SACM                                                      D. Haynes
Internet-Draft                                  The MITRE Corporation
Intended status: Best Current Practice           J. Fitzgerald-McKay
Expires: December 23, 2019                      Department of Defense
                                                         L. Lorenzin
                                                        Pulse Secure
                                                       June 21, 2019

                 Endpoint Posture Collection Profile
                      draft-ietf-sacm-ecp-05

   Abstract

   This document specifies the Endpoint Posture Collection Profile,
   which describes the best practices for the application of IETF, TNC,
   and ISO/IEC data models, protocols, and interfaces to support the on-
   going collection and communication of endpoint posture to a
   centralized server where it can be stored and made available to other
   tools.  This document is an extension of the Trusted Computing
   Group's Endpoint Compliance Profile Version 1.0 specification [ECP].

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 https://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 December 23, 2019.

Copyright Notice

carefully, as they describe your rights and restrictions with respect
to this document.  Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1.  Introduction

   The Endpoint Posture Collection Profile (EPCP) builds on prior work
   from the IETF NEA WG, the IETF NETCONF WG, IETF NETMOD WG, the
   Trusted Computing Group (TCG) Trusted Network Communications [TNC]
   WG, and the International Organization for Standardization/
   International Electrotechnical Commission Joint Technical Committee
   (JTC) 1, Subcommittee (SC) 7, WG 21 (ISO/IEC JTC 1, SC7, WG21) to
   describe the best practices for the collection and communication of
   posture information from network-connected endpoints to a centralized
   server.

   This document focuses on reducing the security exposure of a network
   by enabling event-driven posture collection, standardized querying of
   additional posture information as needed, and the communication of
   that data to a centralized server where it can made available to
   other components.  Thus, eliminating the need for redundant
   collection and agents on endpoints.  Future revisions of this
   document may include support for the collection of posture
   information from other endpoint types as well as a standardized
   interface for storing and querying data in repositories among other
   capabilities.  Additional information about this future work can be
   found in Section 6 of this document.

   To support the collection of posture information from new endpoint
   types, this document is organized such that it first provides a high-
   level overview of EPCP as well as its abstract architectural
   components and transactions that will be realized by implementations
   (Section 3).  This is followed by individual sections that discuss
   the best practices for specific implementations of the EPCP for a
   given endpoint type (e.g., traditional, network device, etc.) along
   with any extensions for supported use cases (software asset
   management, vulnerability management, etc.).

2.  Terminology

   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 [RFC2119].  This
   specification does not distinguish blocks of informative comments and
   normative requirements.  Therefore, for the sake of clarity, note

that lower case instances of must, should, etc. do not indicate
normative requirements.

Furthermore, this document uses terms as defined in
[I-D.ietf-sacm-terminology] unless otherwise specified.

3.  Endpoint Posture Collection Profile

The EPCP describes how IETF, TCG, and ISO/IEC data models, protocols,
and interfaces can be used to support the posture assessment of
endpoints on a network.  This profile does not generate new data
models, protocols, or interfaces; rather, it offers best practices
for a full end-to-end solution for posture assessment, as well as a
fresh perspective on how existing standards can be leveraged against
vulnerabilities.  Rationale for the EPCP solution as well as the
supported and non-supported use cases is available in Appendix A and
Appendix B respectively.

The EPCP makes it possible to perform posture assessments against all
network-connected endpoints by:

1.  uniquely identifying the endpoint;

2.  collecting and evaluating posture based on data from the endpoint
    (asset management, software asset management, vulnerability
    management, and configuration management);

3.  creating a secure, authenticated, confidential channel between
    the endpoint and the posture manager;

4.  enabling the endpoint to notify the posture manager about changes
    to its configuration;

5.  enabling the posture manager to request information about the
    configuration of the endpoint; and

6.  storing the posture information in a repository linked to the
    identifier for the endpoint.

