Security Automation and Continuous Monitoring WG
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

D. Waltermire
NIST
A. Montville
TW
D. Harrington
Effective Software

July 15, 2013

Using Security Posture Assessment to Grant Access to Enterprise Network Resources

draft-waltermire-sacm-use-cases-05

Abstract

This memo documents a sampling of use cases for securely aggregating configuration and operational data and assessing 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 assessing data relevant to security posture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

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 assessing 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 assessing data relevant to security posture.

Using this standard data, tools can analyse the state of endpoints, user activities and behaviour, and assess 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. Terms and Definitions

assessment

Defined in [RFC5209] as "the process of collecting posture for a set of capabilities on the endpoint (e.g., host-based firewall) such that the appropriate validators may evaluate the posture against compliance policy."

Within this document the use of the term is expanded to support other uses of collected posture (e.g. reporting, network enforcement, vulnerability detection, license management). The phrase "set of capabilities on the endpoint" includes: hardware and software installed on the endpoint."

asset

Defined in [RFC4949] as "a system resource that is (a) required to be protected by an information system's security policy, (b) intended to be protect by a countermeasure, or (c) required for a system's mission.

attribute

Defined in [RFC5209] as "data element including any requisite meta-data describing an observed, expected, or the operational status of an endpoint feature (e.g., anti-virus software is currently in use)."

endpoint

Defined in [RFC5209] as "any computing device that can be connected to a network. Such devices normally are associated with

a particular link layer address before joining the network and potentially an IP address once on the network. This includes: laptops, desktops, servers, cell phones, or any device that may have an IP address."

Network infrastructure devices (e.g. switches, routers, firewalls), which fit the definition, are also considered to be endpoints within this document.

Based on the previous definition of an asset, an endpoint is a type of asset.

posture

Defined in [RFC5209] as "configuration and/or status of hardware or software on an endpoint as it pertains to an organization's security policy."

This term is used within the scope of this document to represent the state information that is collected from an endpoint (e.g. software/hardware inventory, configuration settings).

posture attributes

Defined in [RFC5209] as "attributes describing the configuration or status (posture) of a feature of the endpoint. For example, a Posture Attribute might describe the version of the operating system installed on the system."

Within this document this term represents a specific assertion about endpoint state (e.g. configuration setting, installed software, hardware). The phrase "features of the endpoint" refers to installed software or software components.

system resource

Defined in [RFC4949] as "data contained in an information system; or a service provided by a system; or a system capacity, such as processing power or communication bandwidth; or an item of system equipment (i.e., hardware, firmware, software, or documentation); or a facility that houses system operations and equipment.

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Endpoint Posture Assessment

Endpoint posture assessment involves collecting information about the posture of a given endpoint. This 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 posture of a given endpoint;
- Making that posture available to the enterprise for further analysis and action; and
- Assessing that the endpoint's posture is in compliance with enterprise standards and policy.

3.1. Example - Departmental Software Policy Compliance

In order to meet compliance requirements and ensure that corporate finance information is not revealed improperly, all computers in the finance department of Example Corporation are required to run only software contained on an approved list and to be configured to download and install software patches every night. Each computer is checked to make sure it complies with this policy whenever it connects to the network and at least once a day thereafter. These daily compliance checks assess the posture of each computer and report on its compliance with policy.

3.2. Main Success Scenario

1. Define a target endpoint to be assessed
2. Select acceptable state policies to apply to the defined target
3. Identify the endpoint being assessed
4. Collect posture attributes from the target
5. Communicate target identity and collected posture to external system for evaluation
6. Compare collected posture attributes from the target endpoint with expected state values as expressed in acceptable state policies

4. Functional Capabilities and Requirements

The capabilities in this section support assessing endpoint posture in an automated manner as described in Section Section 3.

4.1. Asset Management

Organizations manage a variety of assets within their enterprise including: endpoints, the hardware they are composed of, installed software, hardware/software licenses used, and configurations.

Managing endpoints and the different types of assets that compose them involves initially discovering and characterizing each asset instance, and then identify them in a common way. Characterization may take the form of logical characterization or security characterization, where logical characterization may include business context not otherwise related to security, but which may be used as information in support of decision making later in risk management.

