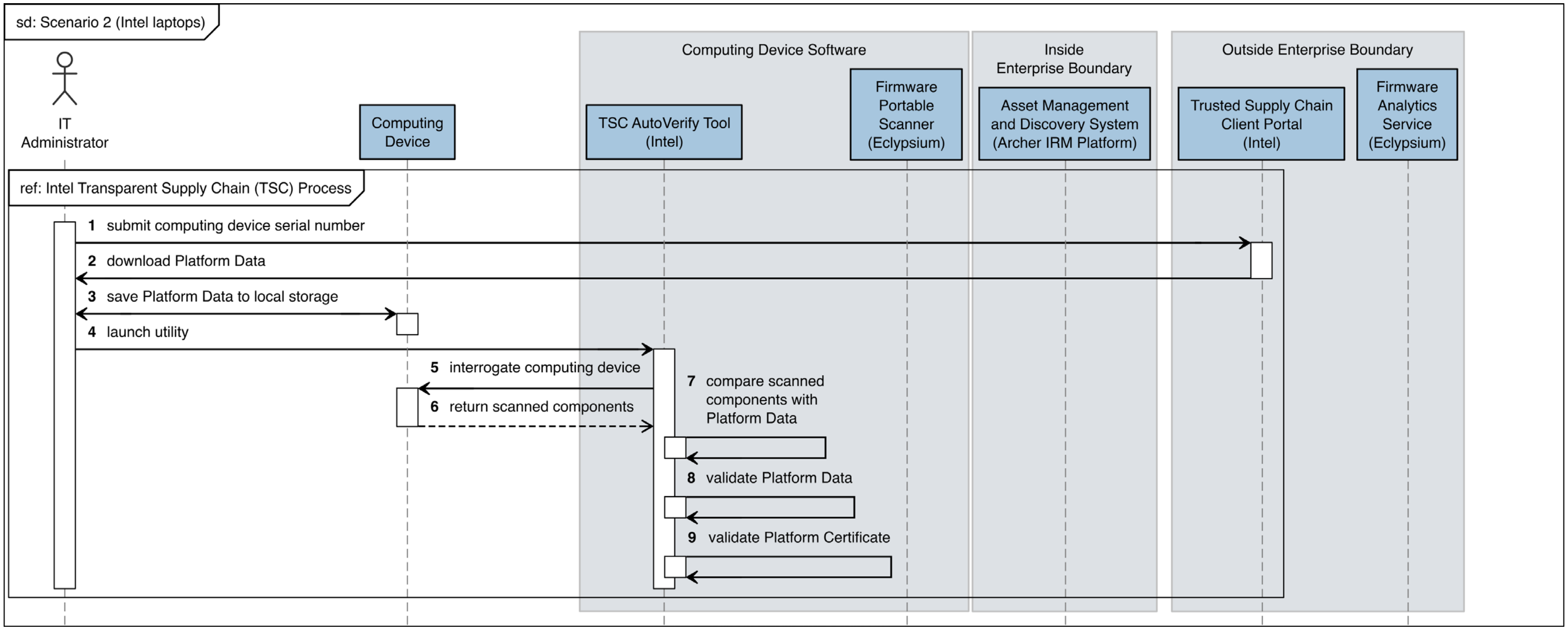


Figure C-4 Intel Laptop Scenario 2 Part 2

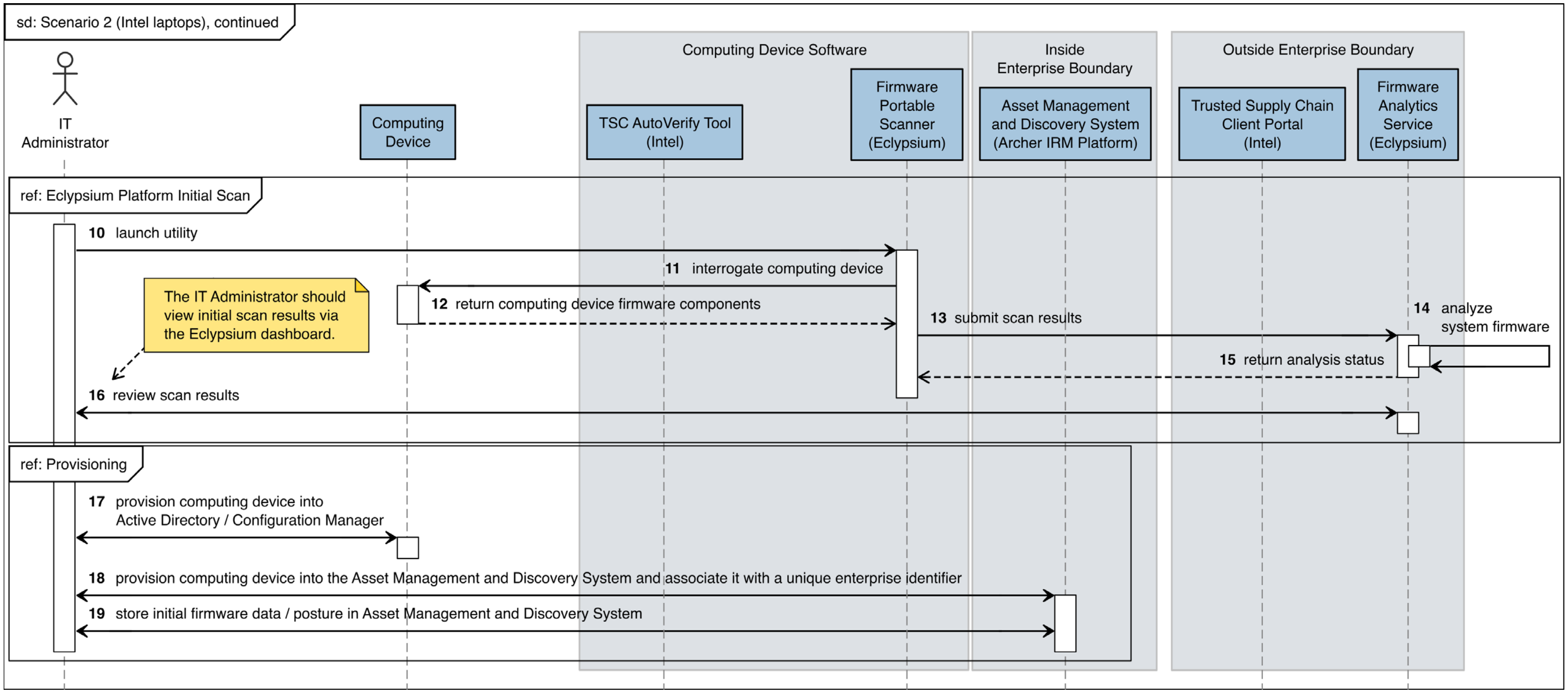


Figure C-5 Intel Server Scenario 2 Part 1

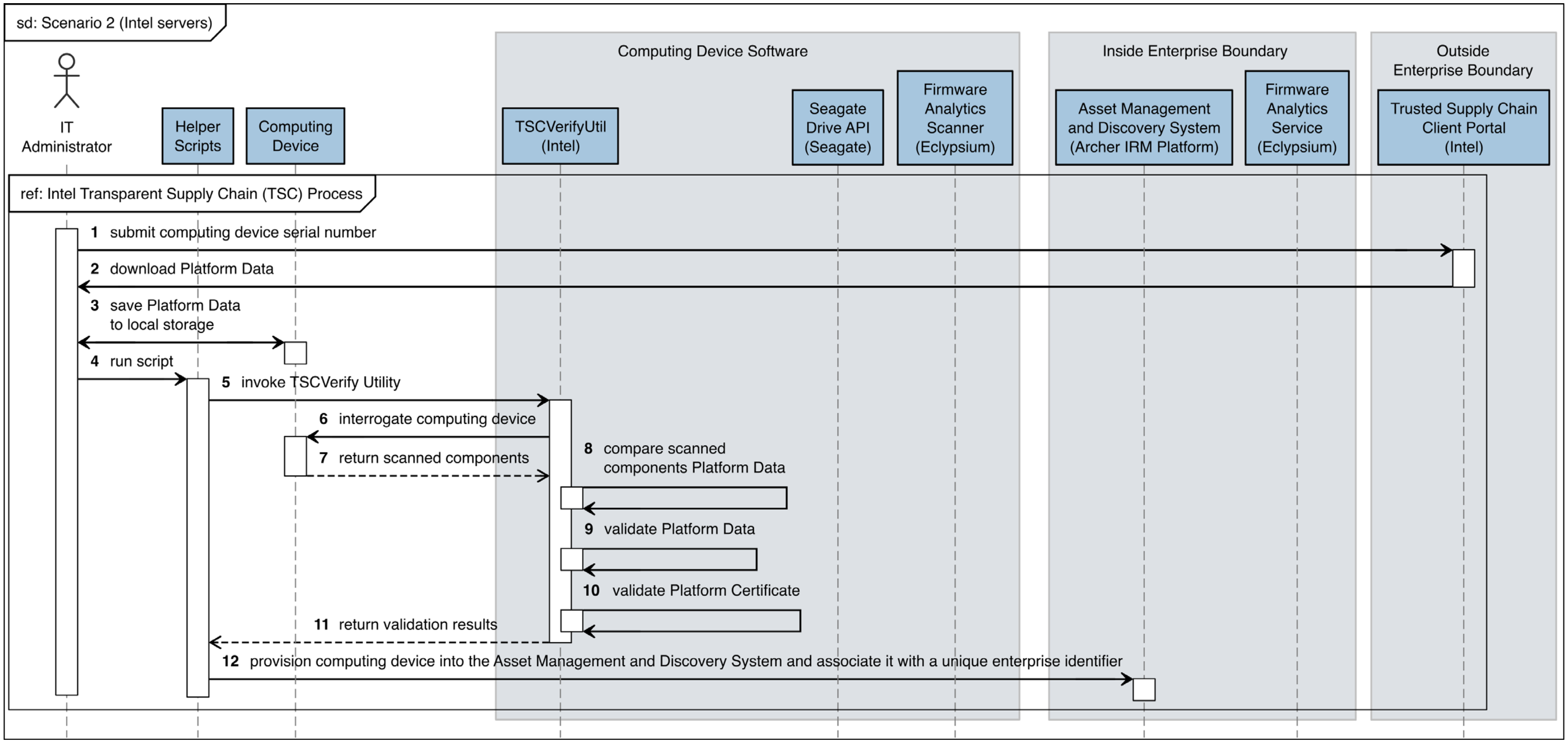


Figure C-6 Intel Server Scenario 2 Part 2

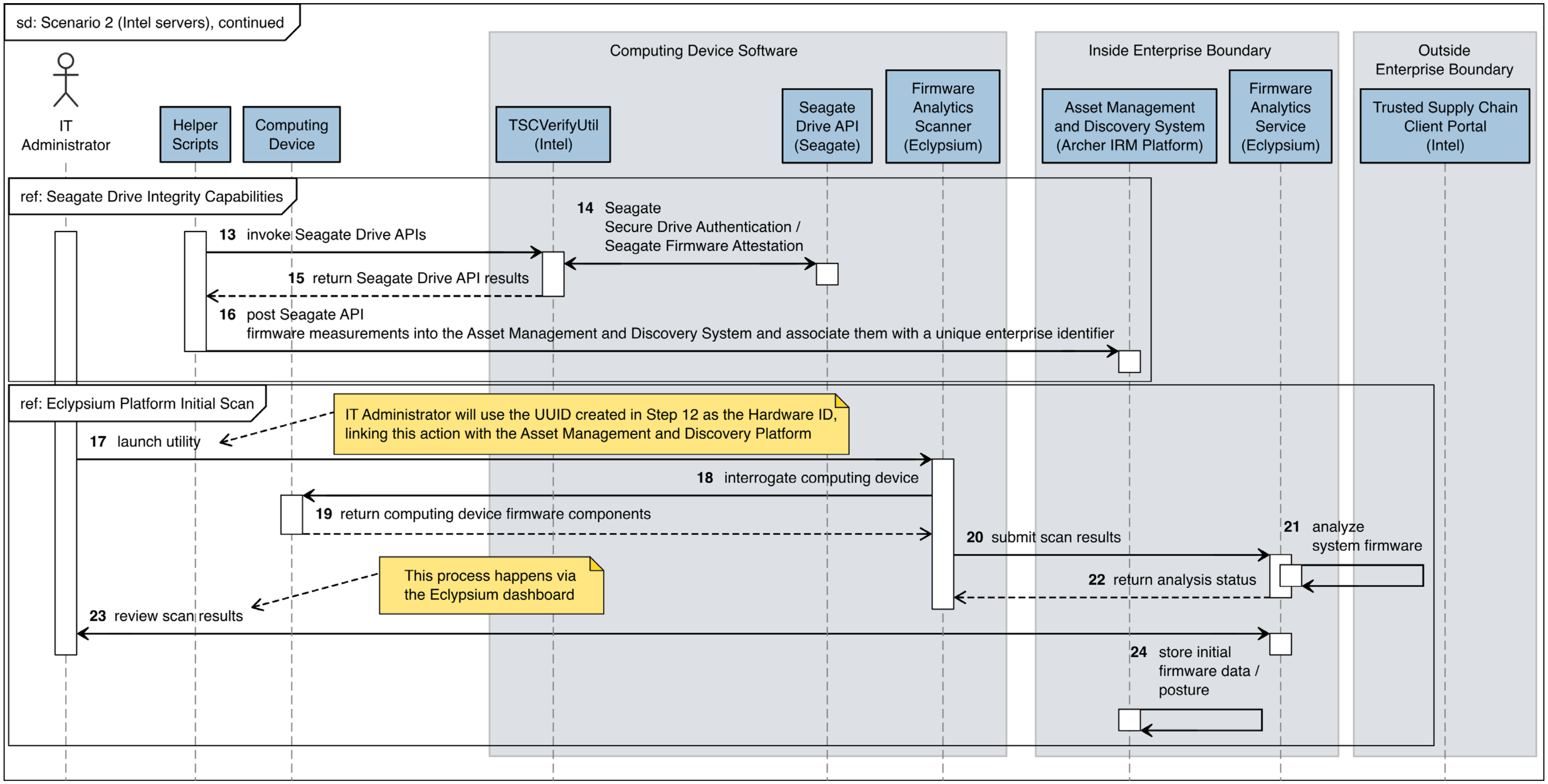


Figure C-7 Dell Server Scenario 2

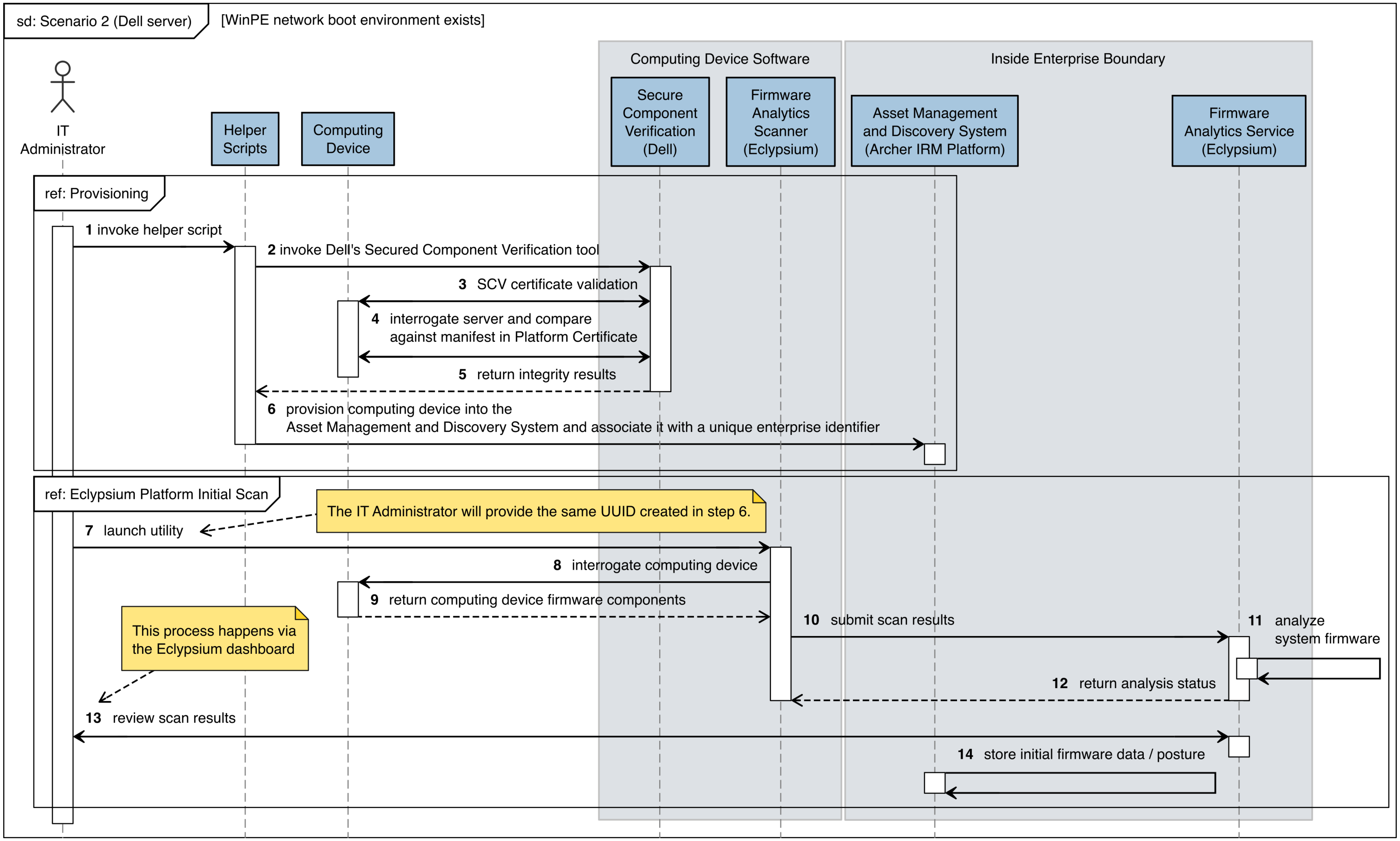


Figure C-8 HPE Server Scenario 2

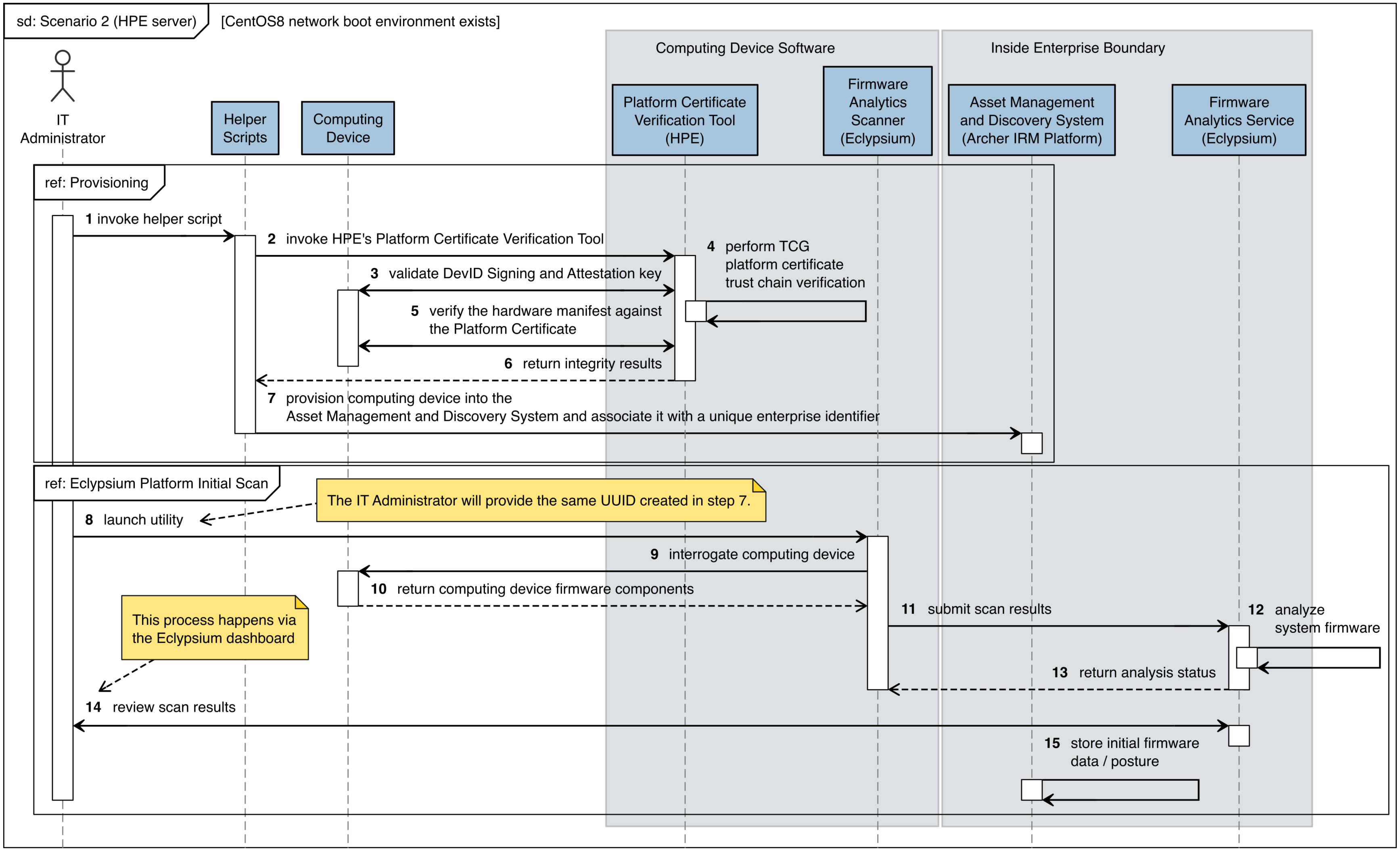


Figure C-9 Intel Laptop Scenario 3

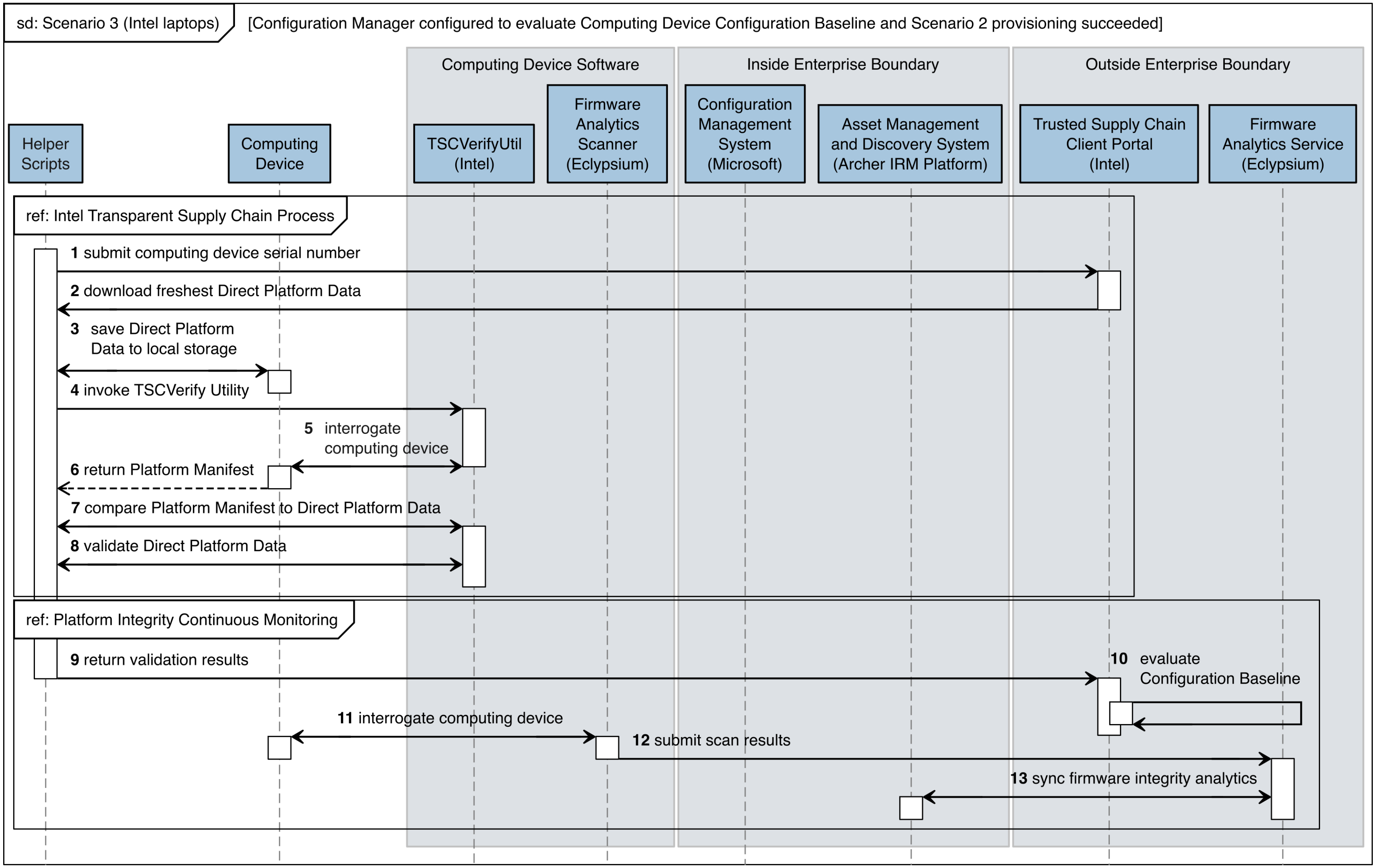


Figure C-10 Dell Laptops Scenario 3

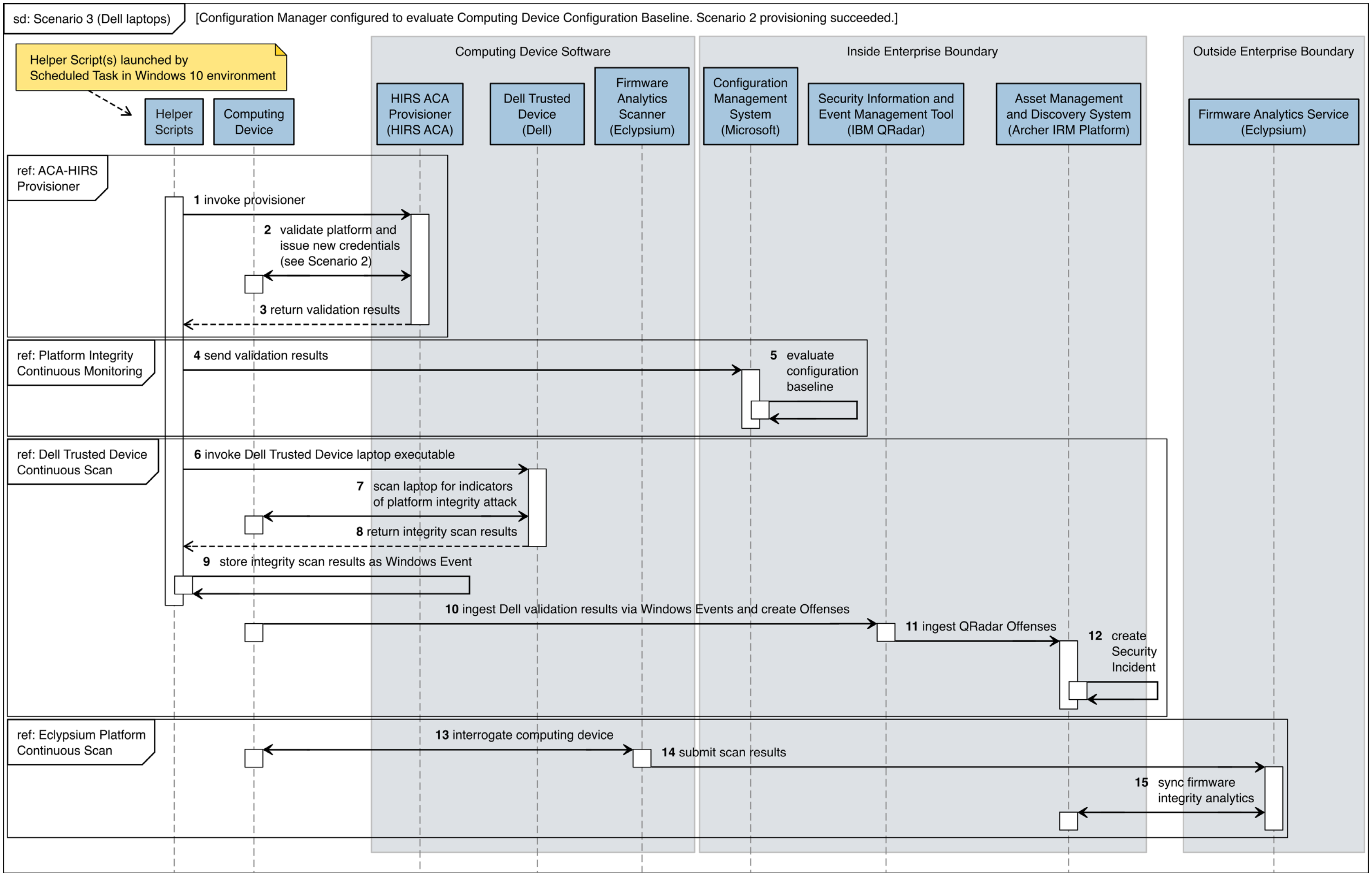
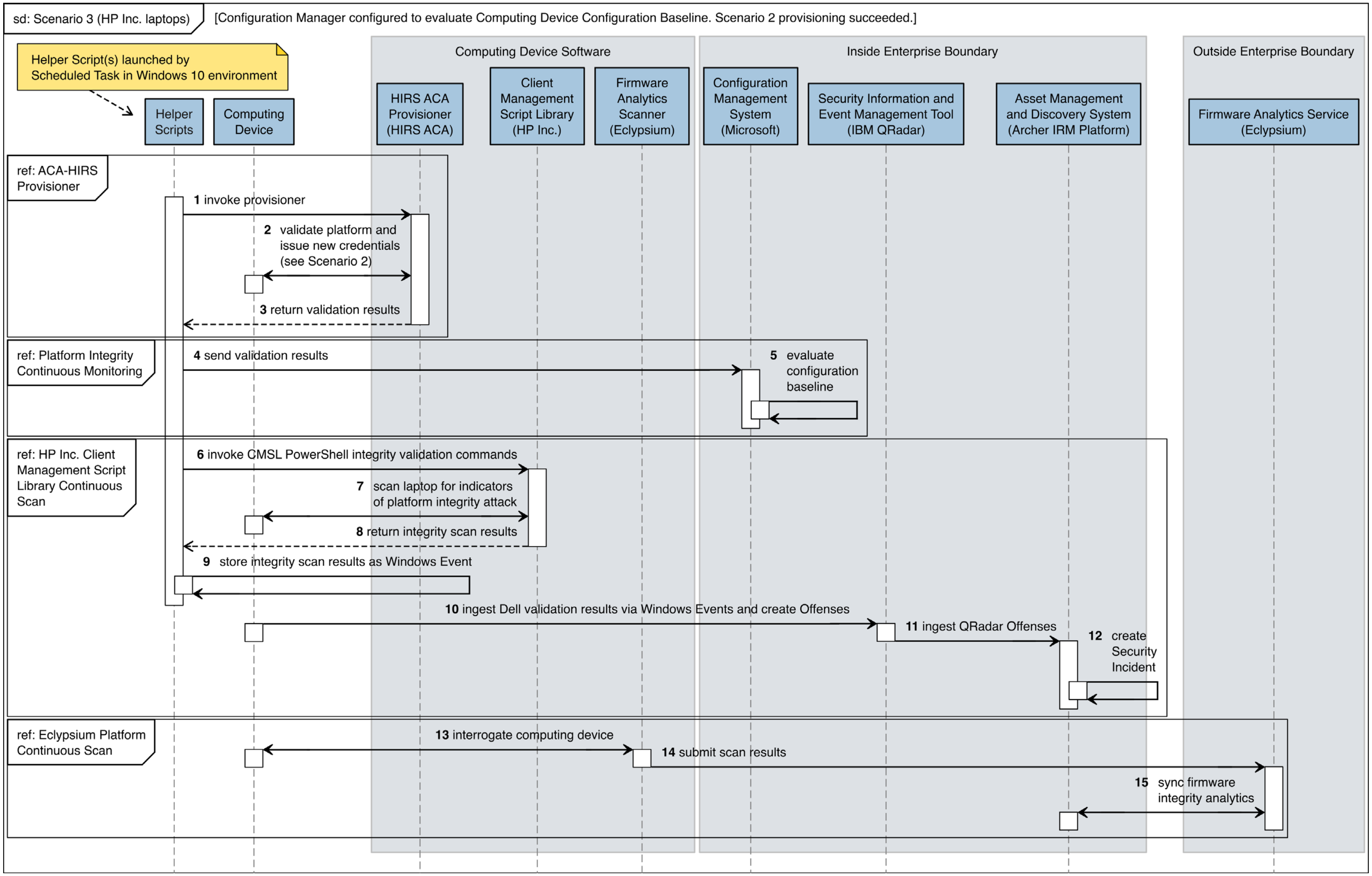


Figure C-11 HP Inc. Laptops Scenario 3



NIST SPECIAL PUBLICATION 1800-34C

Validating the Integrity of Computing Devices

Volume C:
How-To Guides

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

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/building-blocks/supply-chain-assurance

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.

National Institute of Standards and Technology Special Publication 1800-34C, Natl. Inst. Stand. Technol. Spec. Publ. 1800-34C, 141 pages, (June 2022), CODEN: NSPUE2

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.

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.

KEYWORDS

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

ACKNOWLEDGMENTS

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

Name	Organization
Charles Robison	Dell Technologies
Mukund Khatri	Dell Technologies
Rick Martinez	Dell Technologies
Daniel Carroll	Dell Technologies
Jason Young	Dell Technologies
Travis Raines	Eclypsium
John Loucaides	Eclypsium
Jason Cohen	Hewlett Packard Enterprise
CJ Coppersmith	Hewlett Packard Enterprise
Ludovic Jacquin	Hewlett Packard Enterprise
Boris Balacheff	HP Inc.
Jeff Jeansonne	HP Inc.
Joshua Schiffman	HP Inc.
Harmeet Singh	IBM
Tom Dodson	Intel
Jason Ajmo	The MITRE Corporation
Chelsea Deane	The MITRE Corporation

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

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

Technology Partner/Collaborator	Build Involvement
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

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

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

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

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

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

Contents

List of Figures

List of Tables

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.

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

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

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

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

1.1.1 Supplemental Material

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

1.2 Build Overview

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

1.3 Typographic Conventions

The following table presents typographic conventions used in this volume.

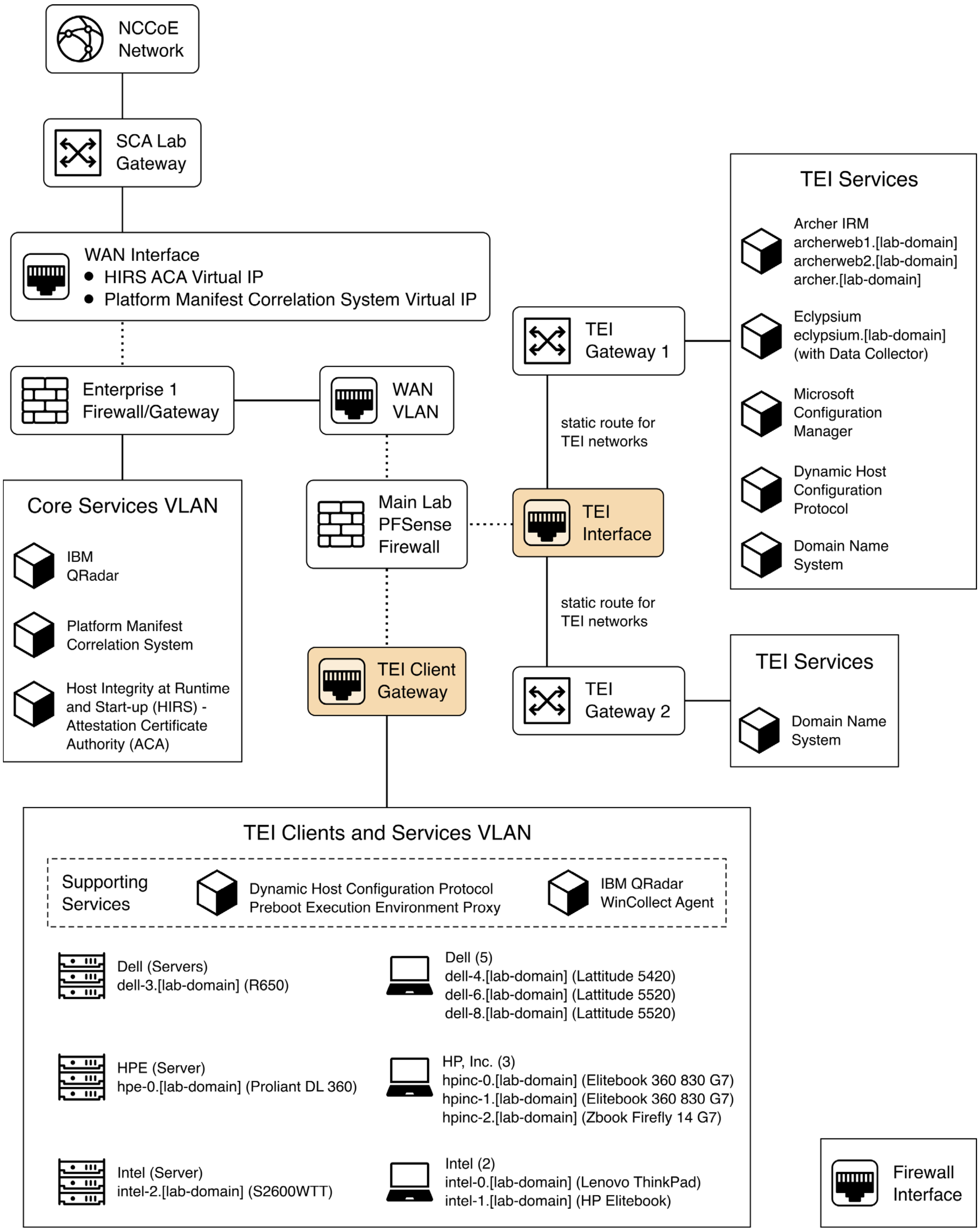
Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For language use and style guidance, see the *NCCoE Style Guide*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit.**
`Monospace`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

1.4 Logical Architecture Summary

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

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

271 **Figure 1-1 Demonstration Network Architecture**



Services specific to the capabilities of this prototype demonstration are instantiated on the Core Services virtual network. This virtual network represents the integration of supply chain risk management (SCRM) requirements into an enterprise architecture to support the SCRM controls, as described in the Risk Assessment section of Volume B.

2 Product Installation Guides

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

2.1 Supporting Systems and Infrastructure

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

2.1.1 Network Boot Services

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

2.1.1.1 Linux-Based Acceptance Testing Image Creation

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

2.1.1.1.1 HIRS TPM Provisioner Configuration

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

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

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

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

```
#*******************************************
#* HIRS site configuration properties file
#*******************************************
```

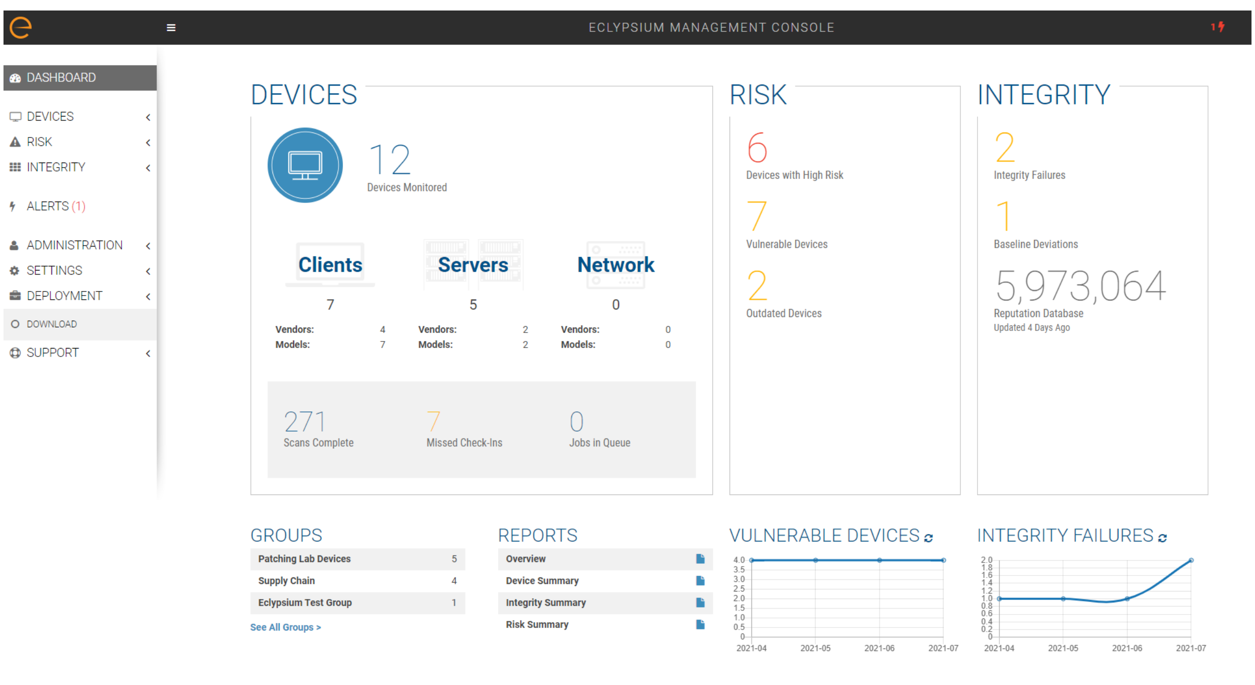
```
# Client configuration
CLIENT_HOSTNAME=localhost
TPM_ENABLED=true
IMA_ENABLED=false

# Site-specific configuration
ATTESTATION_CA_FQDN=hirs-server.yourdomain.test
ATTESTATION_CA_PORT=8443
BROKER_FQDN=hirs-server.yourdomain.test
# Change this port number to your local configuration
BROKER_PORT=61616
PORTAL_FQDN=hirs-server.yourdomain.test
# Change this port number to your local configuration
PORTAL_PORT=8443
```

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

2.1.1.1.2 Eclypsium Agent Configuration

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

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



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
```

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

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

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

2.1.1.1.4 CentOS 8 Image Creation

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

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

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

15. Complete the remaining steps as documented.

2.1.1.2 Windows-Based Acceptance Testing Image Creation

The following procedures will produce a Windows Preinstallation Environment (WinPE) bootable image that can be used in computing device acceptance testing. You will need to have a Windows Server (2016 or above) environment available to complete the following steps.

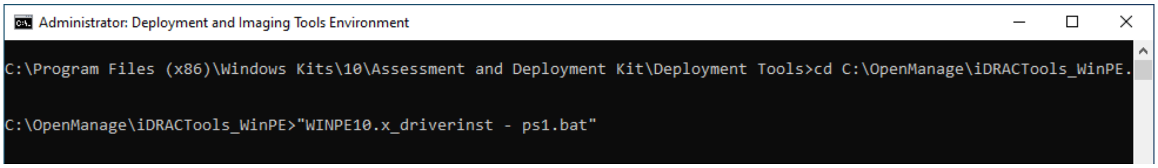
2.1.1.2.1 Build WinPE

1. Download and install the Windows Assessment and Deployment Kit (ADK) and WinPE add-on.

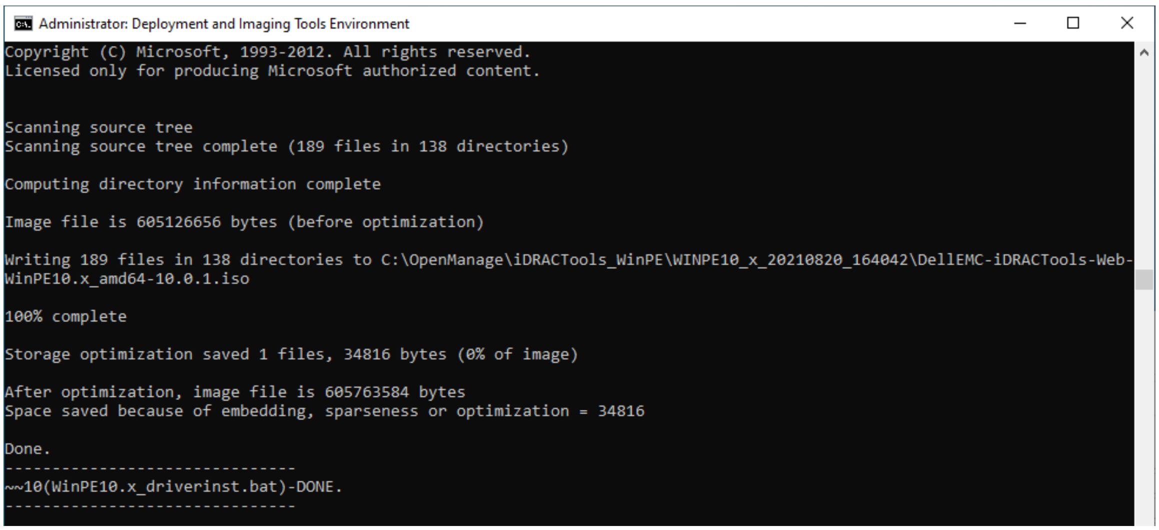
2. Download the Dell EMC iDRAC Tools for Microsoft WinPE (R), v10.1.0.0 software package.

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

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



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



2.1.1.3 Preboot Execution Environment (PXE)

2.1.1.3.1 Dynamic Host Configuration Protocol (DHCP) Proxy

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

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

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

Table 2-1 DHCP Proxy System Information

Operating System	Version	Platform
Ubuntu Server	Release 20.04	Virtual Machine

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

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

7. Create a custom iPXE bootloader that directs iPXE to boot from a fixed URL.
    a. Create a file named *embed.ipxe* with the following contents.

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

    b. [Download](#) and extract the iPXE source files. Install all software dependencies noted on the download page.
    c. Change directory to *ipxe/src* and run the following command.

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

8. Copy the newly built iPXE efi boot file to */var/lib/tftpboot*.
9. Edit the DNSMasq configuration file to suit your environment.
    a. `$ [your favorite editor] /etc/dnsmasq.conf`
    b. Ensure the following configuration variables are set in the configuration file:

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

10. Restart DNSMasq.