Furthermore, the EPCP aims to support data storage and data sharing
capabilities to make the collected posture information available to
authorized parties and components in support of other processes
(analytic, access control, remediation, reporting, etc.).

3.1.  Components

   To perform posture assessment, data storage, and data sharing, the
   EPCP defines several components.  Some of these components reside on
   the target endpoint.  Others reside on a posture manager that manages
   communications with the target endpoint and stores the target
   endpoint's posture information in a repository.

   It should be noted that the primary focus of this document is on the
   communication between the posture manager and endpoints.  While the
   orchestrator, evaluator, repository, and administrative interface and
   API will be discussed in the context of the broader EPCP
   architecture, these components are not strictly defined nor are best
   practices provided for them at this time.  As a result, vendors are
   free to implement these components and interfaces in a way that makes
   the most sense for their products.

```
        *********FUTURE WORK**********  Posture Manager            Endpoint
        *        Orchestrator       *  +----------------+         +--------
--------+
        *      +--------+            *  |                |         |
    |
        *      |        |<------------->|                |         |
    |
        *      |        | publish/   *  |                |         |
    |
        *      |        | subscribe  *  |                |         |
    |
        *      |        |            *  | +------------+ |         | +------
------+ |
        *      +--------+            *  | |            | |         | |
   | |
        *********FUTURE WORK**********  | | Posture    | | report/ | | Postu
re    | |
                                        | | Collection | | publish | | Colle
ction | |
  Evaluator        Repository           | | Manager    | |<----------| | Engin
e     | |
  +------+         +--------+           | |            | |         | |
   | |
       |  |             |        |      | |            | |         | |
   | |
       |  |             |        |      | +------------+ |         | +------
------+ |
       |  |<-------->|  |<---------->|  |                | query/   |
       |
       |  | request/ |  | store    |   |                | subscribe |
       |
       |  | respond  |  |          |   |                |---------->|
       |
       |  |          |  |          |   |                |          |
       |
  +------+         +--------+           +----------------+         +--------
--------+
       |                    query                ^  ^
       +-------------------------------------+   |  |
************************FUTURE WORK***********|************
*                                            |           *
*                     +-----------------------------+ *
*                     | Administrative Interface    | *
*                     | and API                     | *
*                     +-----------------------------+ *
*                                            *
***********************FUTURE WORK***********************
```
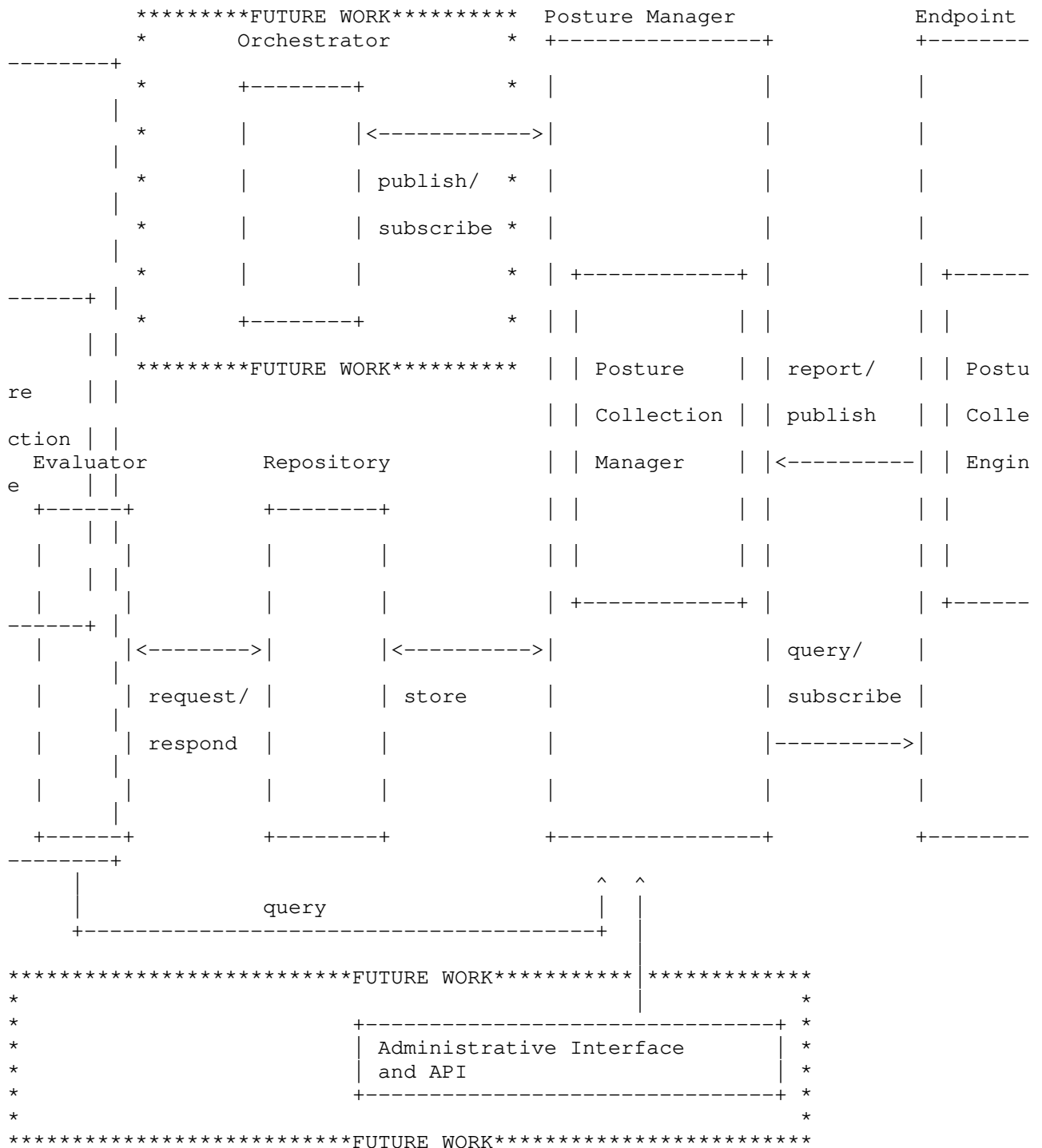
Figure 1: EPCP Components