4.1.1. Example - Asset Discovery within a subnet

Many network management systems detect the presence of assets in a subnet, such as an Ethernet subnet, by monitoring the MAC addresses bradcast within the subnet to determine who responds to broadcasts, and determing the location of the endpoint relative to a bridge. This information is useful for initally discovering and characterizing endpoints belonging to a particular type of network (e.g. Ethernet), and for detecting new nodes in the subnet. This type of information may be accessible by accessing ARP tables [RFC0826], Etherlike-MIB [RFC3535], the Link Layer Discovery Protocol MIB [RFC2922], the Interfaces MIB (IF-MIB) [RFC2863], the YANG module ietf-interfaces , and others.

4.1.2. Example - Asset Discovery by IP Address

Many network management systems periodically test for the presence of endpoints or interfaces in a network by broadcasting ICMP echo commands (pings) to a range of IP addresses and recording the addresses of nodes that respond. This helps discover the endpoints in the network, including endpoints that have suddenly appeared in a network tha are not authorized to be part of the network.

4.1.3. Example - Asset Characterization using system information

The SYSTEM-MIB [RFC1213] contains information to help characterize an endpoint, including a description of the endpoint, an authoritative identifier of the type of endpoint assigned by the vendor of the endpoint, an administrative name for the endpoint, plus the

endpoint's contact person, the location of the endpoint, system time, and an enumerator that identifies the layer of services provided by the endpoint. The system decription includes the vendor, product type, model number, OS version, and networking software version. This is a key MIB module mandated for all SNMP-managed endpoints.

Similar information is available via the YANG module ietf-system . This module includes data node definitions for system identification, time-of-day management, user management, DNS resolver configuration, and some protocol operations for system management.

4.1.4. Example - Asset Characterization using the ENTITY-MIB

The ENTITY-MIB [RFC6933] contains information to describe the components of an endpoint, including physical and logical components, and the relationships between the components. The information about the physical entities includes manufacturer-assigned serial number, manufacture date, administratively-assigned AssetID, and UUID. Logical entities may be defined, and associated with the physical entities using a mapping table.

4.1.5. Example - Asset Characterization using the HOST-RESOURCES-MIB

The HOST-RESOURCES-MIB [RFC2790] contains information to describe the resources of an endpoint, including storage, memory, installed software, running software, software versions, processes, user sessions, devices (processors, disks, printers, network interfaces, etc.). This MIB module also provides monitoring of performance and error states.

4.1.6. Concepts

Managing endpoints and the different types of assets that compose them involves initially discovering and characterizing each asset instance, and then identify them in a common way. Characterization may take the form of logical characterization or security characterization, where logical characterization may include business context not otherwise related to security, but which may be used as information in support of decision making later in risk management.

Coverage involves understanding what and how many assets are under control. Assessing 80% of the enterprise assets is better than assessing 50% of the enterprise assets.

Getting asset details can be comparatively subtle - if an enterprise does not have a precise understanding of its assets, then all acquired data and consequent actions taken based on the data are considered suspect.

Assessing assets (managed and unmanaged) requires that we have visibility into the posture of endpoints, the ability to understand the composition and relationships between different assets types, and the ability to properly characterize them at the outset and over time.

The following list details some requisite Asset Management capabilities:

- Discover assets in the enterprise
- For a given endpoint, understand the composition and relationship of its constituent assets
- Characterize assets according to security and non-security asset properties
- Identify and describe assets using a common vocabulary between implementations
- Reconcile asset representations originating from disparate tools
- Manage asset information throughout the asset's life cycle

4.1.7. Requirements

A method MUST be provided for identifying an endpoint (asset identification) as a unique entity within the its administrative domain.

The endpoint identifier SHOULD be able to be determined in an automated manner.

The endpoint identifier, as communicated between entities, SHOULD be held to a minimal size.

A method MUST be provided for defining an endpoint (asset classification) based on a set of organizationally relevant properties (e.g. organizational affiliation, criticality, function).

4.2. Security Configuration Management

Organizations manage a variety of configurations within their enterprise including: endpoints, the hardware they are composed of, installed software, hardware/software licenses used, and configurations.

4.2.1. Example - ENTITY-MIB

4.2.2. Example - HOST-RESOURCES-MIB

4.2.3. Example - YANG module ietf-interfaces

4.2.4. Concepts

Security configuration management (SCM) deals with the configuration of endpoints, including networking infrastructure devices and computing hosts. Data will include installed hardware and software, its configuration, and its use on the endpoint.