```
$ systemctl restart dnsmasq
```

11. Copy the WinPE, CentOS 7, and CentOS 8 images to the HTTP server.
    a. In the root of your HTTP server, create two directories to store the images.

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

   b. Copy the */media* directory created in Section 2.1.1.2.1 to *images/winpe*.
   c. Copy *initrd.img* and *vmlinuz* created in Section 2.1.1.1.2 to *images/centos7*.
   d. Copy *initrd.img* and *vmlinuz* created in Section 2.1.1.1.4 to *images/centos8*.
   e. Download the latest wimboot binary from the iPXE repository and store it in the *images* directory.

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

2.1.2 Platform Manifest Correlation System (PMCS)

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

13. The system is based on Node.js, an open-source JavaScript runtime built on Chrome's V8 JavaScript engine designed to build scalable network applications. Download and install Node.js on a system best suited for your environment. This demonstration uses an Ubuntu 20.04.2 LTS virtual machine.
14. Install the node package manager (npm).
15. Install git on the platform chosen in Step 1. Git provides source code management capabilities used in later steps.
16. Install Process Manager 2 (PM2). This package will manage the Node.js processes that run the PMCS codebase.

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

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

```
$ cd platform-manifest-collation-system
$ pm2 start index.js
```

The PMCS should now be running as a background process. Consider using a startup script to keep your 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.

23. Create a network bootable CentOS 7 image. This step is detailed in Section 2.1.1.

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

24. Power-on the laptop and boot to the BIOS setup by pressing the Function 2 (F2) key.
25. Clear the TPM to remove Windows ownership of the device. Navigate to *Security > TPM 2.0 Security > Clear* in the main menu. Click the *Clear* radio box and select **Yes** in the dialog box.
26. Turn off *Secure Boot*. Navigate *Secure Boot > Secure Boot Enable* in the main menu. Click the *Clear* radio box and select **Yes** in the dialog box.
27. Reboot the laptop by clicking **Apply** and **Yes** in the dialog box followed by **Exit.**

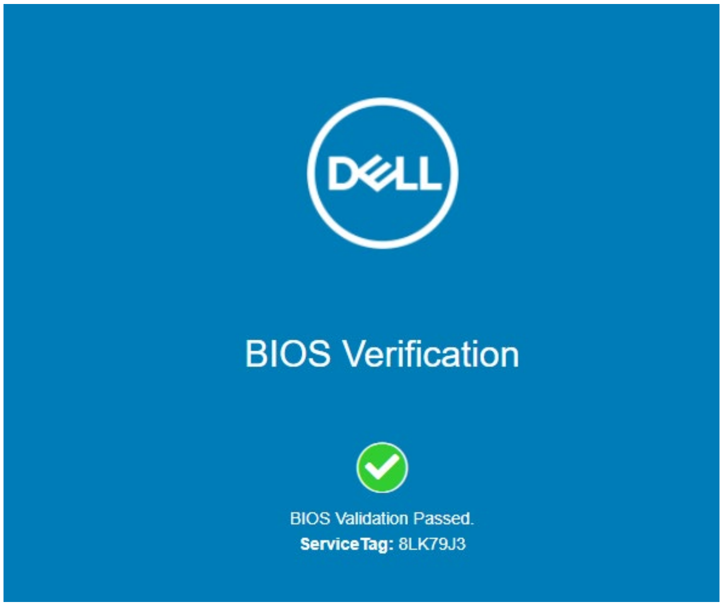
2.2.1.2 Install the Dell Trusted Device Agent

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

28. Download the latest version of the Dell Trusted Agent from the Dell website.
29. Open a command prompt as an Administrator. Install the agent with the following command:

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

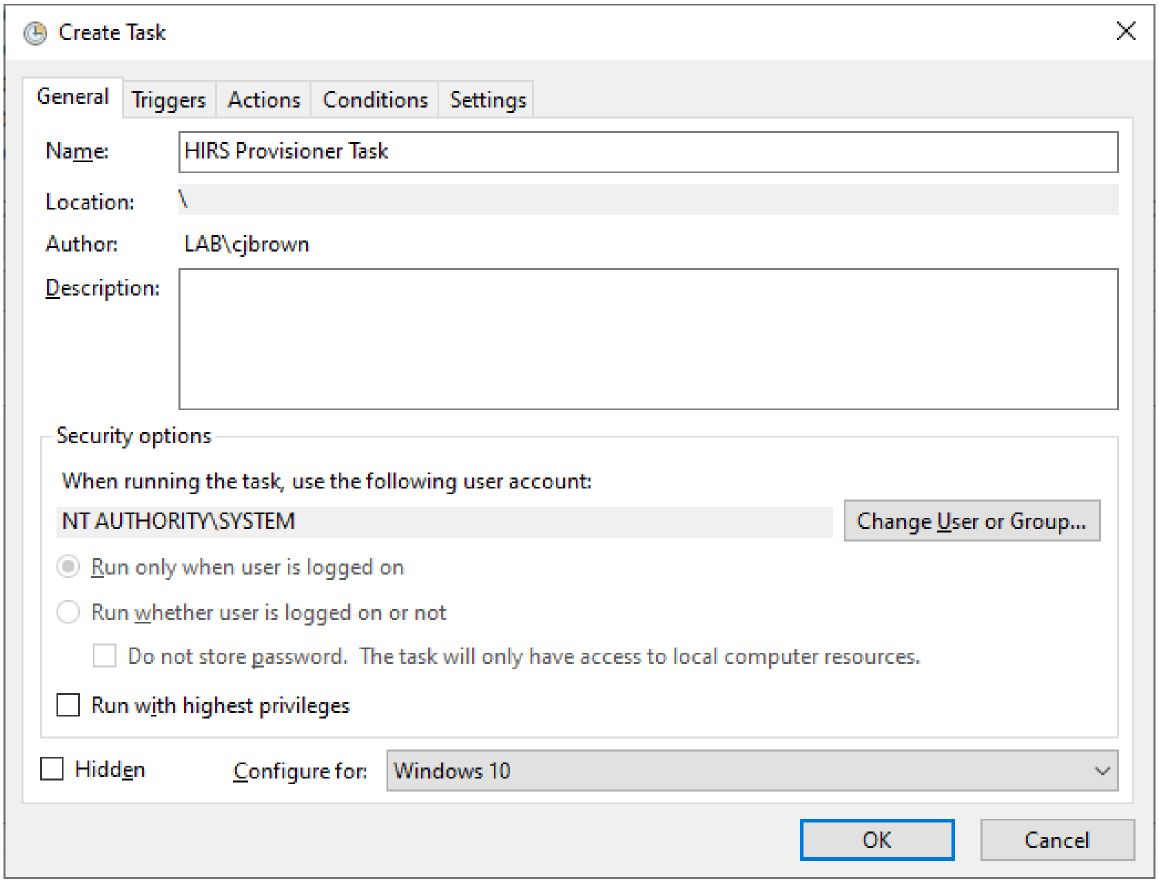
30. An installation wizard will launch. Click **Next** and then the **Install** button. The installation package will warn that the laptop will require a reboot. Accept the warning.
31. Follow the prompt to reboot the laptop. After the reboot, check the installation by manually launching the agent. If successful, a browser window will launch with a message similar to the following.



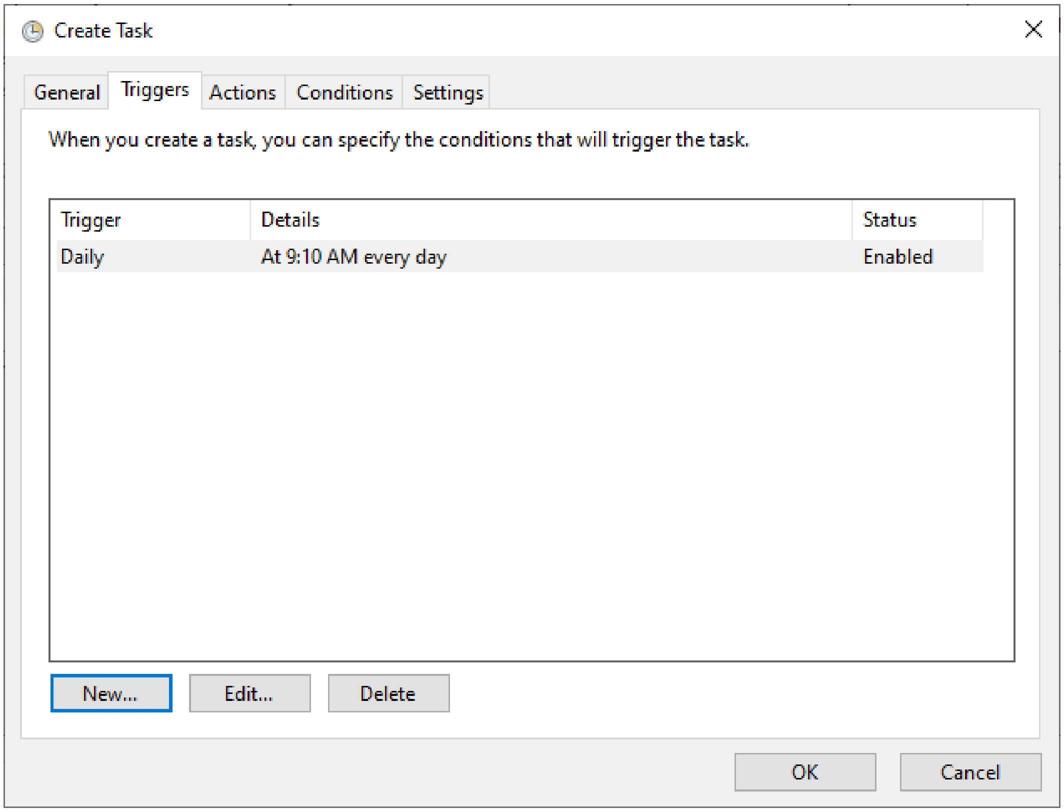
2.2.1.3 Create the Scheduled Tasks

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

32. Open the Task Scheduler as an Administrator on the target laptop.
33. Select **Action > Create New Task.**
34. In the **General** tab, enter a name for the task in the **Name** field. Click the **Change User or Group** button and select the *System* account. Select *Windows 10* from the **Configure for** pull-down menu.



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



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

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

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

2.2.2 Servers

The Dell R650 used in this demonstration does not require any preparatory activities for acceptance testing. All platform validation tools are included in the network-booted acceptance testing environment. Continue with creating the WinPE acceptance testing environment as described in Section 2.1.1.2.

2.3 Eclypsium

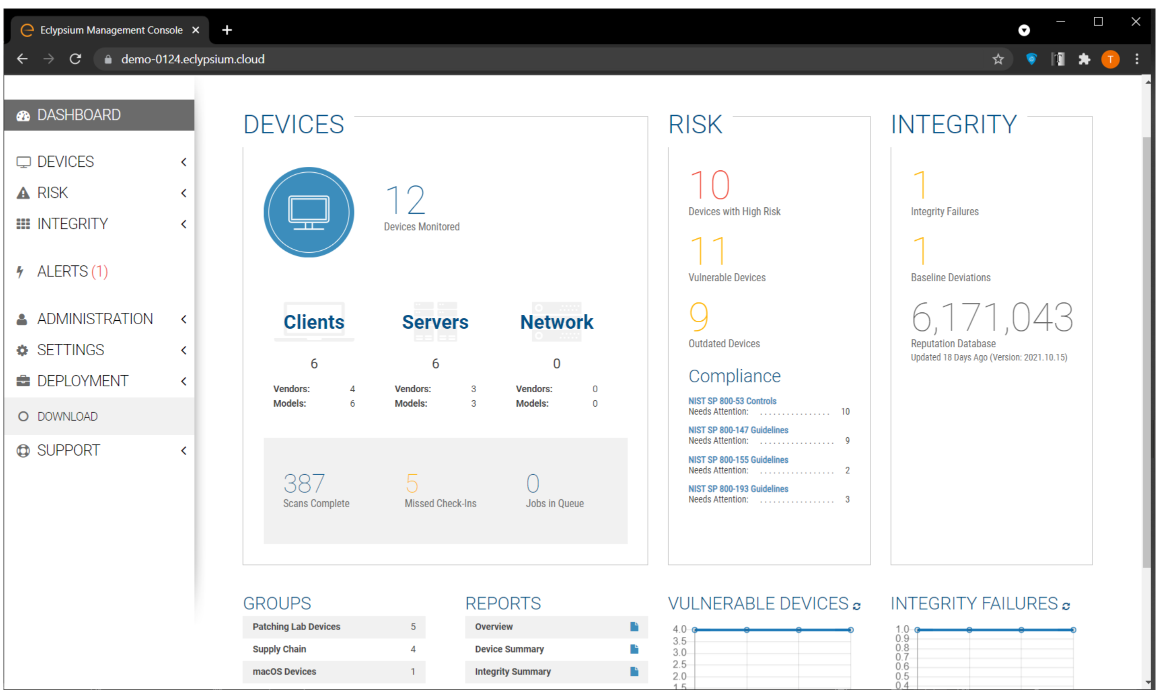
Eclypsium is a firmware security solution with cloud-based and on-premises deployment options. It secures firmware in servers, endpoints, and network devices by:

- identifying devices that contain firmware and creating detailed profiles of each component;
- verifying these profiles are free of vulnerabilities, have maintained their integrity, and are properly configured; and
- fortifying device firmware through a combination of configuration hardening, automated updates, and packaged guidance.

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

2.3.1 Download Eclypsium Agent

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



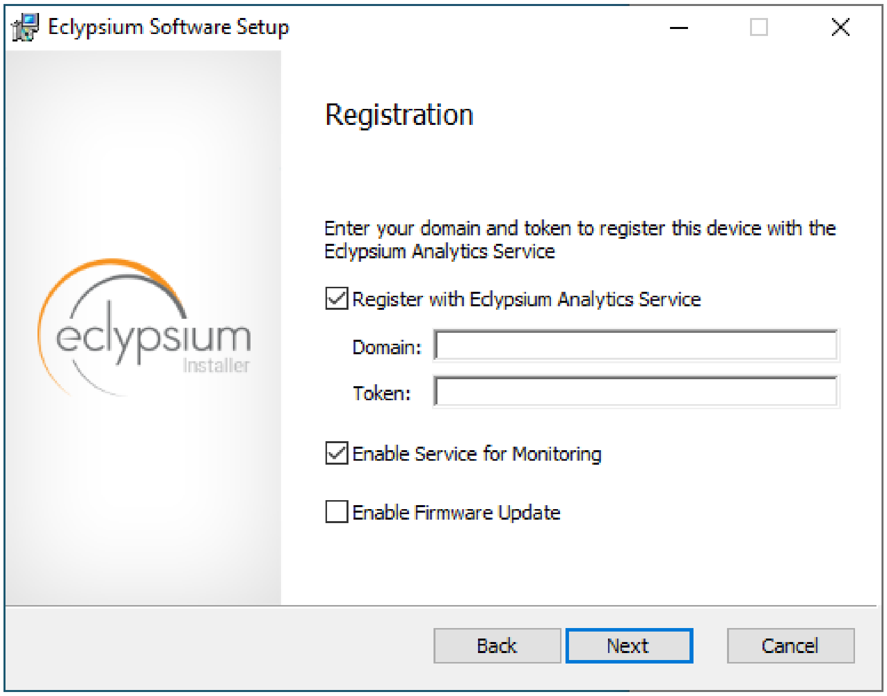
2. Select **Deployment > Download.**
3. Download the installer for the appropriate OS (Windows, macOS, Linux (Deb), or Linux (RPM)).

2.3.2 Install Eclypsium Agent for Windows

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

5. Select **Next**.

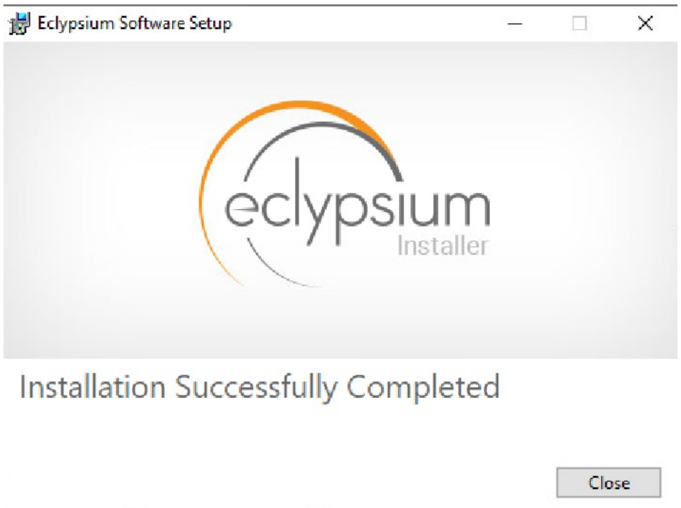
6. Ensure **Register with Eclypsium Analytics Service** and **Enable Service for Monitoring** are selected. Enter the **Domain** and Registration **Token** that can be found on the Download page of the **Eclypsium Management Console**, then select **Next.**



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

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

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



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

2.3.3 Install Eclypsium Agent for Linux

1. Ensure the *App* and *Driver* installation packages that are appropriate for your distribution are available on the host server system. The example below is an Ubuntu distribution.
2. Install the packages with the following command with root privileges. Note that there may be prerequisite packages that are required before installing the Eclypsium packages.

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

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

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

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

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

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

Successful registration.

[Dumping system firmware through SPI] \ 16777KB

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

```
[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.

```
  GNU nano 2.3.1                    File: /etc/hosts

127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.11.5 hirs_aca.ad.ent1.sca.nccoe.nist.gov hirs_aca
```

```
  GNU nano 2.3.1                    File: /etc/hostname                    Modified

hirs-aca
```

```
  GNU nano 2.3.1                    File: /etc/resolv.conf                 Modified

; generated by /usr/sbin/dhclient-script
search ent1.sca.nccoe.nist.gov
nameserver 192.168.11.2
```

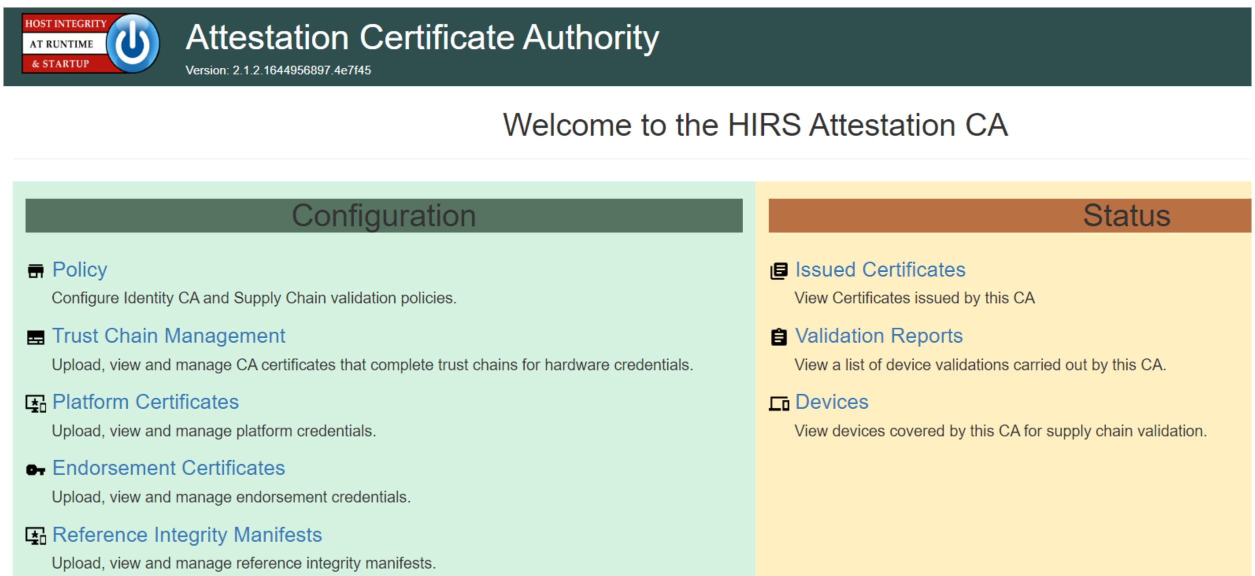
5. Install the HIRS-ACA dependencies using the following command. This will install MySQL/MariaDB, OpenSSL, Tomcat, Java, RPM Dev Tools, GNU Core Utilities, and other Linux commands (initscripts, chkconfig, sed, grep, firewalld, and policycoreutils).

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

6. Download the latest version of HIRS ACA from the [Release](#) page on GitHub and execute the following command to install the HIRS ACA.

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

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



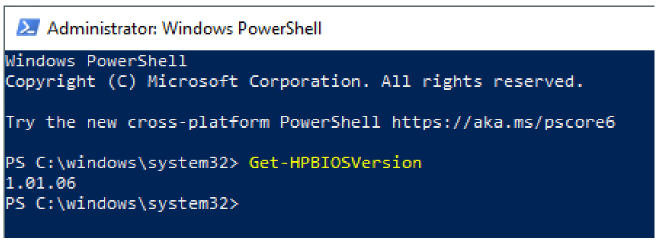
2.5 HP Inc.

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

2.5.1.1 Install the HP CMSL

7. Download the latest CSML from the HP Developers [website](#) onto the target HP Inc. laptop.

8. Launch the executable file and proceed through the wizard. Accept the agreement and click **Next.**
9. Select **Install into PowerShell** path and click **Next.**
10. Click **Install.**
11. Click **Finish.**
12. Test the installation by opening PowerShell as an administrator and executing a CMSL command such as `Get-HPBIOSVersion`.



2.5.1.2 Execute Provisioning Steps

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

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

```
Set-HPSecurePlatformPayload -PayloadFile EKProvisionPayload.dat
Set-HPSecurePlatformPayload -PayloadFile SKProvisionPayload.dat
```

14. Reboot the laptop. A local administrator must accept the *Physical Presence Prompt* to complete provisioning of the Endorsement and Signing Key.
15. Execute the following commands from PowerShell as an administrator.

```
Set-HPSecurePlatformPayload -PayloadFile EnableEBAMPayload.dat
Set-HPSecurePlatformPayload -PayloadFile LAKProvisionPayload.dat
```

16. Reboot the laptop. This will expose settings that require a BIOS administrator be configured before the next step can be completed.
17. Execute the following commands from PowerShell as an administrator.

```
Set-HPSecurePlatformPayload -PayloadFile BIOSsettingsPayloadFile.dat
```

```
Set-HPSecurePlatformPayload -PayloadFile SureRecoverProvision.dat
```

2.6 Hewlett Packard Enterprise (HPE)

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

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

18. Download and extract the source code from the HPE repository.
19. Install the software prerequisites onto the system.

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

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

```
mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
pcvt_v1.0.0/lib/HIRS_Utils-1.1.1.jar -DgroupId=HIRS_Utils -
DartifactId=HIRS_Utils -Dversion=1.1.1 -Dpackaging=jar -
DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
pcvt_v1.0.0/lib/HIRS_Structs-1.1.1.jar -DgroupId=HIRS_Structs -
DartifactId=HIRS_Structs -Dversion=1.1.1 -Dpackaging=jar -
DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
mvn install:install-file -Dfile=/<pcvt_source_directory>/PCVT-
pcvt_v1.0.0/lib/paccor-1.1.3-2.jar -DgroupId=paccor -DartifactId=paccor -
Dversion=1.1.3-2 -Dpackaging=jar -
DlocalRepositoryPath=/<pcvt_source_directory>/.m2/repository
```

21. Build the PCVT.

```
mvn clean compile assembly:single
```

22. Change to the **diskScan** directory.
23. Set the **GOPATH** to a local directory and set **GO11Module** to **off**.

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

```
go env -w GO111MODULE=off
```

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

```
./build/create_install_bundle.sh
```

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

2.7 Intel

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

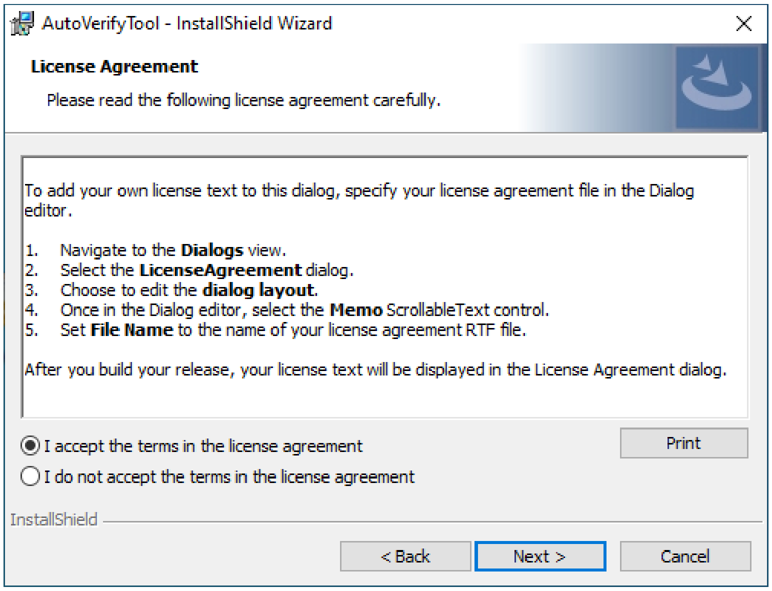
2.7.1 Laptops

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

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

1. Download and install the latest Microsoft Visual C++ Redistributable for Visual Studio.
2. Launch the AutoVerifyTool installation wizard. Click **Next**.
3. Accept the license and client **Next**.



4. Enter your Name and Organization. Click **Next**.
5. Select the **Typical** installation. Click **Next**.
6. Click **Install**.

2.7.2 Servers

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

Table 2-4 Intel-Contributed Server

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

Additionally, to complete the implementation we connected the Seagate enclosure to this server board. 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

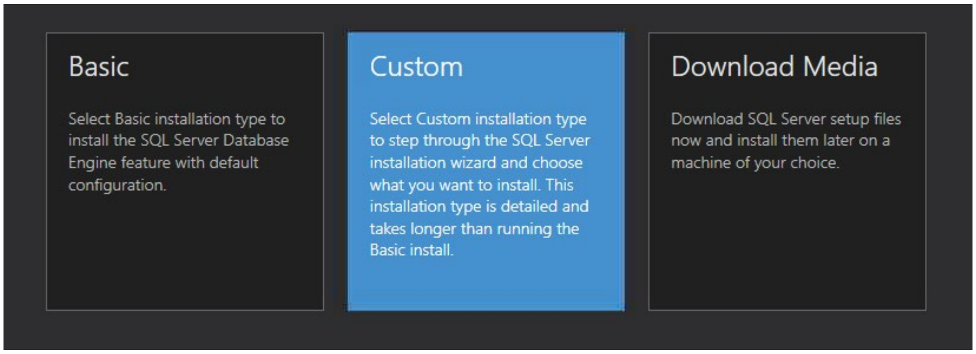
Machine Name	Machine Type	Operating System
Archer Database Server	Virtual	Windows 2019 Server
Archer Services	Virtual	Windows 2019 Server

2.8.1 Prerequisites

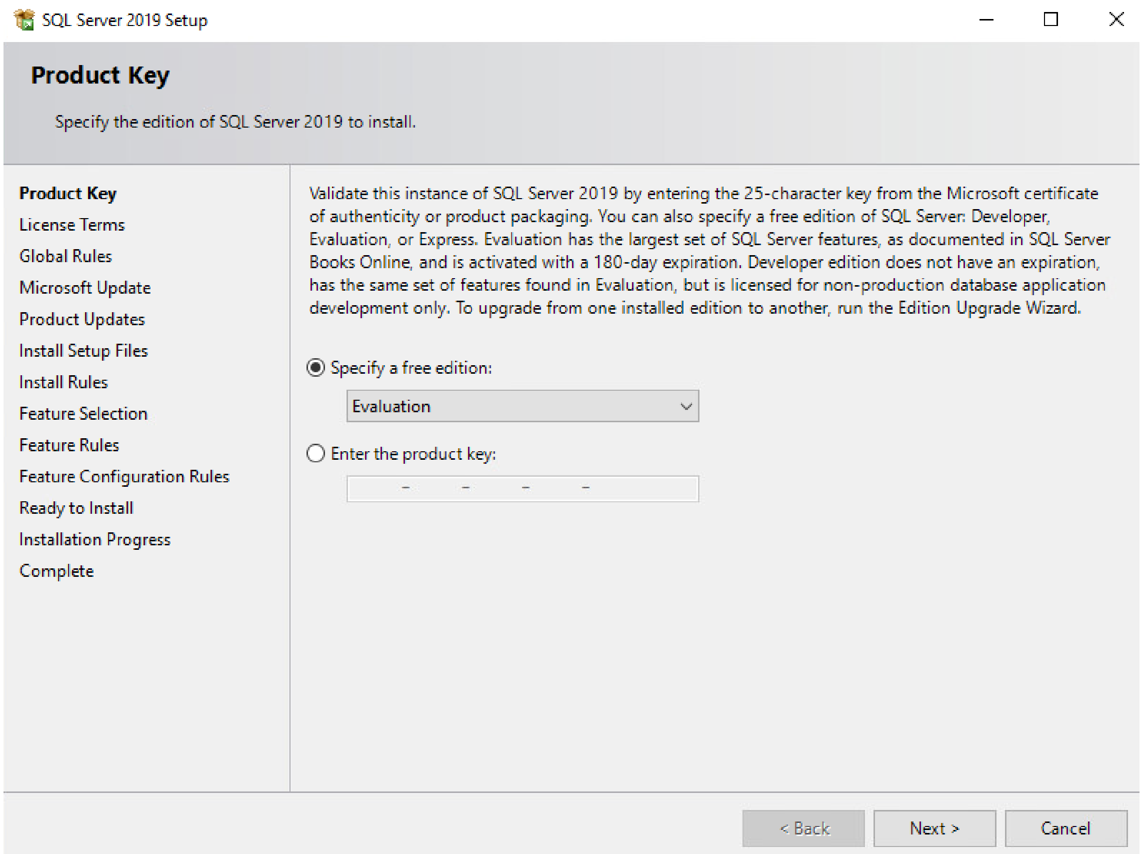
Before installing Archer IRM services, several prerequisites must be fulfilled. In this section, we describe those prerequisites involving the database server and Microsoft's Internet Information Services (IIS) web server.

2.8.1.1 Install SQL Server on Database Server

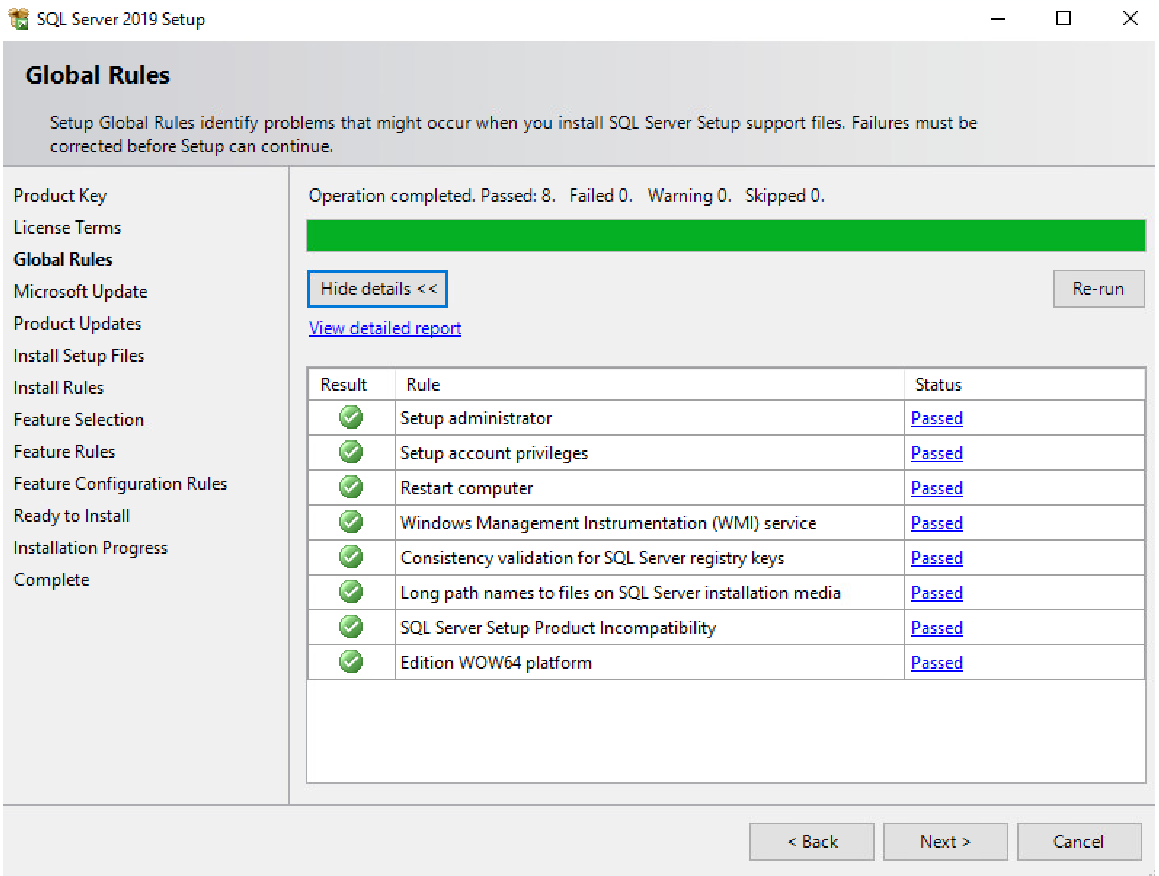
1. Download SQL Server 2019 from https://www.microsoft.com/en-us/sql-server/sql-server-downloads onto the database server.
2. Run the SQL Server 2019 executable.
3. Select the **Custom** installation type.



4. Specify the download location and select **Install.**
5. Allow the installer to download the SQL Server 2019 package.
6. The SQL Server Installation Center should automatically open. From the left menu panel, select **Installation.** Select the option **New SQL Server stand-alone installation or add features to an existing installation**.
7. Enter the product key or select a free edition of the software. Then select **Next.**



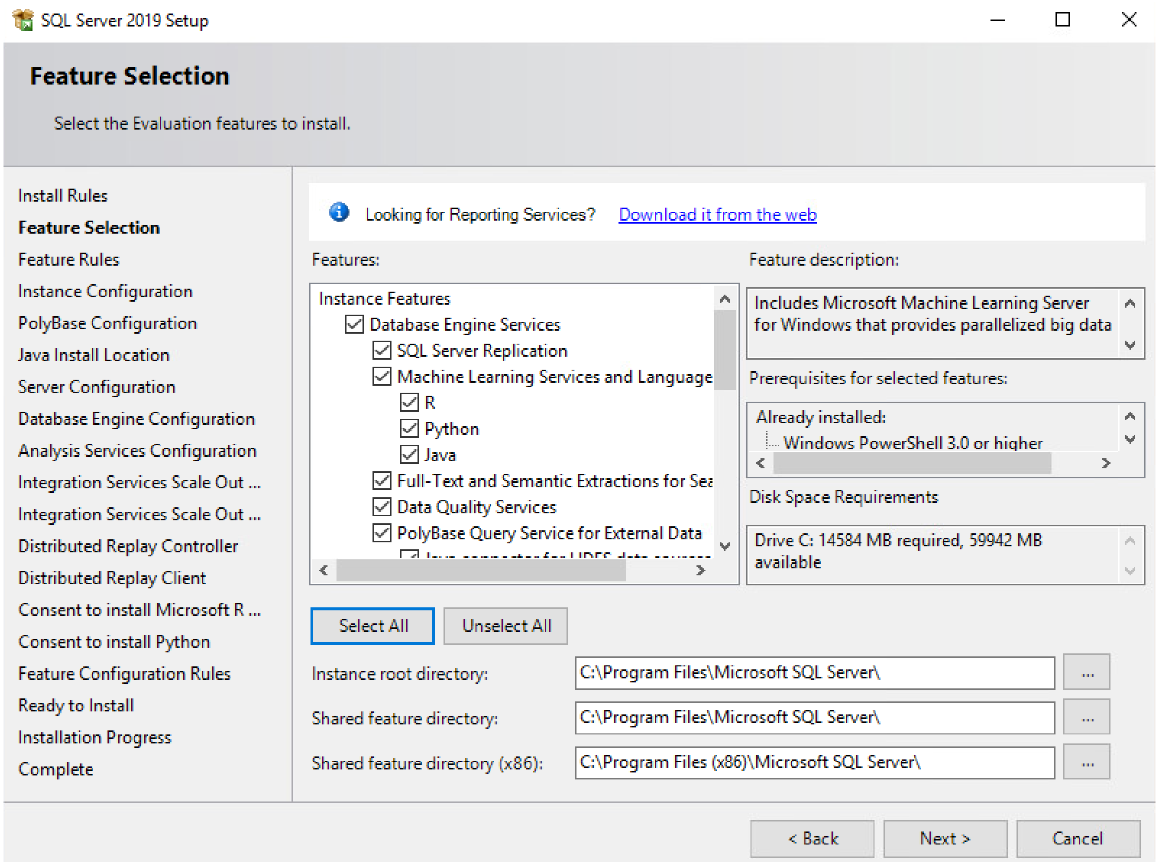
8. Read and accept the License Terms. Then select **Next**.
9. Ensure that all the **Global Rules** have passed. Then select **Next**.



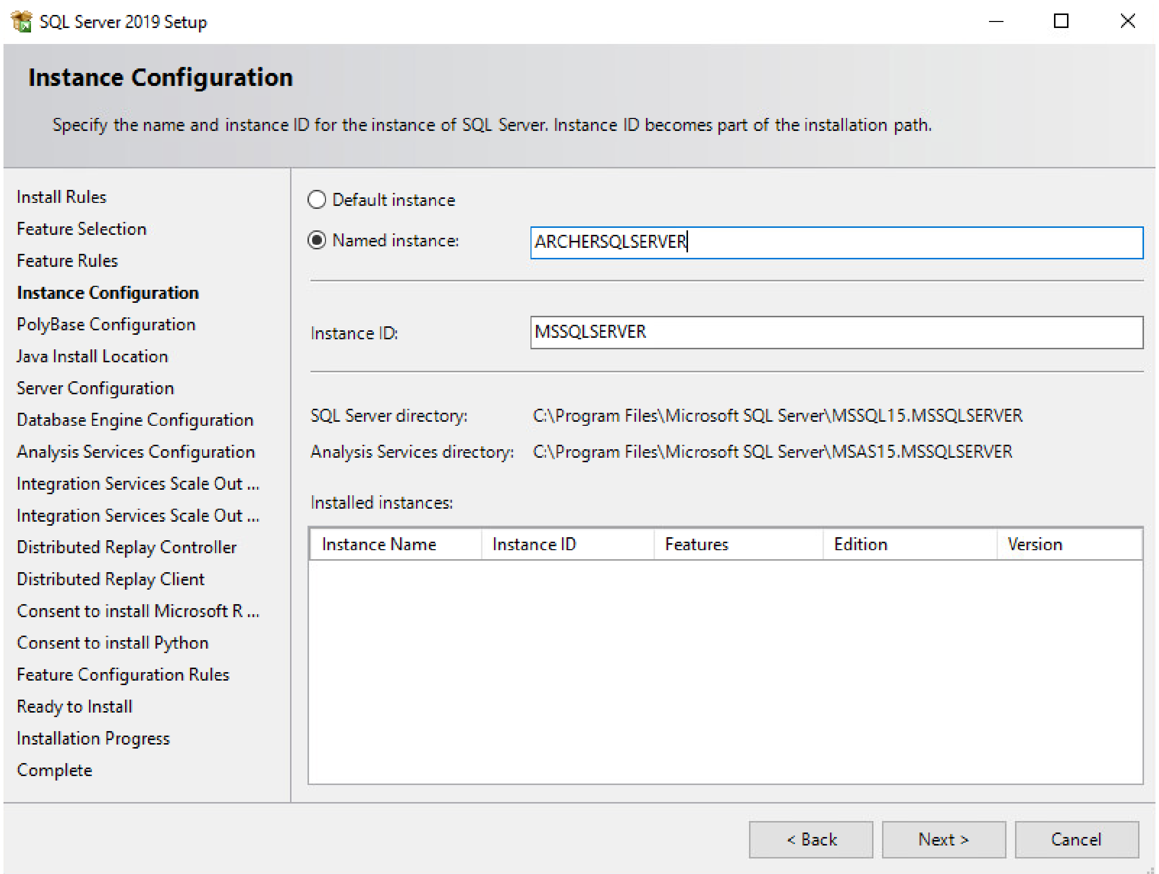
10. To use Microsoft Update to automatically deliver updates, check the box **Use Microsoft Update to check for updates (recommended).** Then select **Next.**

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

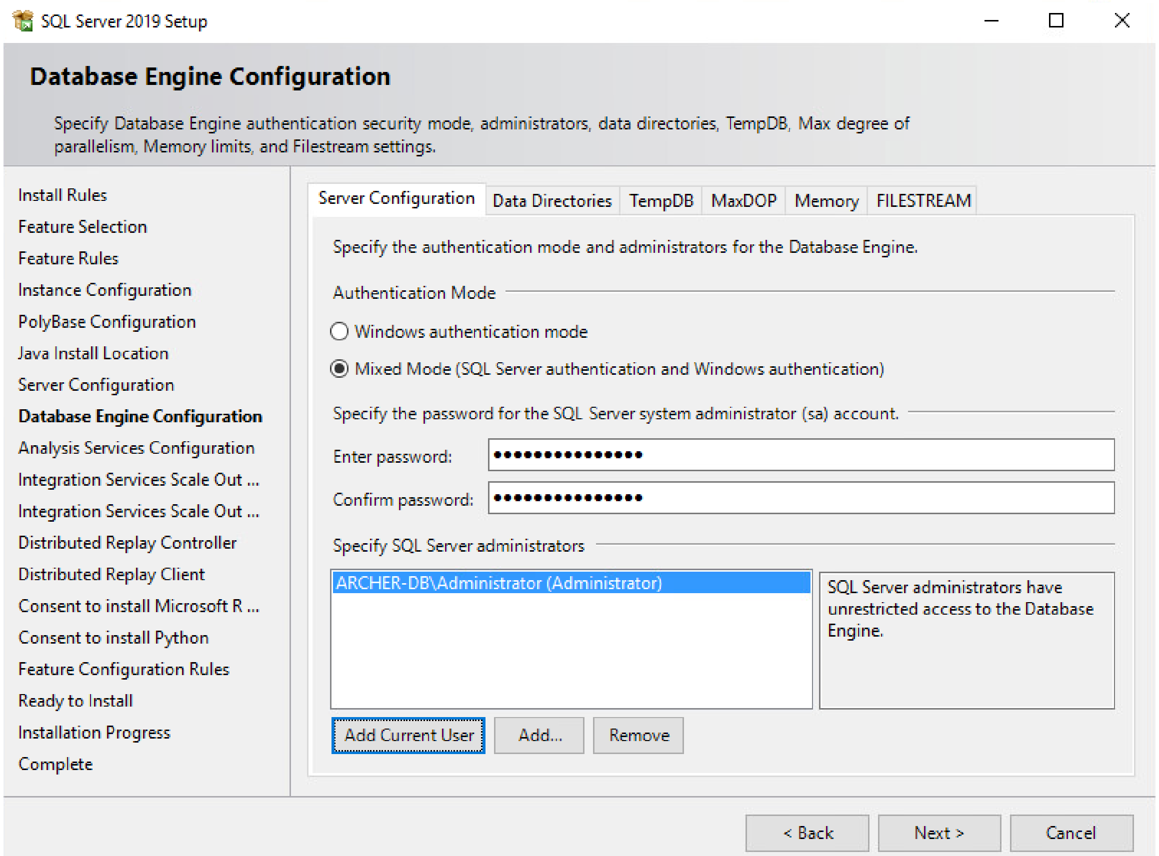
12. Select the desired features to install. Then select **Next.** Complete the sections for the selected features.



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



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



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

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

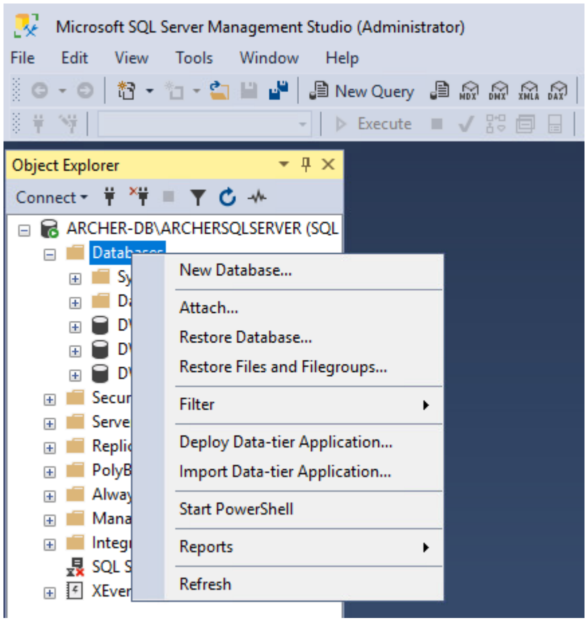
17. Once the installation completes, select **Close.**

2.8.1.2 Create the Archer IRM Databases

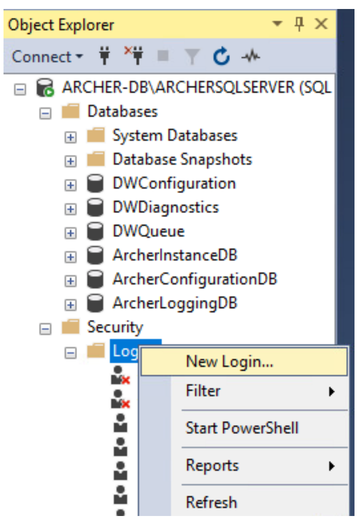
1. Download SQL Server Management Studio (SSMS) from https://aka.ms/ssmsfullsetup. Follow the installation steps.