3.1.1.  Endpoint

   An endpoint is defined in [RFC6876].  In the EPCP, the endpoint is
   monitored by the enterprise and is the target of posture assessments.
   To support these posture assessments, posture information is
   collected via a posture collection engine.

3.1.1.1.  Posture Collection Engine

   The posture collection engine is located on the target endpoint and
   can either receive queries for data from the posture collection
   manager (see Section 3.2.4) or can push data to the posture

collection manager (see Section 3.2.3).  The posture collection
engine sends collected posture information to the posture manager

where it can be sanity checked and stored in the repository.  The
posture collection engine also contains a capability that sets up
exchanges between the target endpoint and posture manager.  This
capability makes the posture collection engine responsible for
performing the client-side portion of encryption handshakes, and for
locating authorized posture managers with which to communicate.

### 3.1.2.  Posture Manager

The posture manager is an endpoint that collects, validates, and
enriches posture information received about a target endpoint.  It
also stores the posture information it receives in the repository
where it can be evaluated.  The posture manager does not evaluate the
posture information.

### 3.1.2.1.  Posture Collection Manager

A posture collection manager is a lightweight and extensible
component that facilitates the coordination and execution of posture
collection requests using collection mechanisms deployed across the
enterprise.  The posture collection manager may query and retrieve
guidance from the repository to guide the collection of posture
information from the target endpoint.

The posture collection manager also contains a capability that sets
up exchanges between the target endpoint and the posture manager, and
manages data sent to and from posture collection engine.  It is also
responsible for performing the server-side portion of encryption
handshakes.

