Security Automation and Continuous Monitoring WG D. Waltermire
Internet-Draft NIST
Intended status: Informational D. Harrington
Expires: April 22, 2014 Effective Software
October 19, 2013

Endpoint Security Posture Assessment - Enterprise Use Cases
draft-ietf-sacm-use-cases-03

Abstract

This memo documents a sampling of use cases for securely aggregating configuration and operational data and evaluating that data to determine an organization's security posture. From these operational use cases, we can derive common functional capabilities and requirements to guide development of vendor-neutral, interoperable standards for aggregating and evaluating data relevant to security posture.

Table of Contents

1. Introduction

Our goal with this document is to improve our agreement on which problems we're trying to solve. We need to start with short, simple problem statements and discuss those by email and in person. Once we agree on which problems we're trying to solve, we can move on to propose various solutions and decide which ones to use.

This document describes example use cases for endpoint posture assessment for enterprises. It provides a sampling of use cases for securely aggregating configuration and operational data and evaluating that data to determine the security posture of individual endpoints, and, in the aggregate, the security posture of an enterprise.

These use cases cross many IT security information domains. From these operational use cases, we can derive common concepts, common information expressions, functional capabilities and requirements to guide development of vendor-neutral, interoperable standards for aggregating and evaluating data relevant to security posture.

Using this standard data, tools can analyze the state of endpoints, user activities and behaviour, and evaluate the security posture of an organization. Common expression of information should enable interoperability between tools (whether customized, commercial, or freely available), and the ability to automate portions of security processes to gain efficiency, react to new threats in a timely manner, and free up security personnel to work on more advanced problems.

The goal is to enable organizations to make informed decisions that support organizational objectives, to enforce policies for hardening systems, to prevent network misuse, to quantify business risk, and to collaborate with partners to identify and mitigate threats.

It is expected that use cases for enterprises and for service providers will largely overlap, but there are additional complications for service providers, especially in handling information that crosses administrative domains.

The output of endpoint posture assessment is expected to feed into additional processes, such as policy-based enforcement of acceptable state, verification and monitoring of security controls, and compliance to regulatory requirements.

2. Endpoint Posture Assessment

Endpoint posture assessment involves orchestrating and performing data collection and evaluating the posture of a given endpoint. Typically, endpoint posture information is gathered and then published to appropriate data repositories to make collected information available for further analysis supporting organizational security processes.

Endpoint posture assessment typically includes:

- Collecting the attributes of a given endpoint;
- Making the attributes available for evaluation and action; and
- Verifying that the endpoint's posture is in compliance with enterprise standards and policy.

As part of these activities it is often necessary to identify and acquire any supporting content that is needed to drive data collection and analysis.

The following is a typical workflow scenario for assessing endpoint posture:

1. Some type of trigger initiates the workflow. For example, an operator or an application might trigger the process with a request, or the endpoint might trigger the process using an event-driven notification.

   QUESTION: Since this is about security automation, can we drop the User and just use Application? Is there a better term to use here? Once the policy is selected, the rest seems like something we definitely would want to automate, so I dropped the User part.

2. A user/application selects a target endpoint to be assessed.

3. A user/application selects which policies are applicable to the target.

4. The application determines which (sets of) posture attributes need to be collected for evaluation.

   QUESTION: It was suggested that mentioning several common acquisition methods, such as local API, WMI, Puppet, DCOM, SNMP, CMDB query, and NEA, without forcing any specific method would be good. I have concerns this could devolve into a "what about my favorite?" contest. OTOH, the charter does specifically call for use of existing standards where

applicable, so the use cases document might be a good neutral location for such information, and might force us to consider what types of external interfaces we might need to support when we consider the requirements. It appears that the generic workflow sequence would be a good place to mention such common acquisition methods.

5. The application might retrieve previously collected information from a cache or data store, such as a data store populated by an asset management system.

6. The application might establish communication with the target, mutually authenticate identities and authorizations, and collect posture attributes from the target.

7. The application might establish communication with one or more intermediary/agents, mutually authenticate their identities and determine authorizations, and collect posture attributes about the target from the intermediary/agents. Such agents might be local or external.

8. The application communicates target identity and (sets of) collected attributes to an evaluator, possibly an external process or external system.

9. The evaluator compares the collected posture attributes with expected values as expressed in policies.

QUESTION: Evaluator generates a report or log or notification of some type?