2. Once installed, open SSMS.

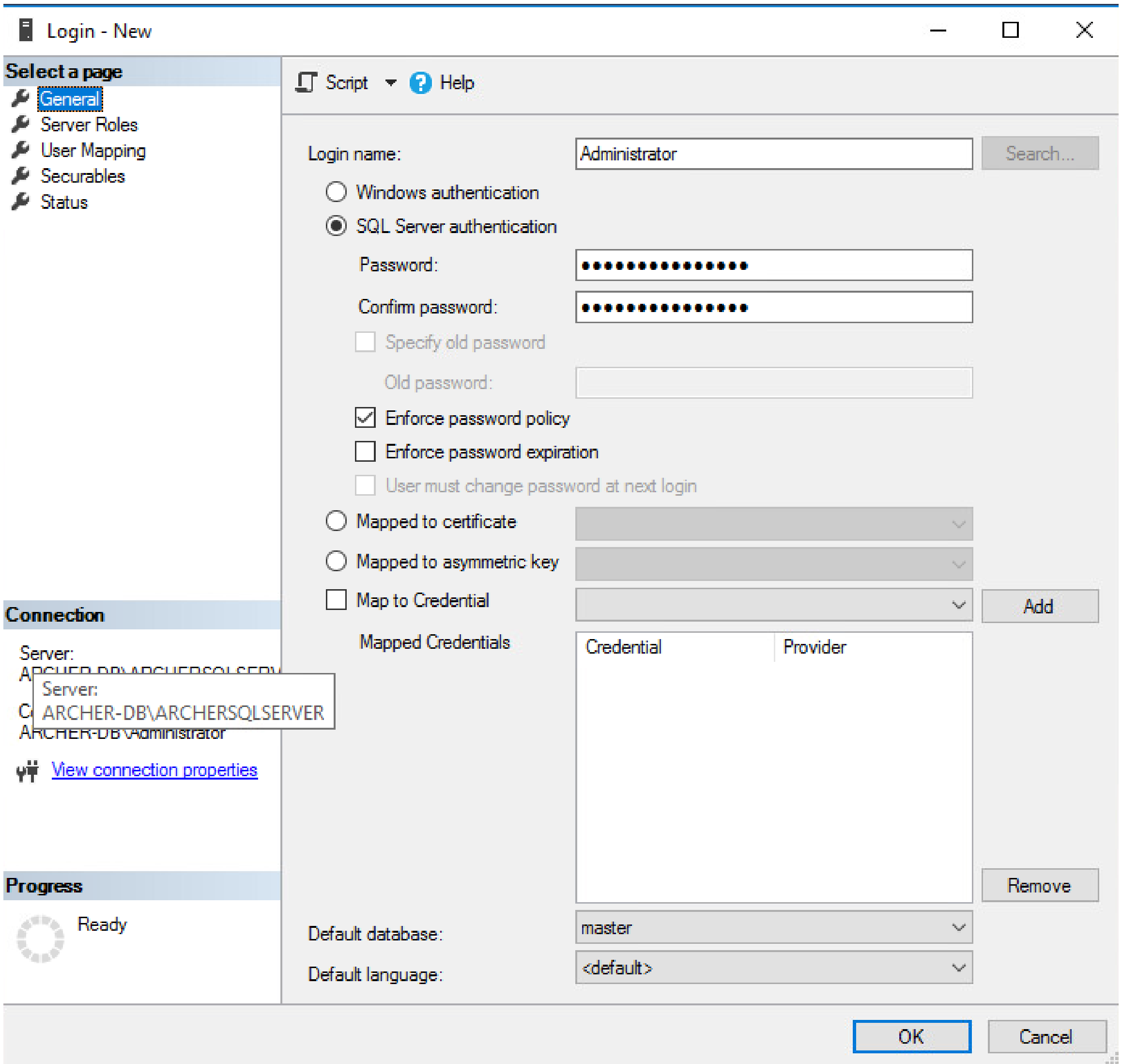
3. Expand the ARCHERSQLSERVER tree. Right-click on **Databases** and select **New Database.** Create three databases: *ArcherInstanceDB, ArcherConfigurationDB,* and *ArcherLoggingDB.*



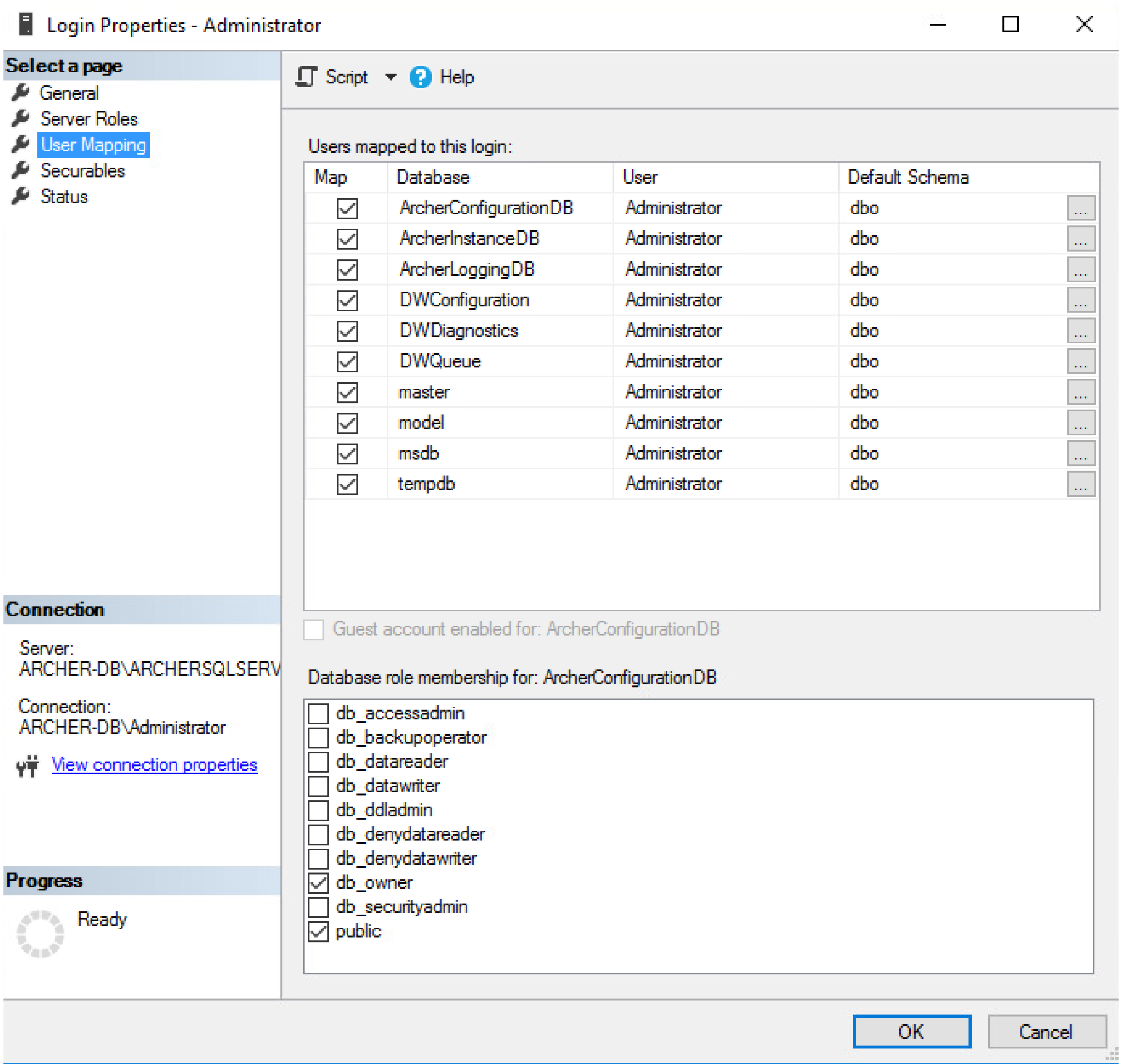
4. Next, create a local Administrator user. Right-click **Security** and select **New Login.**



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

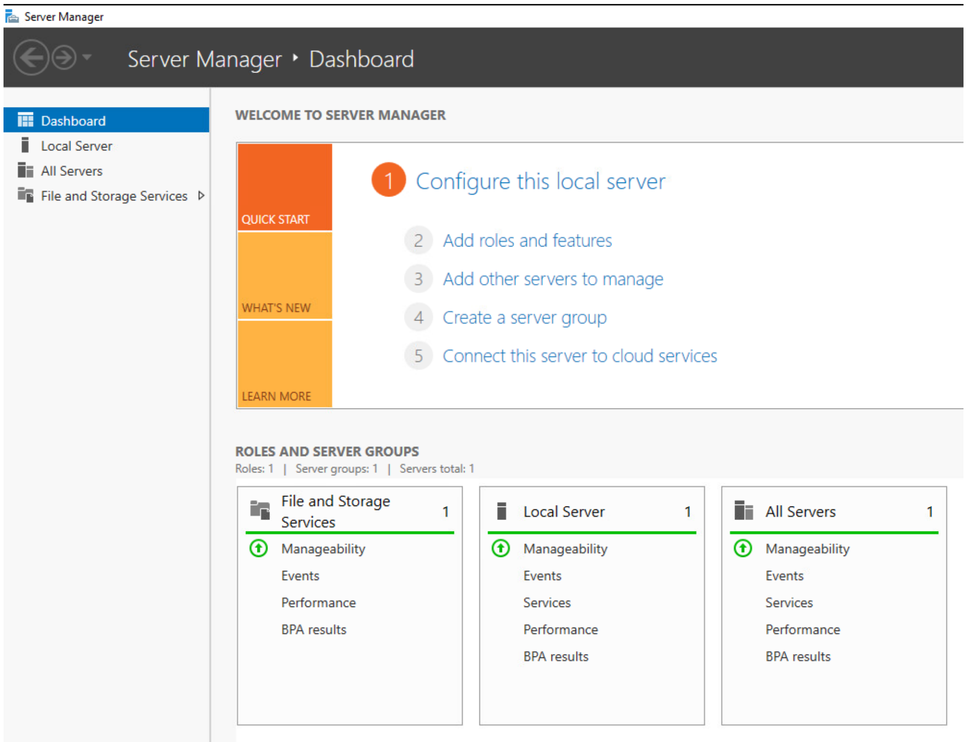


6. Navigate to the **User Mapping** tab. Ensure all the databases have the **Default Schema** set to **dbo.** Also, ensure that **db_owner** is selected for each database under the **Database role membership** section. **Select OK.**



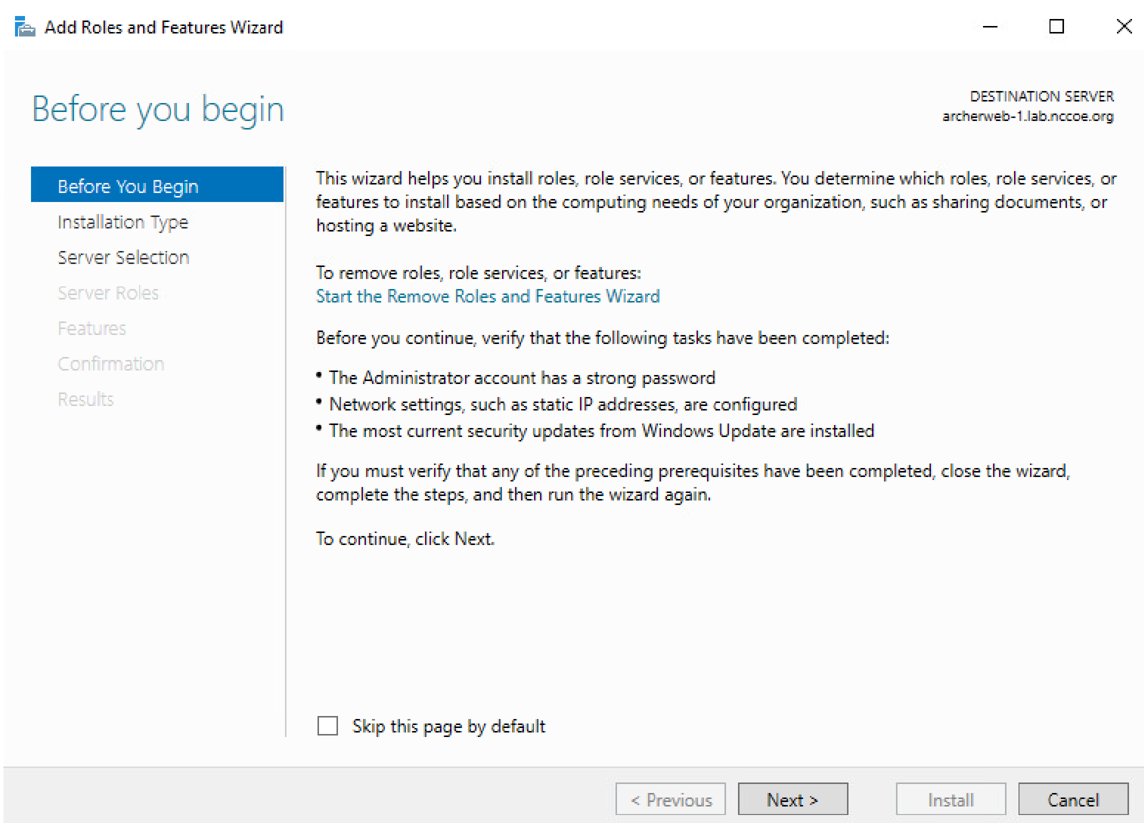
2.8.1.3 Install Internet Information Services on the Web Server

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

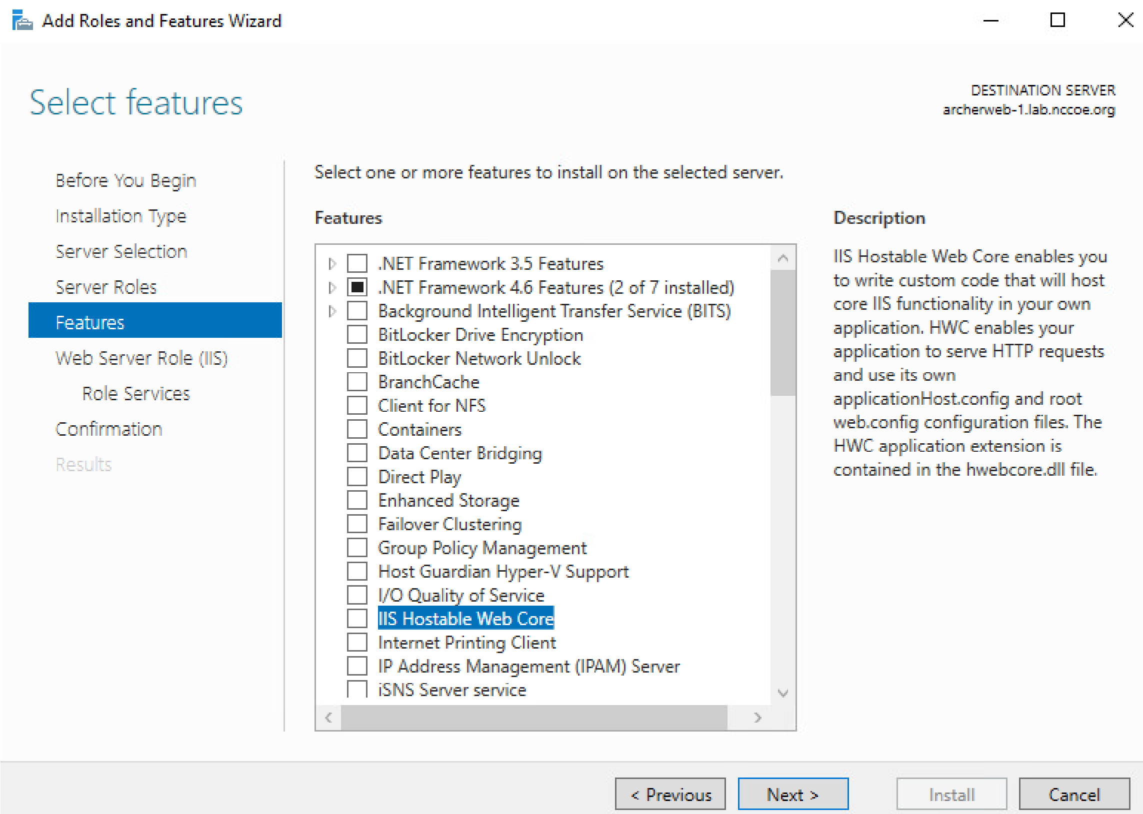


756 2. Under **Manage**, select **Add Roles and Features.**

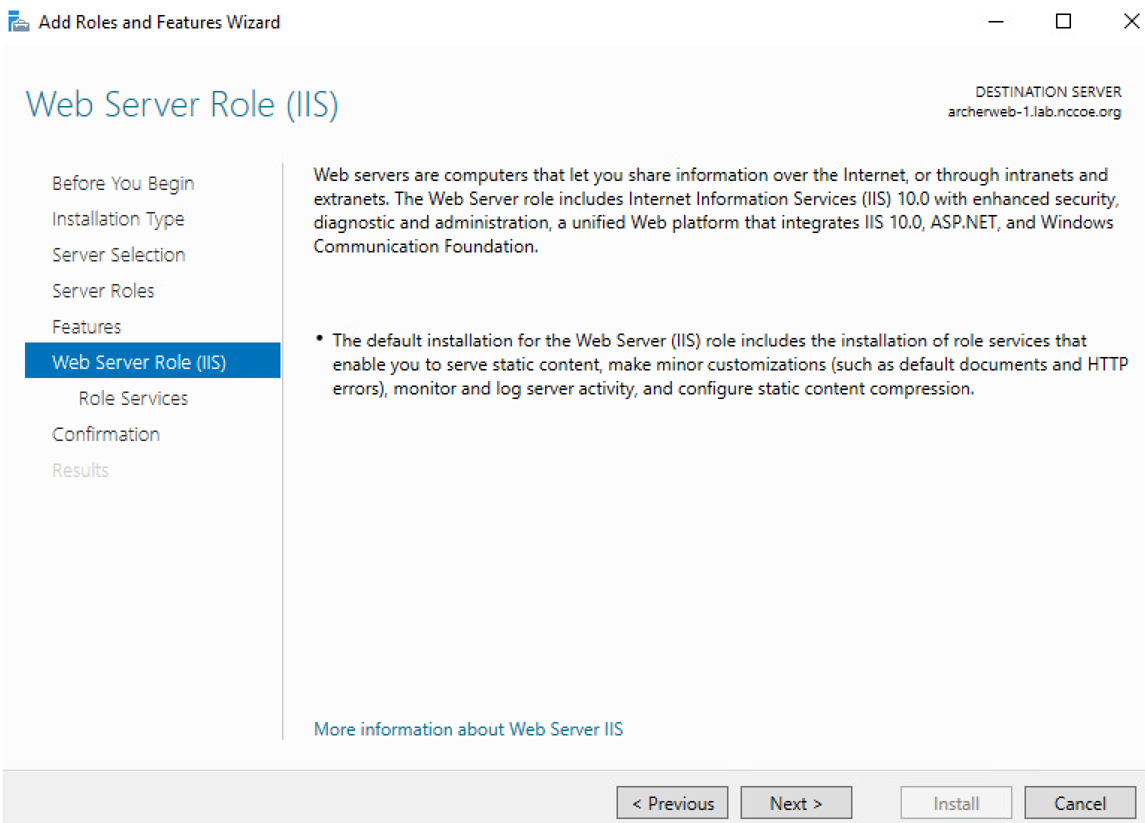
757 3. Select **Next.**



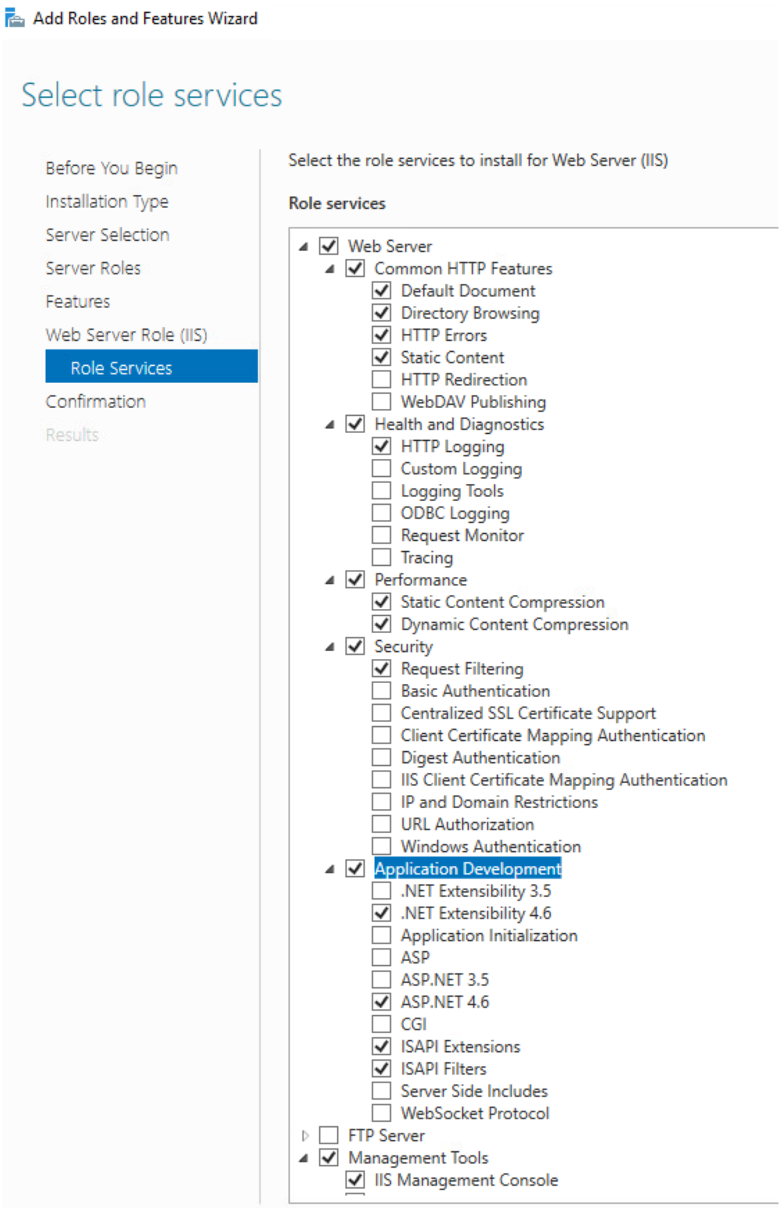
4. Select the **Role-based or feature-based installation** radio button. Select **Next.**
5. Select the **Web Server (IIS)** server role. Then select **Next.**
6. In the pop-up window, select **Add Features.**
7. Select **Next.**



762 8. Select **Next.**



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

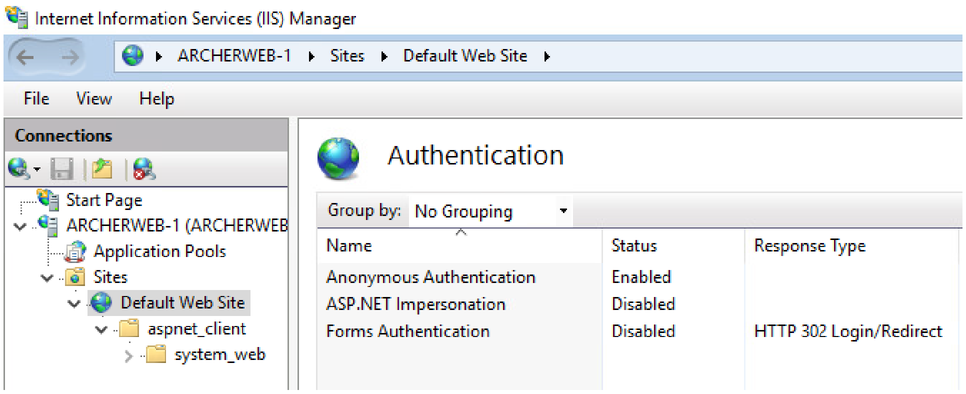


10. Confirm that the selected options are correct and select **Install.**
11. Once the installation completes, select **Close.**
12. Restart the computer.

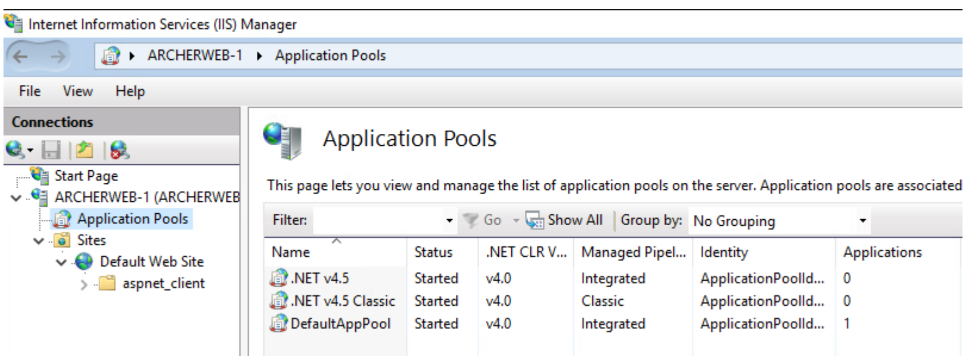
2.8.1.4 Configure IIS

1. Open the IIS application.

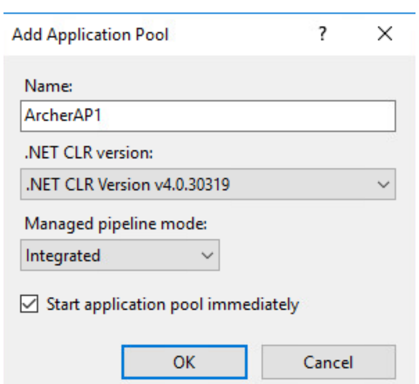
2. Click on the web server in the left pane. **Select Authentication.**
3. Ensure that **Anonymous Authentication** is enabled and **ASP.NET Impersonation** and **Forms Authentication** are disabled for the **Default Web Site.**



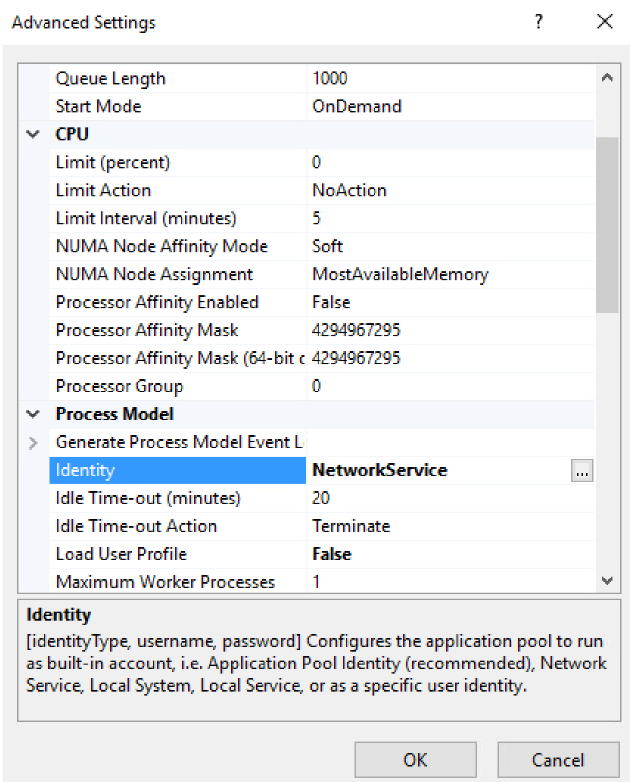
4. Expand the web server tree and select **Application Pools.** In the far-right pane, select **Add Application Pool.**



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



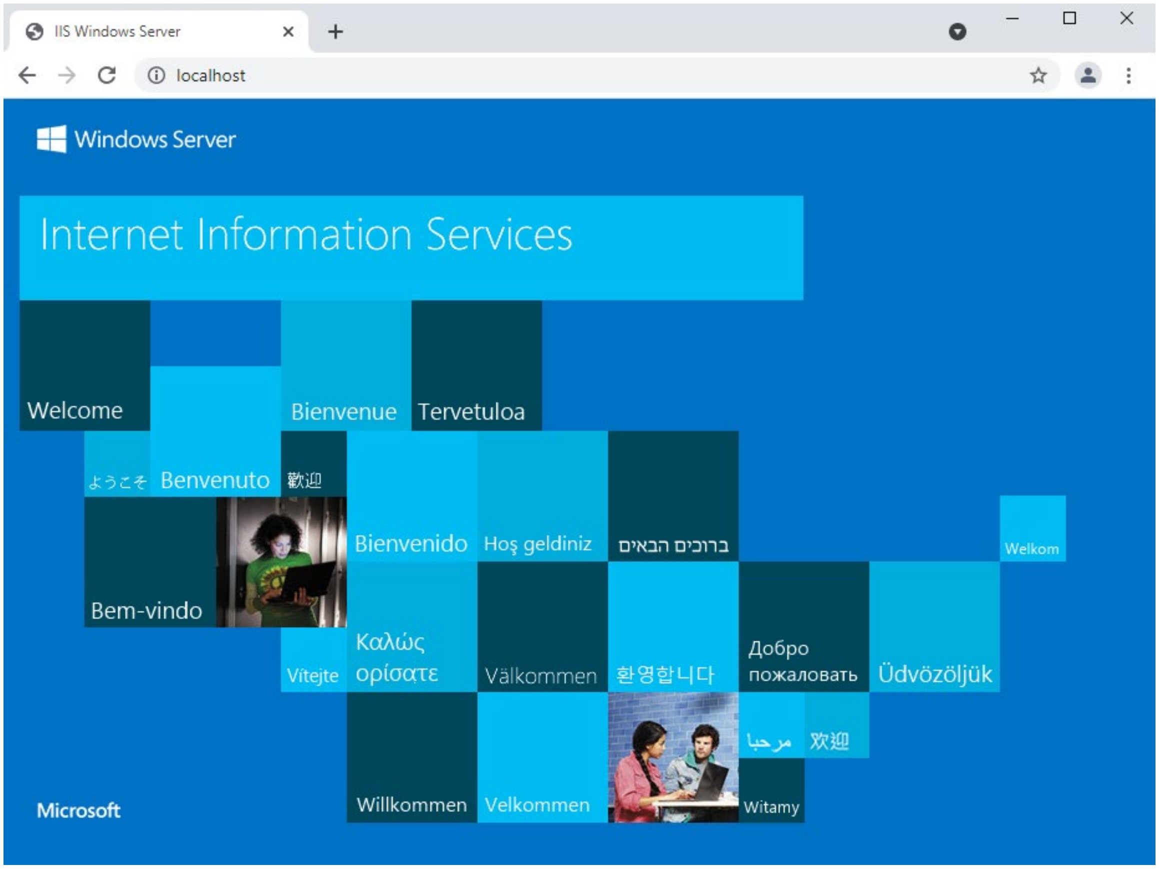
6. Right-click on the newly created application pool and select **Advanced Settings.** Under **Process Model,** select the ellipsis button that is next to the **Identity** field.



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

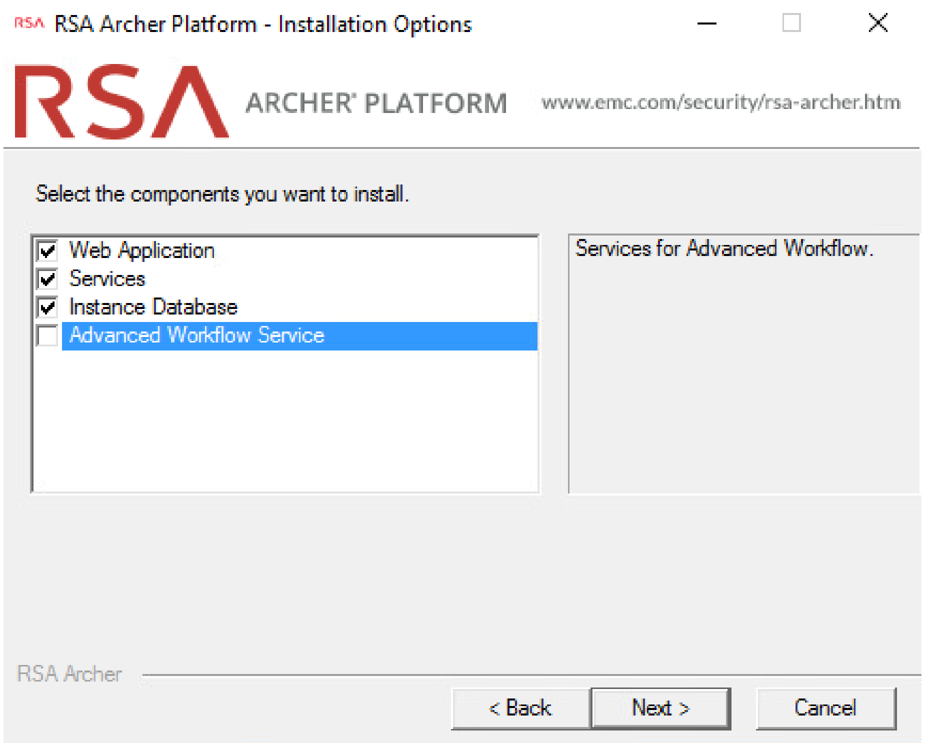
8. Click on the web server. In the far-right pane, select **Restart.**

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

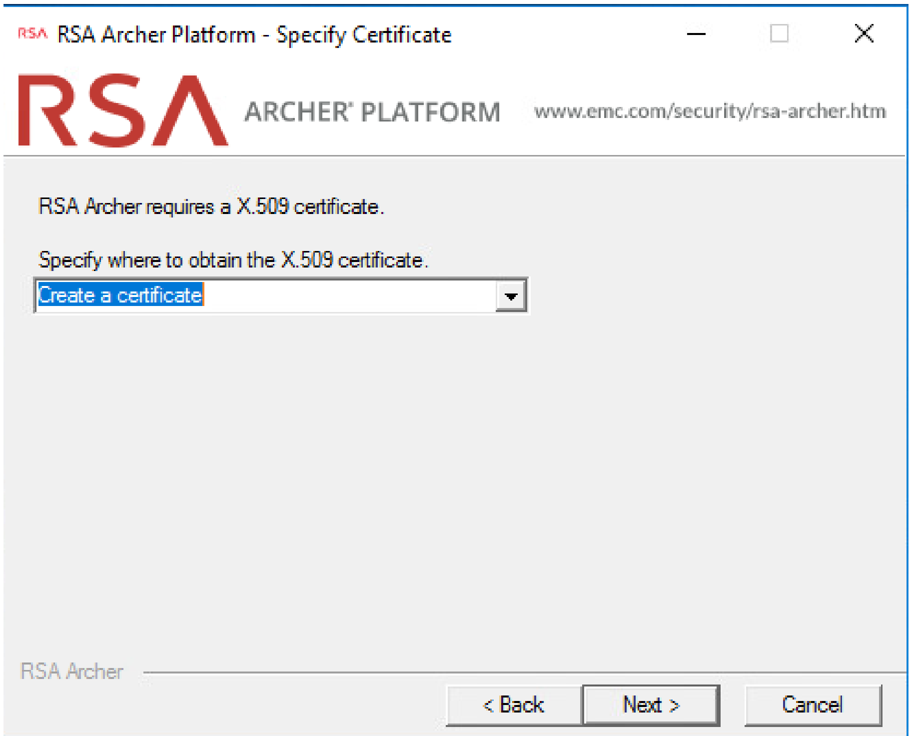


2.8.2 Archer IRM Installation

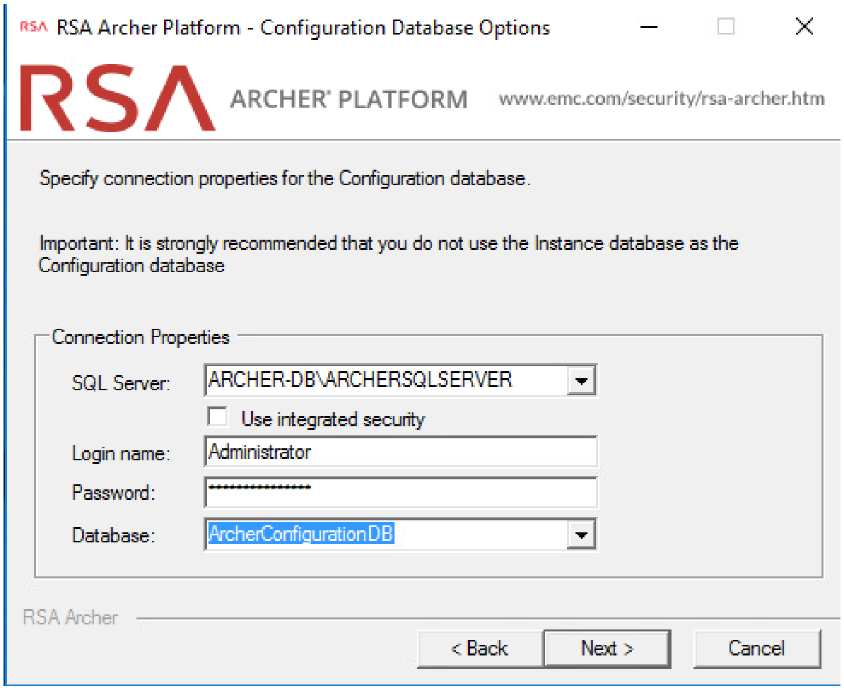
1. Before installing Archer IRM, .NET Framework version 4.7.2 must be installed. It can be downloaded at https://dotnet.microsoft.com/download/dotnet-framework/net472.
2. Extract the zip file that was downloaded from the Archer IRM download page.
3. Open the folder and run the executable **ArcherInstall.**
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.**

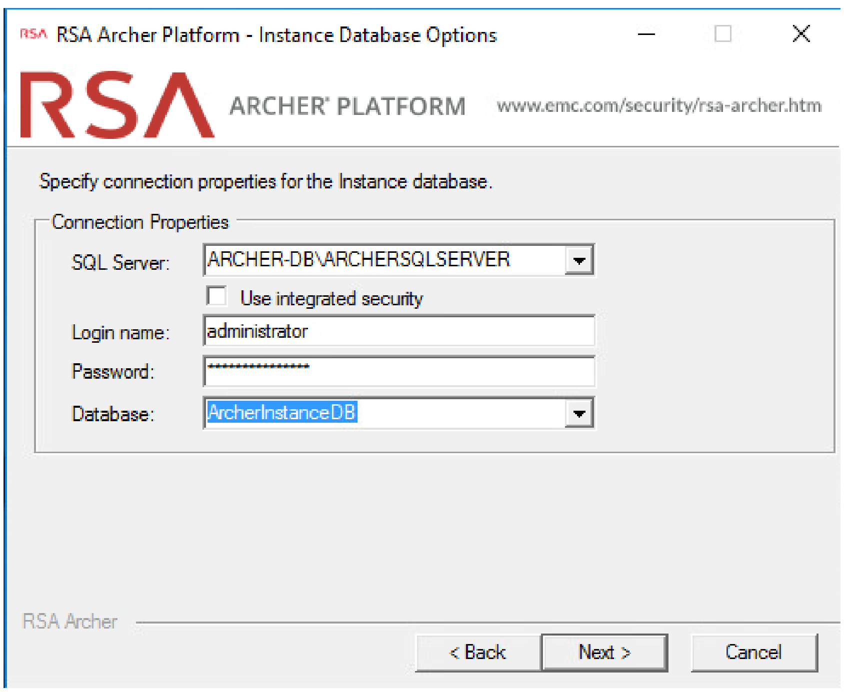


8. Select the database server that was previously created. Enter the credentials that were created in SSMS. Then select the configuration database from the dropdown menu and click **Next.**

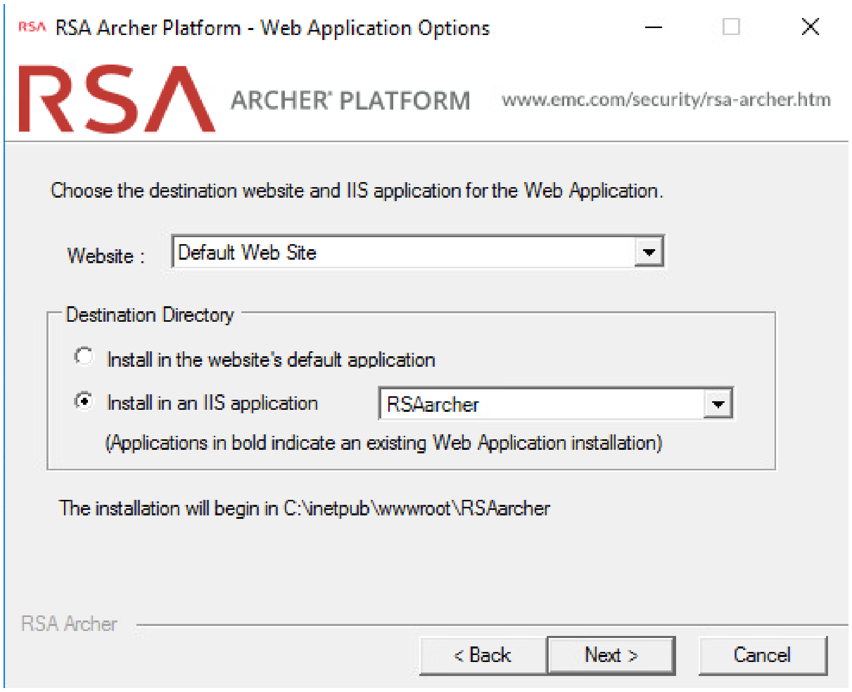


794 9. Select the preferred language from the dropdown menu and select **Next.**

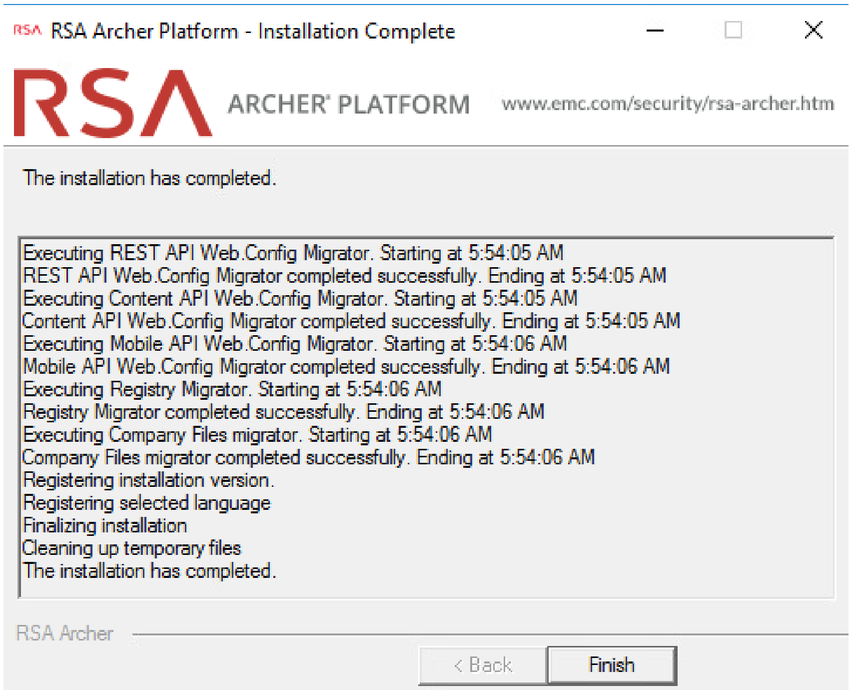
795 10. Repeat step 8 and select the instance database from the dropdown menu. Then select **Next.**



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

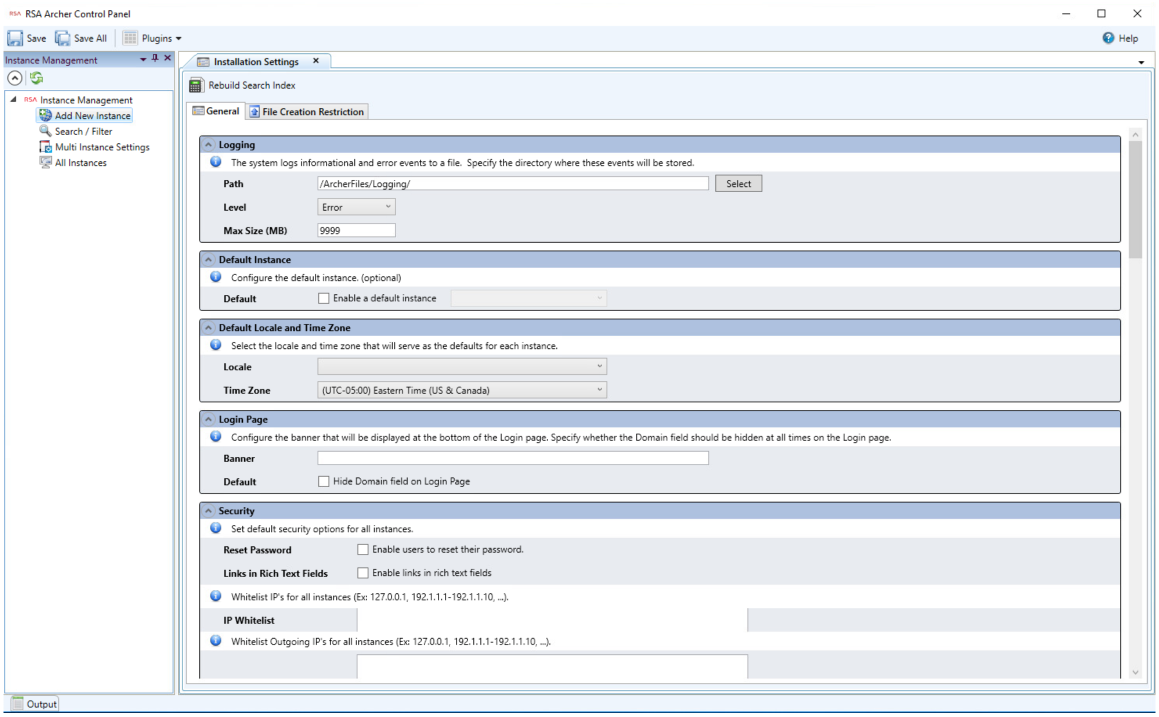


13. To add an Instrumentation Database, repeat step 8 and use the **ArcherLogging** database that was created in SSMS. Otherwise, select **Not using Archer IRM Instrumentation service.** Select **Next.**
14. Specify the account to run the services. Then select **Next.**
15. Confirm or edit the installation paths for the services and application files. Select the **Create Archer IRM program group for all users** radio button. Then select **Next.**
16. Confirm or edit the path for installation logs. Then select **Next.**
17. Select **Install** and wait for the installation to complete. Once completed, select **Finish.**



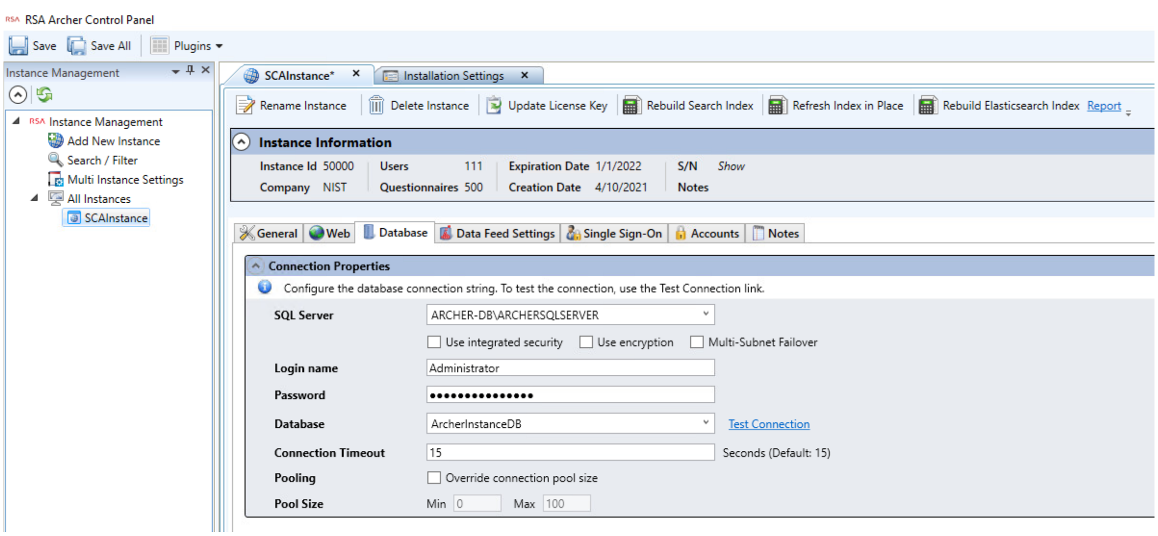
2.8.2.1 Configure Options in the Control Panel

1. Open the RSA Control Panel.
2. In the left pane, select **Add New Instance.**



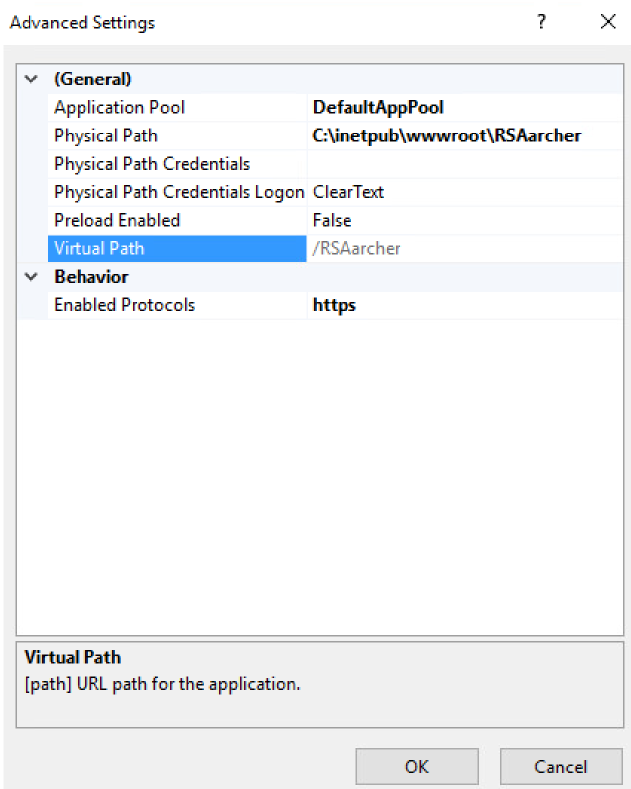
3. Enter a name for the instance in the **Instance Name** field. Select **Go.**

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

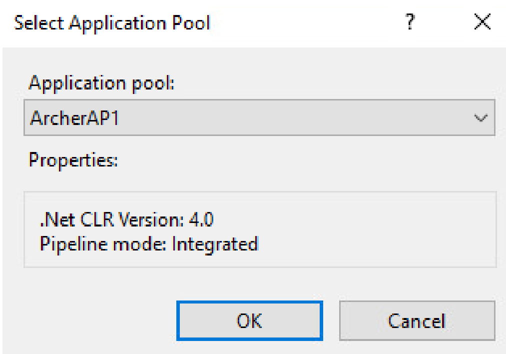


2.8.2.2 Add New Application to Application Pool

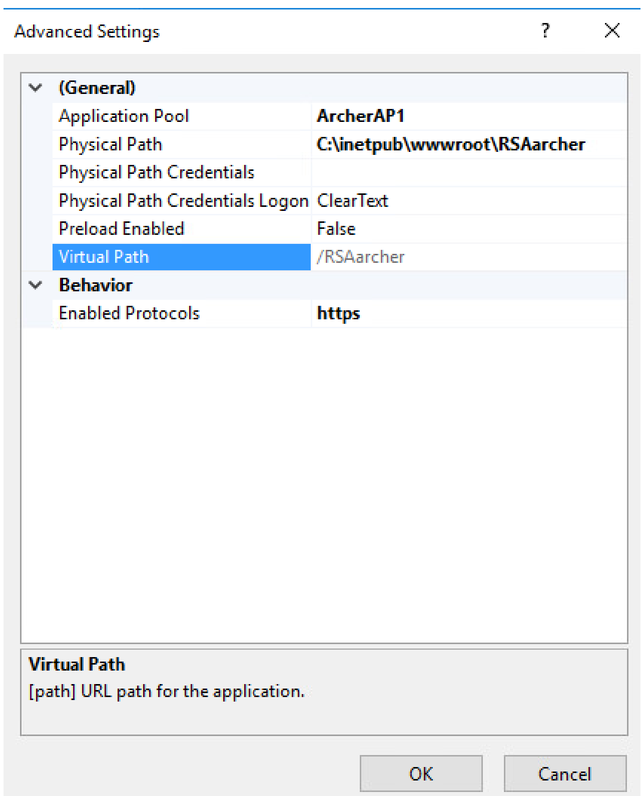
1. Navigate back to IIS. Expand the web server directory, expand the **Sites** directory, and expand the **Default Web Site** directory.
2. Select the RSAarcher site. Click on **Authentication** and ensure that **Anonymous Authentication** is the only thing that is enabled.
3. Right-click on the RSAarcher site and select **Manage Application > Advanced Settings.**
4. Click on **Application Pool** and select the ellipsis button. You will see a screen similar to the following:



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



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



7. Restart the Archer IRM site.

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

2.9 Seagate

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

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

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

```
*-disk:0
                description: SCSI Disk
                product: ST18000NM005J
                vendor: SEAGATE
                physical id: 0.0.0
                bus info: scsi@0:0.0.0
                logical name: /dev/sdb
                version: ET02
```

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

Additionally, we recommend using Seagate's [command line interface tool](#) that communicates with the drives via the Trusted Computing Group (TCG) Storage API to confirm successful integration. Use the following command to print drive information:

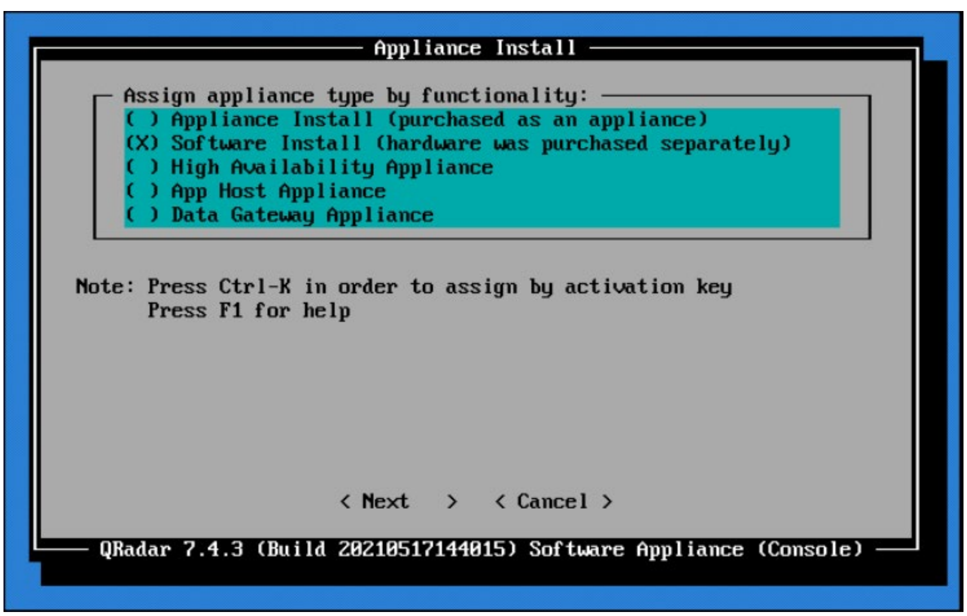
```
python3 sed_cli.py --device=/dev/sdb --operation=printdriveinfo
```

2.10 IBM QRadar

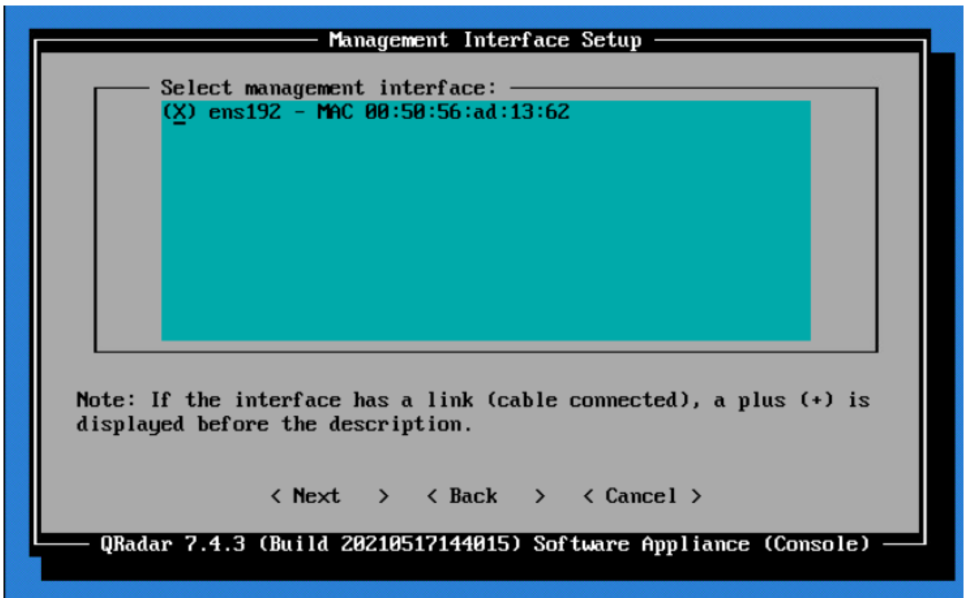
This section describes the installation of the IBM QRadar system for this demonstration. Our instantiation of IBM QRadar is viable for a lab environment, but the reader is encouraged to refer to the architecture planning guide on the IBM website for specific guidance for your environment.