If the posture manager wants to register for continuous collection of
endpoint posture changes with the endpoint, then it must do so in a
scalable way.  Specifically, it will need to create subscriptions
with endpoints in a way which allows the posture data to be securely
pushed.  Effectively this means that the endpoint must be able to
establish secure transport connectivity to the posture collection
manager as needed, and the collection manager must be able to
periodically collect the current state of the endpoint to verify the
expected state of that endpoint.

### 3.1.3.  Repository

The repository hosts guidance, endpoint identification information,
and posture information reported by target endpoints where it is made
available to authorized components and persisted over a period of
time set by the administrator.  Information stored in the repository
will be accessible to authorized parties via a standard
administrative interface as well as through a standardized API.  The

repository may be a standalone component or may be located on the posture manager.  Furthermore, an implementation is not restricted to a single repository and may leverage several repositories to provide this functionality.

### 3.1.4.  Evaluator

The evaluator assesses the posture status of a target endpoint by comparing collected posture information against the desired state of the target endpoint specified in guidance.  The evaluator queries and retrieves the appropriate guidance from the repository as well as queries and retrieves the posture information required for the assessment from the repository.  If the required posture information is not available in the repository, the evaluator may request the posture information from the posture collection manager, which will result in the collection of additional posture information from the target endpoint.  This information is subsequently stored in the repository where it is made available to the evaluator and other components.  The results of the assessment are stored in the repository where they are available to tools and administrators for follow-up actions, further evaluation, and historical purposes.

### 3.1.5.  Orchestrator

The orchestrator provides a publish/subscribe interface for the repository so that infrastructure endpoints can subscribe to and receive published posture assessment results from the repository regarding endpoint posture changes.

### 3.1.6.  Administrative Interface and API

The administrative interface allows administrators to query the repository and manage the endpoints and software used in the EPCP via the posture manager.  Similarly, an API is necessary to allow infrastructure endpoints and software access to the information stored in the repository and to manage the endpoints and software used in the EPCP.  The administrative interface and API provide authorized users, infrastructure endpoints, and software with the ability to query the repository for data, send commands to the posture collection managers requesting information from the associated posture collection engines residing on endpoints, and establish and update the policy that resides on the posture manager.

### 3.2.  Transactions

The following sections describe the transactions associated with the components of the EPCP architecture and may be provided in an implementation.

### 3.2.1.  Provisioning

An endpoint is provisioned with one or more attributes that will
serve as its unique identifier on the network as well as the
components and data models necessary to interact with the posture
manager.  Examples of such identifiers include MAC addresses, serial
numbers, hardware certificates compliant with [IEEE-802-1ar], and the
identities of hardware cryptographic modules among others.  Once
provisioning is complete, the endpoint is deployed on the network.
Over time, components and data models may need to be added to the
endpoint or updated to support the collection needs of an enterprise.

### 3.2.2.  Discovery and Validation

If necessary, the target endpoint finds and validates the posture
manager.  The posture collection engine on the target endpoint and
posture collection manager on the posture manager complete an
encryption handshake, during which endpoint identity information is
exchanged.

### 3.2.3.  Event Driven Collection

The posture assessment is initiated when the posture collector engine
on the target endpoint notices that relevant posture information on
the endpoint has changed.  Then, the posture collection engine
initiates a posture assessment information exchange with the posture
collection manager.

### 3.2.4.  Querying the Endpoint

The posture assessment is initiated by the posture collection
manager.  This can occur because:

1.  policy states that a previous assessment has aged out or become
    invalid, or

2.  the posture collection manager is alerted by a sensor or an
    administrator (via the posture manager's administrative
    interface) that an assessment must be completed.

### 3.2.5.  Data Storage

Once posture information is received by the posture manager, it is
forwarded to the repository.  The repository could be co-located with
the posture manager, or there could be direct or brokered
communication between the posture manager and the repository.  The
posture information is stored in the repository along with past
posture information collected about the target endpoint.