The following list details some requisite Configuration Management capabilities:

o [todo]

4.2.5. Requirements

[todo]

4.3. Security Change Management

Organizations manage a variety of changes within their enterprise including: [todo]

4.3.1. Example - DHCP addressing

4.3.2. Example - RADIUS network access

4.3.3. Example - NAT logging

4.3.4. Example - SYSLOG Authorization messages

SYSLOG [RFC5424] includes facilities for security authorization messages. These messages can be used to alert an analysts that an authorization attempt failed, and the analyst might choose to follow up and assess potential attacks on the relevant endpoint.

4.3.5. Concepts

[todo]

The following list details some requisite Change Management capabilities:

o [todo]

4.3.6. Requirements

[todo]

4.4. Security Vulnerability Management

Vulnerability management involves identifying the patch level of software installed on the device and the identification of insecure custom code (e.g. web vulnerabilities). All vulnerabilities need to be addressed as part of a comprehensive risk management program, which is a superset of software vulnerabilities. Thus, the capability of assessing non-software vulnerabilities applicable to the system is required. Additionally, it may be necessary to support non-technical assessment of data relating to assets such as aspects related to operational and management controls.

policy attribute collection

4.4.1. Example - NIDS response

1. An organization's Network Intrusion Detection System detects a suspect packet received by an endpoint and sends an alert to an analyst. The analyst looks up the endpoint in the asset inventory database, looks up the configuration policy associtaed with that endpoint, and initates an endpoint assessment of installed software and patches on the endpoint to determine if the endpoint is compliant with policy.

The analyst reviews the results of the assessment and takes action according to organization policy and procedures.

4.4.2. Example - Historical vulnerability analysis

When a serious vulnerability or a zero-day attack is discovered, one of the first priorities in any organization is to determine which endpoints may have been affected and assess those endpoints to try to determine whether they were compromised. Checking current endpoint state is not sufficient because an endpoint may have been temporarily compromised due to this vulnerability and then the infection may have removed itself. In fact, the vulnerable software may have been removed or upgraded since the compromise took place. And if the endpoint is still compromised, the malware on the endpoint may cause it to lie about its configuration. In this environment, maintaining historical information about endpoint configuration is essential. Such information can be used to find endpoints that had the vulnerable software installed at some point in time. Those endpoints can be checked for current or past indicators of compromise such as files or behavior linked to a known exploit for this vulnerability.

Endpoints found to be vulnerable can be isolated to prevent infection while remediation is done. Endpoints believed to be compromised can be isolated for analysis and to limit the spread of infection.

4.4.3. Source Address Validation

Source Address Validation Improvement methods were developed to prevent nodes attached to the same IP link from spoofing each other's IP addresses, so as to complement ingress filtering with finer-grained, standardized IP source address validation. The framework document describes and motivates the design of the SAVI methods. Particular SAVI methods are described in other documents.

4.4.4. Concepts

The following list details some requisite Vulnerability Management capabilities:

- Collect the state of non-technical controls commonly called administrative controls (i.e. policy, process, procedure)
- Collect the state of technical controls including, but not necessarily limited to:
  - Software inventory (e.g. operating system, applications, patches)
  - Configuration settings

4.4.5. Requirements

[todo]

4.5. Data Collection

Central to any automated assessment solution is the ability to collect data from, or related to, an endpoint, such as the security state of the endpoint and its constituent assets.

4.5.1. Concepts

The following assessment capabilities support SCM:

- [todo]

4.5.2. Requirements

One or more data formats MUST be identified to describe instructions, data collection methods, to drive data collection (e.g. technical, interrogative).

One or more data formats MUST be identified to instruct what posture attributes need to be collected for a specific set of endpoints.

A method MUST be provided to include OPTIONAL instructions on describing what content must be run on the endpoint.

A method MUST be provided to include OPTIONAL instructions that determine how to collect data supporting any particular test for that endpoint.

A method MUST be provided for retrieving data collection instructions from a remote host (see Section Section 4.7).

One or more data formats MUST be identified to capture the results of data collection.

This expression MUST be capable of supporting the characterization of assets and any related configuration settings that together compose an endpoint.

A mechanism MUST be provided to identify the software and hardware asset instances that compose an endpoint.

An asset identifier SHOULD be able to be determined in an automated manner

An asset identifier, as communicated between entities, SHOULD be held to a minimal size.

An asset identifier SHOULD be able to represented in a simple unambiguous manner, such as a reference, so that its embedded use in places like applicability clauses for individual benchmark tests can be kept from making their usage unwieldy.

A mechanism MUST be provided to associate configuration settings values to the installed software.

A mechanism MUST be provided to identify additional collected posture attribute/value pairs related to an endpoint.