We opted to install the full IBM QRadar suite onto a single virtual machine via an ISO provided by the IBM engineering team. Note that Red Hat Enterprise Linux Server V7.6 (or binary equivalent) must be deployed on the virtual machine before the QRadar installation. Once this prerequisite is met, boot the virtual machine using the ISO provided by IBM. This process will be unique to your environment. Next, follow the instructions provided by the IBM documentation website. The remainder of this section includes example screenshots from the installation wizard we used in our environment.

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



2. For the functionality, select **"All-In-One" Console.**
3. Select **Normal Setup (default)** as the type of setup.
4. Either manually adjust the date and time, or add the name or IP address of a Network Time Protocol (NTP) server to automatically update the date and time.
5. Select the appropriate time zone.
6. Select the appropriate network adapter that will allow communication with the installed system.



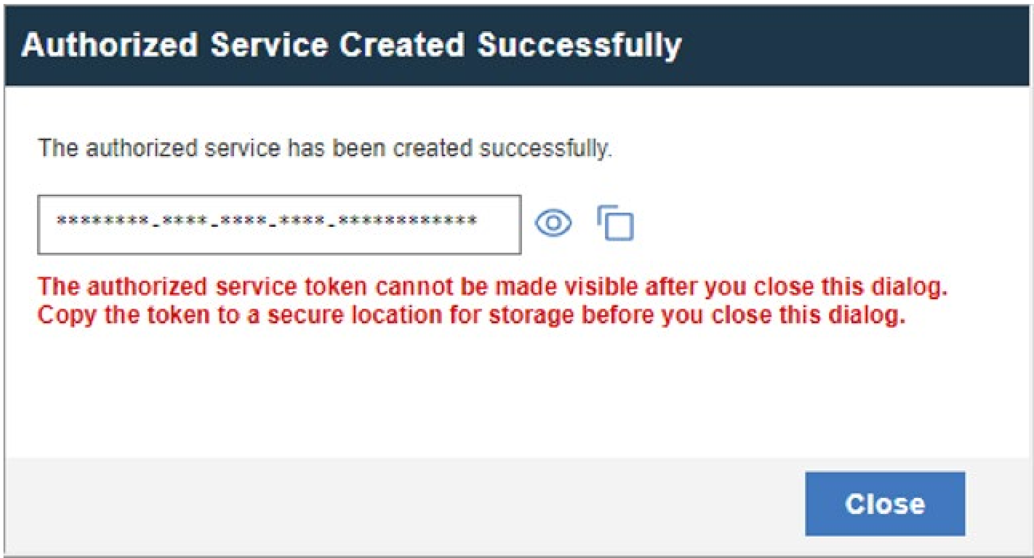
7. Enter the network information for this installation. Note that only static addresses are supported.
8. Set the Admin user password.
9. Set the Root password for console access.

2.10.1 WinCollect Agent

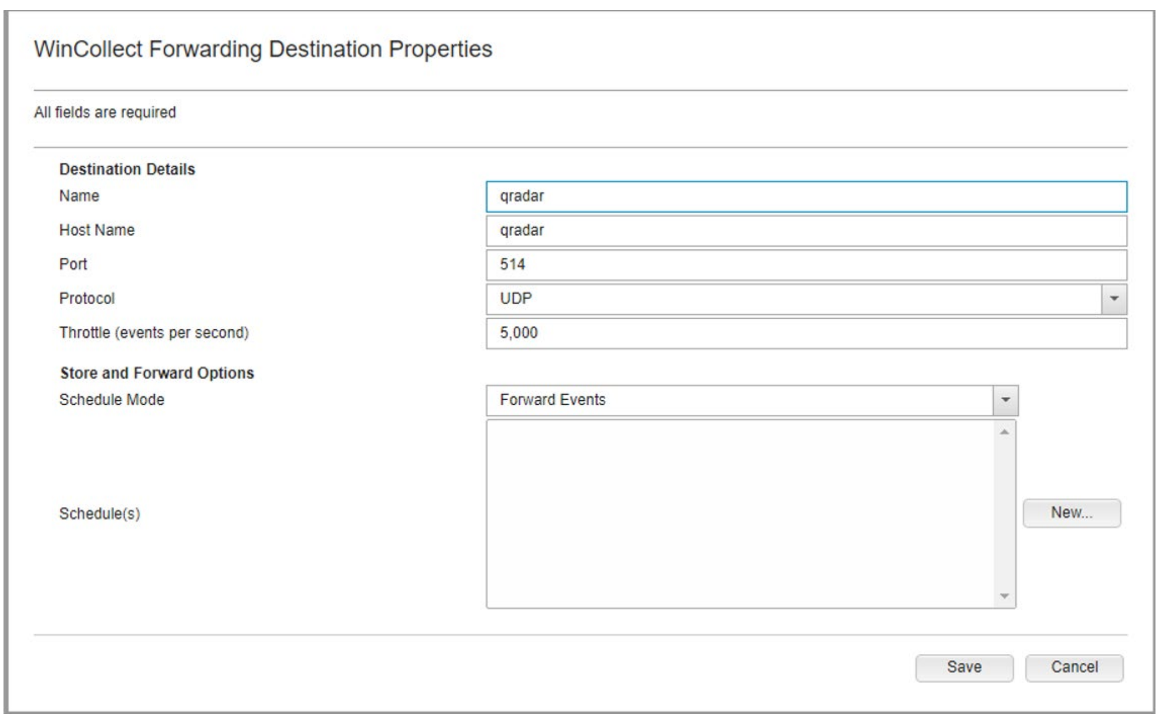
On a separate Windows Server system, configure and install the WinCollect agent. This component polls the remote hosts (laptops), and then sends event information to QRadar.

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

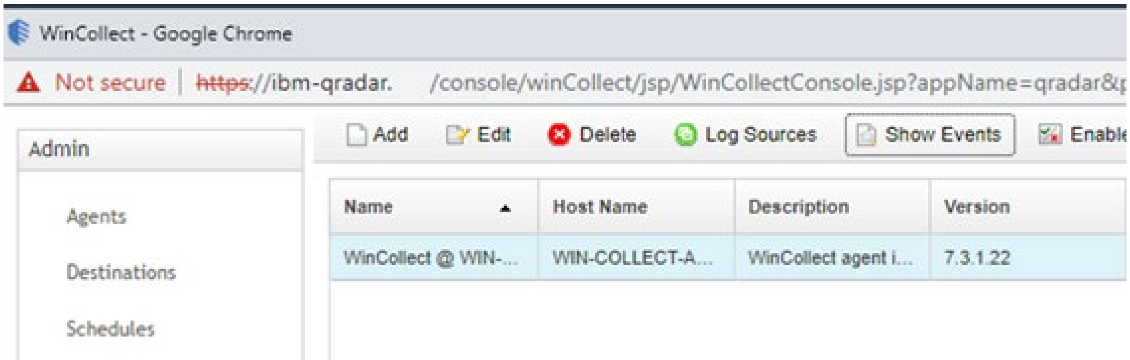
2. Create an authentication token so that the managed WinCollect agents can exchange data with QRadar appliances. This process is documented on the IBM [website](). Note that you will not be able to retrieve the token from QRadar after it has been created.



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



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



2.11 Integrations

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

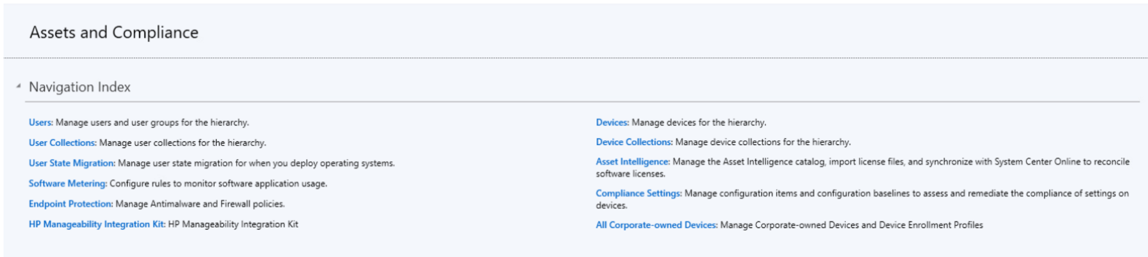
2.11.1 Microsoft Endpoint Configuration Manager and Platform Validation Tools

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

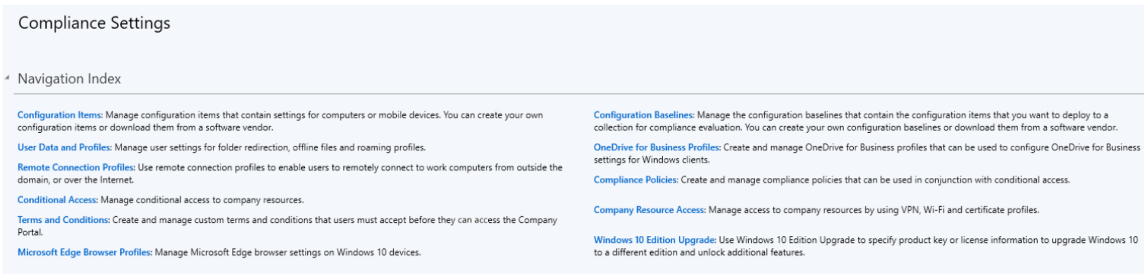
Similarly for HP Inc. and Dell laptops, the HIRS-ACA Windows-based Provisioner periodically monitors the changes to laptop components. We chose to use the same monitoring approach for consistency – the Windows task scheduler captures the result of the Provisioner execution and stores it in the Windows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

2.11.1.1 Set Up Configuration Item

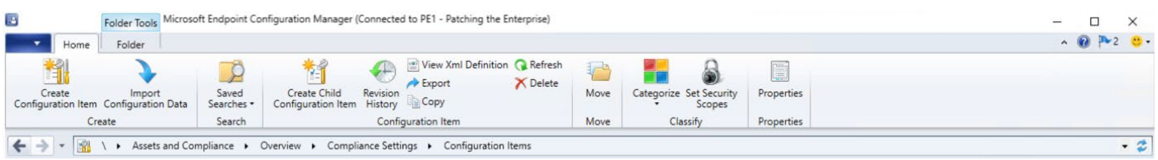
1. In the Microsoft Endpoint Configuration Manager console, under **Assets and Compliance > Overview**, select **Compliance Settings**.



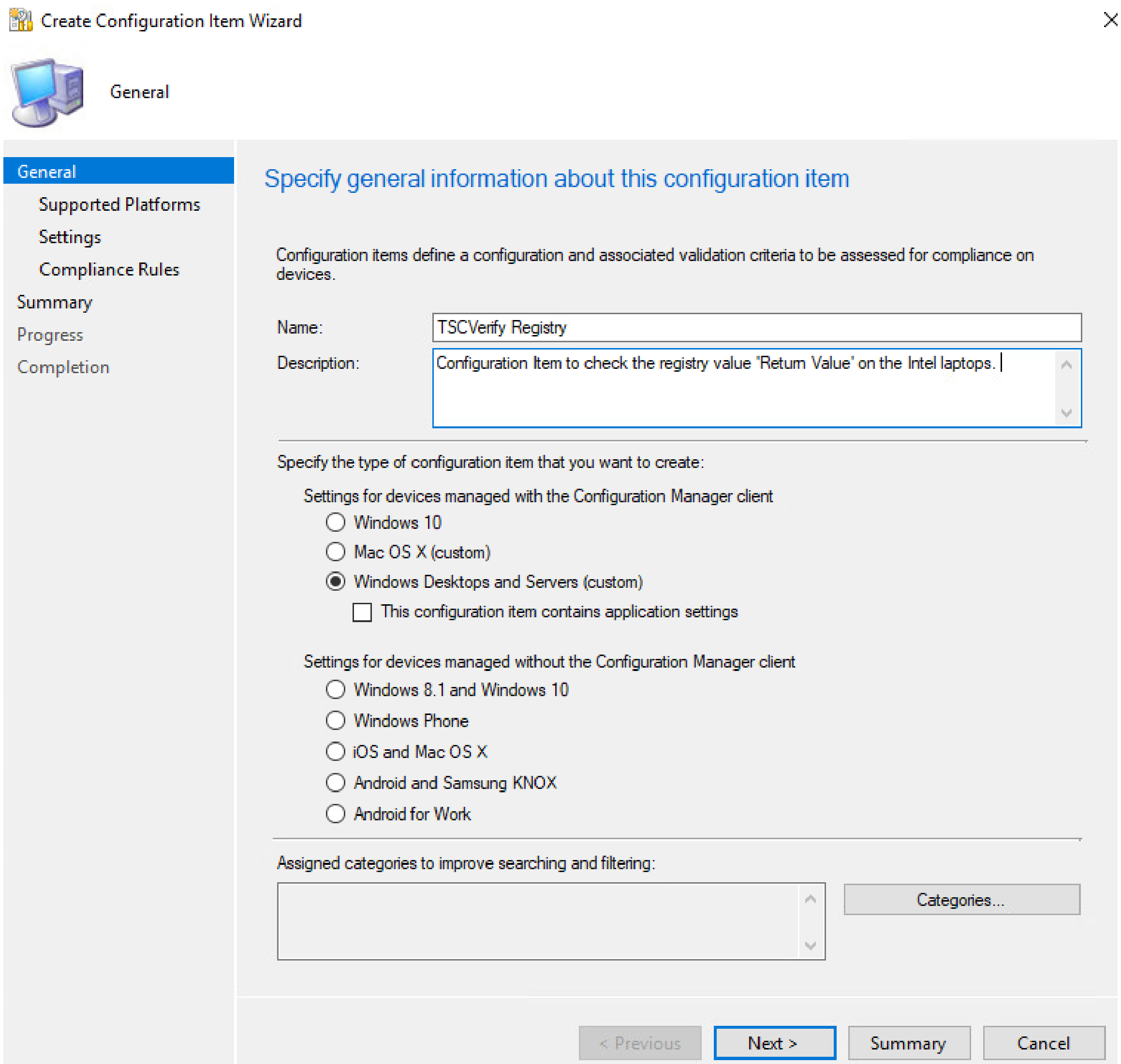
2. Next, select **Configuration Items**.



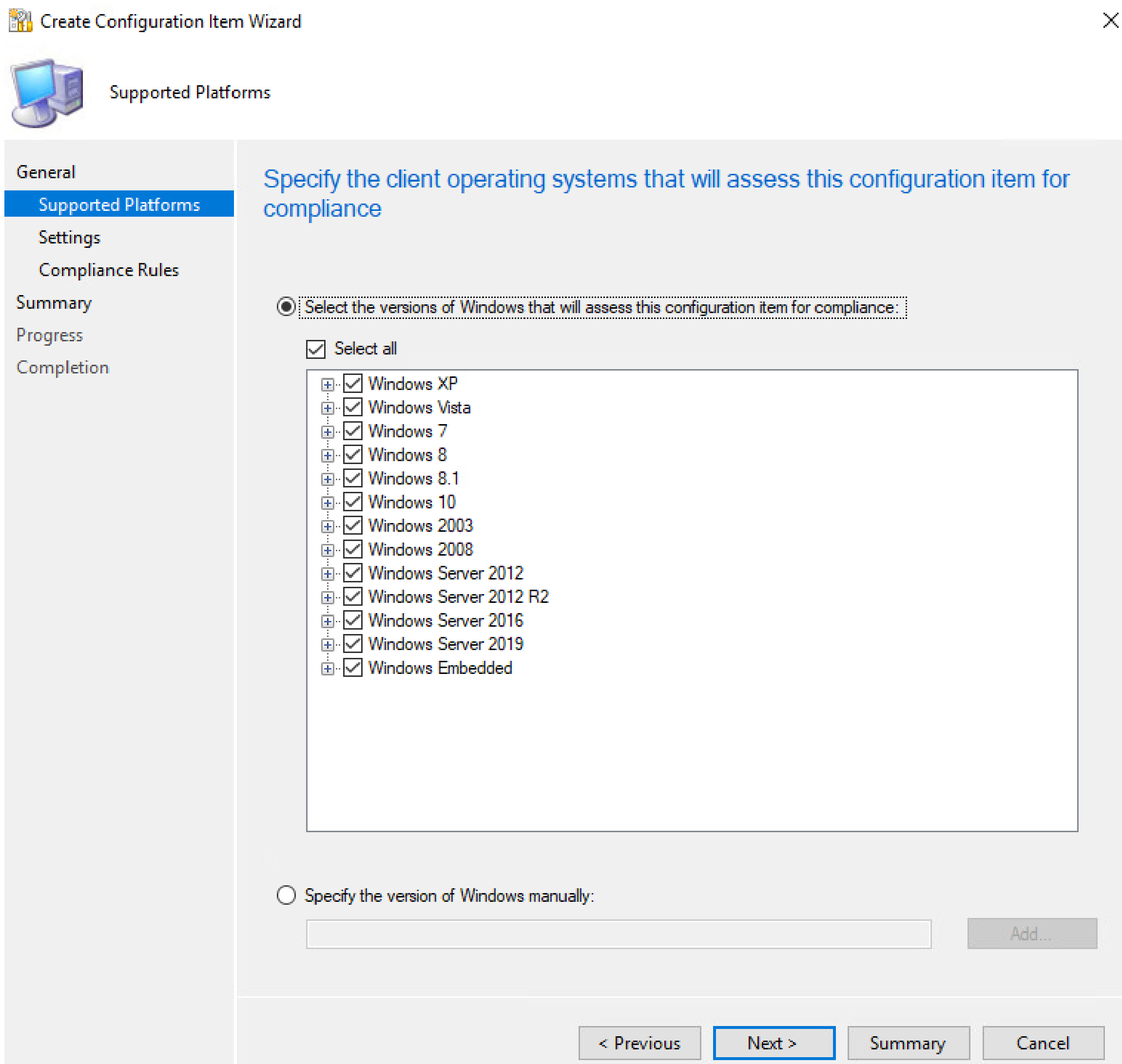
3. From the **Home** panel at the top, select **Create Configuration Item**.



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

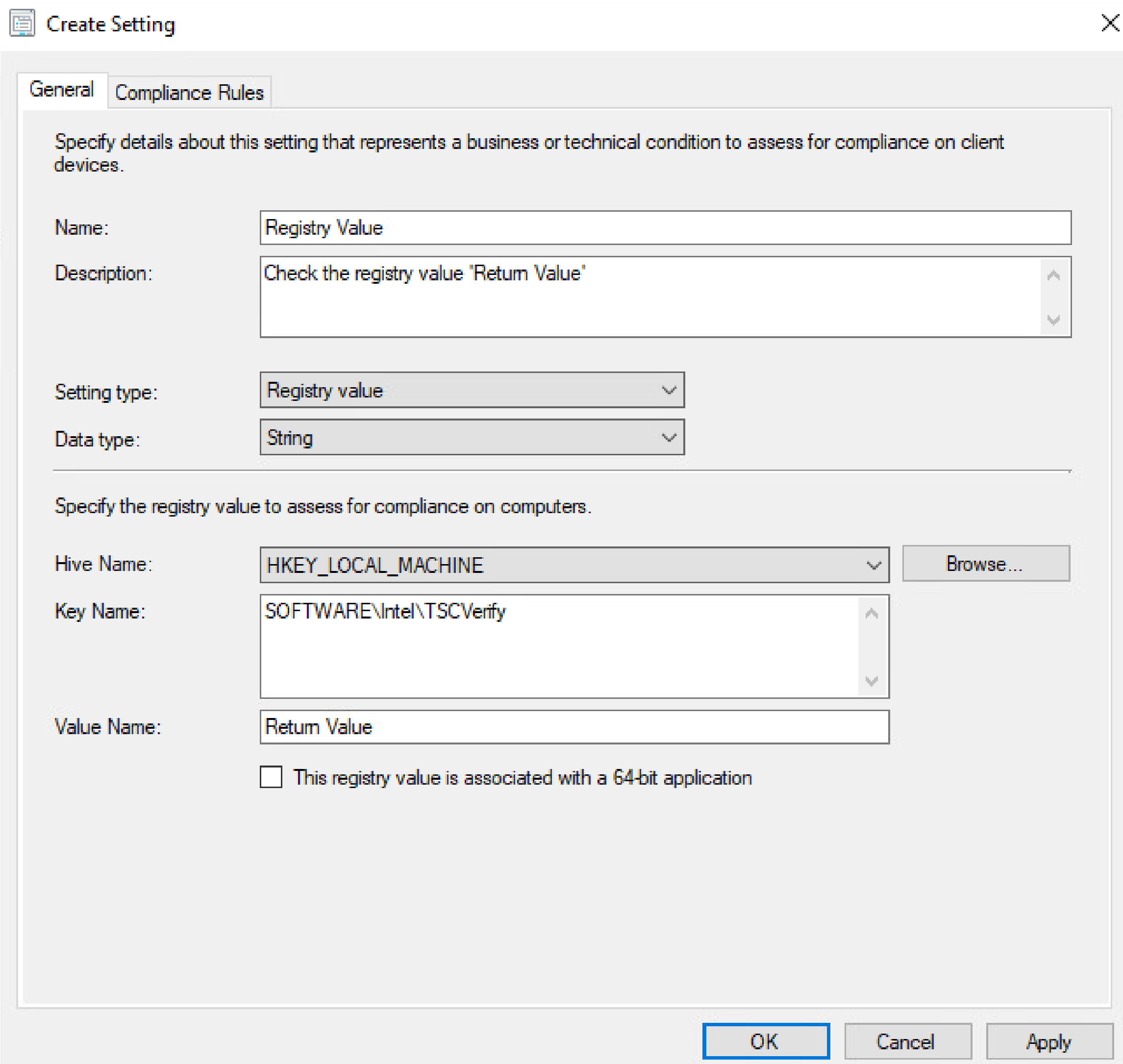


5. Ensure that all versions are selected and click **Next.**



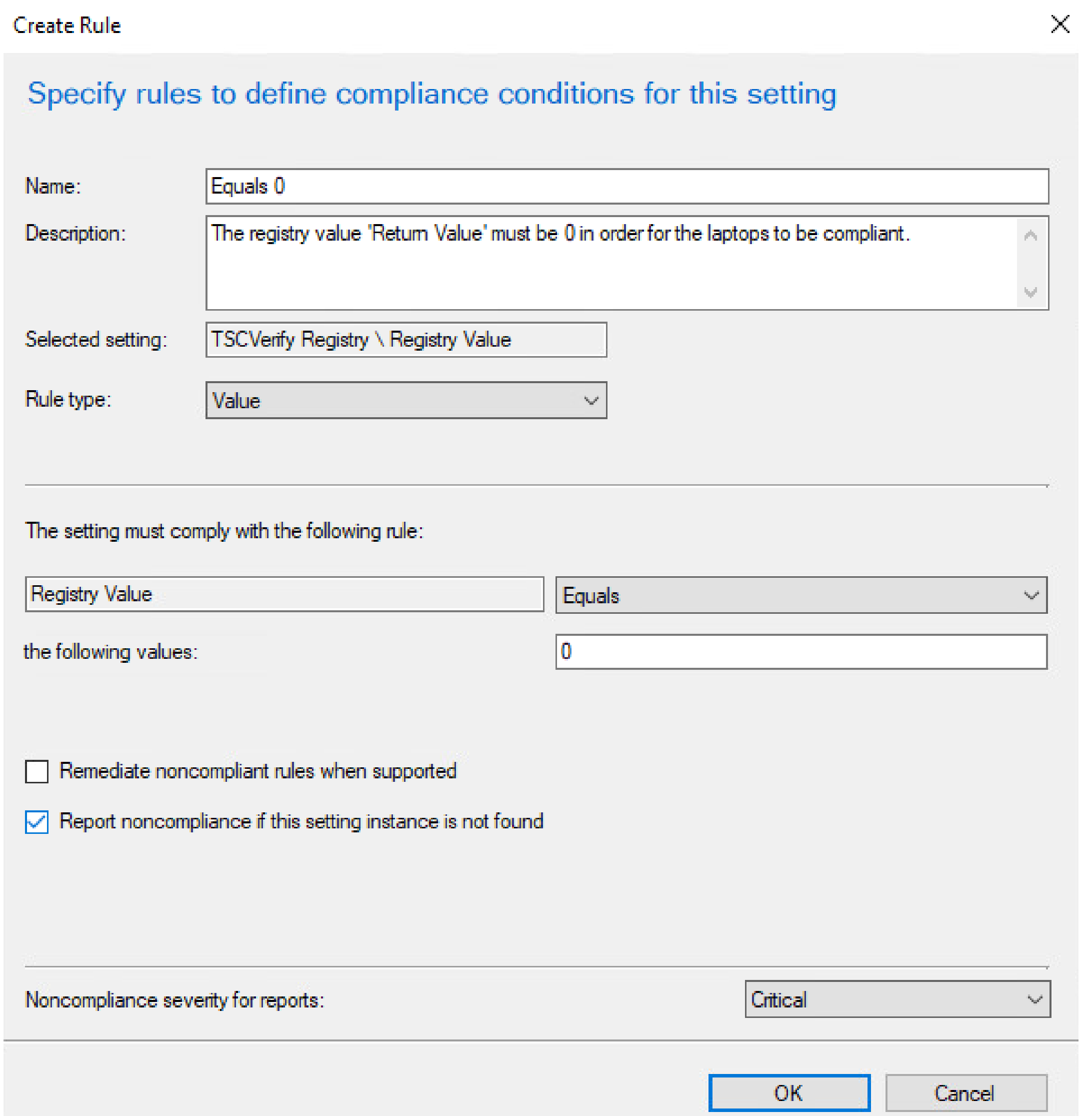
6. On the **Settings** tab, select **New**.

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

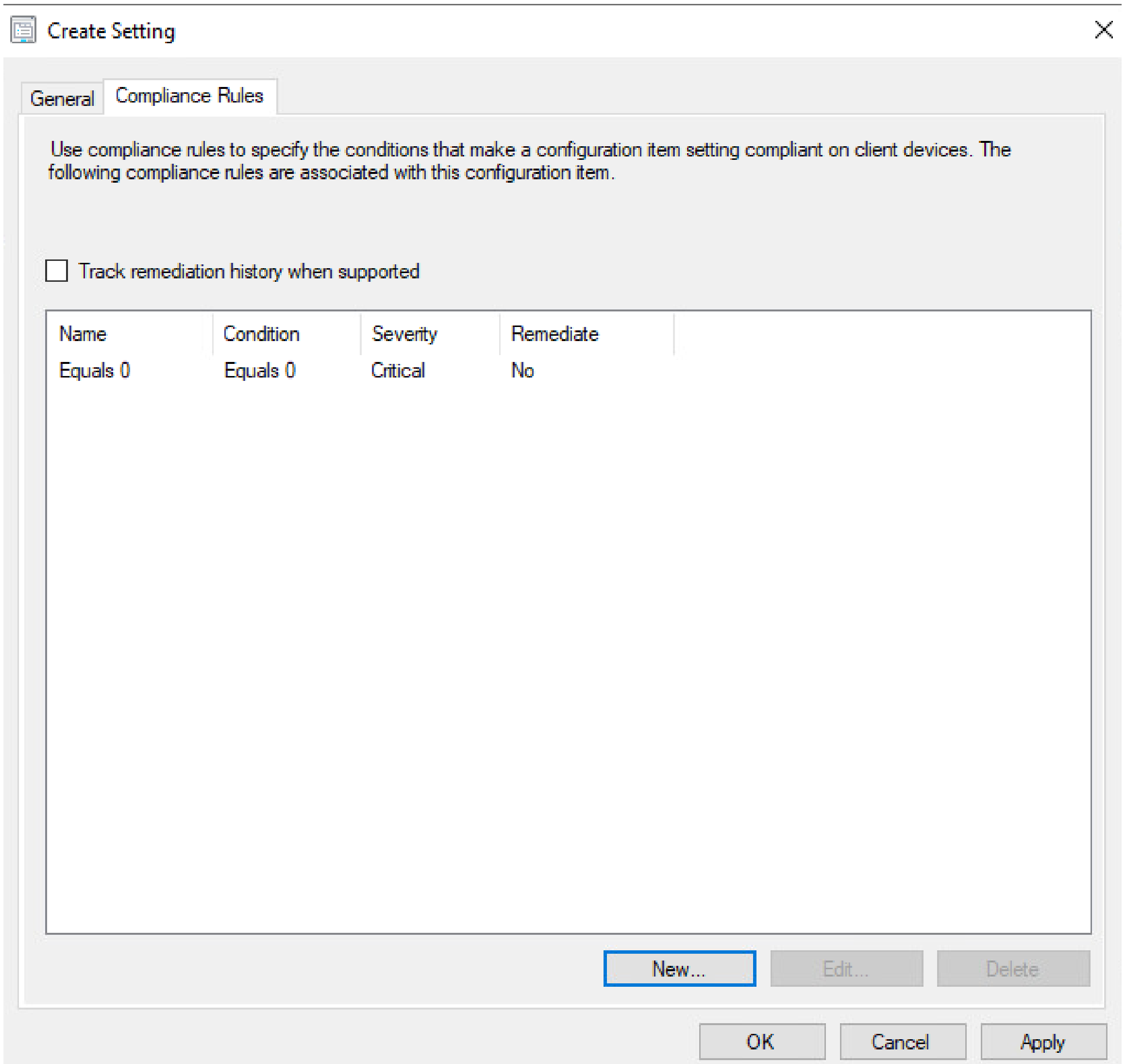


8. Select **New.**

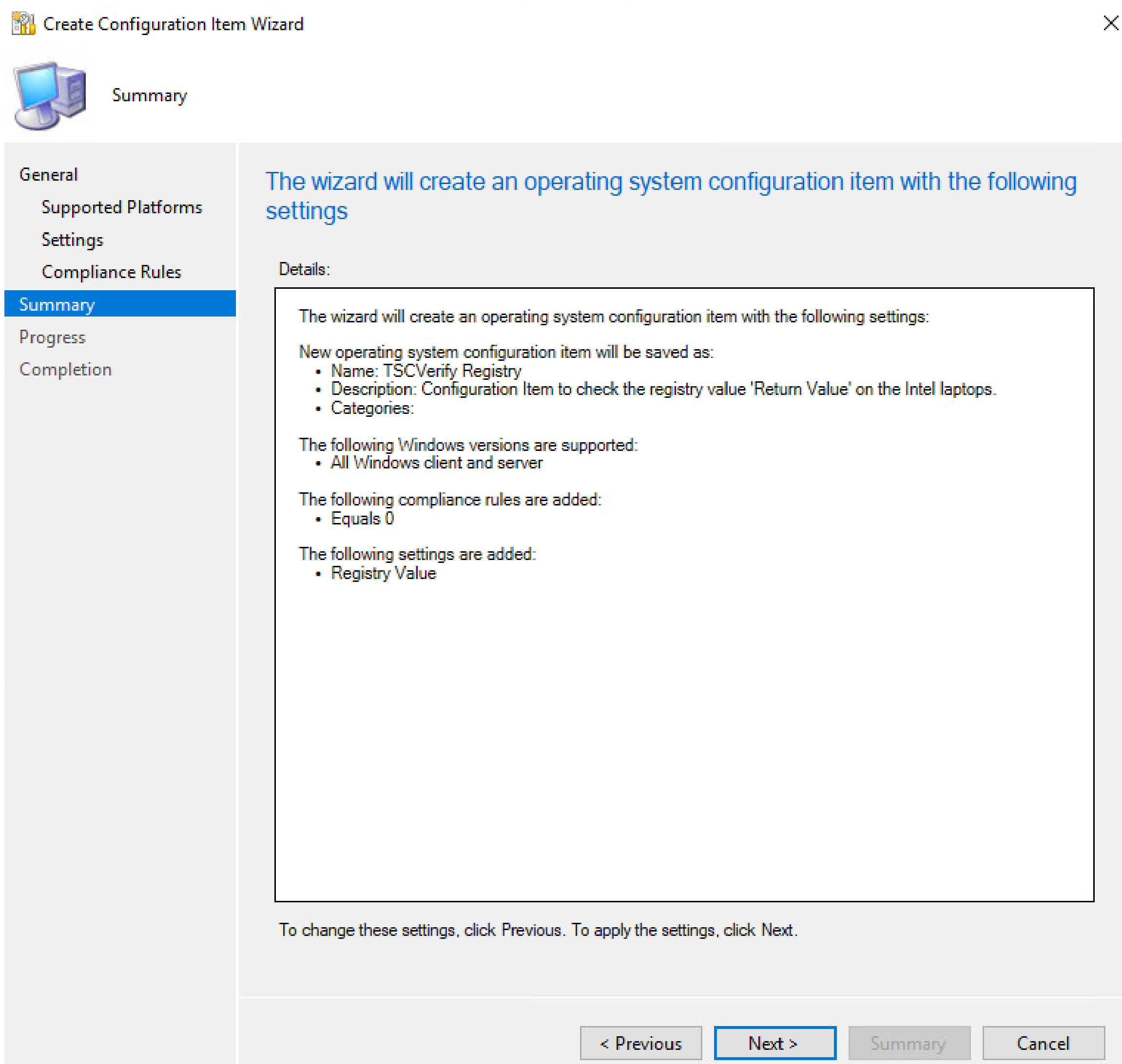
9. Specify the name and description for the rule in the **Name** and **Description** fields. For **Rule type**, select **Value** from the dropdown. Under **The setting must comply with the following rule**, select **Registry Value** and **Equals**, and enter 0 (zero) in **the following values:** field.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



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



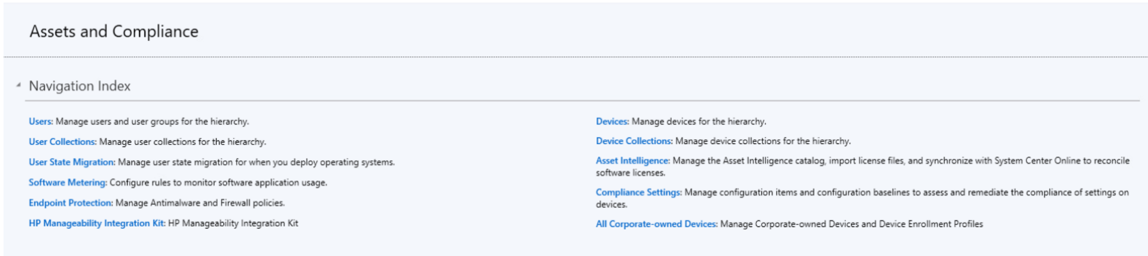
926 11. Review the configurations on the Summary page. After confirming that the configurations are
927 correct, select **Next**.



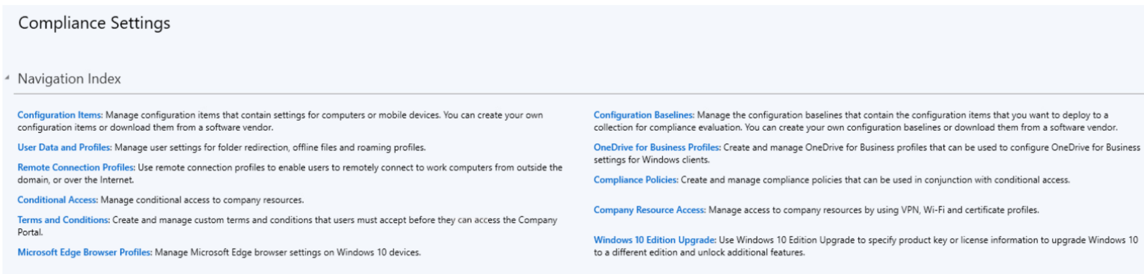
12. After the wizard completes, select **Close.**

2.11.1.2 Set Up Configuration Baseline

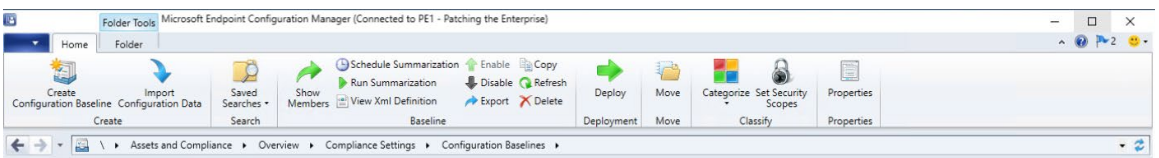
1. In the Microsoft Endpoint Configuration Manager console, under **Assets and Compliance > Overview**, select **Compliance Settings**.



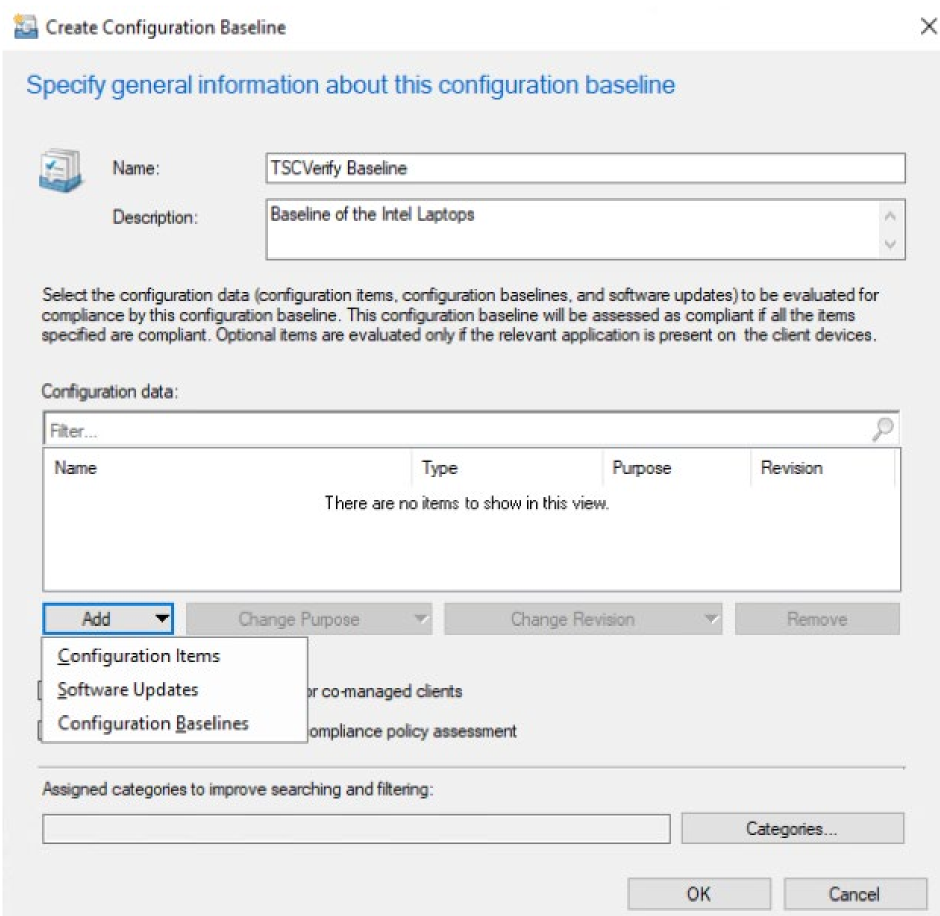
2. Next, select **Configuration Baselines**.



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

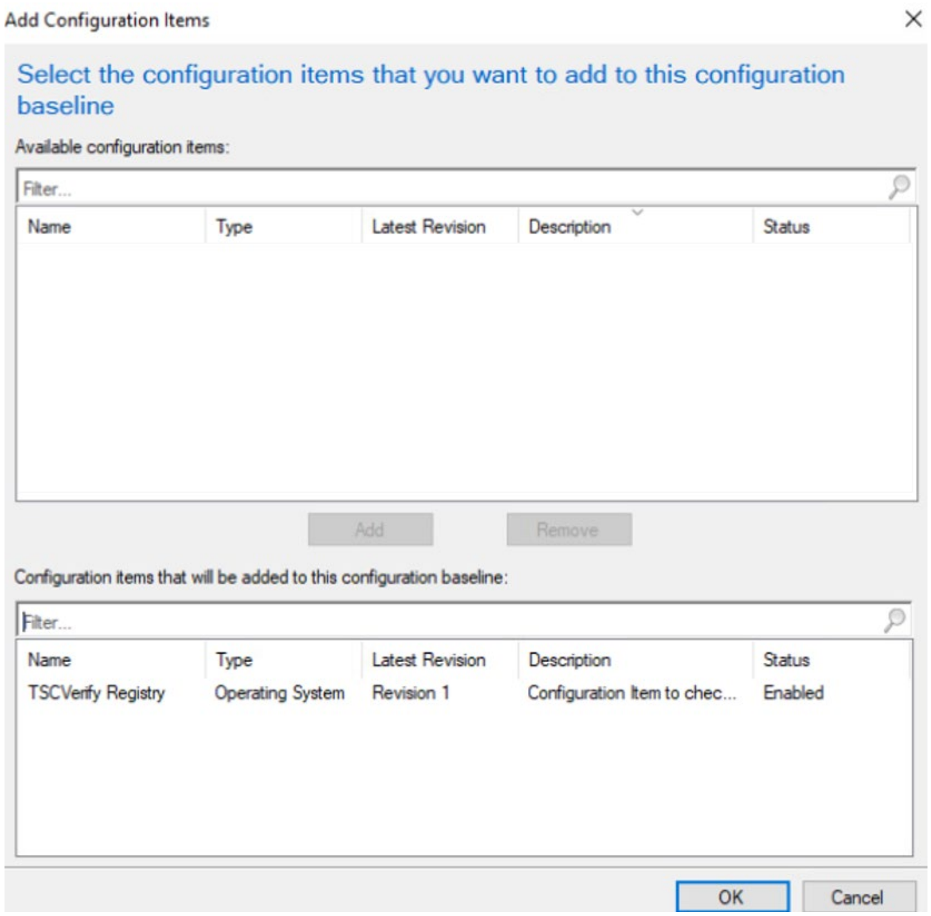


4. Provide a name and description for the configuration baseline in the **Name** and **Description** fields. Next, select **Add** and choose **Configuration Items**.

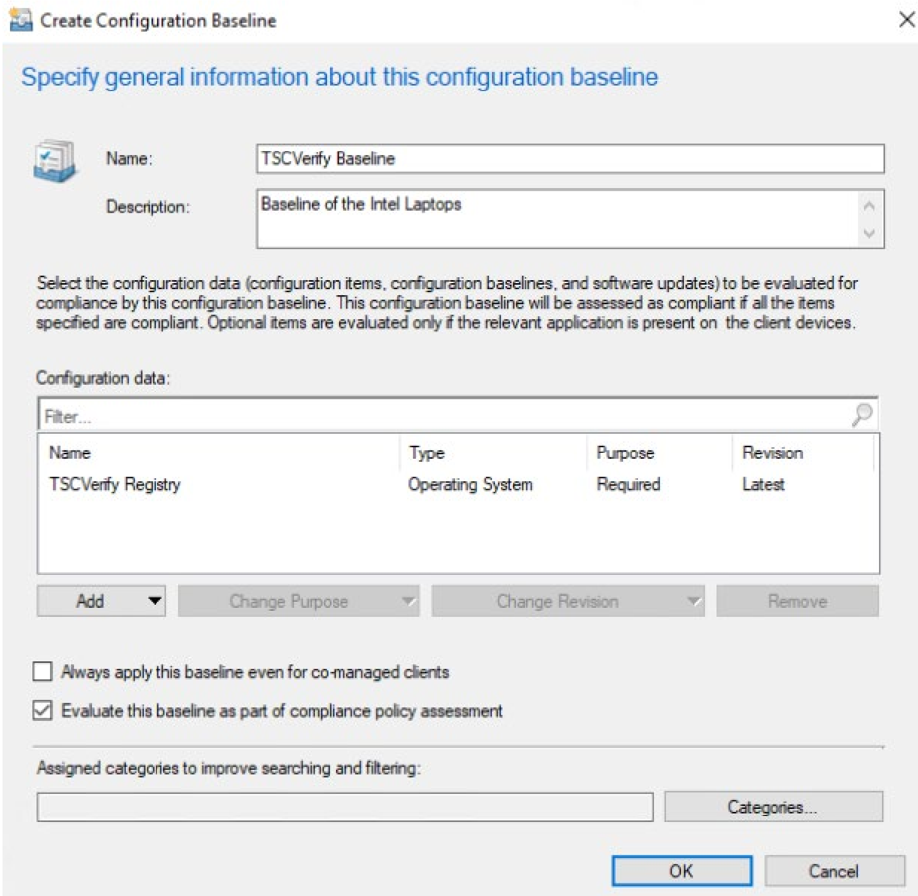


936 5. Select the previously created configuration item from the list and select **Add**.

937 6. Select **OK**.

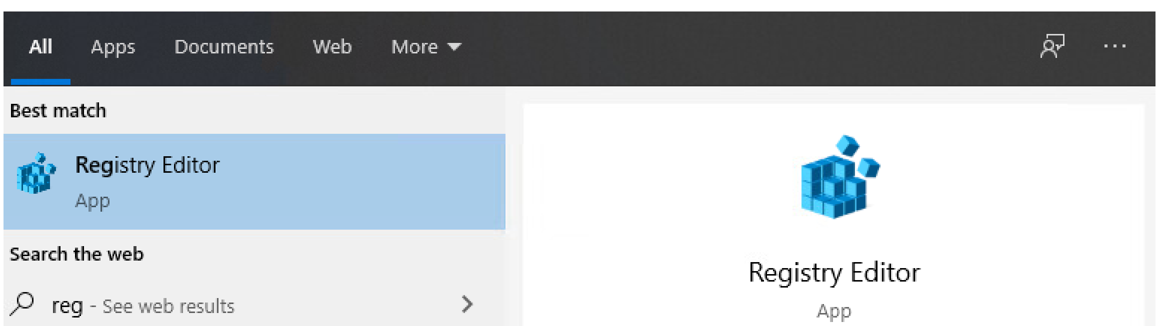


938 7. Select **OK**.

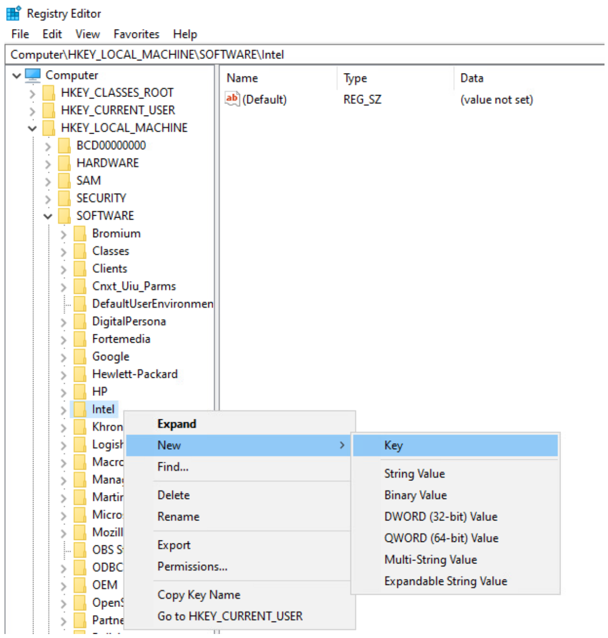


2.11.1.3 Set Up Registry Entry on Intel Devices

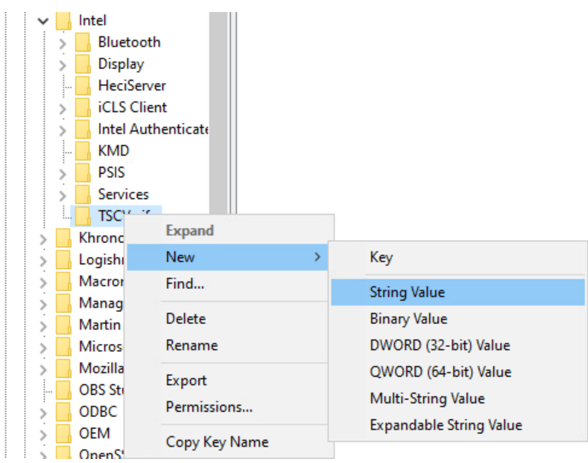
1. On the Windows 10 laptop, go to **Start**, search for the **Registry Editor,** and open that program.



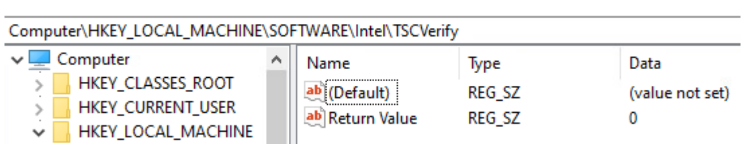
2. Find the Intel folder located in **HKEY_LOCAL_MACHINE\SOFTWARE**. Right click and select **New > Key**. Name the key **TSCVerify**.