The following subsections detail specific use cases for data collection, analysis, and related operations pertaining to the publication and use of supporting content.

2.1. Definition and Publication of Automatable Configuration Guides

A vendor manufactures a number of specialized endpoint devices. They also develop and maintain an operating system for these devices that enables end-user organizations to configure a number of security and operational settings. As part of their customer support activities, they publish a number of secure configuration guides that provide minimum security guidelines for configuring their devices.

Each guide they produce applies to a specific model of device and version of the operating system and provides a number of specialized configurations depending on the devices intended function and what add-on hardware modules and software licenses are installed on the

device. To enable their customers to evaluate the security posture of their devices to ensure that all appropriate minimal security settings are enabled, they publish an automatable configuration checklist using a popular data format that defines what settings to collect using a network management protocol and appropriate values for each setting. They publish these guides to a public content repository that customers can query to retrieve applicable guides for their deployed enterprise network infrastructure endpoints.

Guides could also come from sources other than a device vendor, such as industry groups or regulatory authorities, or enterprises could develop their own checklists.

2.2. Automated Checklist Verification

A financial services company operates a heterogeneous IT environment. In support of their risk management program, they utilize vendor provided automatable security configuration checklists for each operating system and application used within their IT environment. Multiple checklists are used from different vendors to insure adequate coverage of all IT assets.

To identify what checklists are needed, they use automation to gather an inventory of the software versions utilized by all IT assets in the enterprise. This data gathering will involve querying existing data stores of previously collected endpoint software inventory posture data and actively collecting data from reachable endpoints as needed utilizing network and systems management protocols. Previously collected data may be provided by periodic data collection, network connection-driven data collection, or ongoing event-driven monitoring of endpoint posture changes.

Using the gathered software inventory data and associated asset management data indicating the organizational defined functions of each endpoint, they locate and query each vendors content repository for the appropriate checklists. These checklists are cached locally to reduce the need to download the checklist multiple times.

Driven by the setting data provided in the checklist, a combination of existing configuration data stores and data collection methods are used to gather the appropriate posture information from each endpoint. Specific data is gathered based on the defined enterprise function and software inventory of each endpoint. The data collection paths used to collect software inventory posture will be used again for this purpose. Once the data is gathered, the actual state is evaluated against the expected state criteria in each applicable checklist. Deficiencies are identified and reported to the appropriate endpoint operators for remedy.

Checklists could also come from sources other than the application or OS vendor, such as industry groups or regulatory authorities, or enterprises could develop their own checklists.

2.3. Organizational Software Policy Compliance

Example Corporation, in support of compliance requirements, has identified a number of secure baselines for different endpoint types that exist across their enterprise IT environment. Determining which baseline applies to a given endpoint is based on the organizationally defined function of the device.

Each baseline, defined using an automatable standardized data format, identifies the expected hardware, software and patch inventory, and software configuration item values for each endpoint type. As part of their compliance activities, they require that all endpoints connecting to their network meet the appropriate baselines. The configuration settings of each endpoint are collected and compared to the baseline to make sure the configuration complies with the appropriate baseline whenever it connects to the network and at least once a day thereafter. These daily compliance checks evaluate the posture of each endpoint and report on its compliance with the appropriate baseline.

[TODO: Need to speak to how the baselines are identified for a given endpoint connecting to the network.]

2.4. Detection of Posture Deviations

Example corporation has established secure configuration baselines for each different type of endpoint within their enterprise including: network infrastructure, mobile, client, and server computing platforms. These baselines define an approved list of hardware, software (i.e., operating system, applications, and patches), and associated required configurations. When an endpoint connects to the network, the appropriate baseline configuration is communicated to the endpoint based on its location in the network, the expected function of the device, and other asset management data. It is checked for compliance with the baseline indicating any deviations to the device's operators. Once the baseline has been established, the endpoint is monitored for any change events pertaining to the baseline on an ongoing basis. When a change occurs to posture defined in the baseline, updated posture information is exchanged allowing operators to be notified and/or automated action to be taken.