3.2.6.  Data Sharing

   Because the target endpoint posture information was sent in
   standards-based data models over secure, standardized protocols, and
   then stored in a centralized repository linked to unique endpoint
   identifiers, authorized parties are able to access the posture
   information.  Such authorized parties may include, but are not
   limited to, administrators or endpoint owners (via the posture
   manager's administrative interface), evaluators that access the
   repository directly, and orchestrators that rely on publish/subscribe
   communications with the repository.

4.  IETF NEA EPCP Implementation for Traditional Endpoints

   When EPCP is used, posture collectors running on the target endpoint
   gather posture information as changes occur on the endpoint.  The
   data is aggregated by the posture broker client and forwarded to a
   posture manager, over a secure channel, via the posture transport
   client.  Once received by the posture transport server on the posture
   manager, the posture information is directed by the posture broker
   server to the appropriate posture validators where it can be
   processed and stored in a repository.  There the posture information
   can be used by other tools to carry out assessment tasks.  Posture
   collectors can also be queried by posture validators to refresh
   posture information about the target endpoint or to ask a specific
   question about posture information.  This is shown in Figure 2.

```
        Posture                   Posture
        Collection                Collection
        Manager                   Engine
        +--------------+          +--------------+
        |              |          |              |
        | +----------+ | PA-TNC   | +----------+ |
        | | Posture  | | -------- | | Posture  | |
        | | Validator| |          | | Collector| |
        | +----------+ |          | +----------+ |
        |      |       |          |      |       |
        |      | IF-IMV|          |      | IF-IMC|
        |      |       |          |      |       |
        | +----------+ | PB-TNC   | +----------+ |
        | | PB Server| | -------- | | PB Client| |
        | +----------+ |          | +----------+ |
        |      |       |          |      |       |
        |      |       |          |      |       |
        |      |       |          |      |       |
        | +----------+ |          | +----------+ |
        | | PT Server| | <------> | | PT Client| |
        | +----------+ | PT-TLS   | +----------+ |
        |              |          |              |
        +--------------+          +--------------+
```
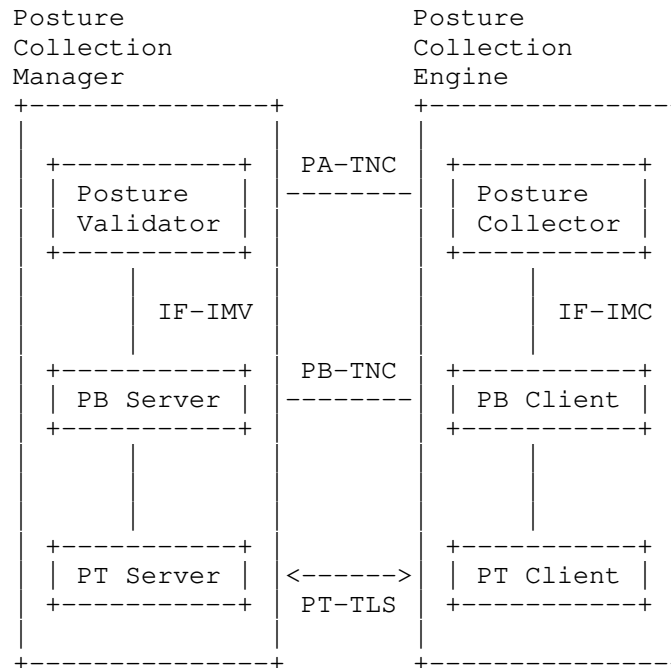
                    Figure 2: NEA Components

   These requirements are written with a view to performing a posture
   assessment on an endpoint; as the EPCP grows and evolves, these
   requirements will be expanded to address issues that arise.  Note
   that these requirements refer to defined components of the NEA
   architecture [RFC5209].  As with the NEA architecture, vendors have
   discretion as to how these NEA components map to separate pieces of
   software or endpoints.