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



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

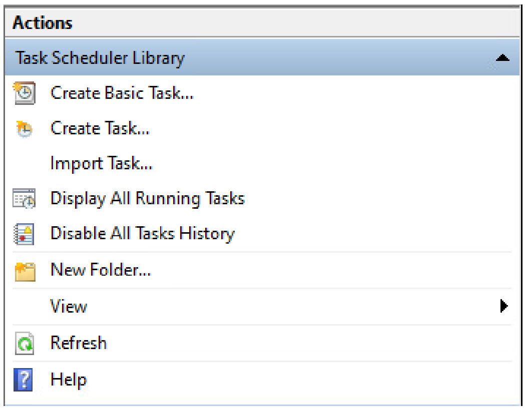


2.11.1.4 Run Script Via Task Manager

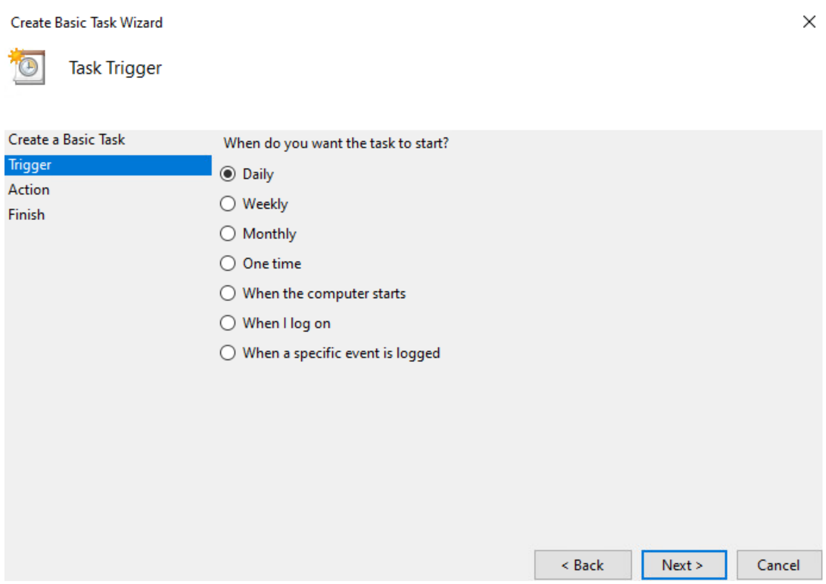
1. Place the script onto the local machine (snippet shown below). A copy of this script can be obtained from our repository.

```
# Run Scan and capture exit code.
# 0=No components have changed and platform certificate validation passed
# 1=At least one component has changed OR platform certificate validation
failed
# 2=At least one component has changed AND Platform Certificate validation
failed

# Write-Output "Starting DPD file scan and compare..."
$tscpinfo = New-Object System.Diagnostics.ProcessStartInfo
$tscpinfo.FileName = "TSCVerifyTool_3.40.exe"
$tscpinfo.WorkingDirectory = $artifactdirectory
$tscpinfo.RedirectStandardError = $true
$tscpinfo.RedirectStandardOutput = $true
$tscpinfo.UseShellExecute = $false
$tscpinfo.Arguments = "SCANREADCOMP -in $dpdfile"
$dpdprocess = New-Object System.Diagnostics.Process
$dpdprocess.StartInfo = $tscpinfo
$dpdprocess.Start() | Out-Null
$stdout = $dpdprocess.StandardOutput.ReadToEnd()
$dpdprocess.WaitForExit()

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

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

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

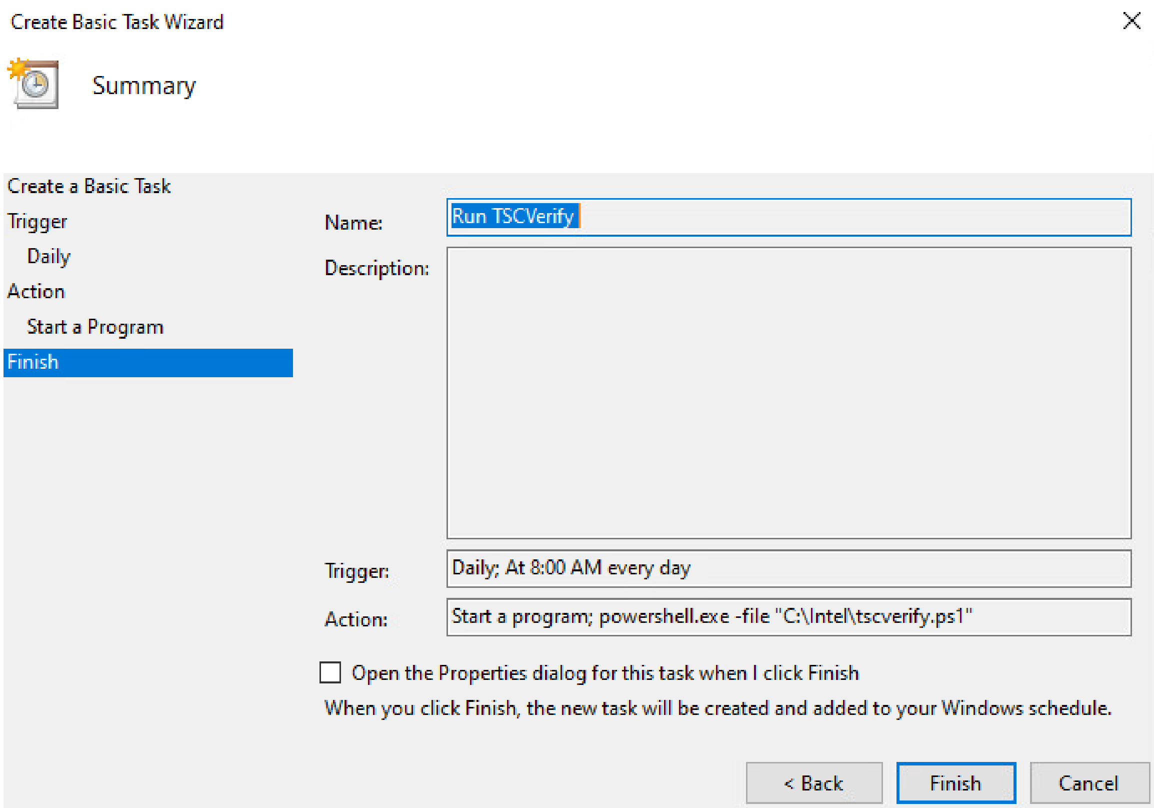
2. From the **Start Menu**, search for **Task Scheduler** and open the program.
3. Under the **Actions** panel, select **Create Basic Task**.



4. Fill in the **Name** and **Description** fields. Then select **Next**.
5. Select the frequency for this task to run. Then select **Next**.

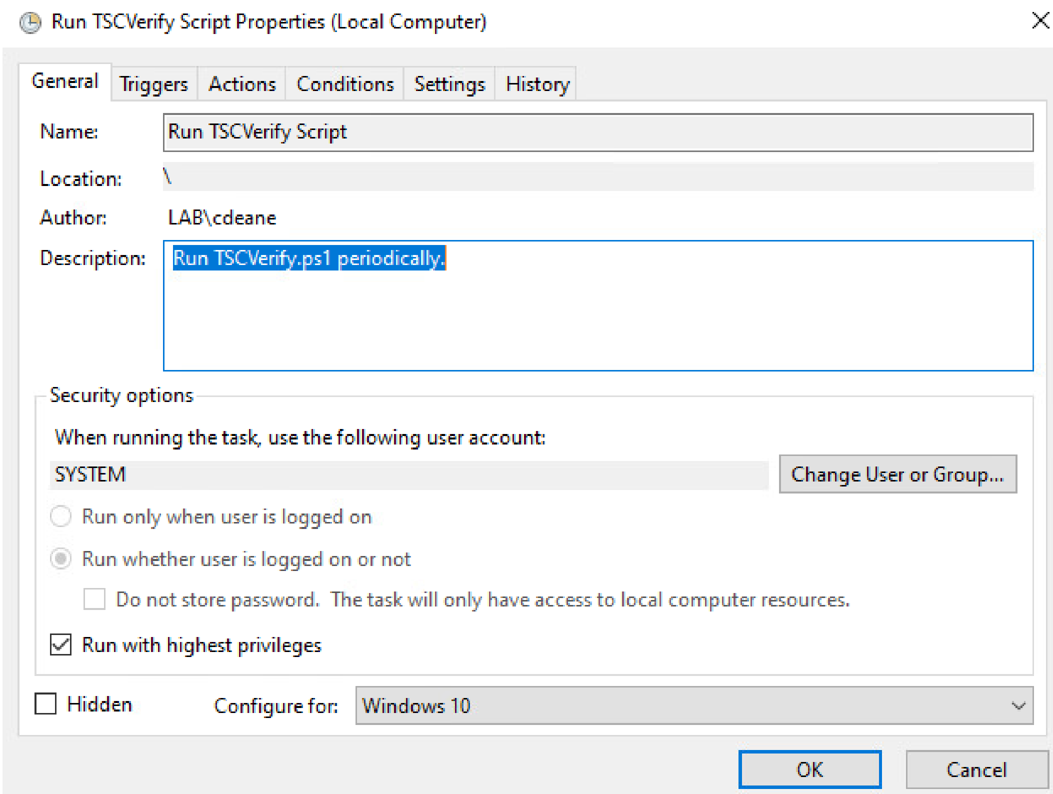


6. Select the start date and time for the task. Then select **Next**.
7. Select the action **Start a program**. Then select **Next**.
8. In the **Start a program** section, type the following in the **Program/script** field: *powershell.exe*. Next, add the following to the add arguments (optional) field: *-file "<Location of script>"*. Then select **Next**.
9. Confirm the settings are correct and select **Finish**.

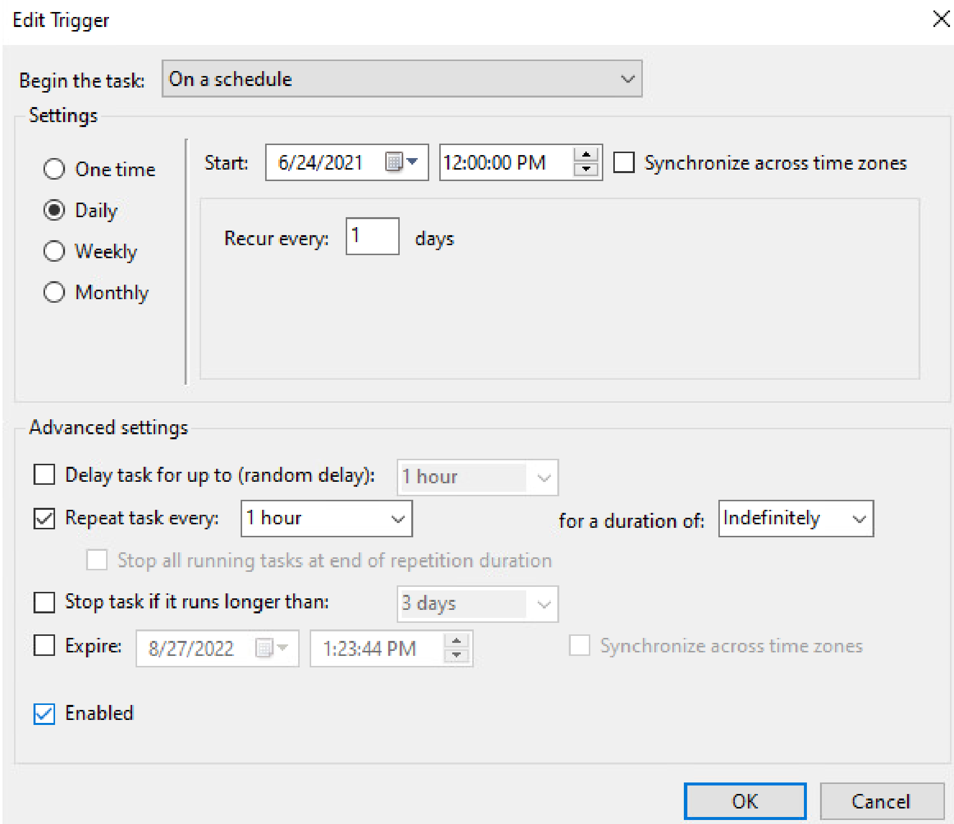


10. On the main page of Task Scheduler, select the newly created task, right-click it, and select **Properties**.

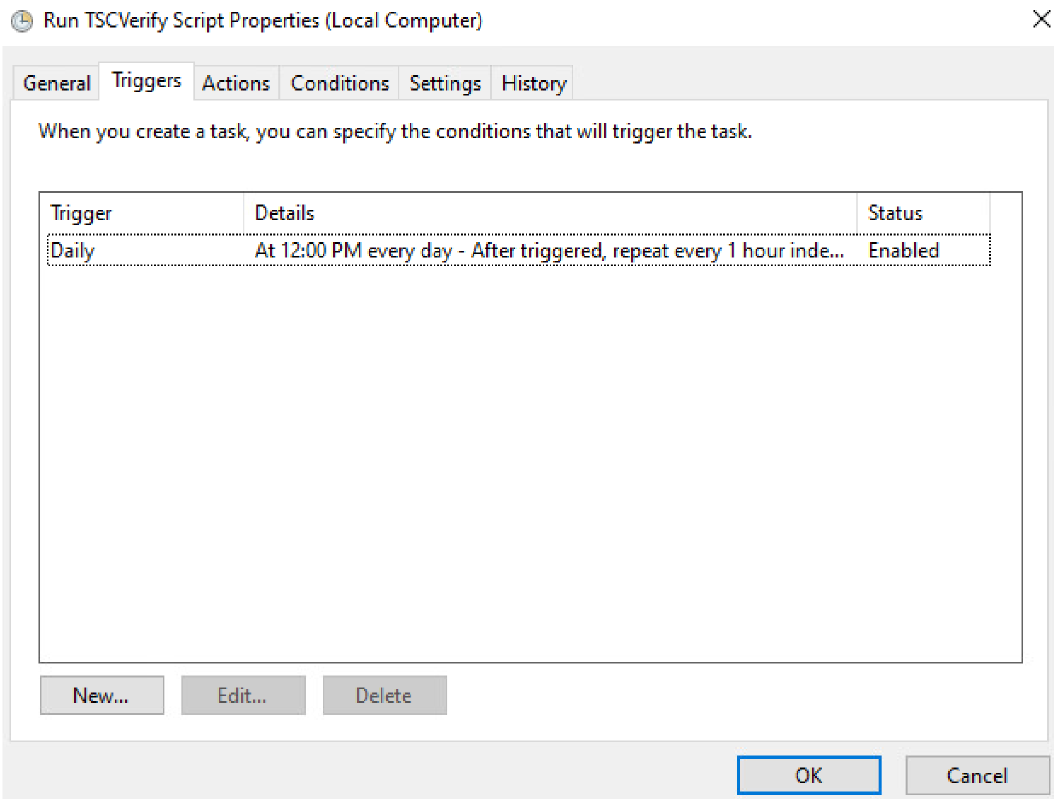
11. On the **General** tab, under **Security Options**, change the user to **SYSTEM**. Next, ensure that the option **Run with highest privileges** is checked.



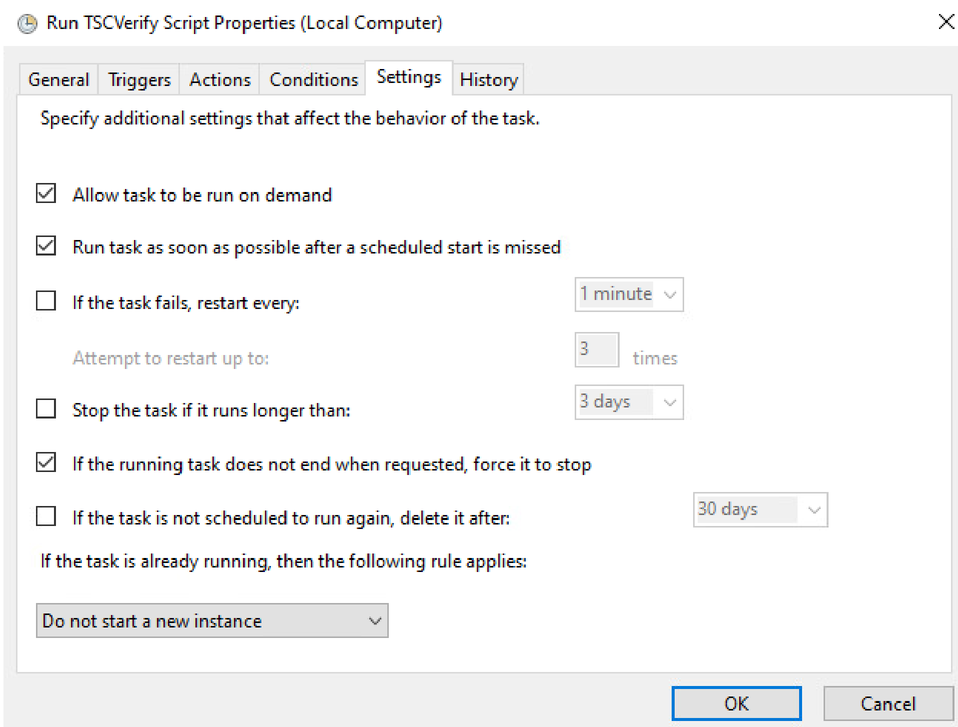
12. Navigate to the **Triggers** tab. Select the existing trigger and select **Edit**.
13. Under the **Advanced Settings** section, ensure that **Repeat task every 1 hour for a duration of Indefinitely** is checked, as well as **Enabled**. Select **OK**.


Edit Trigger
Begin the task:
On a schedule
Settings
One time
Daily
Weekly
Monthly
Start:
6/24/2021
12:00:00 PM
Synchronize across time zones
Recur every:
1
days
Advanced settings
Delay task for up to (random delay):
1 hour
Repeat task every:
1 hour
for a duration of:
Indefinitely
Stop all running tasks at end of repetition duration
Stop task if it runs longer than:
3 days
Expire:
8/27/2022
1:23:44 PM
Synchronize across time zones
Enabled
OK
Cancel

14. Select **OK**.



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



2.11.2 Archer IRM DataFeed Integrations

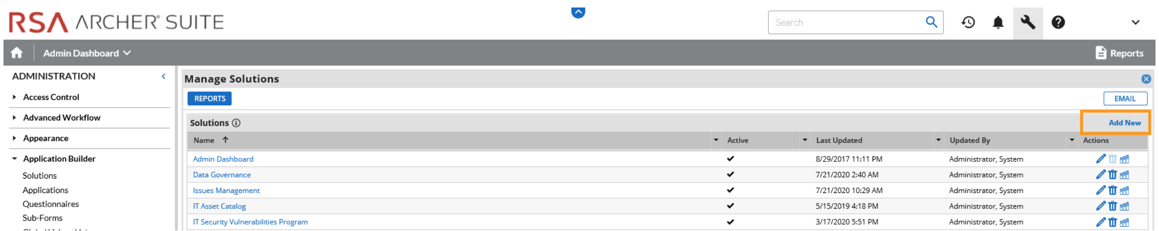
Archer IRM serves a dual role in the prototype demonstration - the Asset Management and Discovery System and the IT Administrator Dashboard. This section will detail the steps necessary to integrate Archer IRM with the PMCS, the Eclypsium Firmware Analytics Platform, and Microsoft Configuration Manager, 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

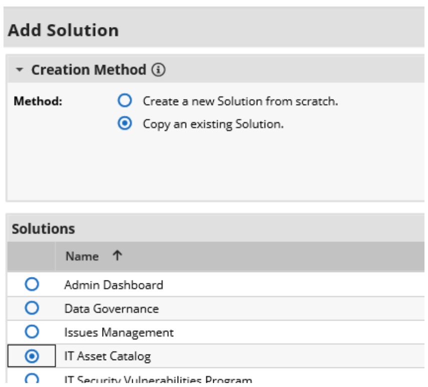
Before platform and firmware data can be stored in the in the Asset Management and Discovery System, the Archer IRM application must be created. For this task, we leverage the default *Devices* application described as *the central repository of knowledge about your business-critical devices.*

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.

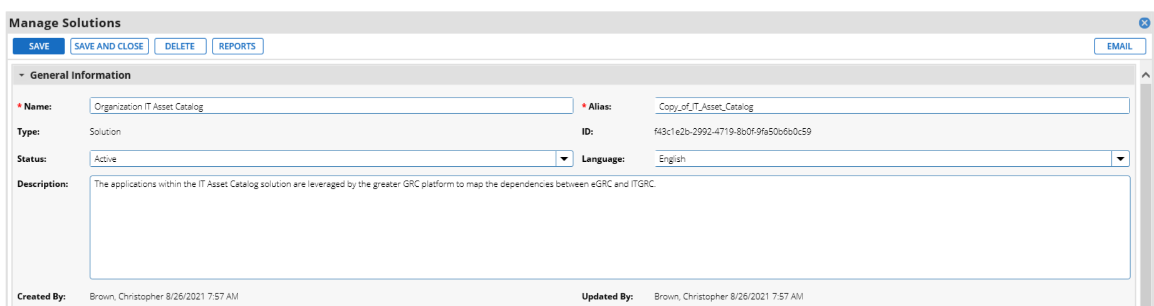
1. In the administration menu, navigate to **Application Builder** > **Solutions**. Select **Add New**.



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



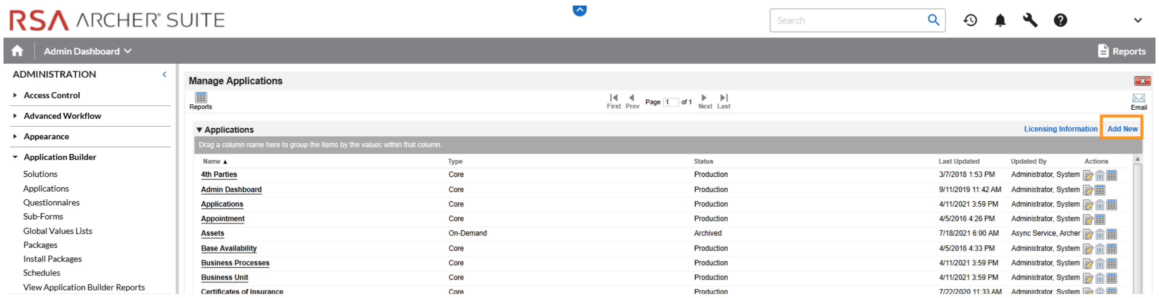
3. Enter an identifier for the catalog in the **Name** field. Click **SAVE AND CLOSE**.



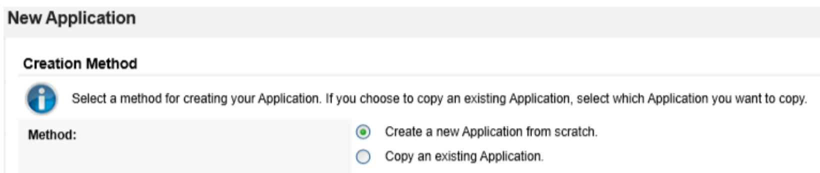
2.11.2.1.1 Create Supporting Applications

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

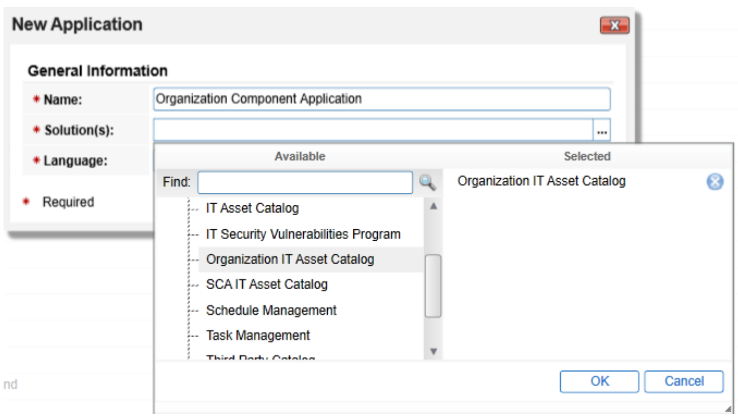
1. In the administration menu, navigate to **Application Builder > Applications.** Select **Add New.**



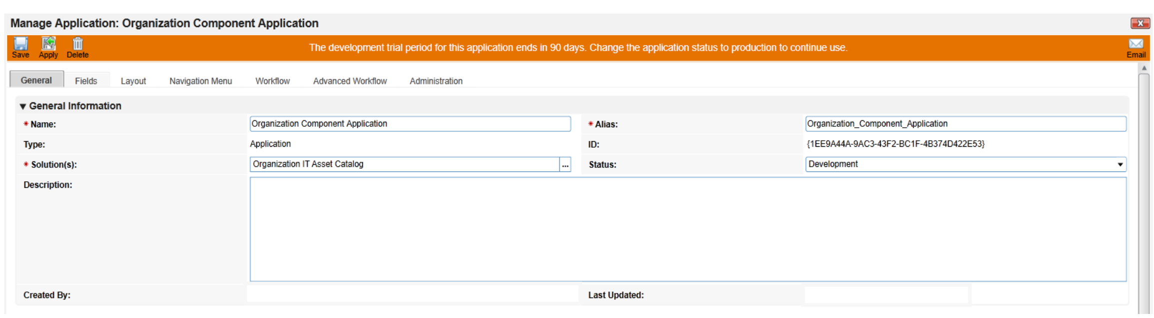
2. Select Create a new **Application from scratch** and click **OK.**



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



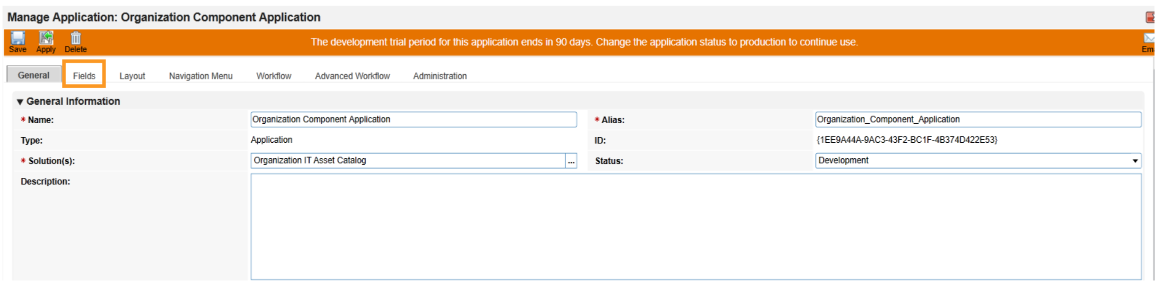
4. Click **Save.**



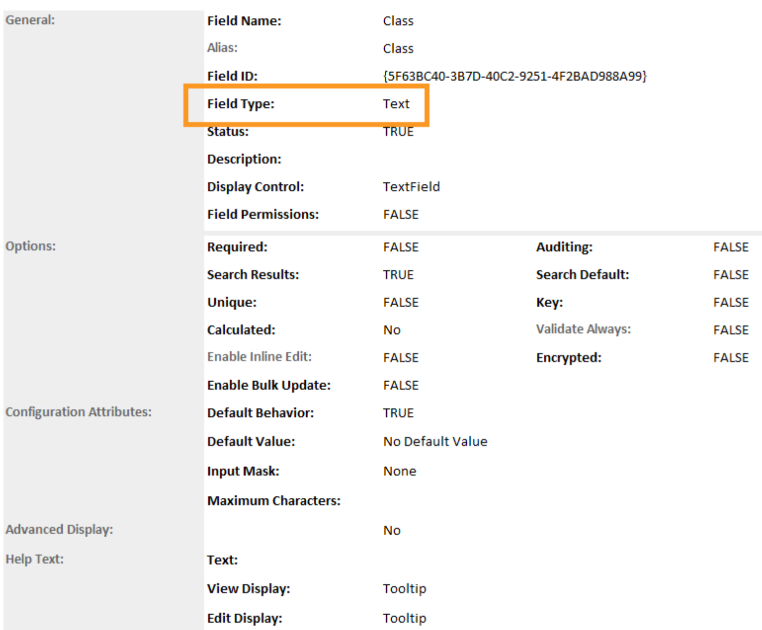
In the next series of steps, we will add several Data Fields to the newly created application. These are like table columns you might define in a relational database. Note that we will only walk through one example, but the steps can be repeated for the remaining data fields. Before starting these steps, download and open the Components application schema from our repository. Some data fields, such as **Tracking ID**, **First Published**, and **Last Updated** are automatically created with each new application and do not need to be repeated.

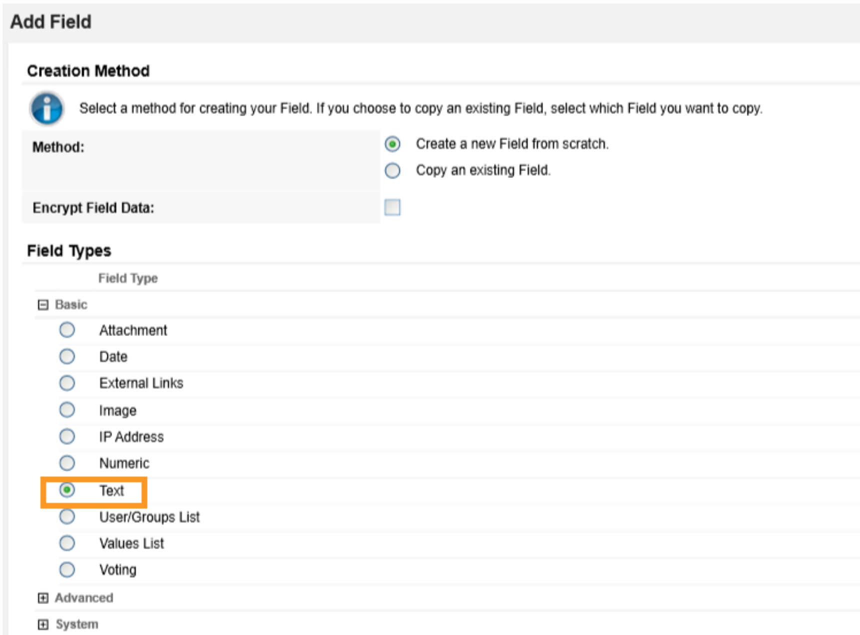
5. Open the target Components application from the Administration menu under **Application Builder > Applications.**

6. Click the **Fields** tab.

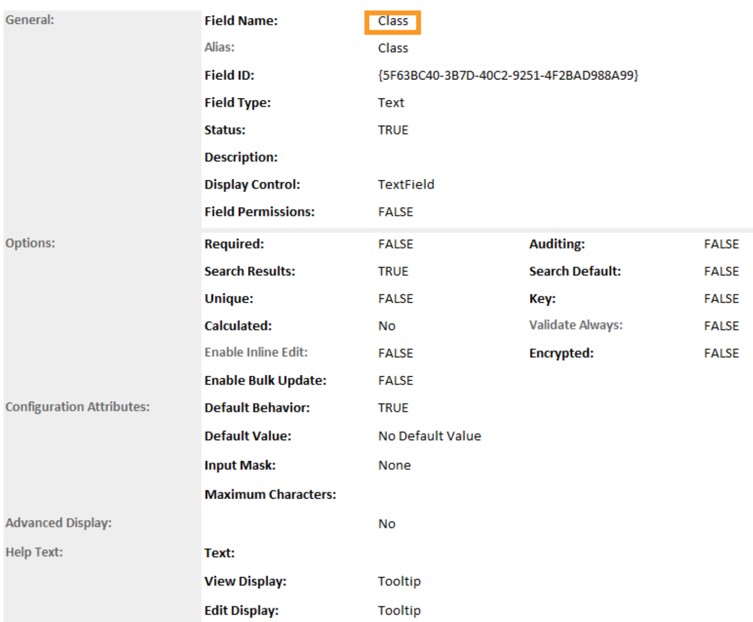


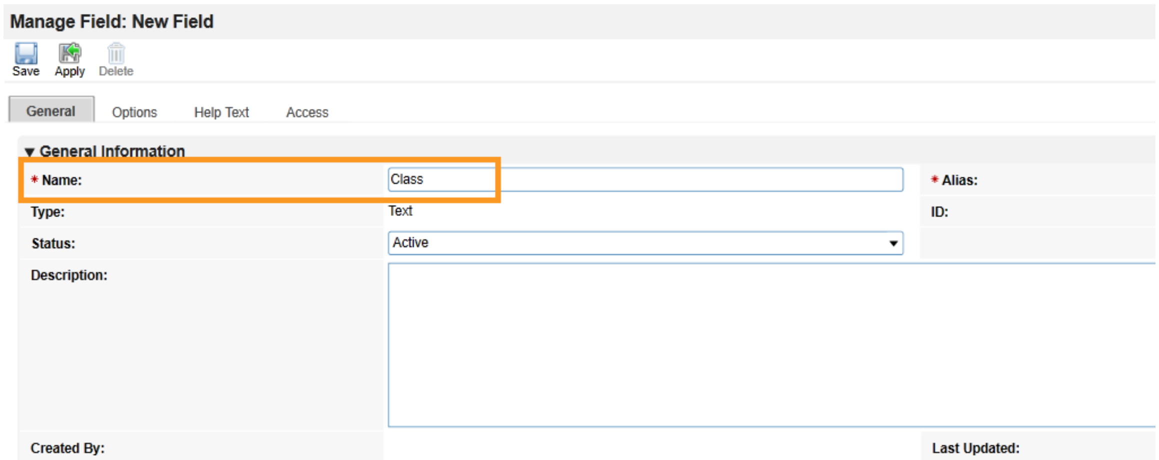
7. Click **Add New.** Match the Field Type from Appendix B to the **Field Type** field in Archer IRM. Click **OK.**





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





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

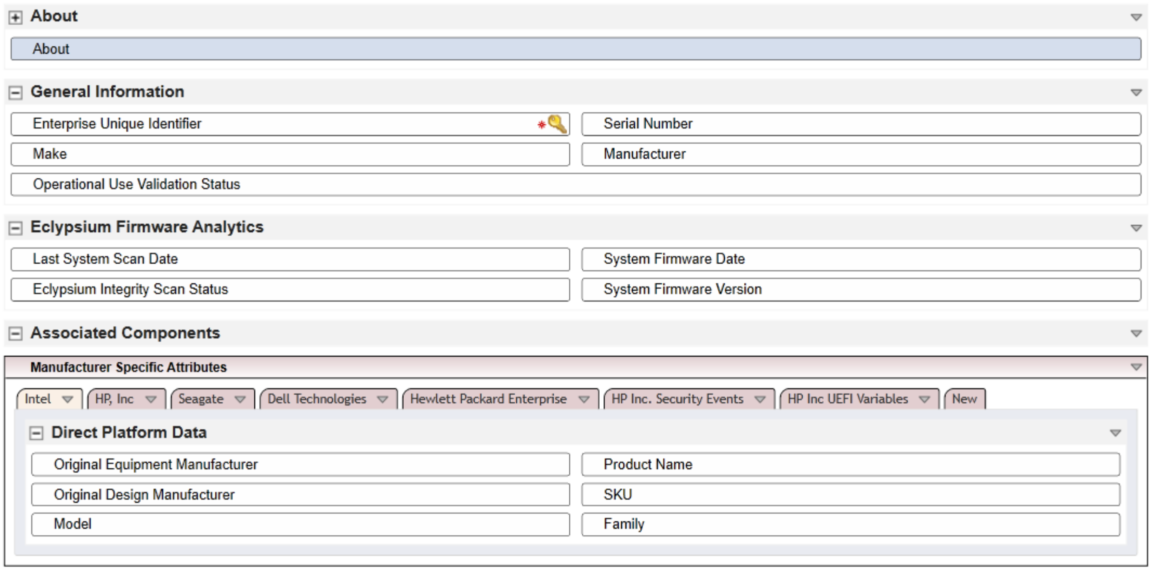
At this point, you have created the first supporting application for the Asset Discovery and Inventory system. Repeat these procedures to create the *HP UEFI Configuration Variables, Seagate Firmware Attestation, and Seagate Firmware Hash* applications. These applications support the demonstration's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

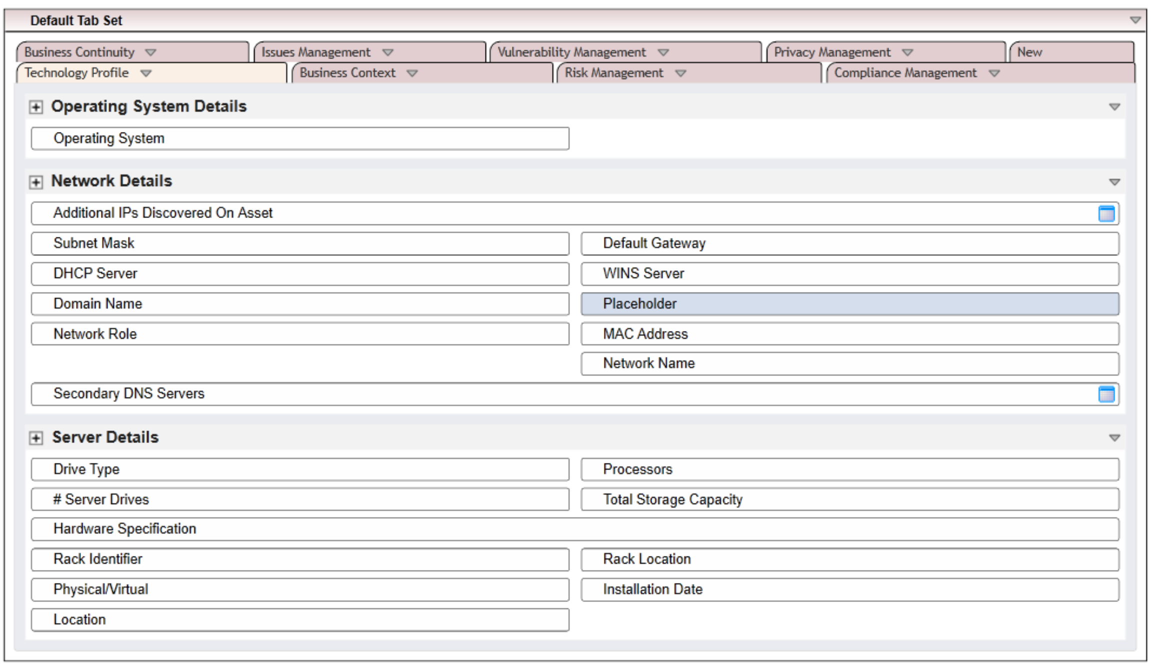
2.11.2.1.2 Modify Default *Devices* Application

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

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

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





2.11.2.1.3 Modify Default Security Incidents Application

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

Table 2-7 Security Incidents Application Custom Data Fields

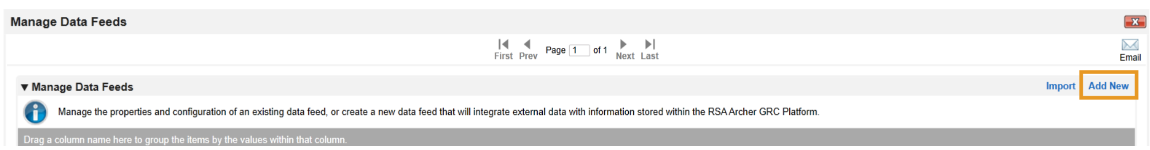
Data Field Name	Data Field Type	Notes
Date/Time QRadar LastUpdate	Date	Stores the date from each *QRadar Offense*
Incident ID (QRadar)	Text	Stores the *QRadar Offense* unique identifier
SCA Computing Device	Cross-Reference	Links to the *Devices* application computing device unique identifier

2.11.2.2 Create Data Feed Integrations

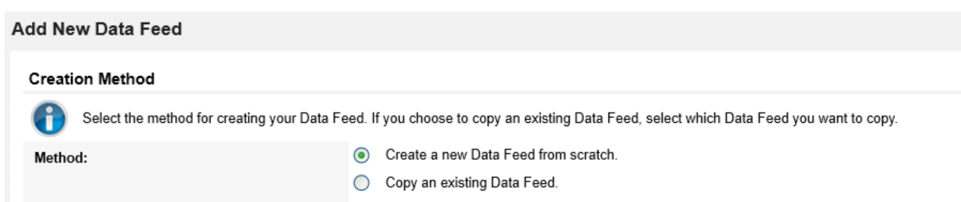
In this section, the implementer will create *data feeds* in Archer IRM that will complete the integration with the PMCS, Microsoft Configuration Manager, IBM QRadar, and Eclypsium. The data feeds will periodically pull data from the three data sources and map it to the *Devices* application created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

2.11.2.2.1 Create Eclypsium Data Feeds

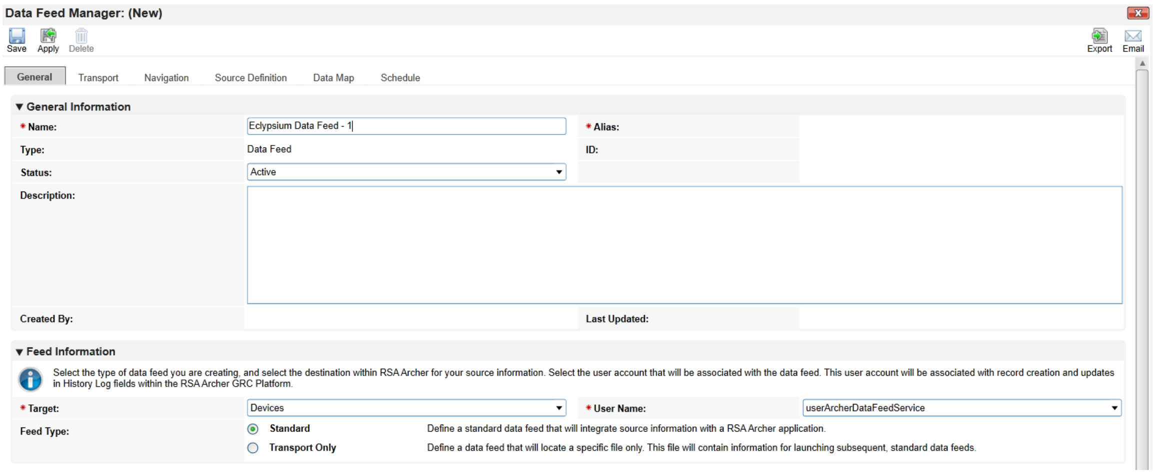
1. In the Administration menu, navigate to **Integration > Data Feeds**. Click **Add New**.



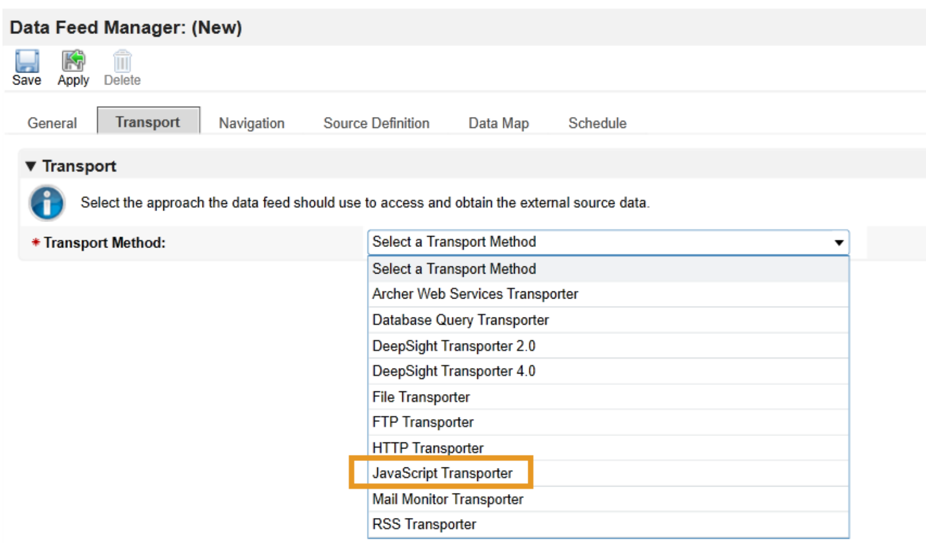
2. Select **Create a new Data Feed from scratch**. Click **OK**.



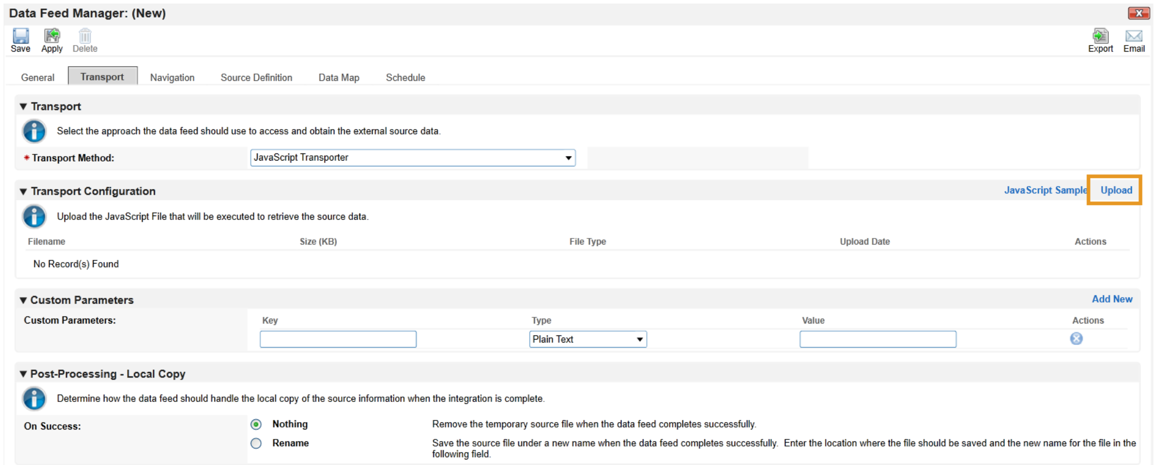
3. Create an identifier in the **Name** field. Select the **Devices** application created in Section 2.11.2.1 in the **Target** field.



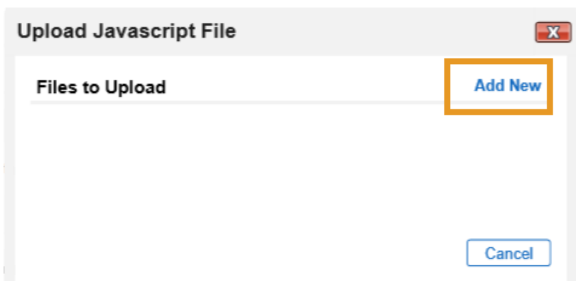
4. Click the **Transport** tab. Select **JavaScript Transporter**.



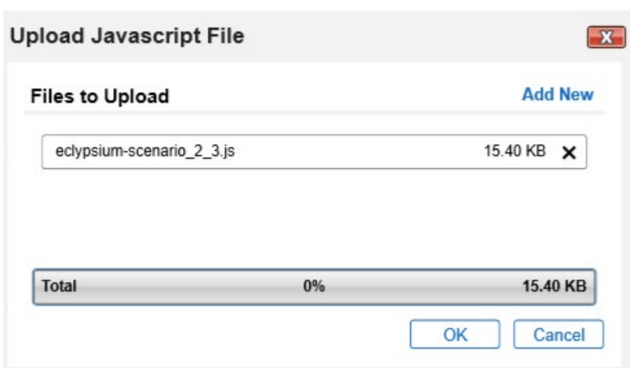
5. Click **Upload** in the **Transport Configuration** section.



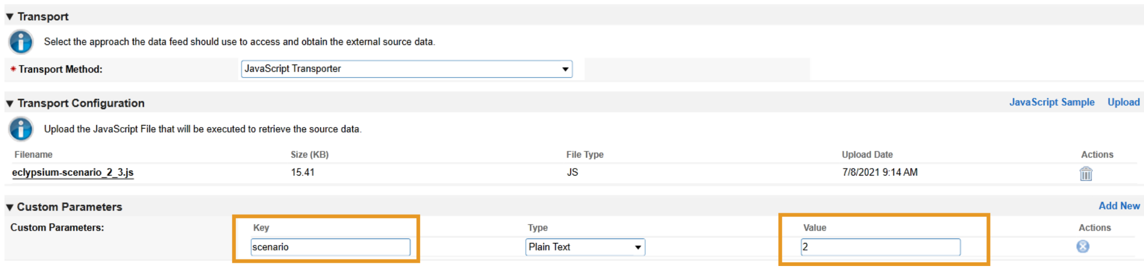
6. Click **Add New**.



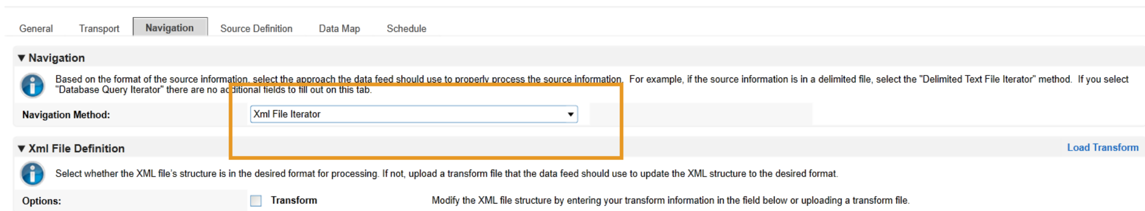
7. In the file selection modal, select the Eclypsium JavaScript data feed file from the repository. Click **OK**.



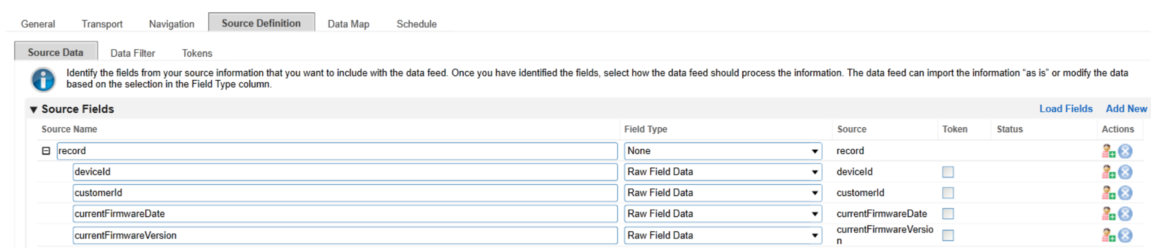
8. Enter “scenario” in the **Key** field and “2” in the **Value** field.



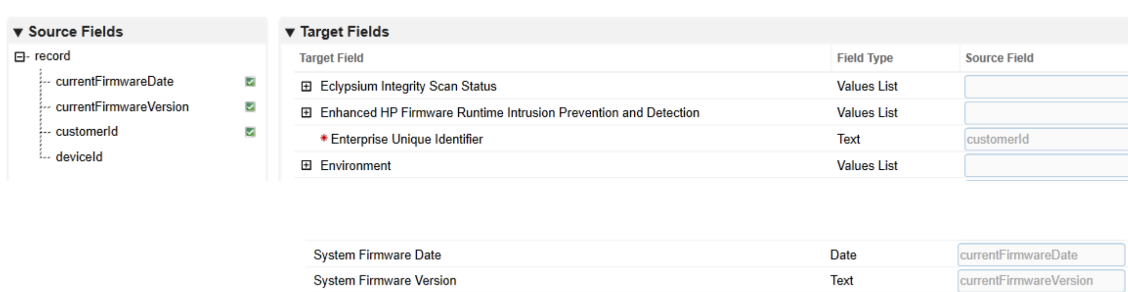
9. Click the **Navigation** tab. Ensure **XML File Iterator** is selected in the **Navigation Method** dropdown menu.



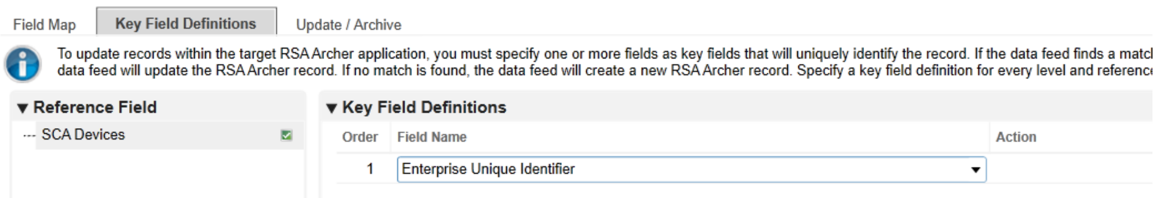
10. Click the **Source Definition** tab. In the **Source Data** sub-tab, select **Load Fields**. Select the Eclypsium example XML file. The configuration in Archer should populate the **Source Fields** as follows.



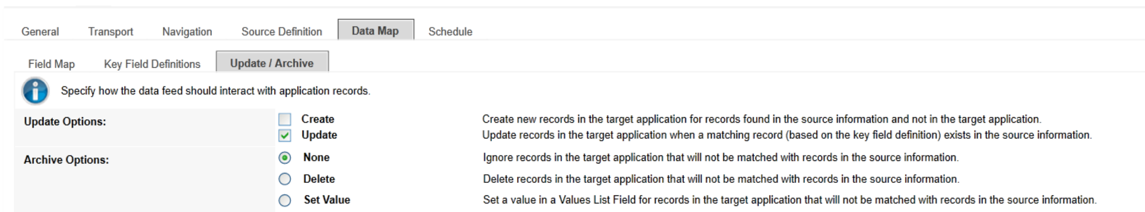
11. Click the **Data Map** and tab which will default to the **Field Map** sub-tab. Drag and drop the source fields onto the application data fields. Due to the large amount of data fields in the Devices application, below we present a truncated view of the mapping.



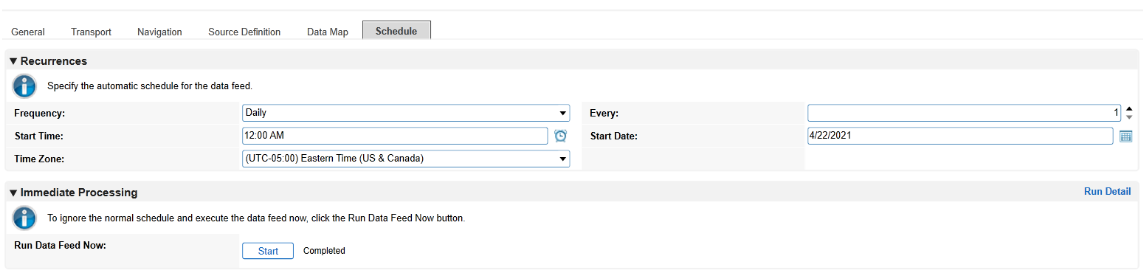
12. Click the **Key Field Definitions** tab. Select **Enterprise Unique Identifier** in the Field Name column.



13. Click the **Update / Archive** tab. Ensure only the **Update** option is selected. Choose **None** for the **Archive Options**.



14. Click the **Schedule** tab. Select a cadence appropriate for your organization. In this example, we've chosen to run the data feed on a daily frequency at 12:00AM.



At this point, the data feed for Eclypsium (Scenario 2) is configured. Scenario 3 is configured with the same process, except a "3" is used in the Value field in Step 8. Click the **Start** button to confirm that the data feed has been properly configured. Archer IRM will report any errors that are useful for debugging.

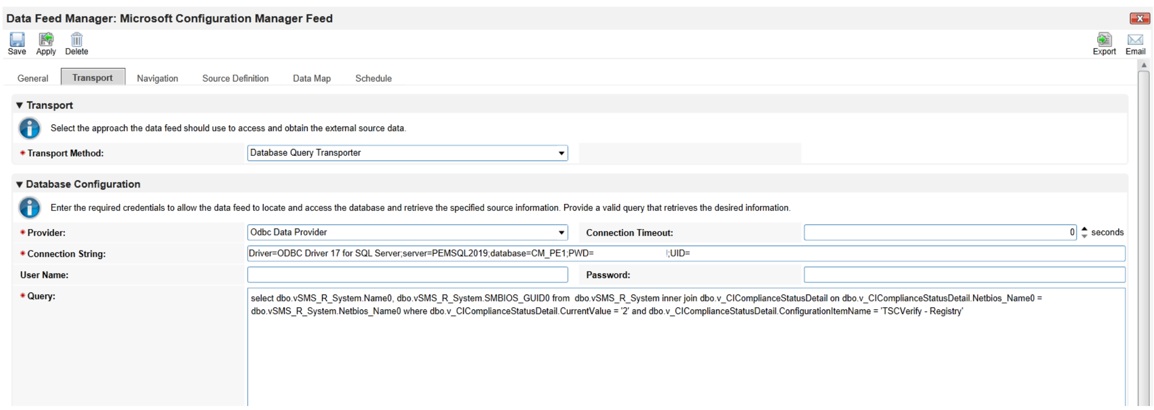
2.11.2.2.2 Create Microsoft Configuration Manager Data Feed

Repeat the preceding steps to add the Microsoft Configuration Manager Data Feed with the following modifications:

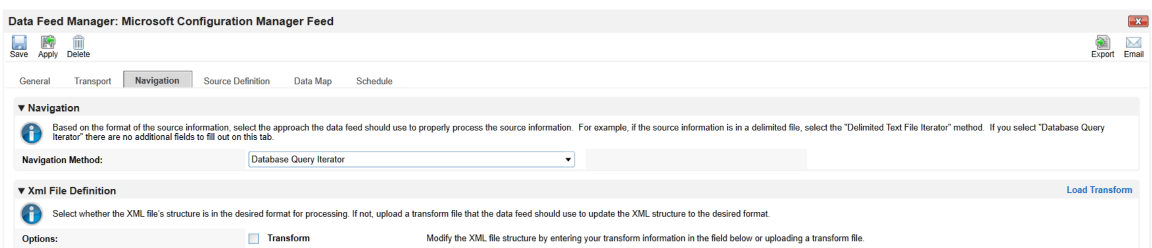
15. In the **Transport** tab, select **Database Query Transporter**. Insert the following values in the form:

Provider	`Odbc Data Provider`