2.5. Search for Signs of Infection

The Example Corporation carefully manages endpoint security with tools that implement the SACM standards. One day, the endpoint security team at Example Corporation learns about a stealthy malware package. This malware has just been discovered but has already spread widely around the world. Certain signs of infection have been identified (e.g. the presence of certain files). The security team would like to know which endpoints owned by the Example Corporation have been infected with this malware. They use their tools to search for the signs of infection and generate a list of infected endpoints.

The search for infected endpoints may be performed by gathering new endpoint posture information regarding the presence of the signs of infection. However, this might miss finding endpoints that were previously infected but where the infection has now erased itself. Such previously infected endpoints may be detected by searching a database of posture information previously gathered for the signs of infection. However, this will not work if the malware hides its presence carefully or if the signs of infection were not included in previous posture assessments. In those cases, the database may be used to at least detect which endpoints previously had software vulnerable to infection by the malware.

2.6. Remediation and Mitigation

When Example Corporation discovers that one of its endpoints is vulnerable to infection, a process of mitigation and remediation is triggered. The first step is mitigating the impact of the vulnerability, perhaps by placing the endpoint into a safe network or blocking network traffic that could infect the endpoint. The second step is remediation: fixing the vulnerability. In some cases, these steps may happen automatically and rapidly. In other cases, they may require human intervention either to decide what response is most appropriate or to complete the steps, which are sometimes complex.

These same steps of mitigation and remediation may be used when Example Corporation discovers that one of its endpoints has become infected with some malware. Alternatively, the infected endpoint may simply be monitored or even placed into a honeynet or similar environment to observe the malware's behavior and lead the attackers astray.

QUESTION: Is remediation and mitigation within the scope of the WG, and should the use case be included here?

2.7. Endpoint Information Analysis and Reporting

Freed from the drudgery of manual endpoint compliance monitoring, one of the security administrators at Example Corporation notices (not

using SACM standards) that five endpoints have been uploading lots of data to a suspicious server on the Internet. The administrator queries the SACM database of endpoint posture to see what software is installed on those endpoints and finds that they all have a particular program installed. She then searches the database to see which other endpoints have that program installed. All these endpoints are monitored carefully (not using SACM standards), which allows the administrator to detect that the other endpoints are also infected.

This is just one example of the useful analysis that a skilled analyst can do using the database of endpoint posture that SACM can provide.

2.8. Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra

A university team receives a grant to do research at a government facility in the arctic. The only network communications will be via an intermittent low-speed high-latency high-cost satellite link. During their extended expedition they will need to show continue compliance with the security policies of the university, the government, and the provider of the satellite network as well as keep current on vulnerability testing. Interactive assessments are therefore not reliable, and since the researchers have very limited funding they need to minimize how much money they spend on network data.

Prior to departure they register all equipment with an asset management system owned by the university, which will also initiate and track assessments.

On a periodic basis -- either after a maximum time delta or when the content repository has received a threshold level of new vulnerability definitions -- the university uses the information in the asset management system to put together a collection request for all of the deployed assets that encompasses the minimal set of artifacts necessary to evaluate all three security policies as well as vulnerability testing.

In the case of new critical vulnerabilities this collection request consists only of the artifacts necessary for those vulnerabilities and collection is only initiated for those assets that could potentially have a new vulnerability.

[Optional] Asset artifacts are cached in a local CMDB. When new vulnerabilities are reported to the content repository, a request to the live asset is only done if the artifacts in the CMDB are incomplete and/or not current enough.

The collection request is queued for the next window of connectivity. The deployed assets eventually receive the request, fulfill it, and queue the results for the next return opportunity.

The collected artifacts eventually make it back to the university where the level of compliance and vulnerability expose is calculated and asset characteristics are compared to what is in the asset management system for accuracy and completeness.

2.9. Vulnerable Endpoint Identification

Typically vulnerability reports identify an executable or library that is vulnerable, or worst case the software that is vulnerable. This information is used to determine if an organization has one or more endpoints that have exposure to a vulnerability (i.e., what endpoints are vulnerable?). It is often necessary to know where you are running vulnerable code and what configurations are in place on the endpoint and upstream devices (e.g., IDS, firewall) that may limit the exposure. All of this information, along with details on the severity and impact of a vulnerability, is necessary to prioritize remedies.

2.10. Compromised Endpoint Identification

Along with knowing if one or more endpoints are vulnerable, it is also important to know if you have been compromised. Indicators of compromise provide details that can be used to identify malware (e.g., file hashes), identify malicious activity (e.g. command and control traffic), presence of unauthorized/malicious configuration items, and other indicators. While important, this goes beyond determining organizational exposure.