   Furthermore, it should be noted that the posture broker client and
   posture transport client components of the posture collection engine
   and the posture broker server and posture transport server components
   of the posture collection manager would likely need to be implemented
   by a single vendor because there are no standardized interfaces
   between the respective components and would not be interoperable.

   Examples of the EPCP as implemented using the components from the NEA
   architecture are provided in Appendix C.

4.1.  Endpoint Provisioning

   An endpoint is provisioned with a machine certificate that will serve
   as its unique identifier on the network as well as the components
   necessary to interact with the posture manager.  This includes a
   posture collection engine to manage requests from the posture manager
   and the posture collectors necessary to collect the posture
   information of importance to the enterprise.  The endpoint is
   deployed on the network.

   The target endpoint SHOULD authenticate to the posture manager using
   a machine certificate during the establishment of the outer tunnel
   achieved with the posture transport protocol defined in [RFC6876].
   [IF-IMV] specifies how to pull an endpoint identifier out of a
   machine certificate.  An endpoint identifier SHOULD be created in
   conformance with [IF-IMV] from a machine certificate sent via
   [RFC6876].

   In the future, the identity could be a hardware certificate compliant
   with [IEEE-802-1ar]; ideally, this identifier SHOULD be associated
   with the identity of a hardware cryptographic module, in accordance
   with [IEEE-802-1ar], if present on the endpoint.  The enterprise
   SHOULD stand up a certificate root authority; install its root
   certificate on endpoints and on the posture manager; and provision
   the endpoints and the posture manager with machine certificates.  The
   target endpoint MAY authenticate to the posture manager using a
   combination of the machine account and password; however, this is
   less secure and not recommended.

4.2.  Endpoint

   The endpoint MUST conform to [RFC5793], which levies several
   requirements against the endpoint.  An endpoint that complies with
   these requirements will be able to:

   1.  attempt to initiate a session with the posture manager if the
       posture makes a request to send an update to posture manager;

   2.  notify the posture collector if no PT-TLS session with the
       posture manager can be created;

   3.  notify the posture collector when a PT-TLS session is
       established; and

   4.  receive information from the posture collectors, forward this
       information to the posture manager via the posture collection
       engine.

4.2.1.  Posture Collector

   Any posture collector used in an EPCP solution MUST be conformant
   with the TCG TNC Integrity Measurement Collector interface [IF-IMC].

4.2.2.  Posture Broker Client

   The posture broker client MUST conform to [IF-IMC] to enable
   communications between the posture broker client and the posture
   collectors on the endpoint.

4.2.3.  Posture Transport Client

   The posture transport client MUST implement PT-TLS.

   The posture transport client MUST support the use of machine
   certificates for TLS at each endpoint consistent with the
   requirements stipulated in [RFC6876] and [Server-Discovery].

   The posture transport client MUST be able to locate an authorized
   posture manager, and switch to a new posture manager when required by
   the network, in conformance with [Server-Discovery].

4.3.  Posture Manager

   The posture manager MUST conform to all requirements in the
   [RFC5793].

4.3.1.  Posture Validator

   Any posture validator used in an EPCP solution MUST be conformant
   with the TCG TNC Integrity Measurement Verifier interface [IF-IMV].

4.3.2.  Posture Broker Server

   The posture broker server MUST conform to [IF-IMV].  Conformance to
   [IF-IMV] enables the posture broker server to obtain endpoint
   identity information from the posture transport server, and pass this
   information to any posture validators on the posture manager.

4.3.3.  Posture Transport Server

   The posture transport server MUST implement PT-TLS.

   The posture transport server MUST support the use of machine
   certificates for TLS at each endpoint consistent with the
   requirements stipulated in [RFC6876] and [Server-Discovery].

4.4.  Repository

   EPCP requires a simple administrative interface for the repository.
   Posture validators on the posture manager receive the target endpoint
   posture information via PA-TNC [RFC5792] messages sent from
   corresponding posture collectors on the target endpoint.  The posture
   validators store this information in the repository linked to the
   identity of the target endpoint where the posture collectors are
   located.