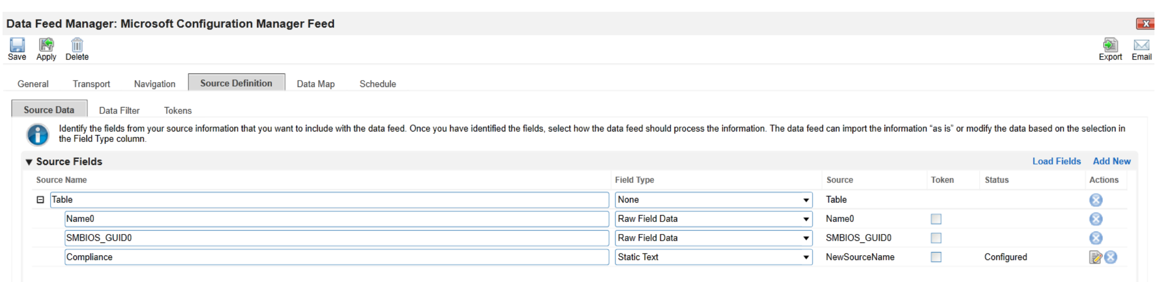
Connection String	`Driver=ODBC Driver 17 for SQL Server;server=PEMSQL2019;database=CM_PE1;PWD=[SQL USER PASSWORD];UID=[SQL USER]`
Query	`select dbo.vSMS_R_System.Name0, dbo.vSMS_R_System.SMBIOS_GUID0 from dbo.vSMS_R_System inner join dbo.v_CIComplianceStatusDetail on dbo.v_CIComplianceStatusDetail.Netbios_Name0 = dbo.vSMS_R_System.Netbios_Name0 where dbo.v_CIComplianceStatusDetail.CurrentValue = '2' and dbo.v_CIComplianceStatusDetail.ConfigurationItemName = 'TSCVerify - Registry'`



16. In the **Navigation** tab, select **Database Query Iterator**.

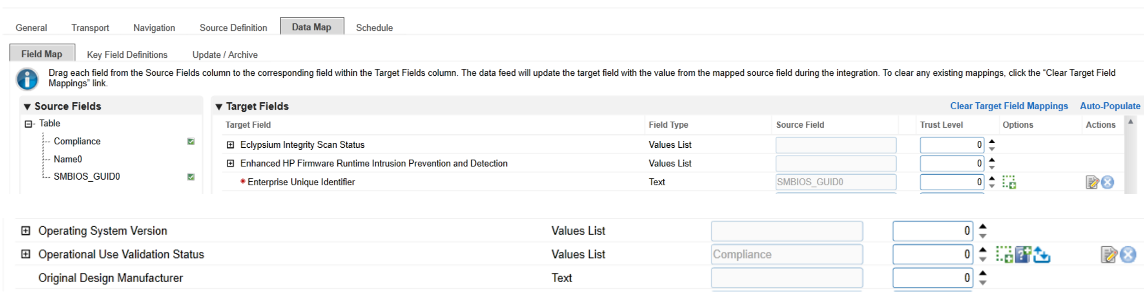


17. In the **Source Definition** tab, add a new **Source Field** named Compliance.

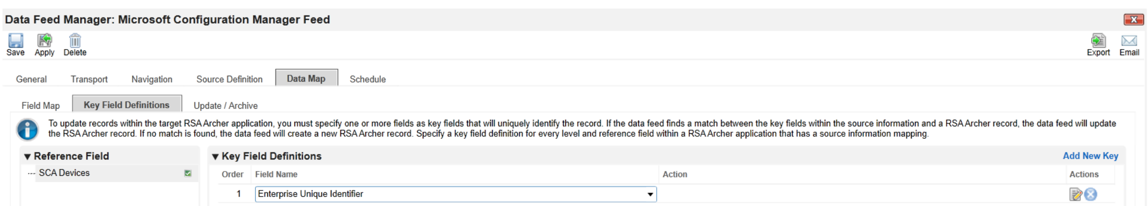


18. Edit the new **Source Field** with the static text “Out of Policy”.

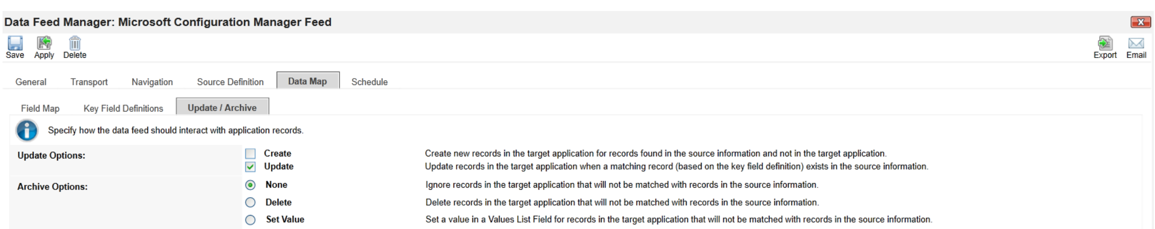
19. In the **Field Map** sub-tab in the **Data Map** tab, drag and drop the **Source Fields** onto the **Target Fields** as shown in the images below.



20. In the **Key Field Definitions** sub-tab in the **Data Map** tab, select **Enterprise Unique Identifier**.



21. In the **Update / Archive** sub-tab in the **Data Map** tab, ensure only **Update** is selected.

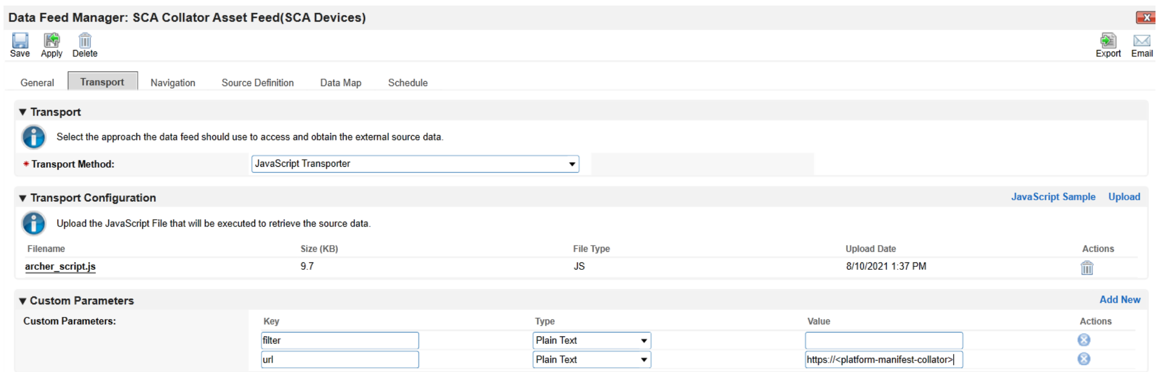


At this point, the Data Feed for the Microsoft Configuration Manager is configured. Click the **Start** button to confirm that the Data Feed has been properly configured. Archer will report any errors that are useful for debugging.

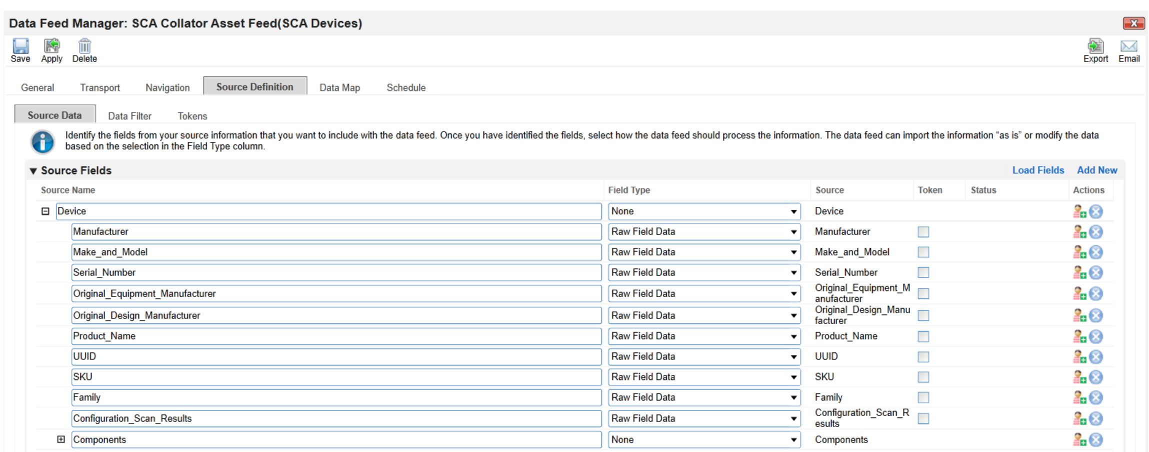
2.11.2.2.3 Create the PMCS Data Feed

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.



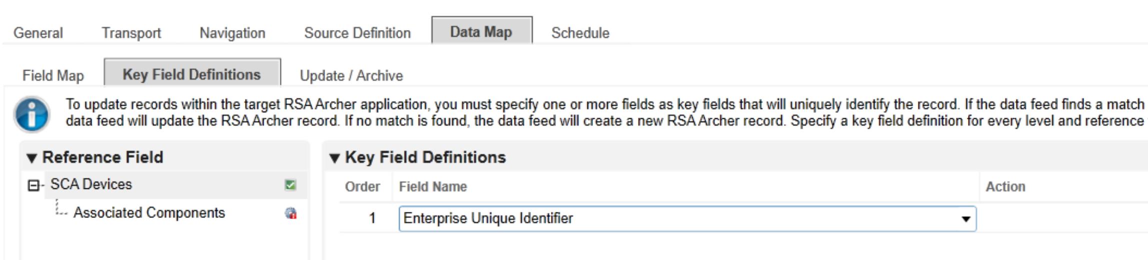
24. Map the **Source Fields** to the **Target Fields** and the **Field Map** sub-tab in the **Data Map** tab. Use Table 2-8 for reference.

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

25. In the **Key Field Definitions** sub-tab in the **Data Map** tab, choose Enterprise Unique Identifier as the **Key Field** definition.

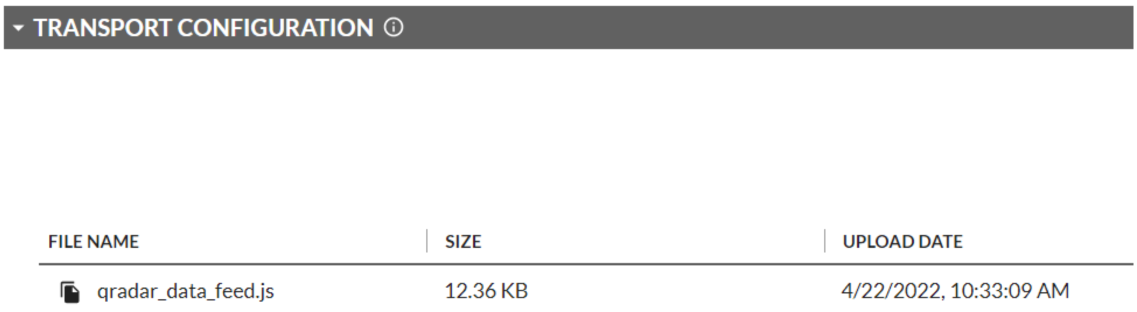


The Data Feed for the PMCS is configured. Click the **Start** button to confirm that the Data Feed has been properly configured. Archer will report any errors that are useful for debugging.

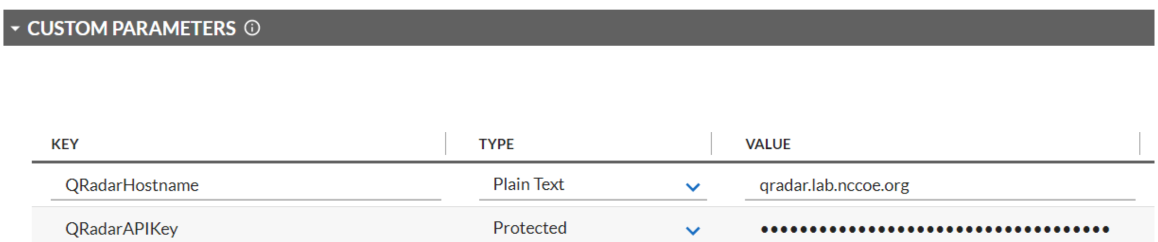
2.11.2.2.4 Create IBM QRadar Offenses Data Feed

Repeat the steps from Section 2.11.2.2.1 to add the Data Feed for IBM QRadar with the following modifications:

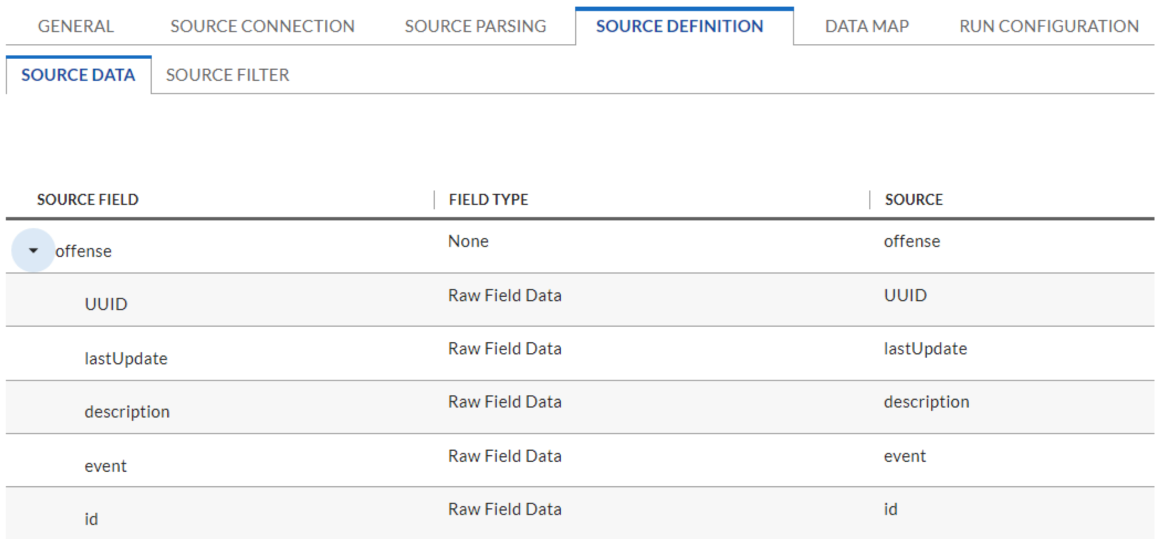
26. In the **Transport Settings** section of **Source Settings,** choose the IBM QRadar script *(Integration-Scripts\Archer Integrated Risk Management Data Feed Integrations\IBM QRadar\app.js)* from the project repository.



27. In the Custom Parameters section of the Source Connection tab, enter the hostname of the QRadar system and the API key created in Section 2.11.3.2.4. Ensure that the QRadarAPIKey is of type Protected.



28. In the **Source Data** section of the **Source Definition** tab, upload the example XML QRadar response file.

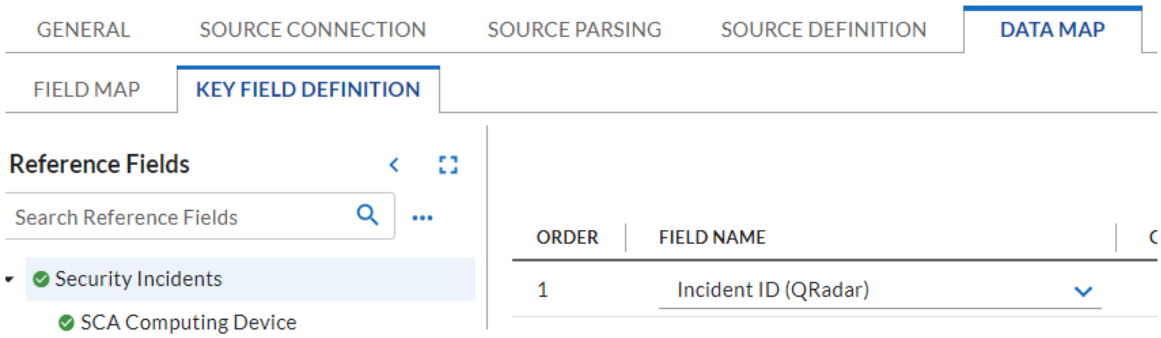


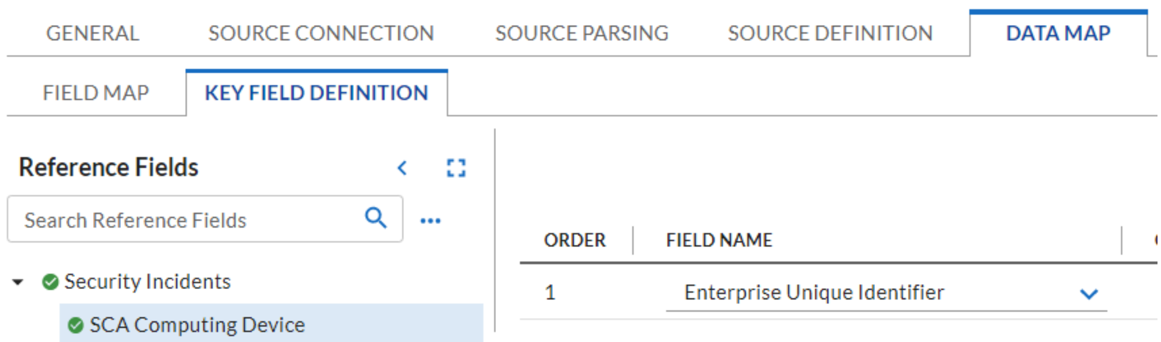
29. Map the **Source Fields** to the **Target Fields** in the **Field Map** sub-tab in the **Data Map** tab. Use Table 2-10 for reference.

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)

30. In the **Key Field Definition** sub-tab in the **Data Map** tab, choose **Incident ID (QRadar)** as the Key Field Definition. Additionally, choose **Enterprise Unique Identifier** as the **Key Field** definition for the **SCA Computing Device** reference field.

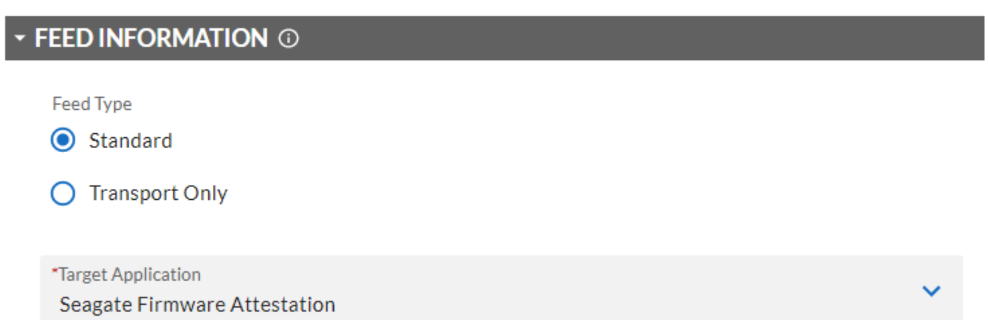




2.11.2.2.5 Create Seagate API Data Feeds

Repeat steps from Section 2.11.2.2.1 to add the Data Feed for Seagate drive firmware attestation and firmware hash data with the following modifications:

31. Enter *Seagate Attestation Feed* in the **Name** field section of the **General** tab. In the **Feed Information** section of the same tab, select *Seagate Firmware Attestation* from the **Target Application** pull-down menu.



32. In the **Transport Configuration** section of **Source Settings,** choose the Seagate script from the project repository.

TRANSPORT CONFIGURATION

FILE NAME	SIZE	UPLOAD DATE
archer_script.js	9.7 KB	2/10/2022, 3:42:17 PM

33. In the **Custom Parameters** section of **Source Connection** tab, enter the PMCS URL and the **filter** value of *seagate.fw.attestation*.

CUSTOM PARAMETERS

KEY	TYPE	VALUE
filter	Plain Text	seagate.fw.attestation
url	Plain Text	http://

34. In the **Source Data** section of the **Source Definition** tab, upload the example Seagate Firmware Attestation XML response file.

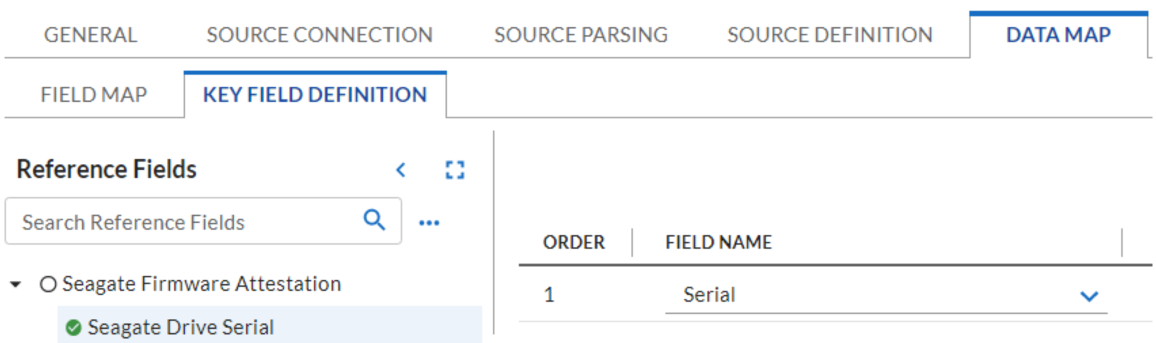
SOURCE FIELD	FIELD TYPE	SOURCE
SeagateDriveFirmwareAttestation	None	SeagateDriveFirmwareAttestation
device_uuid	Raw Field Data	device_uuid
drive_serial	Raw Field Data	drive_serial
assessor_id	Raw Field Data	assessor_id
root_of_trust_id	Raw Field Data	root_of_trust_id
root_of_trust_nonce	Raw Field Data	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

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.

Table 2-10 Seagate Drive Data Feed Field Mapping

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

36. In the **Key Field Definition** tab within the **Data Map** tab, select *Serial* in the pull-down **Field Name** column.

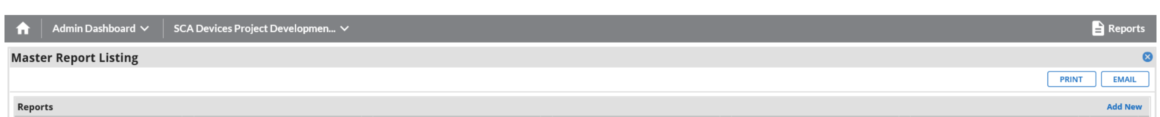


37. Save the new Data Feed.

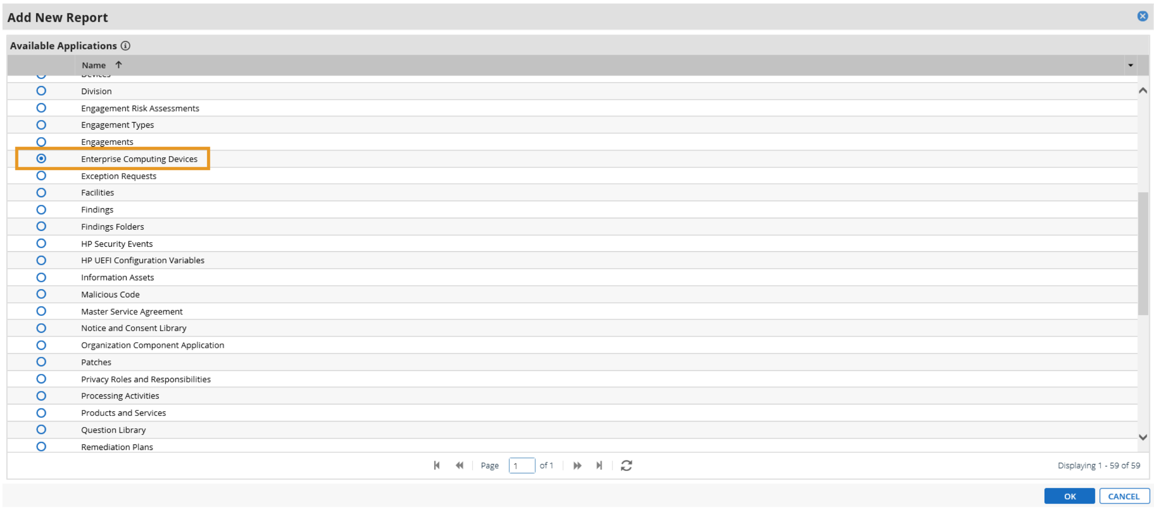
Repeat the procedures in this section to create a Data Feed that will collect the Seagate drive firmware hash values. Note that this Data Feed will target the *Seagate Firmware Hash* application.

2.11.2.3 Create the Dashboard

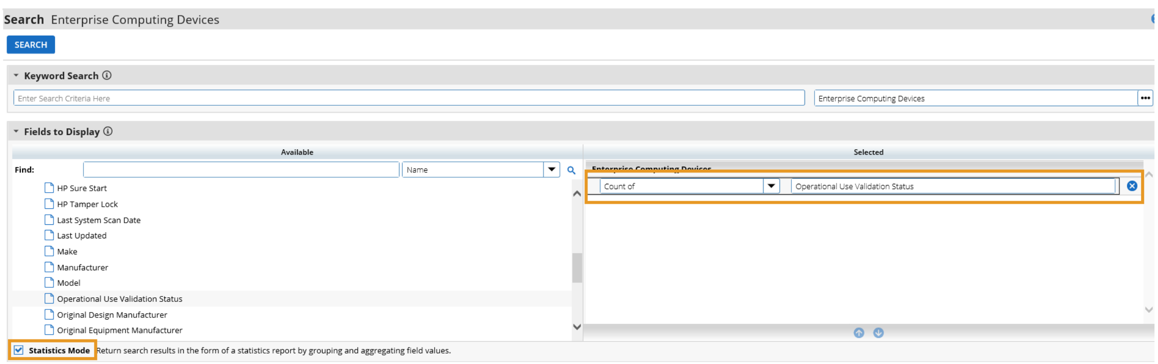
1. Create a new report by clicking **Reports** in the administrative console and **Add New.**



2. Select the Devices application that was created in the preceding steps—in this case, **Enterprise Computing Devices.**

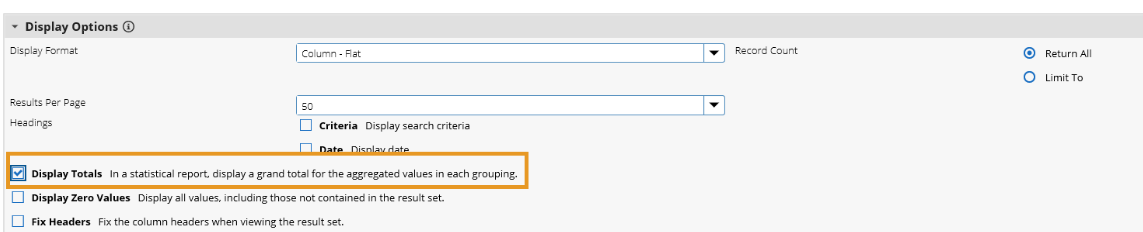


3. Click the **Statistics Mode** option. In the **Fields to Display** section, select **Operational Use Validation Status** and remove the default selections.

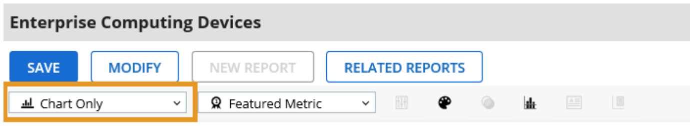


4. In the **Filters** section, select *Operational Use Validation Status* for **Field to Evaluate**, *Equals* for **Operator,** and *Policy violation* for **Value(s).**

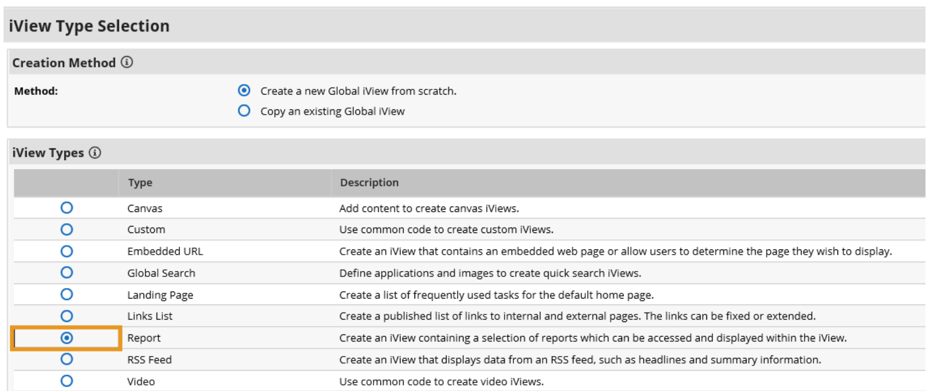
5. Select **Display Totals** in the **Display Options** section.



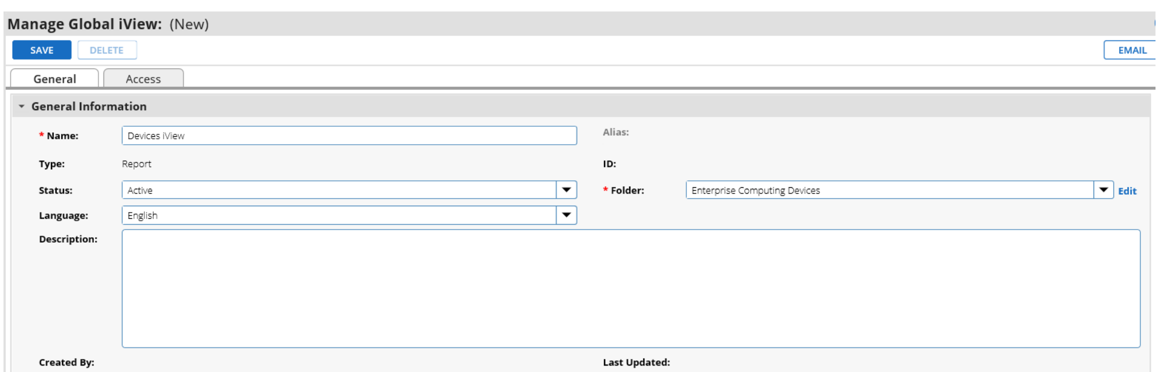
6. Select **Chart Only** and click **Save** and supply a unique name for the report.



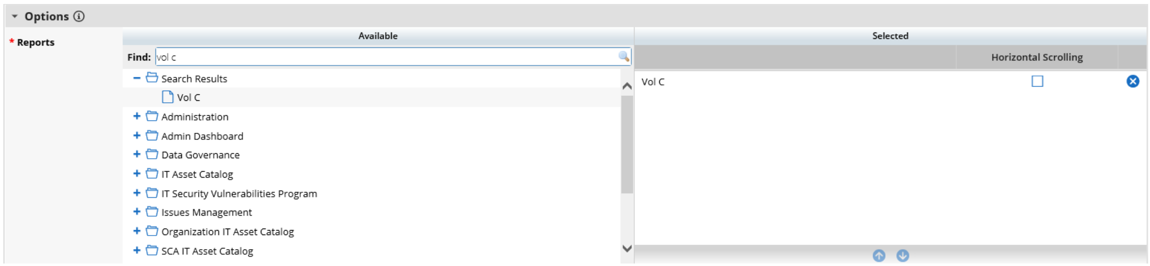
7. Create a new iView by navigating to **Workspaces and Dashboards > Global iViews** in the administrative menu. Click **Add New.**
8. In the **iView Types** section, select **Report** and click **OK.**



9. In the **General Information** section, supply a name and a folder to store the new iView.

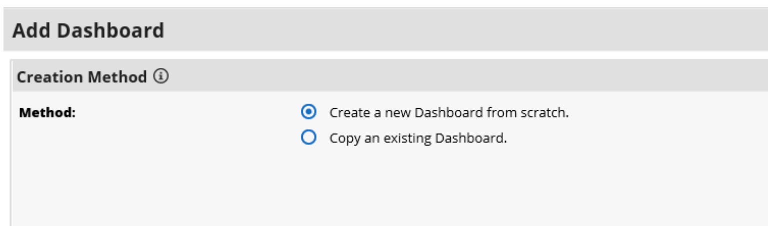


10. In the **Options** section, choose the report that was created in the preceding steps and save the iView.



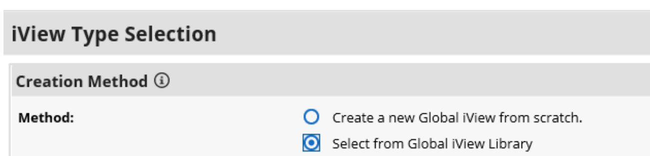
11. Create a new Dashboard by navigating to **Workspaces and Dashboards > Dashboards** in the administration menu. Click **Add New.**

12. Select **Create a new Dashboard from scratch** and click **OK.**



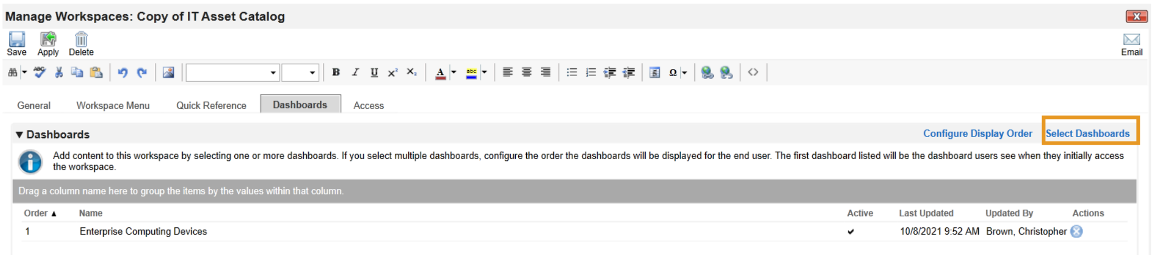
13. In the **General** tab, supply a name for the Dashboard.

14. In the **Layout** tab, click **Select iViews.** Choose *Select* from **Global iView Library** for the **Creation Method.** Choose the iView created in the preceding steps and click **OK.**



15. The selected iView will appear in the layout. Save the Dashboard.

16. Open the solution workspace by navigating to **Workspaces and Dashboards > Workspaces** in the administration menu. In the **Dashboards** tab, choose the Dashboard created in the preceding steps by clicking **Select Dashboards.**



17. Save the workspace. At this point, the new Dashboard will appear as part of the workspace. For further customization options, refer to the RSA website.

18. Repeat the steps in this section to create a report that tracks platform integrity issues that are 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

2.11.3 IBM QRadar Integrations

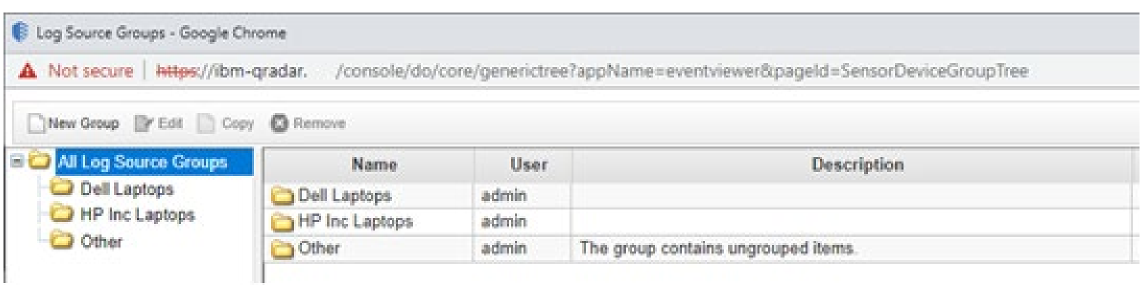
The following sections describe how to integrate Dell and HP Inc. laptops with QRadar so that the laptops transmit continuous monitoring event logs to the QRadar console.

2.11.3.1 Dell and HP Inc. Laptops

Perform the prerequisite steps in Section 2.2.1.3, then on each target laptop:

1. Ensure Remote Event Log Management is enabled for each laptop.

2. (Optional) In the QRadar console, create a new log source group which may be desirable to help organize target laptops. In our demonstration, we created a group for each manufacturer.



3. Create a new log source for the WinCollect Agent (see Section 2.10.1). Note that when configuring the Log Source parameters, a Windows account is required to retrieve the relevant

security event. This demonstration created a domain account with privileges limited to the scope of this capability (Manage auditing and security log permission enabled).

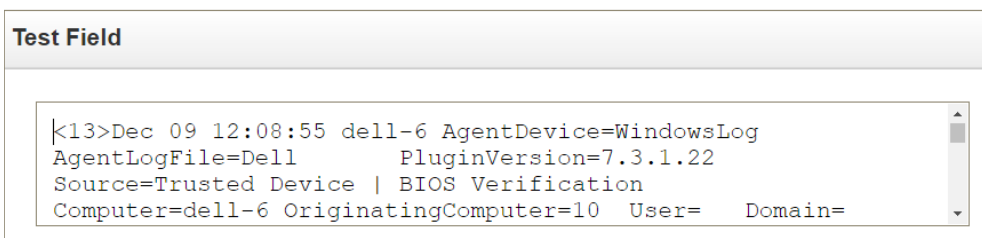
2.11.3.2 IBM QRadar

The section describes the procedures that will create *Offenses* generated from detected laptop platform integrity security events. Additionally, it also describes an API key that is used to access the QRadar REST API. The key is used as input to Section 2.11.2.2.4.

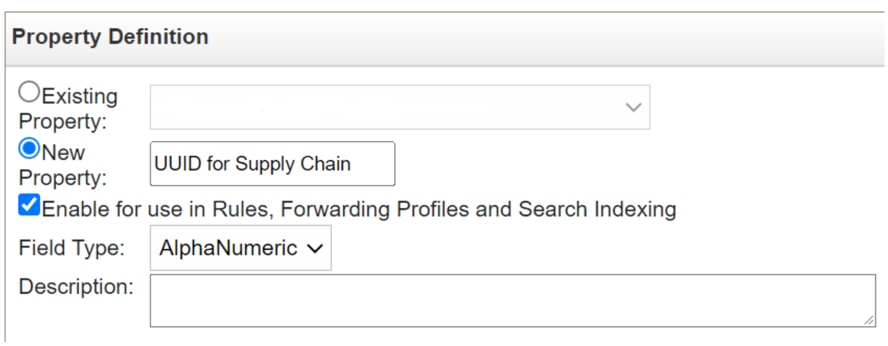
2.11.3.2.1 Create Custom Event Property (UUID)

This property uses a regular expression (regex) to identify universally unique identifiers (UUIDs) that are embedded in Windows 10 Event Logs that are sent from laptops when a platform integrity issue is detected.

4. In the QRadar console, navigate to **Admin** > **Custom Event Properties**. Click **Add** and a new window pops up. In the **Test Field,** paste in the example event log.

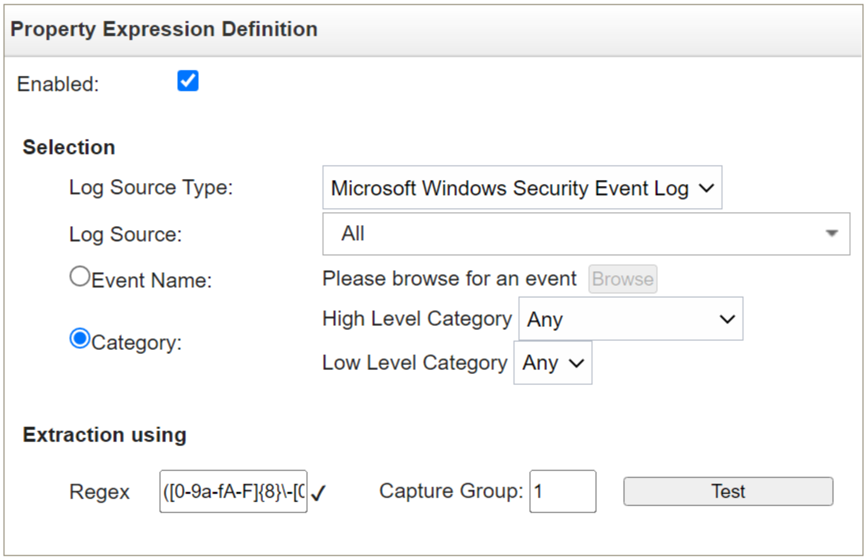


5. In the **Property Definition** section, select **New Property** and enter *UUID for Supply Chain.* Check *Enable for use in Rules, Forwarding Profiles and Search Indexing*.



6. In the **Property Expression Definition** section, ensure *Enabled* is checked. In **the Log Source Type** pull-down, select *Microsoft Windows Security Event Log* and select *All* in the **Log Source** pull-down. Select the *Category* radio button. Choose *Any* in both the **High Level Category** and **Low Level Category** pull-downs. In the **Regex** field, insert the value below.