2.11. Suspicious Endpoint Behavior

This Use Case describes the collaboration between specific participants in an information security system specific to detecting a connection attempt to a known-bad Internet host by a botnet zombie that has made its way onto an organization's Information Technology systems. The primary human actor is the Security Operations Center Analyst, and the primary software actor is the configuration assessment tool. Note, however, the dependencies on other tools, such as asset management, intrusion detection, and messaging.

2.12. Traditional endpoint assessment with stored results

An external trigger initiates an assessment of an endpoint. The Controller uses the data in the Datastore to look up authentication information for the endpoint and passes that along with the assessment request details to the Evaluator. The Evaluator uses the Endpoint information to request taxonomy information from the Collector on the endpoint, which responds with those attributes. The Evaluator uses that taxonomy information along with the information in the original request from the Controller to request the appropriate content from the Content Repository. The Evaluator uses the content to derive the minimal set of endpoint attributes needed to perform the assessment and makes that request. The Evaluator uses the Collector response to do the assessment and returns the results to the Controller. The Controller puts the results in the Datastore.

2.13. NAC/NAP connection with no stored results using an endpoint evaluator

A mobile endpoint makes a VPN connection request. The NAC/NAP broker requests the results of the VPN connection assessment from the Controller. The Controller requests the VPN attributes from a Content Repository. The Controller requests an evaluation of the collected attributes from the Evaluator on the endpoint. The endpoint performs the assessment and returns the results. The Controller completes the original assessment request by returning the results to the NAC/NAP broker, which uses them to set the level of network access allowed to the endpoint.

QUESTION: I edited these from Gunnar's email of 9/11, to try to reduce the use of "assessment", to focus on collection and evaluation, and deal with use cases rather than architecture. I am not sure I got all the concepts properly identified.

2.14. NAC/NAP connection with no stored results using a third-party evaluator

A mobile endpoint makes a VPN connection request. The NAC/NAP broker requests the results of the VPN connection assessment from the Controller. The Controller requests the VPN attributes from a Content Repository. The Controller requests an evaluation of the collected attributes from an Evaluator in the network (rather than trusting an evaluator on the endpoint). The evaluator performs the evaluation and returns the results. The Controller completes the original assessment request by returning the results to the NAC/NAP broker, which uses them to set the level of network access allowed to the endpoint.

QUESTION: I edited these from Gunnar's email of 9/11, to try to reduce the use of "assessment", to focus on collection and evaluation, and deal with use cases rather than architecture. I am not sure I got all the concepts properly identified.

2.15. Repository Interaction

Additional use cases will be identified as we work through other domains.

2.16. Others...

Additional use cases will be identified as we work through other domains.

3. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

This memo documents, for Informational purposes, use cases for security automation. While it is about security, it does not affect security.

5. Acknowledgements

The National Institute of Standards and Technology (NIST) and/or the MITRE Corporation have developed specifications under the general term "Security Automation" including languages, protocols, enumerations, and metrics.

Adam Montville edited early versions of this draft.

Kathleen Moriarty and Stephen Hanna contributed text describing the scope of the document.

Steve Hanna provided use cases for Search for Signs of Infection, Remediation and Mitigation, and Endpoint Information Analysis and Reporting.

Gunnar Engelbach provided the use case about Ice Station Zebra.

6. Change Log

6.1. -02- to -03-

Expanded the workflow description based on ML input.

Changed the ambiguous "assess" to better separate data collection from evaluation.

Added use case for Search for Signs of Infection.

Added use case for Remediation and Mitigation.

Added use case for Endpoint Information Analysis and Reporting.

Added use case for Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra.

Added use case for Traditional endpoint assessment with stored results.

Added use case for NAC/NAP connection with no stored results using an endpoint evaluator.

Added use case for NAC/NAP connection with no stored results using a third-party evaluator.

Added use case for Compromised Endpoint Identification.

Added use case for Suspicious Endpoint Behavior.

Added use case for Vulnerable Endpoint Identification.

Updated Acknowledgements

6.2. -01- to -02-

Changed title

removed section 4, expecting it will be moved into the requirements document.

removed the list of proposed caabilities from section 3.1

Added empty sections for Search for Signs of Infection, Remediation and Mitigation, and Endpoint Information Analysis and Reporting.