A mechanism MUST be provided to identify the endpoint the results pertain to (see Section Section 4.1.

A mechanism MUST be provided to associate the data collection method with the collected value.

A mechanism MUST be provided to include provenance information describing what sensor of software collected the data.

A mechanism MUST be provided to include entailment information, perhaps by reference, describing the methodology used to collect the data.

A method of communicating data collection results to another system for further analysis MUST be identified.

TODO: Communicate, unambiguously and to the necessary level of detail**, the asset details between software components

4.6. Assessment Result Analysis

The data collected needs to be analyzed for compliance to a standard stipulated by the enterprise. Analysis methods may vary between enterprises, but commonly take a similar form.

4.6.1. Concepts

The following capabilities support the analysis of assessment results:

- Comparing actual state to expected state
- Scoring/weighting individual comparison results
- Relating specific comparisons to benchmark-level requirements
- Relating benchmark-level requirements to one or more control frameworks

4.6.2. Requirements

A method MUST be provided for selecting acceptable state policy, describing how to evaluate collected information, based on characteristics of the endpoint and organizational policy.

A method MUST be provided for comparing collected data to expected state values (test evaluation).

Any results produced by analysis processes MUST be capable of being transformed into a human-readable format.

4.7. Content Management

The capabilities required to support risk management state measurement will yield volumes of content. The efficacy of risk management state measurement depends directly on the stability of the driving content, and, subsequently, the ability to change content according to enterprise needs.

4.7.1. Concepts

Capabilities supporting Content Management should provide the ability to create/define or modify content, as well as store and retrieve said content of at least the following types:

- Configuration checklists
- Assessment rules
- Data collection rules and methods
- Scoring models
- Vulnerability information
- Patch information
- Asset characterization data and rules

Note that the ability to modify content is in direct support of tailoring content for enterprise-specific needs.

4.7.2. Requirements

A protocol MUST be identified for retrieving SACM content from a content repository

A protocol MUST be identified for querying SACM content held in a content repository. The protocol MUST support querying content by applicability to asset characteristics.

TODO: Determine what content can or must be run on the endpoint

A protocol MUST be identified for curating SACM content in a content repository. Note: This might be an area where we can limit the scope of work relative to the initial SACM charter.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

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

7. 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.

The authors would like to thank Kathleen Moriarty and Stephen Hanna for contributing text to this document. The author would also like to acknowledge the members of the SACM mailing list for their keen and insightful feedback on the concepts and text within this document.

8. Change Log

8.1. -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 Waltmire'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.

9. References

9.1. Normative References

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

9.2. Informative References

[I-D.ietf-nea-pt-eap]
Cam-Winget, N. and P. Sangster, "PT-EAP: Posture Transport (PT) Protocol For EAP Tunnel Methods", draft-ietf-nea-pt-eap-06 (work in progress), December 2012.

[I-D.ietf-nea-pt-tls]

Sangster, P., Cam-Winget, N., and J. Salowey, "PT-TLS: A TLS-based Posture Transport (PT) Protocol", draft-ietf-nea-pt-tls-08 (work in progress), October 2012.

[I-D.ietf-netmod-interfaces-cfg]
Bjorklund, M., "A YANG Data Model for Interface Management", draft-ietf-netmod-interfaces-cfg-12 (work in progress), July 2013.

[I-D.ietf-netmod-system-mgmt]
Bierman, A. and M. Bjorklund, "YANG Data Model for System Management", draft-ietf-netmod-system-mgmt-08 (work in progress), July 2013.

[I-D.ietf-savi-framework]
Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, "Source Address Validation Improvement Framework", draft-ietf-savi-framework-06 (work in progress), January 2012.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets:MIB-II", STD 17, RFC 1213, March 1991.

[RFC2790] Waldbusser, S. and P. Grillo, "Host Resources MIB", RFC 2790, March 2000.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

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

[RFC2922] Bierman, A. and K. Jones, "Physical Topology MIB", RFC 2922, September 2000.

[RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, June 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[RFC5792] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5792, March 2010.

[RFC5793] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5793, March 2010.

[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

[RFC6933] Bierman, A., Romascanu, D., Quittek, J., and M. Chandramouli, "Entity MIB (Version 4)", RFC 6933, May 2013.

Authors' Addresses

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

Email: david.waltermire@nist.gov

Adam W. Montville
Tripwire, Inc.
101 SW Main Street, Suite 1500
Portland, Oregon 97204
USA

Email: amontville@tripwire.com

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

Email: ietfdbh@comcast.net