```
([0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-
F]{12})
```



7. Click the **Test** button. If successful, a message will appear that the expression has been highlighted in the payload. Click the **Save** button.

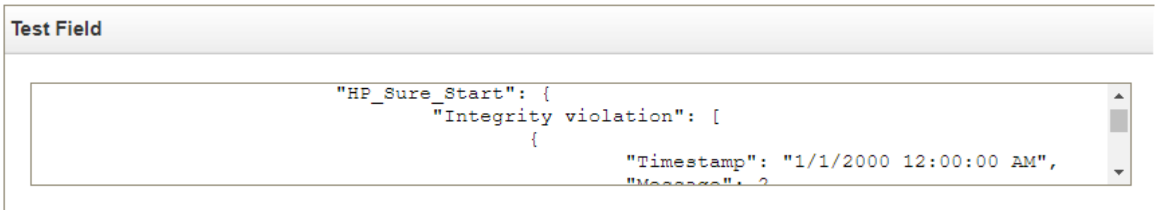
2.11.3.2.2 Create Custom Event Properties (Security Events)

This section describes how to create filters that will identify the individual HP Inc. and Dell platform integrity events that have been detected and reported to QRadar. Use Table 2-11 as a guide. We used existing QRadar Categories which group manufacturer security events. These procedures also require an example of the security event payload that is created on the manufacturer's laptop when a platform integrity issue is detected. For HP Inc laptops, the payloads are generated by custom PowerShell scripts which consume the output from the CMSL Get-HPFirmwareAuditLog command. Dell security event payloads are generated directly by the Dell Trusted Devices platform.

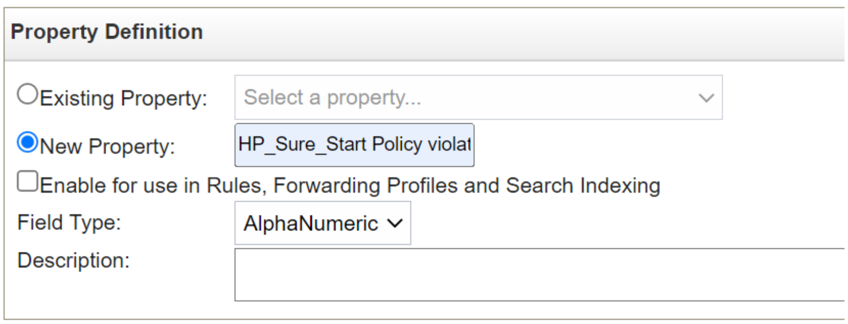
1242 **Table 2-11 QRadar Security Event Mapping**

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*.



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*.

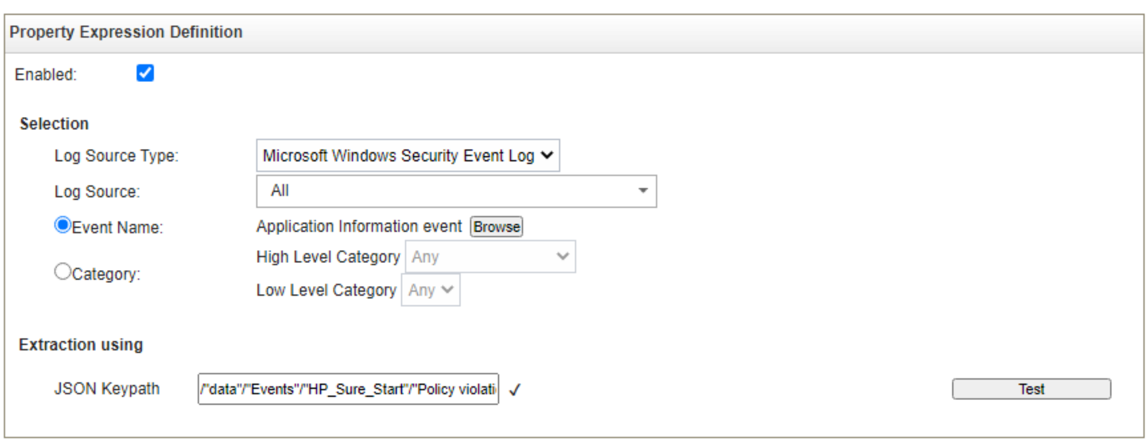


3. In the **Property Expression Definition** section, make sure **Enabled** is checked. In **Log Source Type**, select *Microsoft Windows Security Event Log*. In **Log Source** select **All**. Select the *Event Name* radio button.

    a. Click **Browse** and search for *"Application Information Event"* (with quotes) in the **QID/Name** field. Select it and click **OK**.

    b. Select **Extraction using JSON Keypath**. *"HP_Sure_Start Policy violation"* will look like the following as an example:

    `/"data"/"Events"/"HP_Sure_Start"/"Policy violation"[]`



4. Click the **Test** button. If successful, the security event is found in the **Test Field**. Click **Save**.

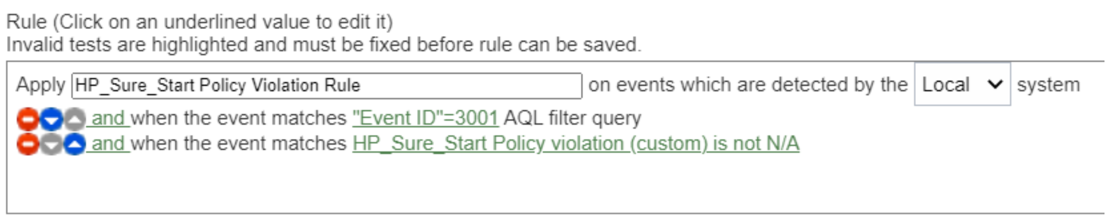
Continue the process for all events listed in Table 2-11.

2.11.3.2.3 Create QRadar Rules

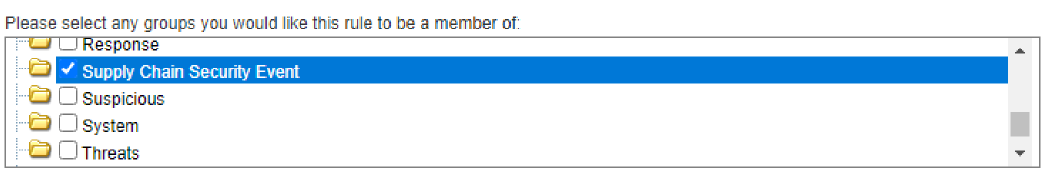
5. In the QRadar console, click **Log Activity**. Select **Rules** > **Rules** then **Actions** > **New Rule**.
6. Ensure **Events** is selected, then click **Next.**
7. Enter a name for the rule. We used the following pattern: *"[Event Category] [Event Value] rule".*
8. In the rules editor, search for *"event matches this AQL filter query"*. Click the *"this"* hyperlink to launch the Ariel Query Language (AQL) filter query. Enter the query below and click **Submit**.

   ```
   "Event ID"=3001
   ```

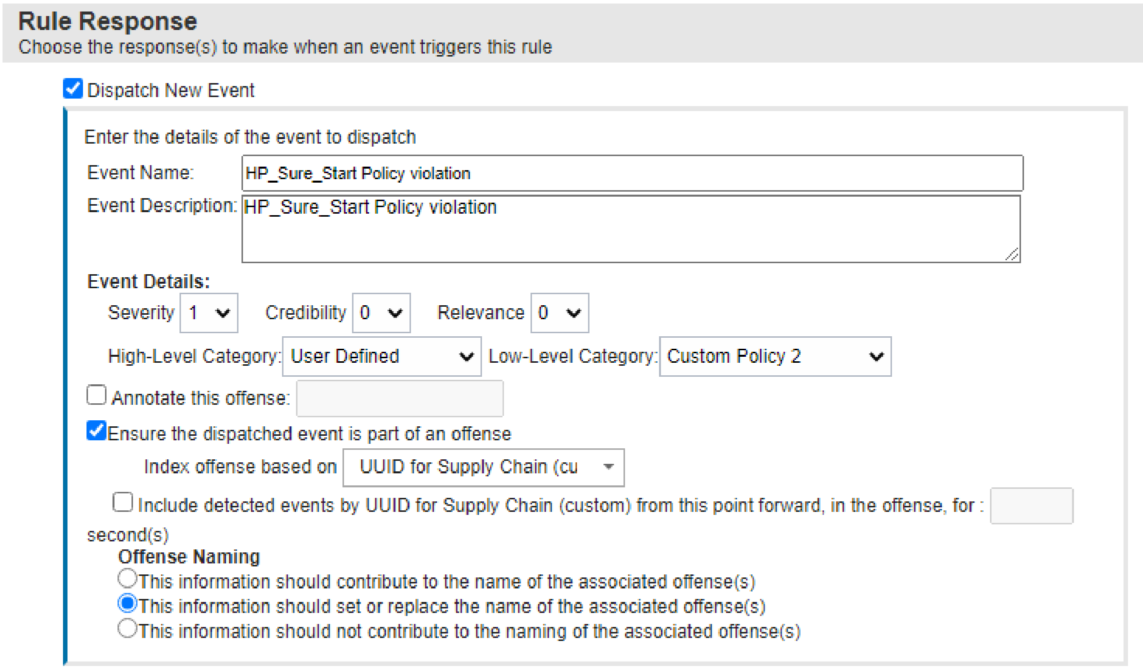
9. Create another criteria by using *"when the event matches this search filter"*. Click *"this search filter"* and locate the matching **Custom Property**. Select *"is not N/A"* and click **Add**. Click **Submit**.



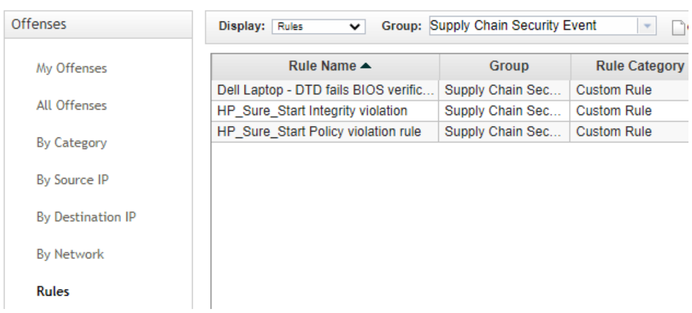
10. (Optional) Make the rule part of a group to organize platform integrity offenses. We created a custom group named *"Supply Chain Security Event"*.



11. Click **Next**. In the **Rule Response** section, select **Dispatch New Event**. Create an **Event Name** and **Event Description** following the same pattern as above.
12. In the **Event Details** section, select the **High-Level Category** of *"User Defined"* and choose the **Low-Level Category** noted in Table 2-11.
13. Check *"Ensure the dispatched event is part of an offense"*. Index offense based on *"UUID for Supply Chain"* in the pull-down menu.
14. In the **Offense Naming** section, select the second option (replace).



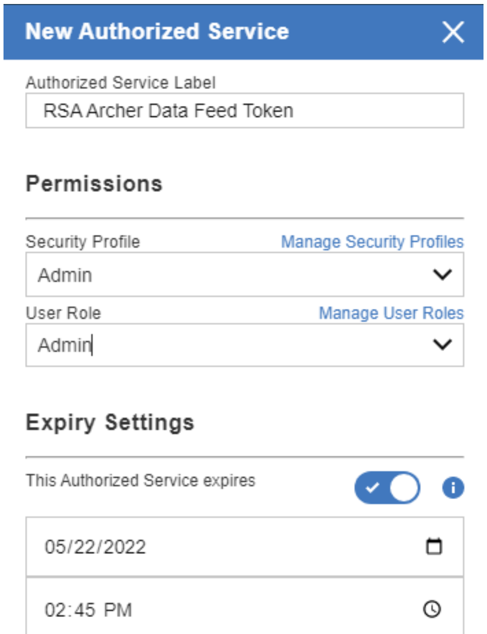
15. Click **Finish.** The new rule will appear in the **Offenses** > **Rules** tab.



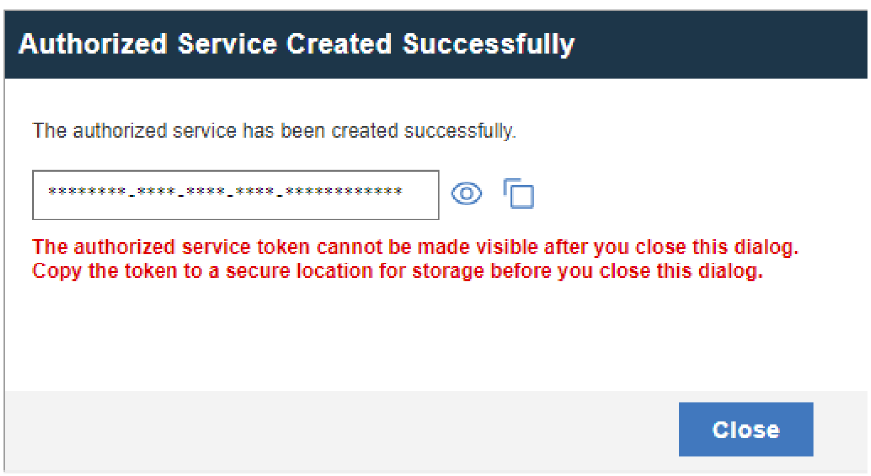
Repeat this section for every security event listed in Table 2-11.

2.11.3.2.4 Create an Authorized Service Token

1. In the administration console, click **Authorized Services**, then **Add New**. Enter an **Authorized Service Label** and appropriate **Security Profile** and **User Role** for your environment. Click **Save**.



2. The QRadar console will display the following dialog. Click the "eye" to reveal the secret token. Store the token securely.



3 Operational Considerations

This section describes the execution steps of an IT administrator assigned to the acceptance testing or monitoring of computing devices during their operational lifecycle. Each subsection restates the scenarios from the project description, but this prototype demonstration does not address each scenario in totality. This preliminary draft will be updated later with additional guidance for laptops and servers.

Create an environment as described in Section 2 before attempting to use the proof-of-concept tools below.

3.1 Scenario 2: Verification of Components During Acceptance Testing

In this scenario, an IT administrator receives a computing device through nonverifiable channels (e.g., off the shelf at a retailer) and wishes to confirm its provenance and authenticity to establish an authoritative asset inventory as part of an asset management program.

The general execution steps are as follows:

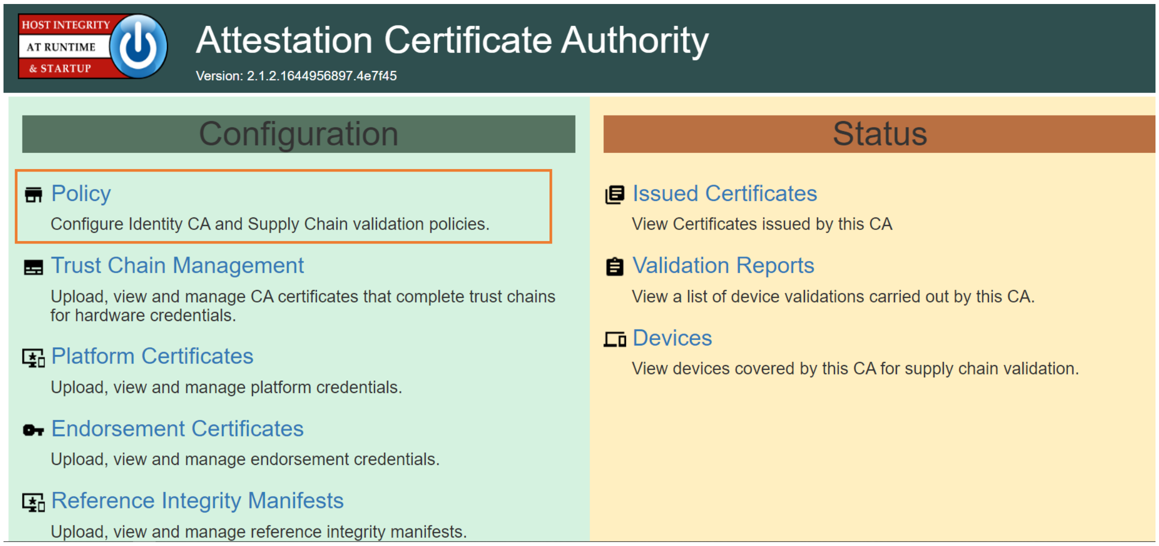
1. As part of the acceptance testing process, the IT administrator uses tools to extract or obtain the verifiable platform artifact associated with the computing device.
2. The IT administrator verifies the provenance of the device's hardware components by validating the source and authenticity of the artifact.
3. The IT administrator validates the verifiable artifact by interrogating the device to obtain platform attributes that can be compared against those listed in the artifact.
4. The computing device is provisioned into the physical asset management system and is associated with a unique enterprise identifier. If the administrator updates the configuration of the platform (e.g., adding hardware components, updating firmware), then the administrator might create new platform artifacts to establish a new baseline.

3.1.1 Technology Configurations

3.1.1.1 Configure the HIRS ACA

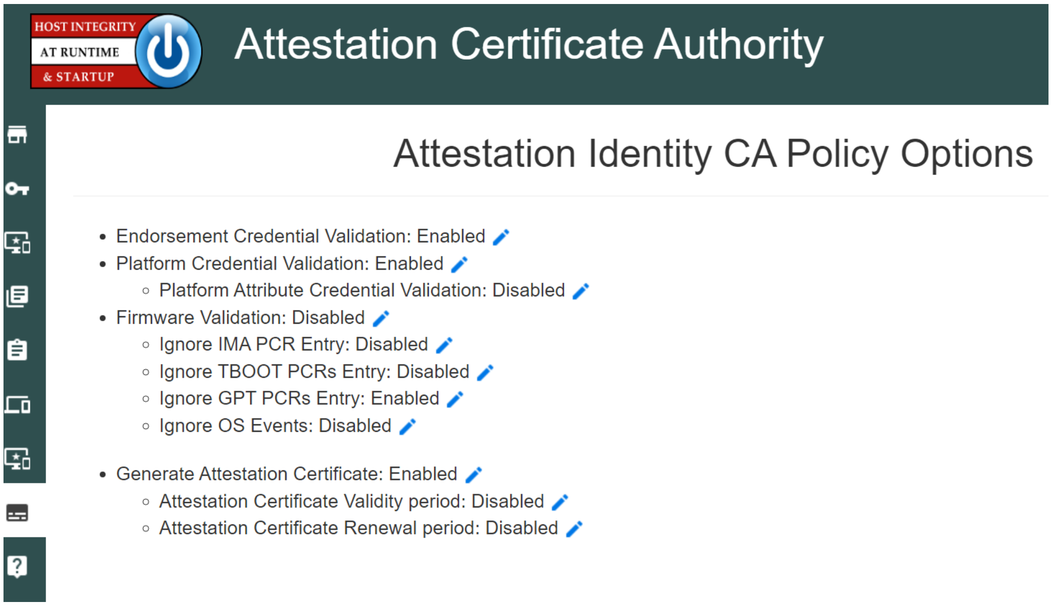
Before running the acceptance test on Dell and HP Inc. laptops, the HIRS ACA must be configured with the target laptop's platform attribute certificate and any trust chains associated with the platform attribute certificate and endorsement credential.

1. On the HIRS ACA web portal, under the **Configuration** panel, select **Policy.**

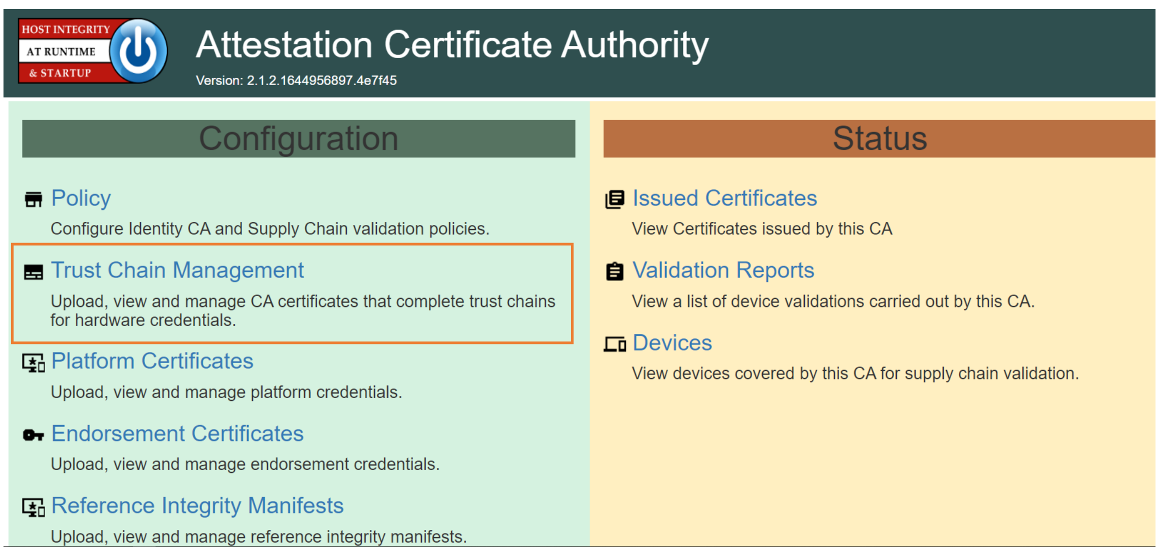


2. For this prototype demonstration, make sure the following policy options are set as listed in the table below.

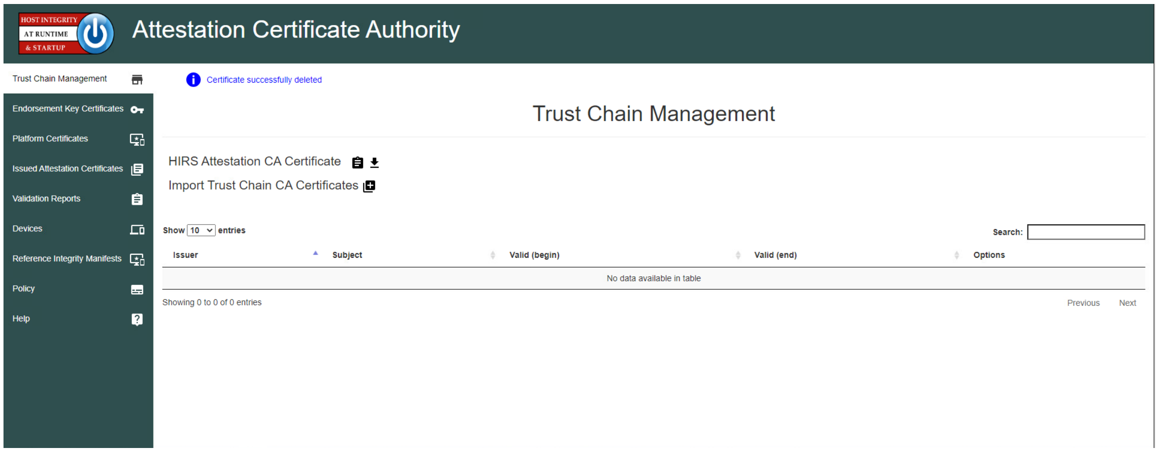
Policy Option	Setting
Endorsement Credential Validation	Enabled
Platform Credential Validation	Enabled
Platform Attribute Credential Validation	Enabled
Firmware Validation	Disabled
Ignore IMA PCR Entry	Disabled
Ignore TBOOT PCRs Entry	Disabled
Ignore GPT PCRs Entry	Disabled
Ignore OS Events	Disabled
Generate Attestation Certificate	Enabled
Attestation Certificate Validity period	Disabled
Attestation Certificate Renewal period	Disabled



3. Upload the trust chain certificates by navigating to the **Configuration** panel, then selecting **Trust Chain Management**.

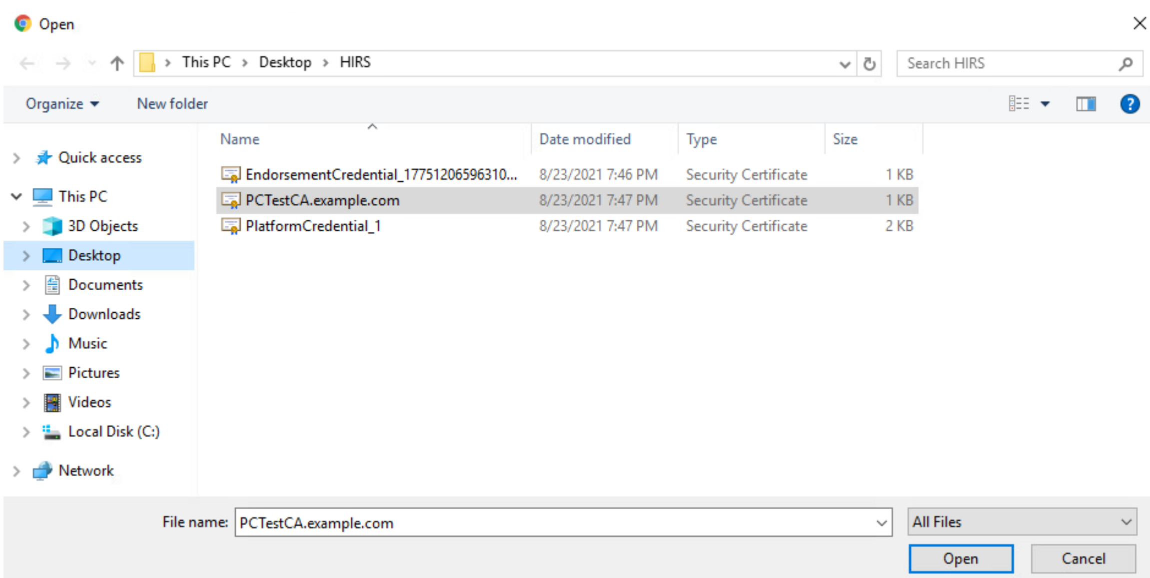


4. Select the icon beside **Import Trust Chain CA Certificates**.



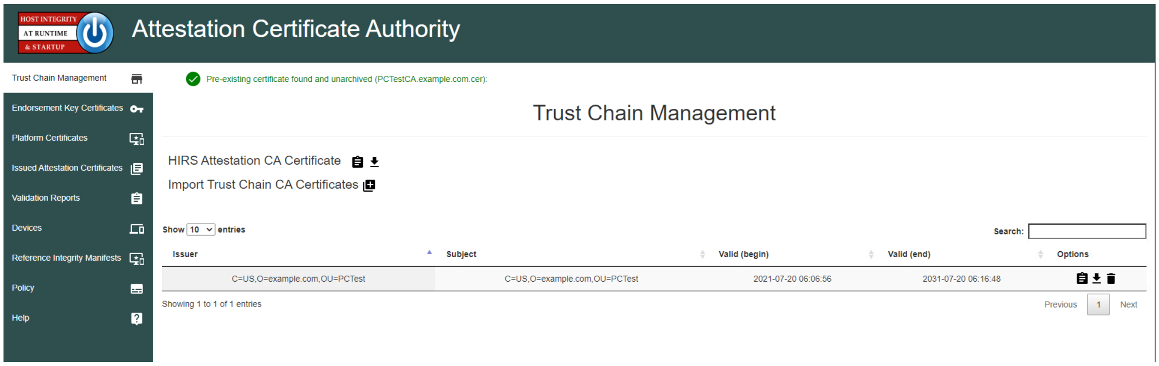
5. Select **Choose Files**.

6. Select the Trust Chain Certificate from the local computer. In the example below, the .crt file is named *PCTestCA.example.com*. Optionally, select multiple certificates if your implementation includes computing devices from distinct manufacturers. Click **Open**.

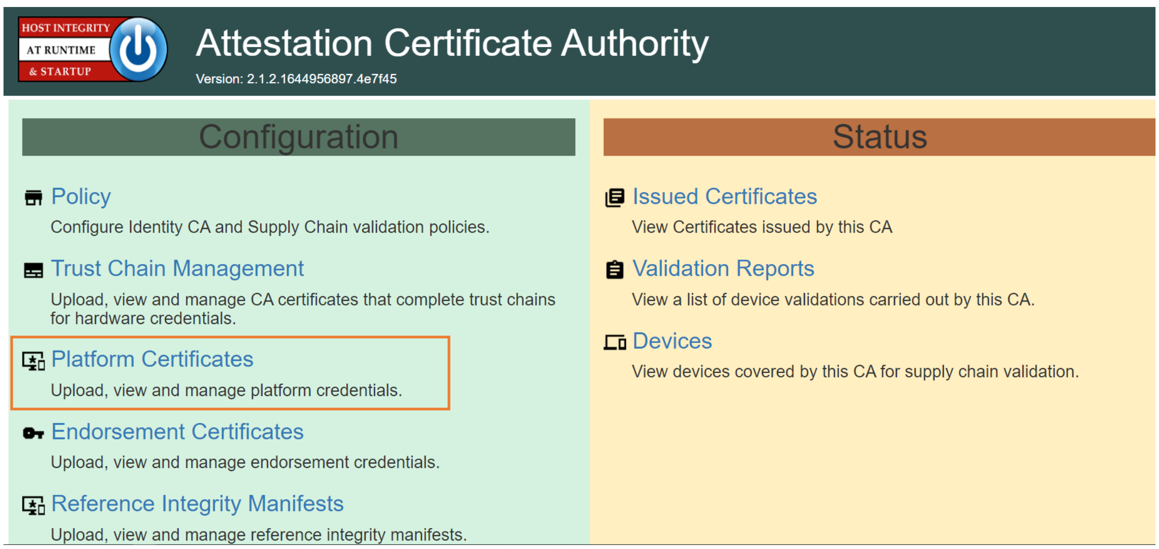


7. Select **Save.**

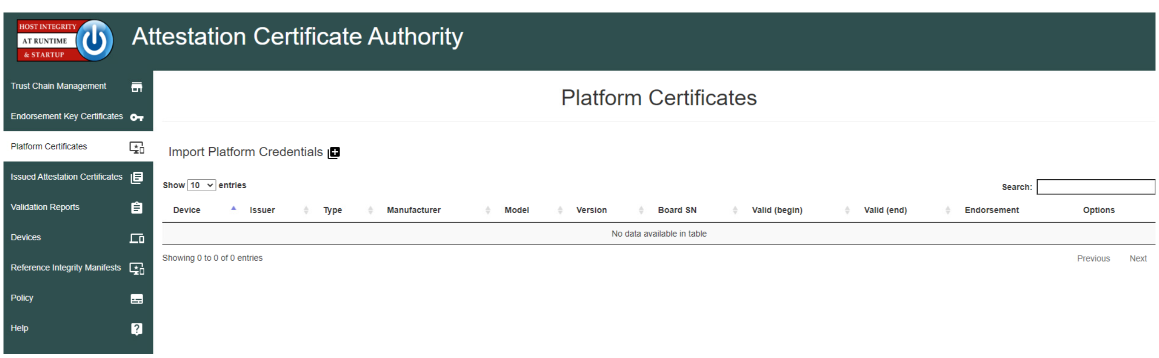
8. The Trust Chain certificate should appear under the **Trust Chain Management** tab. Repeat this process for all root and intermediate certificates.



9. Update the Platform Attribute certificates by navigating to the **Configurations** panel, then selecting **Platform Certificates**.

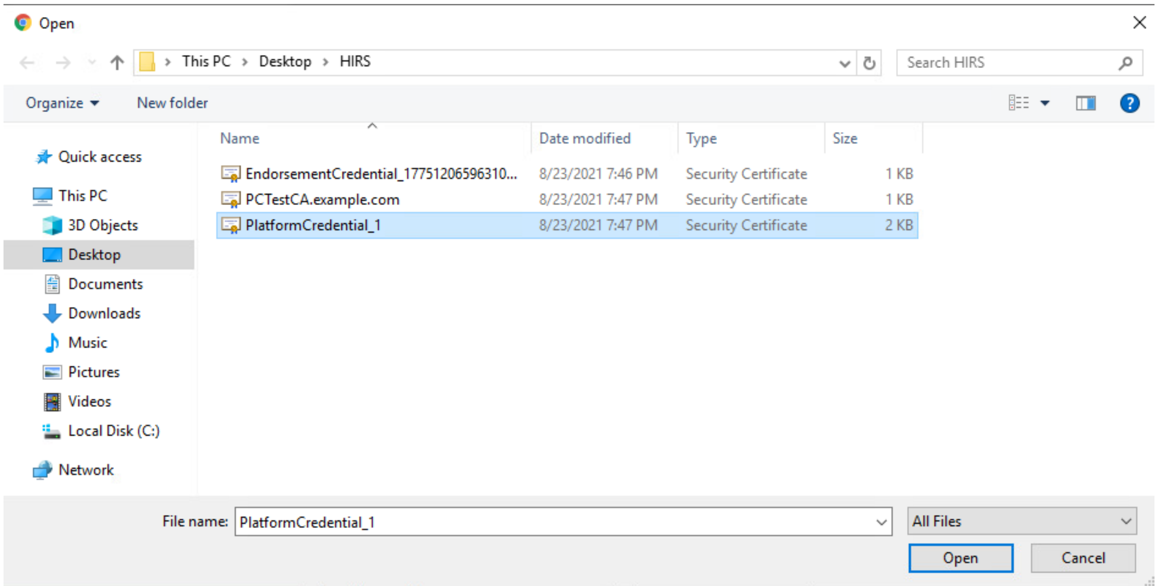


10. Select the icon beside **Import Platform Certificates**.



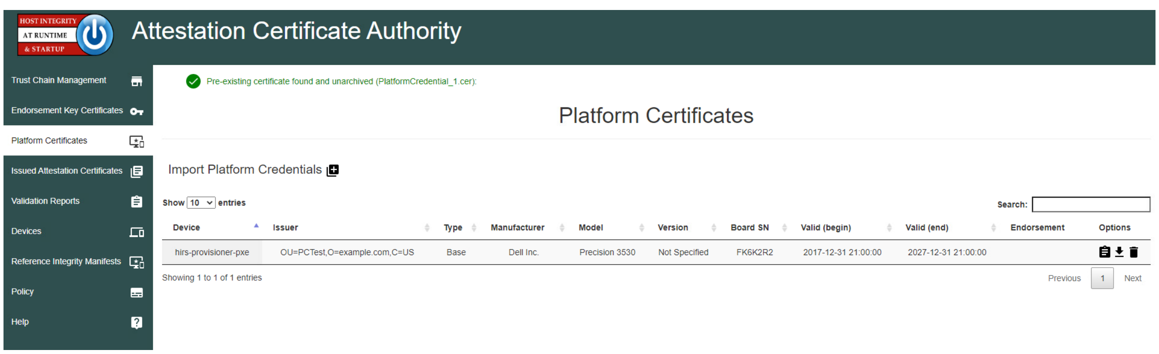
11. Select **Choose Files.**

12. Select the Platform Certificate from the local computer. In the example below, the .crt file is named **PlatformCredential_1**. Select the file and click **Open**.

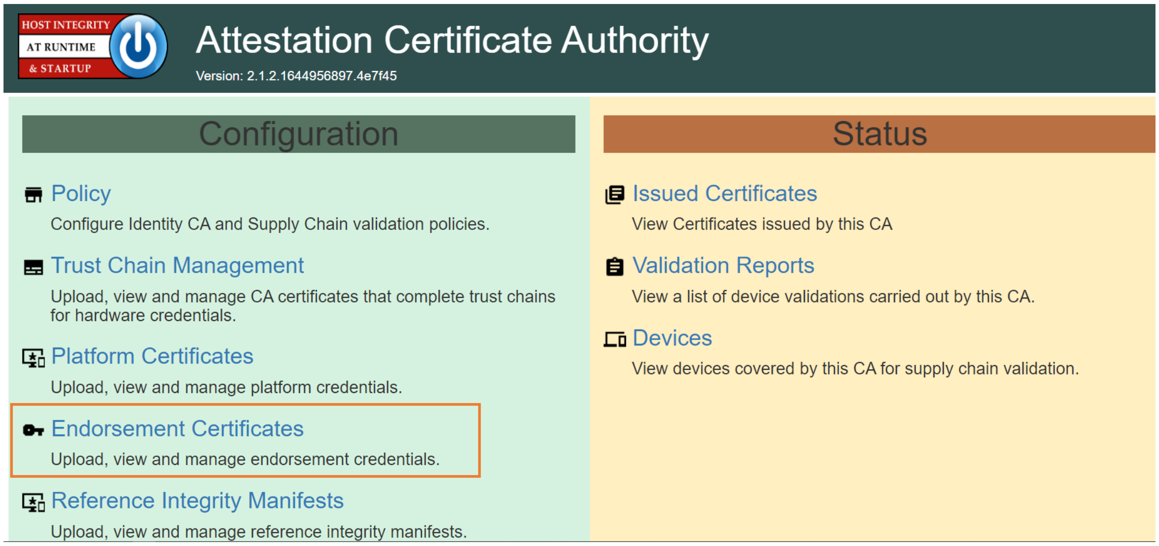


13. Select **Save**.

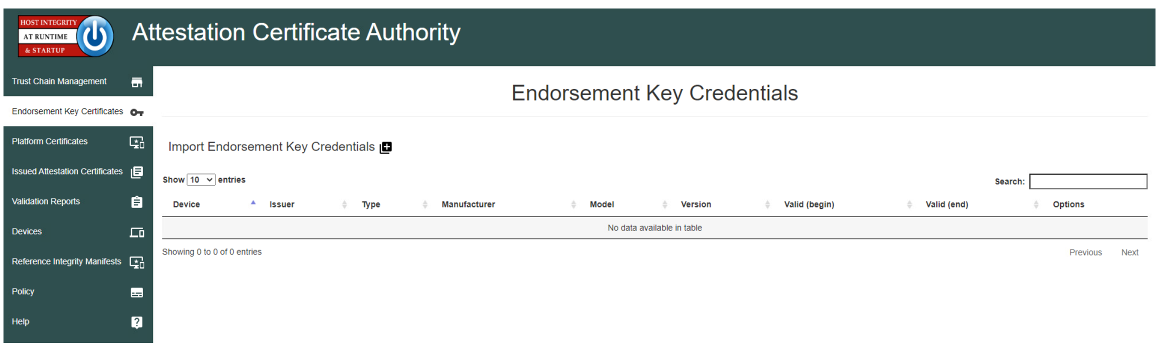
14. The Platform certificate should appear under the **Platform Certificates** tab.



15. Upload the Endorsement Key certificate by navigating to the **Configuration** panel, then selecting **Endorsement Certificates**.

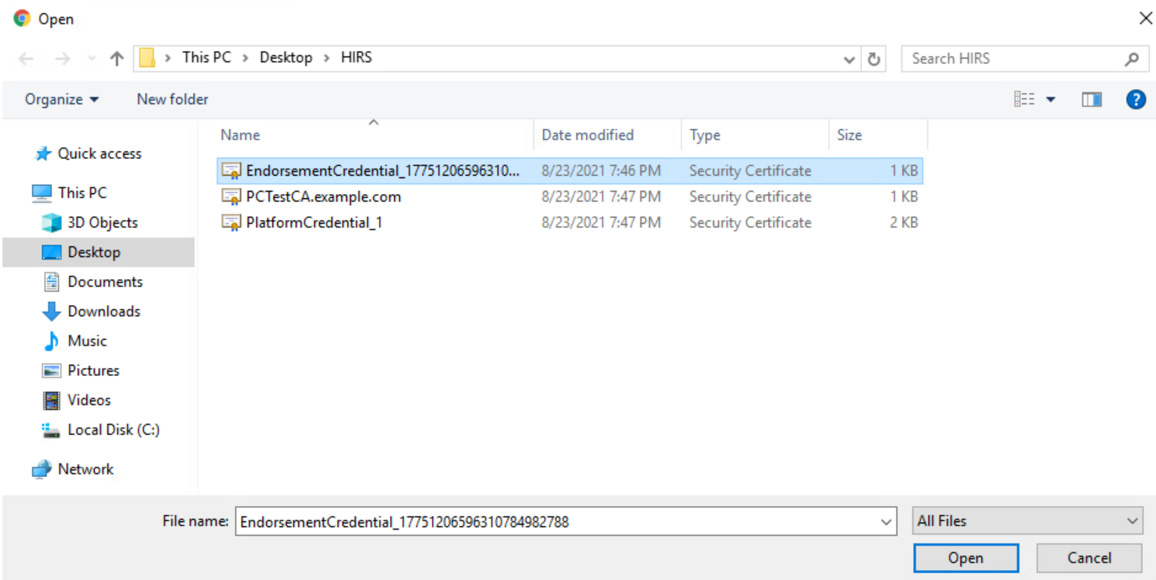


16. Select the icon beside **Import Endorsement Key Certificates**.



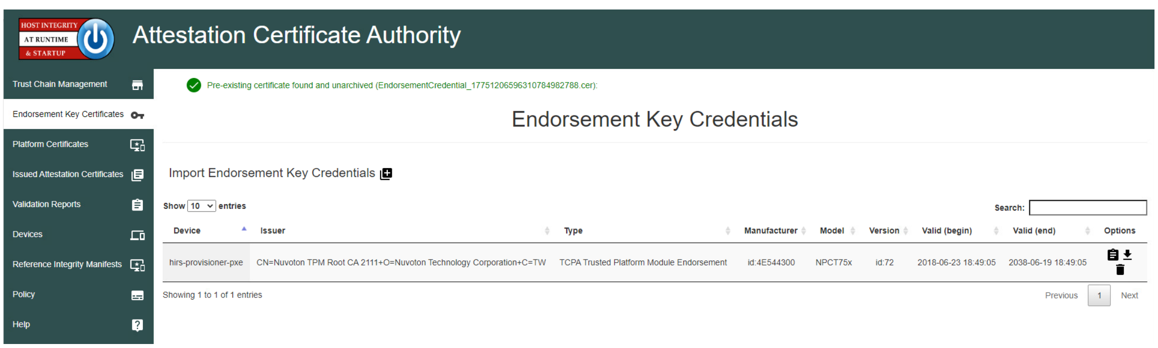
17. Select **Choose Files**.

18. Select the Endorsement Credential from the local computer. For this project, the .crt file is *EndorsementCredential_177512065963107849827 88*. Select the file and click **Open**.



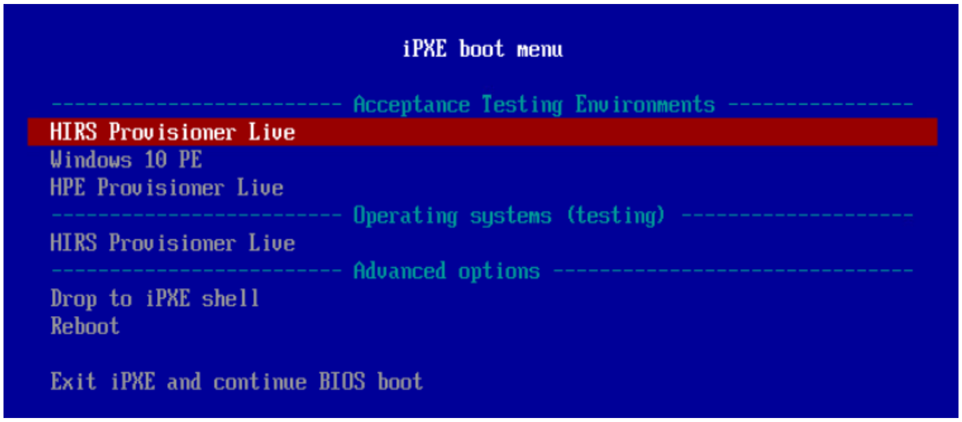
19. Select **Save.**

20. The Endorsement Key certificate should appear under the **Endorsement Key Credentials** tab.



3.1.1.2 Dell and HP Inc. Laptops

1. Boot the target laptop into the CentOS 7 acceptance testing environment via iPXE. This typically requires a one-time boot execution to prevent the laptop from loading the native OS. Consult the manufacturer's documentation for the appropriate steps. Choose HIRS Provisioner Live from the iPXE boot menu.



2. Once the live environment has loaded, log in as a user with root privileges. Run the provision.sh script. The script will attempt to:
    a. Change the hostname of the live environment. This assists the administrator in locating the target machine in the Eclypsium console.
    b. Run the Eclypsium scanner and submit results to the Eclypsium Analytic cloud platform.
    c. Run the HIRS provisioning script. If successful, post the results to the PMCS.

    The script will exit at any point an error is detected. Refer to the comments in the script to set this up in your own environment. Up-to-date information related to debugging the HIRS provisioning process can be found on the project site.

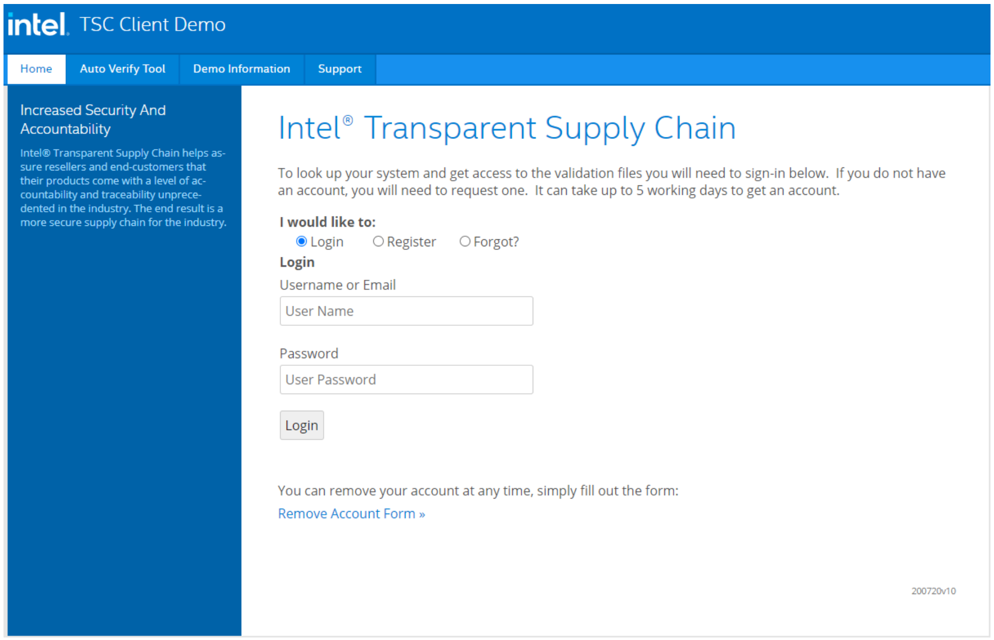
3.1.1.3 Intel-Contributed Laptops

The Auto Verify tool is central to scenario 2 acceptance testing. The tool compares the Direct Platform Data (DPD), allowing the customer to identify certain system changes from the time of manufacturing to the time of first boot. Install the Auto Verify Tool on the target system before attempting to execute the steps in this section.

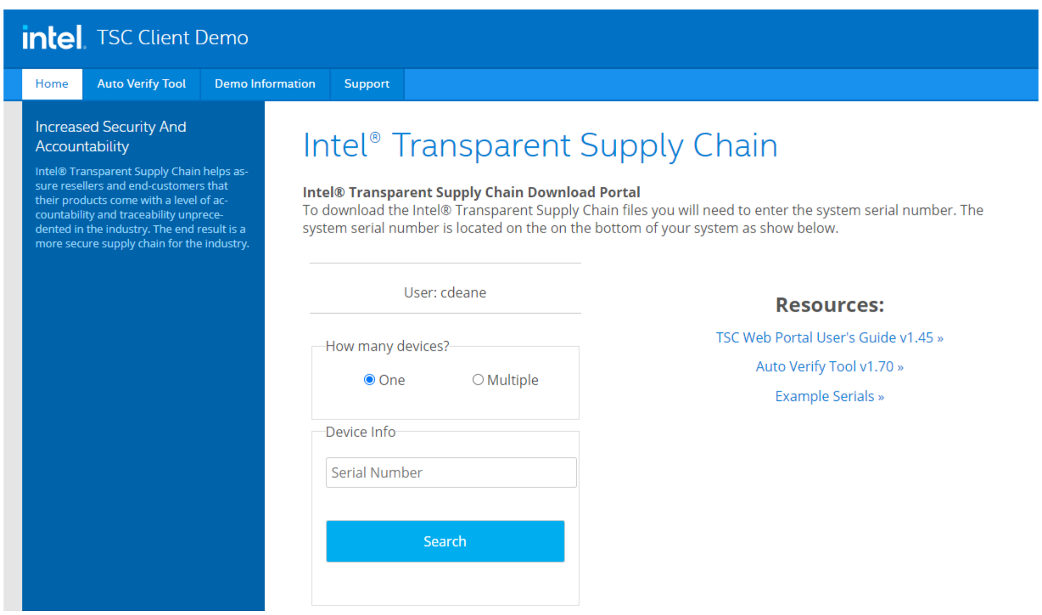
The DPD files and platform certificate files for the target laptop are available from Intel's Transparent Supply Chain demo page, https://tsc.intel.com/client-demo/. Work with your Intel representative to obtain credentials for your organization.

3.1.1.3.1 Download DPD File and Platform Certificate

1. Authenticate to the Intel TSC Client Demo portal page.



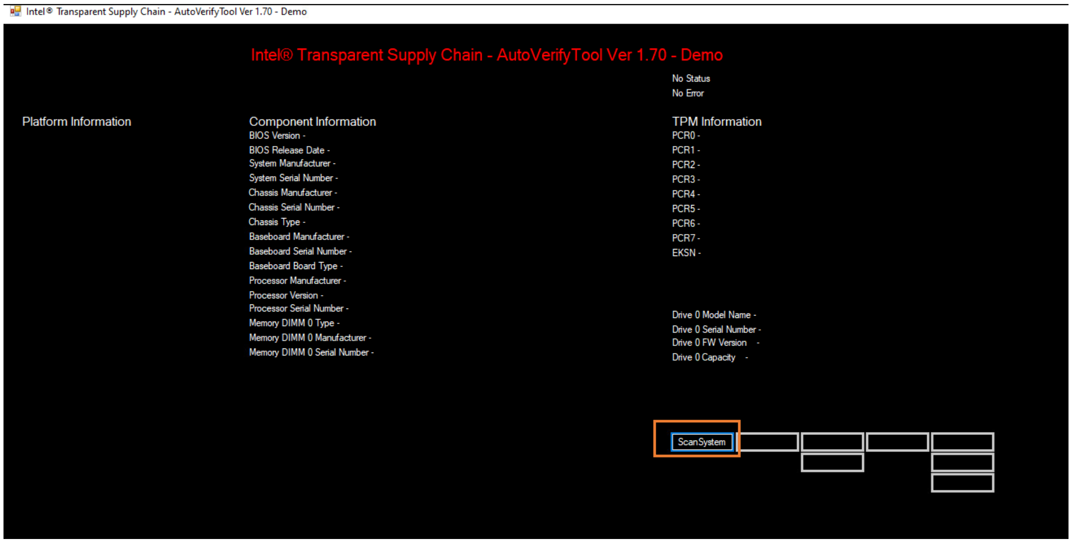
2. Enter the serial number of the Intel laptop. Select **Search.**



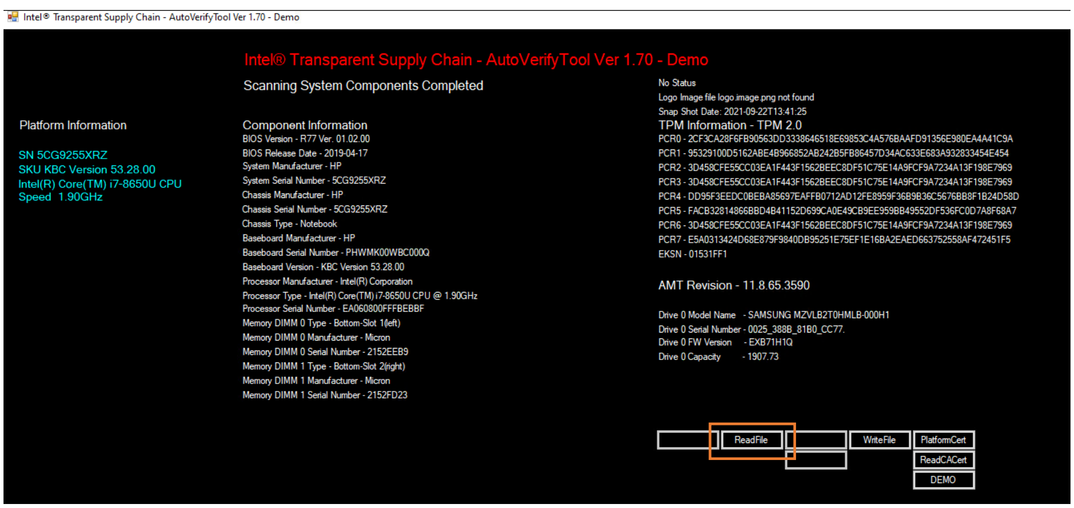
3. Download the zip file containing the DPD files and platform certificate. Save and unzip the file on the target laptop. These files will be used with the Auto Verify tool to determine if any components have been changed.

4. Launch the Auto Verify Tool.

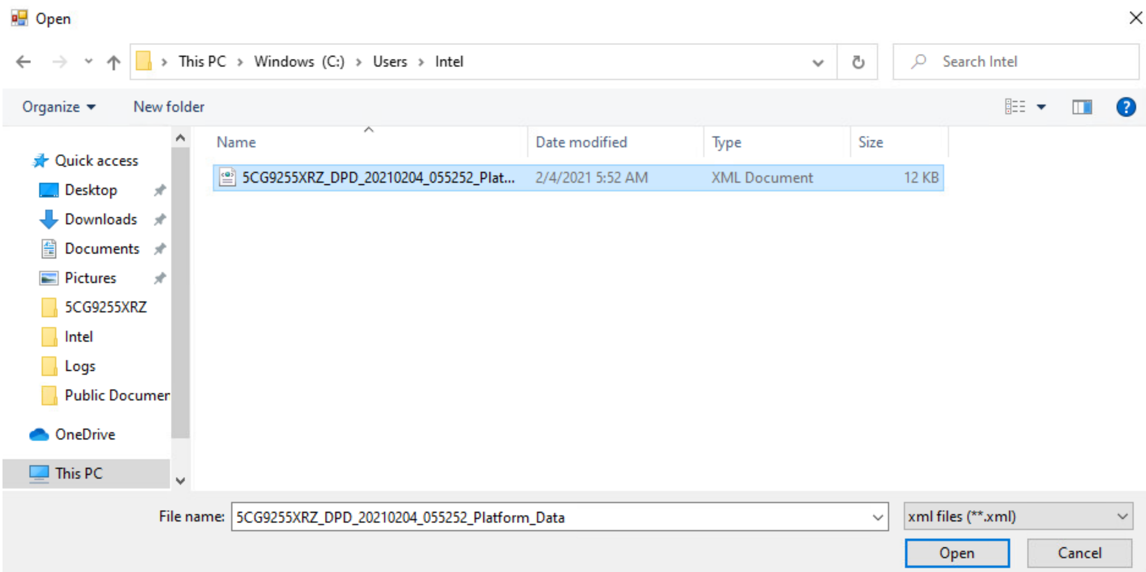
5. Click the **Scan System** button.



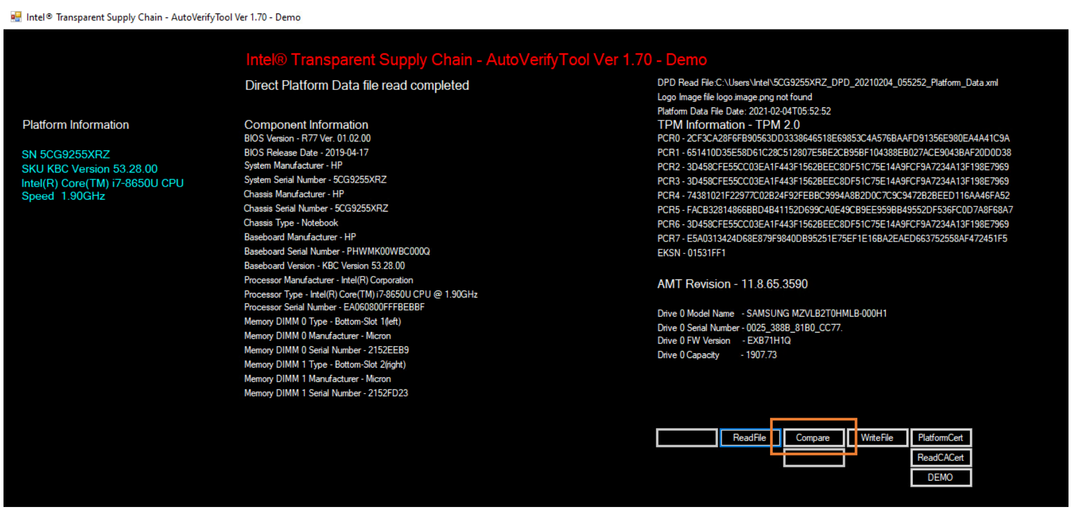
6. The Auto Verify Tool should populate the Component Information entries with the platform details of the computer. To compare the data to the DPD file stored on the local computer, click **ReadFile**.



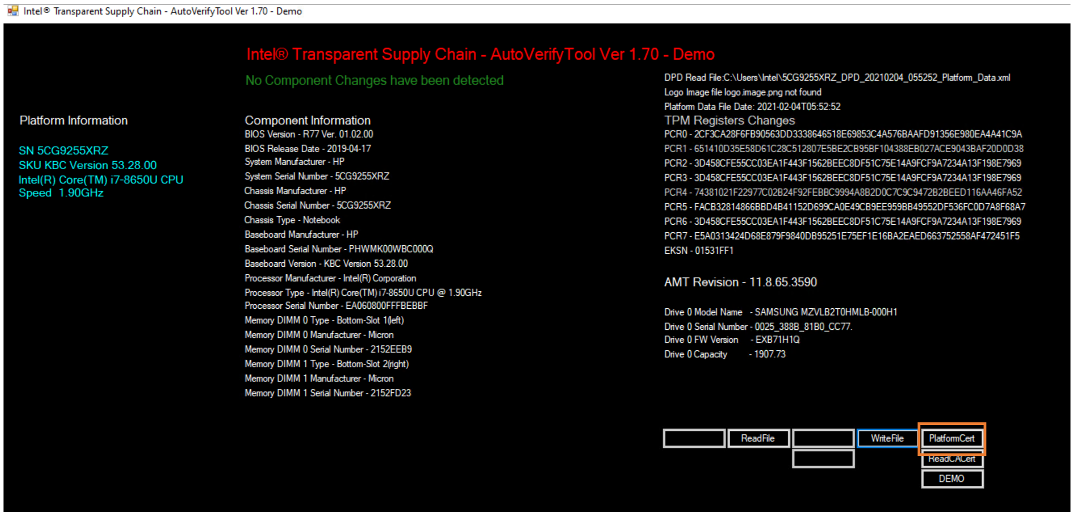
7. Navigate to the downloaded DPD file and select **Open.**



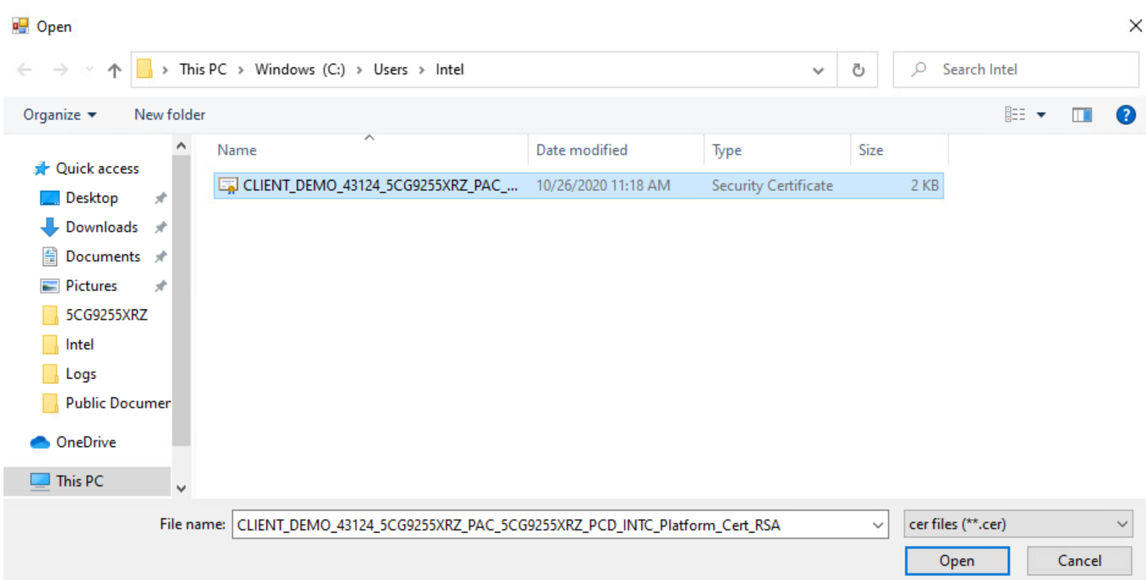
1374 8. Next, click the **Compare** button.



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.



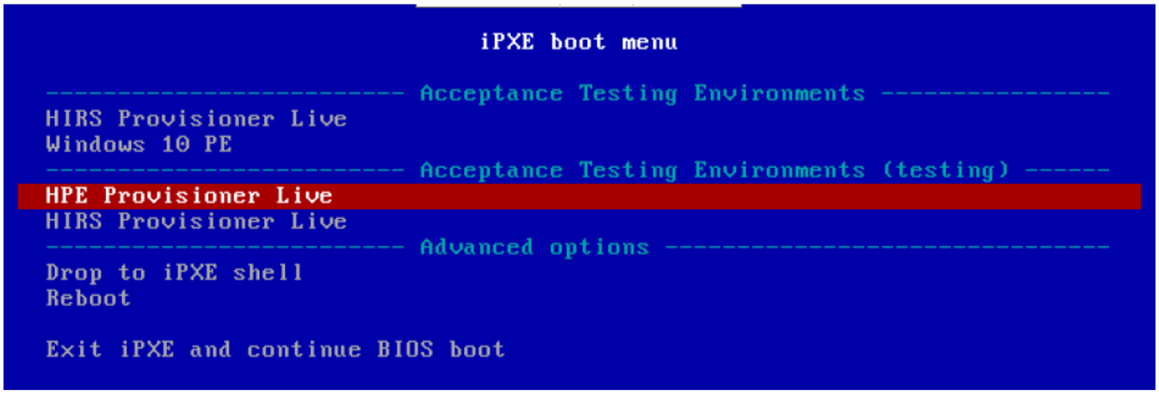
10. Navigate to the location of the platform certificate and select **Open.**



11. If the certificate matches the certificate that the AutoVerify tool detected, the tool will output another green message that reads "*Platform Certificate Matches*."

3.1.1.4 HPE Servers

1. Boot the target HPE server into the CentOS 8 acceptance testing environment via iPXE. This requires a one-time boot execution to prevent the server from loading the native OS. Press F11 in the POST screen after a server reboot to access the one-time boot menu and choose the appropriate network interface card. Then choose *HPE Provisioner Live* from the iPXE boot menu.

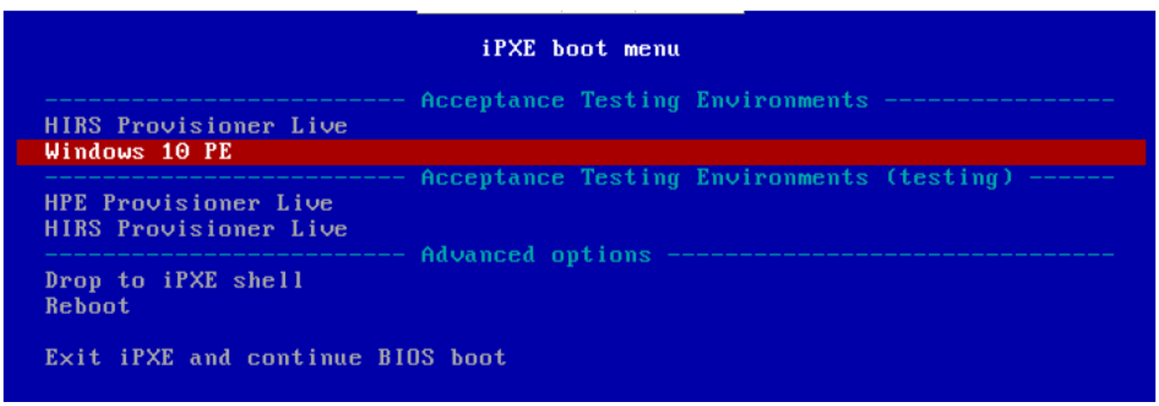


2. Once the live environment has loaded, log in as a user with root privileges. Run the *hpe_provision.sh script*. The script will attempt to execute the PCVT against the verifiable artifacts stored in the image. If successful, the script posts the platform manifest to the PMCS.

   The script will exit when an error is detected. Refer to the comments in the script to set this up in your own environment.

3.1.1.5 Dell Servers

1. Boot the target Dell server into the Windows PE acceptance testing environment via iPXE. This requires a one-time boot execution to prevent the server from loading the native OS. Press F12 in the POST screen after a server reboot to access the one-time PXE boot option and choose the appropriate network interface card. Then choose *Windows 10 PE* from the iPXE boot menu.



2. Once the live environment has loaded, log in as a user with root privileges. Run the *dell-server-scv.ps1* script. The script will attempt to execute the Dell Secured Component Verification (SCV) tool against the verifiable artifacts stored on the server. If successful, the script posts the platform manifest to the PMCS.

The script will exit when an error is detected. Refer to the comments in the script to set this up in your own environment.

3.1.1.6 Intel Server

3. Boot the Intel Server into the CentOS 8 host OS environment. Note that for the demonstration Intel server, a network-booted acceptance testing environment was not implemented.

4. Once the operating system has completed booting, log in as a user with root privileges. Run the *provision.sh* script. The script will attempt to execute the *TSCVerifyUtil* against the verifiable artifacts stored on the server. If successful, the script posts the platform manifest to the PMCS.

   The script will run *TSCVerifyUtil* again with different command arguments which directs the program to access the Seagate drive APIs. If successful, the drive attestation data and measurements are posted to the PMCS.

   The script will exit when an error is detected. Refer to the comments in the script to set this up in your own environment.

3.1.2 Asset Inventory and Discovery

Organizational members with access to the enterprise database of computing devices can access a listing by authenticating to the Archer system. We have configured our instance to display only the relevant Archer solution menus. In Figure 3-1, the administrator clicks the *SCA Devices* menu link to retrieve the listing.

Figure 3-1 Archer Solution Menu

Figure 3-2 shows a listing of all enterprise computing devices that have had their platform validated in accordance with Scenario 2. The computing device *Enterprise Unique Identifier* is hyperlinked and when clicked displays additional data, as described below.

Figure 3-2 Enterprise Computing Devices Listing

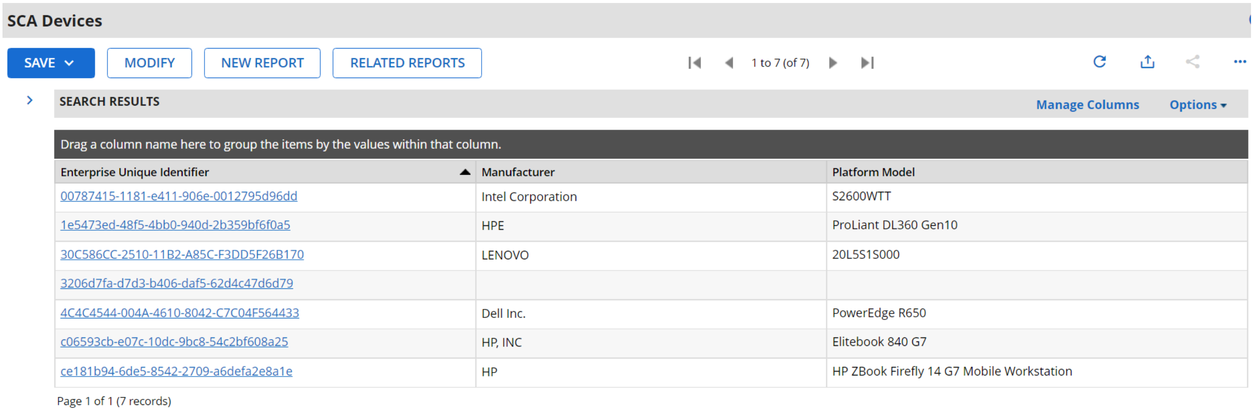
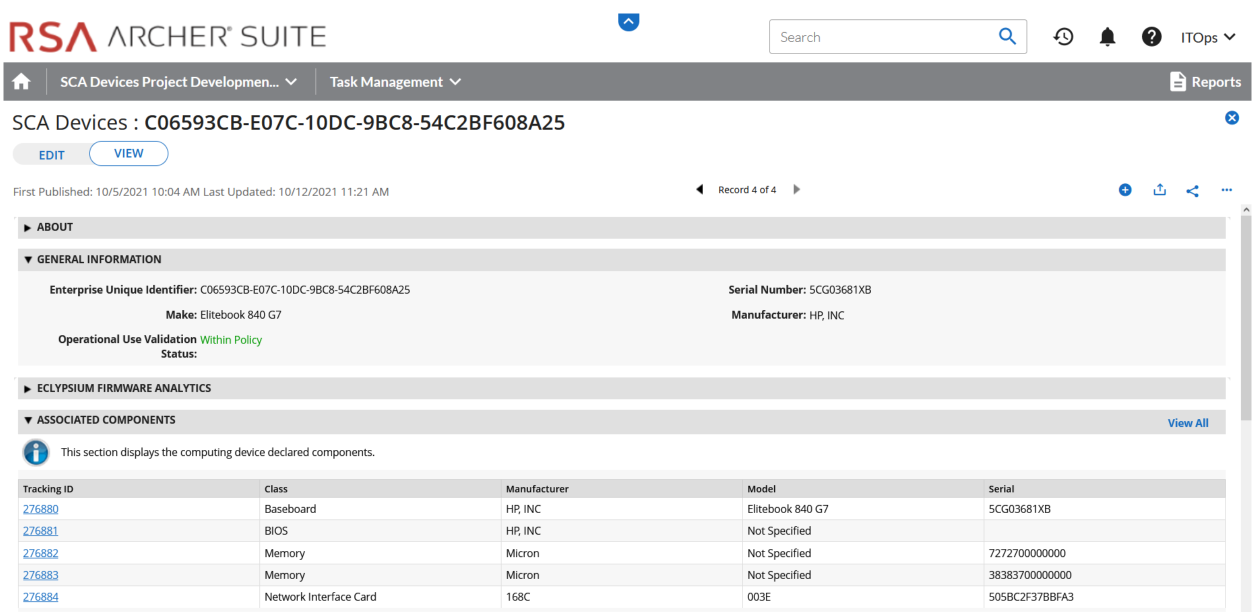


Figure 3-3 shows a representative laptop computing device that has completed the acceptance testing process by an IT administrator. In the **General Information** section, we have opted to display characteristics that are common across all the manufacturers in our project such as the serial number and the make of the computing device. Separately in the **Associated Components** section, we store and track the components from the initial manufacturer manifest. We will continue to iterate on the asset inventory user interface to surface meaningful and easily understandable information that is appropriate for individuals responsible for IT security.

Figure 3-3 Asset Inventory Screenshot



For those computing devices that support Eclypsium during acceptance testing, Archer retrieves the initial firmware data from the Eclypsium backend (cloud or on-premises) and displays it in the Eclypsium Firmware Analytics section of the record as shown in Figure 3-4.

Figure 3-4 Eclypsium Acceptance Testing Firmware Data

▼ ECLYPSIUM FIRMWARE ANALYTICS

Integrity data from the Eclypsium platform.

Last System Scan Date:		**System Firmware Date:**	4/26/2021
Eclypsium Integrity Scan Status:		**System Firmware Version:**	S70 Ver. 01.05.00

3.1.2.1 *Manufacturer-Specific Attributes*

As described in Volume B of this guide, this demonstration also collects manufacturer-specific platform integrity attributes in addition to the agnostic data described above. For HP Inc. laptops, BIOS configuration settings, represented as UEFI variables, are associated with the laptop in the asset inventory when available. From this perspective the security operator is able to view each variable value, description, and the recommended setting for each value. The operator is also alerted if the variable value has changed since the initial baseline (column 2), where a remediation action could be initiated.

HP Inc UEFI Variables

▼ HP UEFI CONFIGURATION VARIABLES (ASSOCIATED COMPUTING DEVICE) View All

UEFI Variable Friendly Name	HP Inc BIOS Configuration Status	Value	UEFI Variable Description	UEFI Variable Possible Values	UEFI Variable Recommended Values
Enhanced HP Firmware Runtime Intrusion Prevention and Detection	No Change Detected	Enable	Utilizes specialized hardware in the platform chipset to prevent, detect, and remediate anomalies in the Runtime HP SMM BIOS.	[Disable, Enable]	Enable
Cover Removal Sensor	No Change Detected	Not found	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.	[Disable, Notify user, Administrator Credential, Administrator Password]	Administrator Credential or Administrator Password

Computing devices that use the Intel Transparent Supply Chain platform declare (if present) additional attributes such as values for the OEM, original design manufacturer (ODM), model, product name, stock keeping unit (SKU), and product family. The screenshot below is an example from a demonstration laptop asset inventory record.

▼ INTEL HARDWARE PROPERTIES

Original Equipment Manufacturer:	LENOVO	**Product Name:**	20L5S1S000
Original Design Manufacturer:	LENOVO	**SKU:**	LENOVO_MT_20L5_BU_Think_FM_ThinkPad T480
Model:		**Family:**	ThinkPad T480

Finally, each Seagate drive asset inventory entry displays associated data from its firmware attestation and measurement capabilities. The security operator can view the currently running version of the firmware and click on the Tracking ID hyperlink for more details associated with the firmware. In the lower section, the Firmware Hash Status column informs the operator if measurement values have 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

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

The general execution steps are as follows:

1. The end user takes ownership of the computing device from the IT department and uses it to perform daily work tasks within the scope of normal duties.
2. The computing device creates a report that attests to the platform attributes, such as device identity, hardware components, and firmware measurements that can be identified by interrogating the platform.
3. The attestation is consumed and validated by existing configuration management systems used by the IT organization as part of a continuous monitoring program.
4. The measured state of the device is maintained and updated as the authorized components of the device are being maintained and associated firmware is updated throughout the device's operational life cycle.
5. Optionally, the IT administrator takes a remediation action against the computing device if it is deemed out of compliance. For example, the computing device could be restricted from accessing certain corporate network resources.

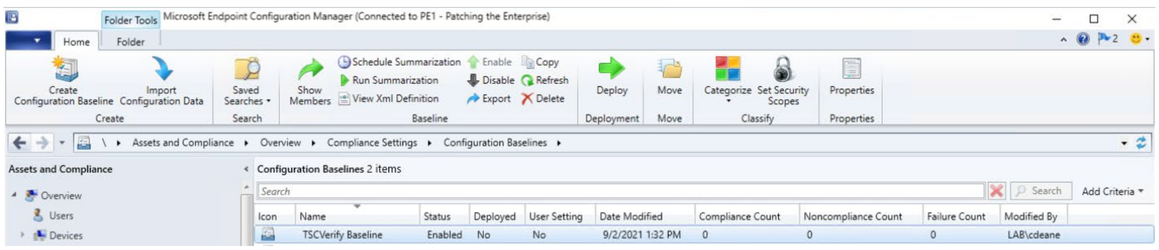
3.2.1 Technology Configurations

3.2.1.1 Monitoring Using Intel and HIRS-ACA Validation Clients

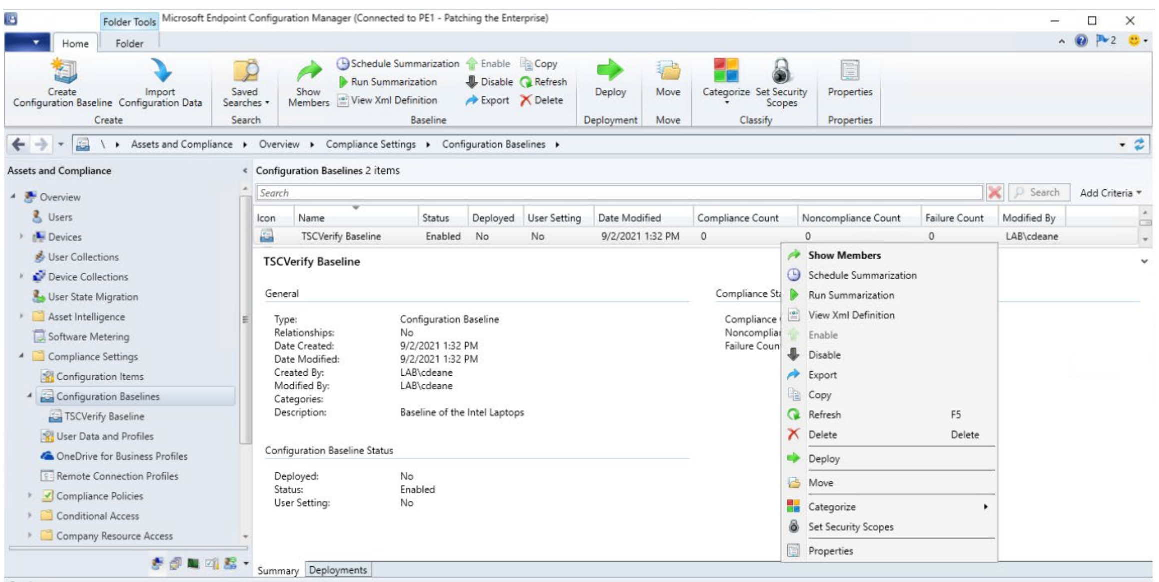
This section describes the steps that monitor for unexpected component changes using Intel TSC/HIRS-ACA tooling and Microsoft Configuration Manager capabilities.

3.2.1.1.1 Deploy Baseline

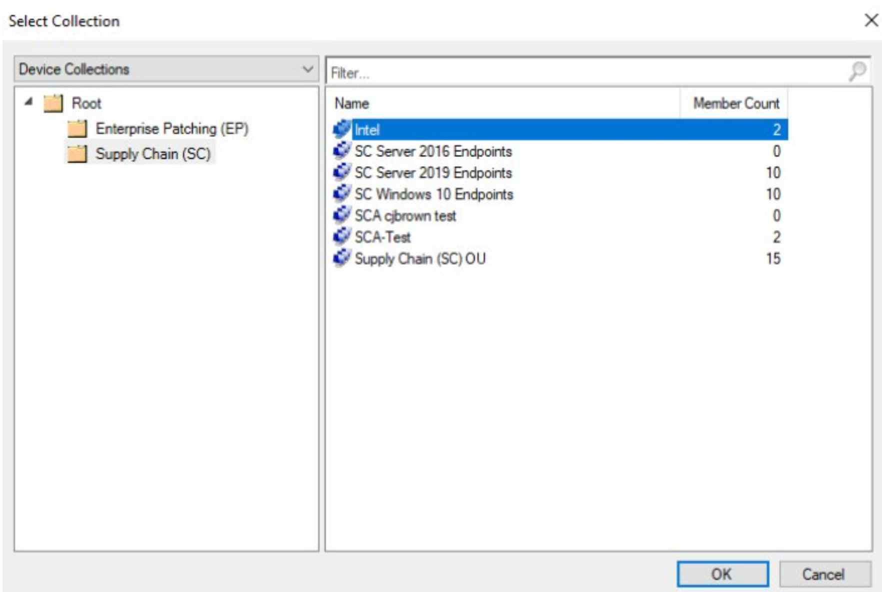
1. Navigate to the newly created configuration baseline located at **Assets and Compliance > Overview > Compliance Settings > Configuration Baselines.**



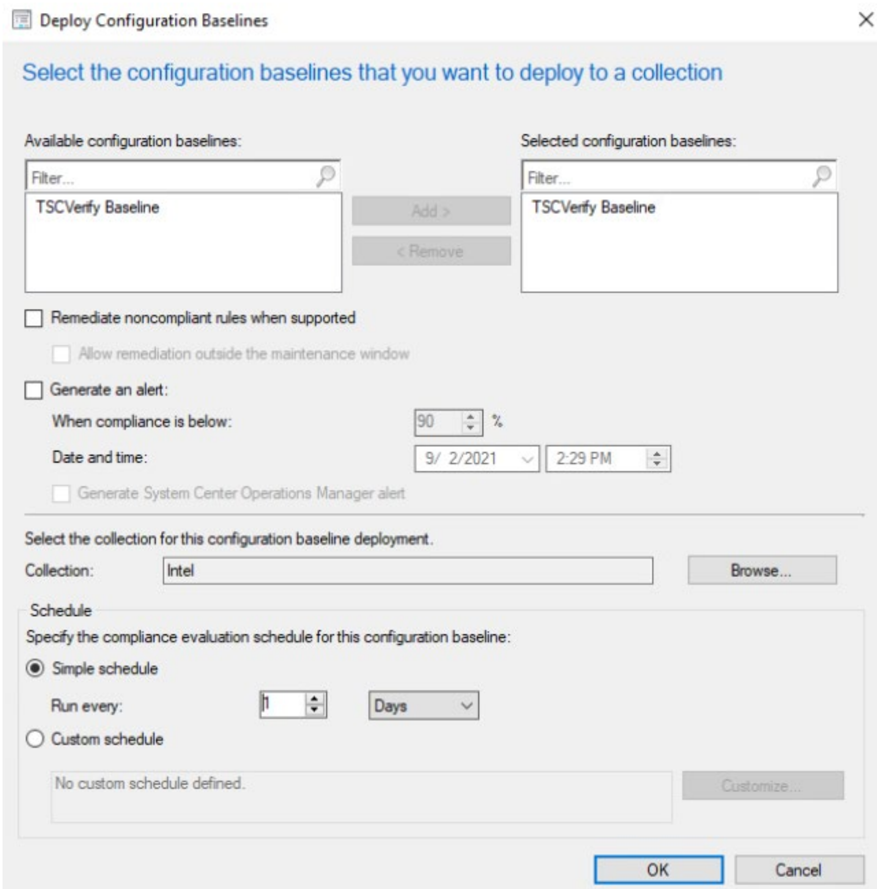
2. Right-click on the configuration baseline and select **Deploy.**



3. Select the device collection for the Intel TSC-supported machines. For this project, the device collection is named **Intel.** Select **OK.**



4. Ensure that the baseline is selected and then select the desired frequency of when to run the baseline. Select **OK.**

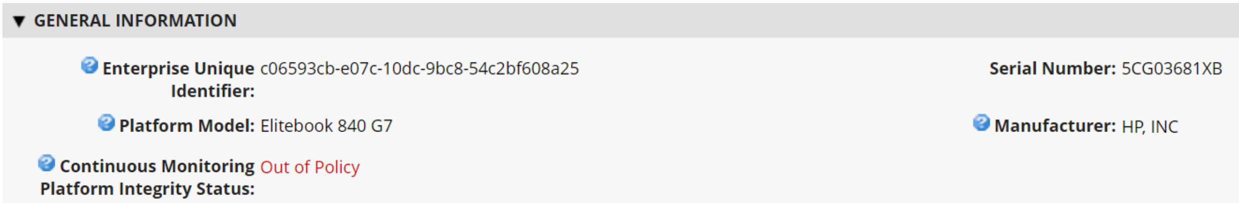


This completes the configuration for Intel TSC platform validation tools. Repeat this section to create a similar baseline for Dell and HP Inc. laptops that leverage HIRS-ACA platform validation tools.

3.2.1.2 Updating the Platform Verifiable Artifact During Operational Use

During the operational use of a computing device, a member of security operations may observe a warning in a computing device's asset record that it is out of compliance. This could indicate that the platform has been updated but the change has not been reflected in the verifiable artifact. Archer will continue to display this warning until the verifiable artifact is updated with the new platform manifest. Figure 3-5 illustrates this scenario.

Figure 3-5 Out of Policy Computing Device



Address the policy warning by using the following procedures to create a Delta Platform Certificate on HP Inc. and Dell laptops which reflects changes in the target platform components. A Delta Platform Certificate can be created in Linux or Windows; however, this demonstration only exercises creation on the Windows platform.

Ensure the following prerequisites are met:

- The administrator has installed PACCOR onto the target laptop.
- A base Platform Certificate has been created and configured in the HIRS ACA. Creation of a Delta Platform Certificate is dependent on the existence of another base Platform Certificate for the same laptop.

Next, complete the following procedures to create a Delta Platform Certificate.

5. Open a command prompt as an Administrator on the target laptop. Change directories to the following:

   ```
   <paccor install folder>\scripts\windows
   ```

6. Create a directory named *pc_testgen* in the working directory from the previous step if it does not already exist.
7. Retrieve the base Platform Certificate from the HIRS ACA portal or other means. Change the filename of the Platform Certificate to *holder.crt* and place it into the *pc_testgen* directory.
8. Execute PACCOR's component gathering script and capture the output with the following command.

   ```
   powershell -ExecutionPolicy Bypass ./allcomponents.ps1 components.json
   ```

9. The component list needs to be manually edited to reflect added, modified, or removed components of the system. Using a JSON file editor, open the *components.json* file.
   a. In the **COMPONENTS** object, identify the objects that represent components to be saved in the new Delta Platform Certificate. Add a **STATUS** field at the end of these components with a value of **ADDED, MODIFIED,** or **REMOVED**. For example, to modify the chassis serial number, create a **COMPONENTS** entry similar to the following.