Removed Requirements Language section and rfc2119 reference.

Removed unused references (which ended up being all references).

6.3. -00- to -01-

- Work on this revision has been focused on document content relating primarily to use of asset management data and functions.

- Made significant updates to section 3 including:

  * Reworked introductory text.

  * Replaced the single example with multiple use cases that focus on more discrete uses of asset management data to support hardware and software inventory, and configuration management use cases.

  * For one of the use cases, added mapping to functional capabilities used. If popular, this will be added to the other use cases as well.

  * Additional use cases will be added in the next revision capturing additional discussion from the list.

- Made significant updates to section 4 including:

  * Renamed the section heading from "Use Cases" to "Functional Capabilities" since use cases are covered in section 3. This section now extrapolates specific functions that are needed to support the use cases.

  * Started work to flatten the section, moving select subsections up from under asset management.

  * Removed the subsections for: Asset Discovery, Endpoint Components and Asset Composition, Asset Resources, and Asset Life Cycle.

  * Renamed the subsection "Asset Representation Reconciliation" to "Deconfliction of Asset Identities".

  * Expanded the subsections for: Asset Identification, Asset Characterization, and Deconfliction of Asset Identities.

  * Added a new subsection for Asset Targeting.

  * Moved remaining sections to "Other Unedited Content" for future updating.

6.4. draft-waltermire-sacm-use-cases-05 to draft-ietf-sacm-use-cases-00

- Transitioned from individual I/D to WG I/D based on WG consensus call.
- Fixed a number of spelling errors. Thank you Erik!
- Added keywords to the front matter.
- Removed the terminology section from the draft. Terms have been moved to: draft-dbh-sacm-terminology-00
- Removed requirements to be moved into a new I/D.
- Extracted the functionality from the examples and made the examples less prominent.
- Renamed "Functional Capabilities and Requirements" section to "Use Cases".
  - Reorganized the "Asset Management" sub-section. Added new text throughout.
    - Renamed a few sub-section headings.
    - Added text to the "Asset Characterization" sub-section.
- Renamed "Security Configuration Management" to "Endpoint Configuration Management". Not sure if the "security" distinction is important.
  - Added new sections, partially integrated existing content.
  - Additional text is needed in all of the sub-sections.
- Changed "Security Change Management" to "Endpoint Posture Change Management". Added new skeletal outline sections for future updates.

6.5. waltermire -04- to -05-

- Are we including user activities and behavior in the scope of this work? That seems to be layer 8 stuff, appropriate to an IDS/IPS application, not Internet stuff.
- I removed the references to what the WG will do because this belongs in the charter, not the (potentially long-lived) use cases document. I removed mention of charter objectives because the

charter may go through multiple iterations over time; there is a website for hosting the charter; this document is not the correct place for that discussion.

- I moved the discussion of NIST specifications to the acknowledgements section.
- Removed the portion of the introduction that describes the chapters; we have a table of concepts, and the existing text seemed redundant.
- Removed marketing claims, to focus on technical concepts and technical analysis, that would enable subsequent engineering effort.
- Removed (commented out in XML) UC2 and UC3, and eliminated some text that referred to these use cases.
- Modified IANA and Security Consideration sections.
- Moved Terms to the front, so we can use them in the subsequent text.
- Removed the "Key Concepts" section, since the concepts of ORM and IRM were not otherwise mentioned in the document. This would seem more appropriate to the arch doc rather than use cases.
- Removed role=editor from David Waltermire's info, since there are three editors on the document. The editor is most important when one person writes the document that represents the work of multiple people. When there are three editors, this role marking isn't necessary.
- Modified text to describe that this was specific to enterprises, and that it was expected to overlap with service provider use cases, and described the context of this scoped work within a larger context of policy enforcement, and verification.
- The document had asset management, but the charter mentioned asset, change, configuration, and vulnerability management, so I added sections for each of those categories.
- Added text to Introduction explaining goal of the document.
- Added sections on various example use cases for asset management, config management, change management, and vulnerability management.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

Authors' Addresses

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Email: david.waltermire@nist.gov

David Harrington
Effective Software
50 Harding Rd
Portsmouth, NH 03801
USA

Email: ietfdbh@comcast.net