```
{
  "COMPONENTS": [
    {
      "COMPONENTCLASS": {
        "COMPONENTCLASSREGISTRY": "2.23.133.18.3.1",
        "COMPONENTCLASSVALUE": "00020001"
      },
      "MANUFACTURER": "Example Manufacturer",
      "MODEL": "1",
      "SERIAL": "1234",
      "STATUS": "MODIFIED"
    }
  ]
}
```

   b. Delete all other objects under **COMPONENTS.**

   c. Once finished editing the *components.json* file, move it to the **pc_testgen** folder.

10. Using a text editor, modify the **pc_certgen** script header variables.

    a. Set the **ekcert** variable to point to **holder.crt** from step 3.

    b. Set the **componentlist** variable to point to **components.json** from step 5.

    c. Change the value of **serialnumber** to 0002.

    d. If you have a specific signing key and cert, move those files to **pc_testgen** as well and update the **sigkey** and **pcsigncert** variables to point to them.

11. Execute the *pc_certgen.ps1* script using the following command:

    ```
    powershell -ExecutionPolicy Bypass ./pc_certgen.ps1
    ```

12. The resulting Delta Platform Certificate will be stored in the **pc_testgen** folder.

13. Upload the new Delta Platform Certificate to the HIRS-ACA portal.

Note that laptops that are continuously monitored with the Windows-based HIRS Provisioner will be evaluated against this new baseline.

3.2.2 Dashboards

The dashboard created in Section 2.11.2.3 attempts to consolidate and communicate potential integrity issues to the IT administrator while computing devices are in operational use. The timeliness of this information will depend on the cadence that your organization chooses to update the various data feeds from Microsoft Configuration Manager and the Eclypsium Analytic platform. This demonstration displays to the administrator if there are detected component swaps from computing devices that can leverage

Intel TSC and HIRS-ACA platforms. Further, it displays any detected firmware platform integrity issues from the Eclypsium Analytic cloud and on-premises platform across all manufacturers in this prototype.

The Archer IRM dashboard should resemble the screenshots below, where a count of computing devices with potential integrity issues is displayed (Figure 3-6 and Figure 3-7). Your organization's security operations team may also want to access the Eclypsium Analytic platform directly to obtain detailed information, including remediation actions, for computing devices with detected integrity issues.

Figure 3-6 Dashboard with No Integrity Issues Detected

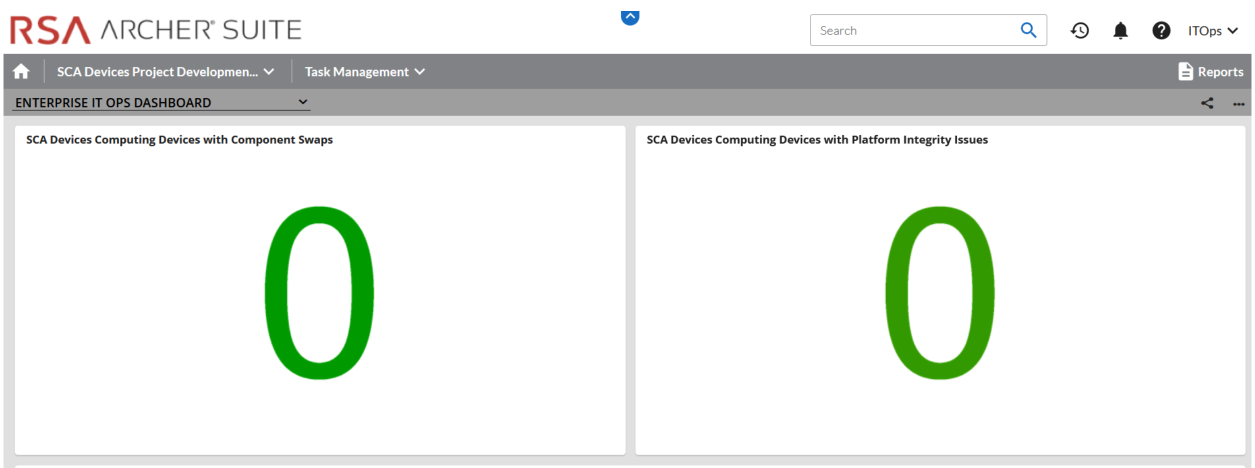
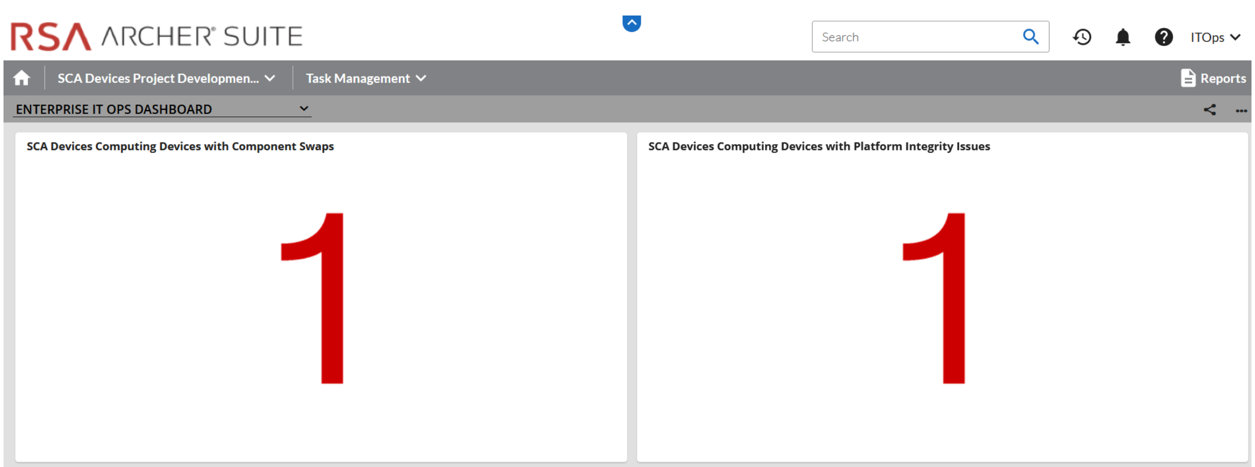
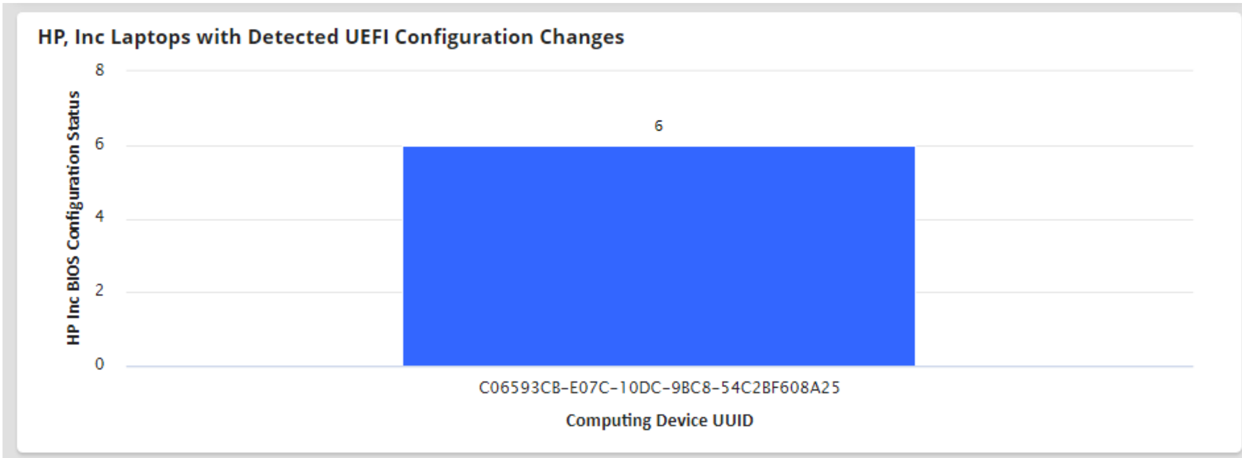


Figure 3-7 Dashboard with Integrity Issues Detected



The demonstration dashboards are also capable of monitoring manufacturer-specific platform integrity datapoints. In Figure 3-8, we show a dashboard component that captures the number of UEFI configuration parameters that have changed from the baseline values (Y-axis) for each HP Inc. computing device (X-axis).

Figure 3-8 HP Inc. Laptop Continuous Monitoring

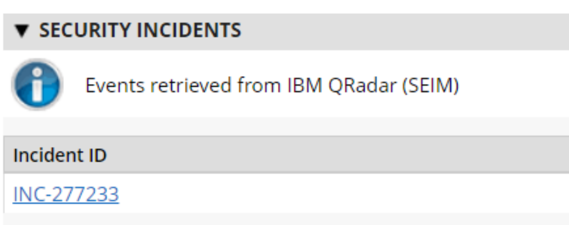


In the final dashboard component, the security operator can display the number of Seagate drives with firmware hash values that have changed since the initial acceptance testing baseline. In a production setting, it could be more useful to compare the drive measurements against known values communicated directly from the manufacturer (Seagate).

3.2.3 Platform Integrity Incident Management

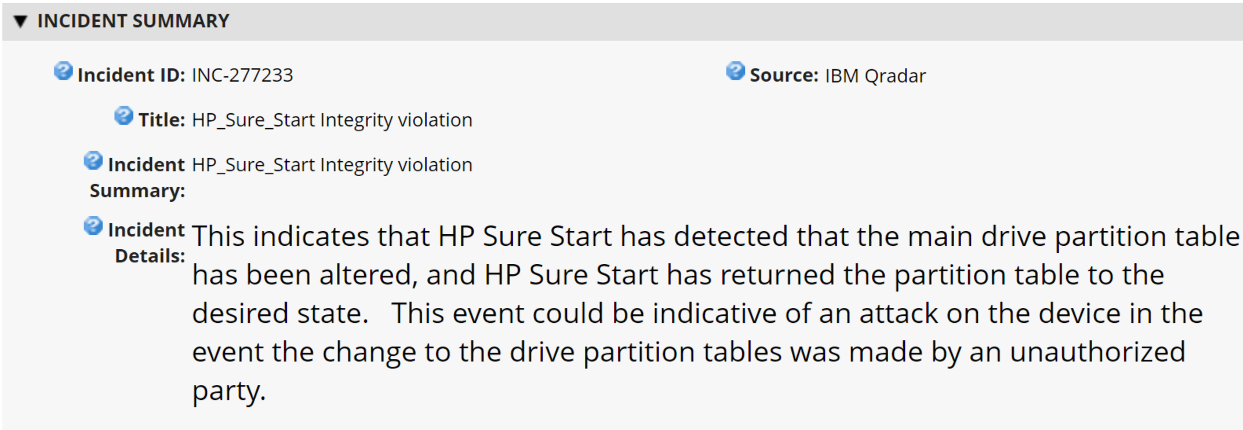
The final continuous monitoring scenario we demonstrate is the automated creation of Archer *Incidents* when the QRadar continuous monitoring data feed (Section 2.11.2.2.4) retrieves a platform integrity issue. In the asset inventory record shown in Figure 3-9, we have triggered a platform integrity issue in one of our demonstration HP Inc. laptops, which has automatically created an Archer *Security Incident*. Note that the Archer platform offers workflow customization options that are not documented here that can support more complex organizational structures.

Figure 3-9 New Security Incident



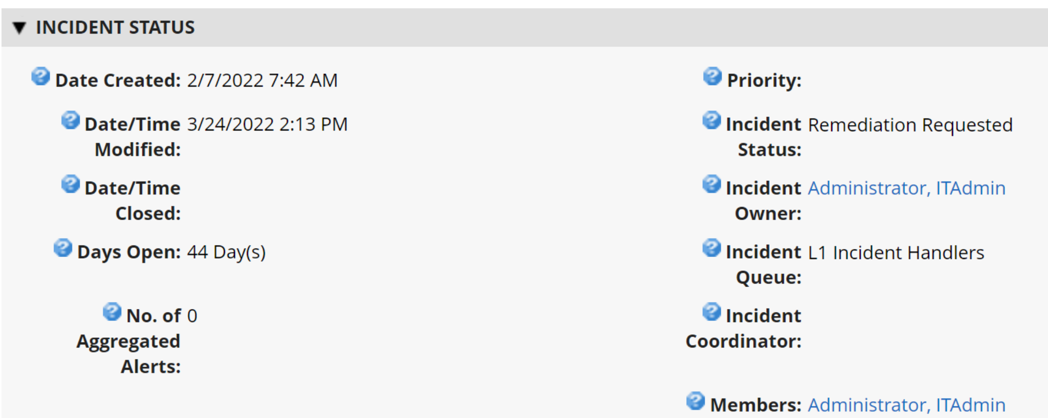
The security operator can click the hyperlink, which displays more detailed information about the issue. In the case depicted in Figure 3-10, the *HP Sure Start* capability has flagged a potential issue.

Figure 3-10 Incident Summary



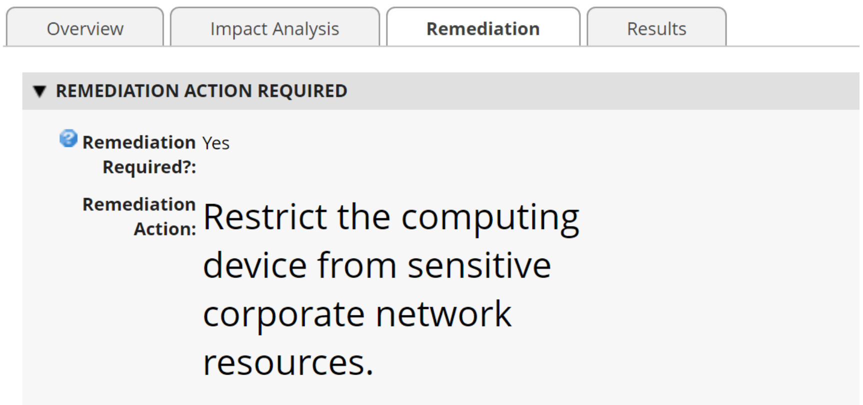
In the *Incident Status* section, metadata associated with the incident is displayed, including whether remediation is requested by the security operator. Figure 3-11 shown an example of this.

Figure 3-11 Incident Status



If remediation is requested, the security operator clicks the *Remediation* tab within the *Security Incident* where a suggested action is displayed (see Figure 3-12).

1586 **Figure 3-12 Incident Remediation Action**



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

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 Section 2.11.2.1. 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"), 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"), IF ([Firmware Integrity Check Status]=1, VALUEOF([Eclypsium Integrity Scan Status], "No Integrity Issues Detected"), IF ([Firmware Integrity Check Status]=0, VALUEOF([Eclypsium Integrity Scan Status], "Integrity Issue Detected - Action Recommended"))))`
3	`IF (ISEMPTY([Continuous Monitoring Platform Integrity Status]), 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

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	

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", 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.",
	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.",
	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.",
	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.",
	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.",
	IF ([UEFI Variable Name]="DMA_Protection", "BIOS will configure IOMMU hardware for use by operating systems that support DMA protection.",
	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.",
	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.",
	IF ([UEFI Variable Name]="", "No Description", "No Description")
	)))))))))

4	IF ([UEFI Variable Name]="SS_SB_KeyProt", "[Disable, Enable]",
	IF ([UEFI Variable Name]="FW_RIPD", "[Disable, Enable]",
	IF ([UEFI Variable Name]="TL_Power_Off", "[Disable, Enable]",
	IF ([UEFI Variable Name]="TL_Clear_TPM", "[Disable, Enable]",
	IF ([UEFI Variable Name]="SS_GPT_HDD", "[Disable, Enable]",
	IF ([UEFI Variable Name]="SS_GPT_Policy", "[Local user control, Recover in event of corruption]",
	IF ([UEFI Variable Name]="DMA_Protection", "[Disabled, Enabled]",
	IF ([UEFI Variable Name]="PreBoot_DMA", "[Thunderbolt Only, All PCI-e Devices]",
	IF ([UEFI Variable Name]="Cover_Sensor", "[Disable, Notify user, Administrator Credential, Administrator Password]",
	IF ([UEFI Variable Name]="", "No Possible Values", "No Possible Values")
	)))))))))

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

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**

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**	

Table 3-8 Calculated Fields (Seagate Firmware Hash)

Index	Calculation
1	`IF ([First Published]<>[Last Updated], "Change Detected",` `IF ([First Published]=[Last Updated], "No Change Detected"))`