4.5.  IETF SACM SWAM Extension to the IETF NEA EPCP Implementation

   This section defines the requirements associated with the software
   asset management extension [RFC8412] to the IETF NEA EPCP
   implementation.

4.5.1.  Endpoint Pre-Provisioning

   This section defines the requirements associated with implementing
   SWIMA.

   The following requirements assume that the platform or OS vendor
   supports the use of SWID tags and has identified a standard directory
   location for the SWID tags to be located as specified by [SWID].

4.5.2.  SWID Tags

   The primary content for the EPCP is the information conveyed in the
   elements of a SWID tag.

   The endpoint MUST have SWID tags stored in a directory specified in
   [SWID].  The tags SHOULD be provided by the software vendor; they MAY
   also be generated by:

   o  the software installer; or

   o  third-party software that creates tags based on the applications
      it sees installed on the endpoint.

   The elements in the SWID tag MUST be populated as specified in
   [SWID].  These tags, and the directory in which they are stored, MUST
   be updated as software is added, removed, or updated.

4.5.3.  SWID Posture Collectors and Posture Validators

4.5.3.1.  The SWID Posture Collector

   For the EPCP, the SWID posture collector MUST be conformant with
   [RFC8412], which includes requirements for:

   1.  Collecting SWID tags from the SWID directory;

   2.  Monitoring the SWID directory for changes;

   3.  Initiating a session with the posture manager to report changes
       to the directory;

   4.  Maintaining a list of changes to the SWID directory when updates
       take place and no PT-TLS connection can be created with the
       posture manager;

   5.  Responding to a request for SWID tags from the SWID Posture
       Validator on the posture manager; and

   6.  Responding to a query from the SWID posture validator as to
       whether all updates have been sent.

   The SWID posture collector is not responsible for detecting that the
   SWID directory was not updated when an application was either
   installed or uninstalled.

4.5.3.2.  The SWID Posture Validator

   Conformance to [RFC8412] enables the SWID posture validator to:

   1.  Send messages to the SWID posture collector (at the behest of the
       administrator at the posture manager console) requesting updates
       for SWID tags located on endpoint;

   2.  Ask the SWID posture collector whether all updates to the SWID
       directory located at the posture manager have been sent; and

   3.  Perform any validation and processing on the collected SWID
       posture information prior to storage.

   In addition to these requirements, a SWID posture validator used in
   conformance with this profile MUST be capable of passing this SWID
   posture information as well as the associated endpoint identity to
   the repository for storage.

### 4.5.4.  Repository

The administrative interface SHOULD enable an administrator to:

1.  Query which endpoints have reported SWID tags for a particular
    application

2.  Query which SWID tags are installed on an endpoint; and

3.  Query tags based on characteristics, such as vendor, publisher,
    etc.

## 5.  IETF NETCONF EPCP Implementation for Network Device Endpoints

When EPCP is used, a NETCONF client that implements the posture
collection manager sends a query to target network device endpoint
requesting posture information over a secure channel.  Once the
NETCONF server on the endpoint receives the request, it queries one
or more datastores for the posture information.  The NETCONF server
then reports the information back to the NETCONF client where it can
be stored in a repository for use by other tools.  This is shown in
Figure 3.

```
              Posture                  Posture
              Collection               Collection
              Manager                  Engine
              +--------------+         +--------------+
              |              |         |              |
              |              |         |  +----------+|
              |              |         |  | Data     ||
              |              |         |  | Store(s) ||
              |              |         |  +----------+|
              |              |         |       |      |
              |              |         |       |      |
              |  +----------+|         |  +----------+|
              |  | NETCONF  ||         |  | NETCONF  ||
              |  | Client   || <------>|  | Server   ||
              |  +----------+|  NETCONF|  +----------+|
              |              |         |              |
              +--------------+         +--------------+
```
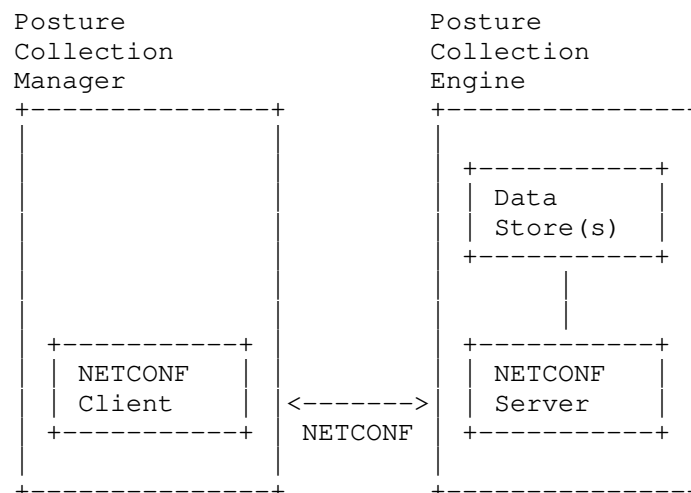
                  Figure 3: NETCONF Components

These requirements are written with a view to performing a posture
assessment on network device endpoints (routers, switches, etc.); as
the EPCP grows and evolves, these requirements will be expanded to
address issues that arise.

Note that these requirements refer to defined components of the
NETCONF architecture and map back to EPCP.  As with the NETCONF
architecture, vendors have discretion as to how these NETCONF
components map to separate pieces of software or endpoints.

5.1.  Endpoint Provisioning

For the posture manager to be able to query the datastores on the
endpoint, the endpoint MUST be configured to grant the posture
manager access to its datastores as described in [RFC6241].  The
posture manager is identified by its NETCONF username.  The endpoint
is deployed on the network.

5.2.  Posture Manager Provisioning

For the posture manager to be able to query the datastores on the
endpoint, the posture manager MUST be provisioned with a NETCONF
username that will be used to authenticate the posture manager to the
endpoint as described in [RFC6241].  The username generated will be
determined by the selected transport protocol.  The posture manager
is deployed on the network.

5.3.  Endpoint

An endpoint MUST conform to the requirements outlined for servers in
the NETCONF protocol as defined in [RFC6241].  This requires the
implementation of NETCONF over SSH [RFC6242].  An endpoint MAY
support the NETCONF protocol over other transports such as TLS
[RFC7589] as well as the RESTCONF protocol as defined in [RFC8040].

5.3.1.  Datastore

A NETCONF datastore on an endpoint MUST support the operations
outlined in [RFC6241], but, the actual implementation of the
datastore is left to the endpoint vendor.

Datastores MUST support the YANG data modeling language [RFC7950] for
expressing endpoint posture information in a structured format.  In
addition, datastores MAY support other data models such as XML (via
YIN) for representing posture information.

Datastores MUST support the compliance posture information specified
in [RFC7317].  Datastores MAY support other models standardized or
proprietary as deemed appropriate by the endpoint vendor.

5.4.  Posture Manager

   A posture manager MUST conform to the requirements specified for
   clients in the NETCONF protocol as defined in [RFC6241].  This
   requires the implementation of NETCONF over SSH [RFC6242].  A posture
   manager MAY also support the NETCONF protocol over other transports
   such as TLS [RFC7589].  In addition, a posture manager MAY support
   the RESTCONF protocol as defined in [RFC8040].

   While ad-hoc fetch/polling via NETCONF and RESTCONF is useful for
   assessing endpoint compliance, such solutions by themselves are not
   able to detect changes as they occur on the endpoint.  As a result, a
   future revision of this document will support
   [I-D.ietf-netconf-yang-push] to receive updates on YANG-modeled
   posture information.  Similarly, because not all posture information
   is modeled in YANG, a future revision of this document will reference
   [I-D.ietf-netconf-subscribed-notifications] once it is a standard to
   support continuous streams of unstructured data from the endpoint to
   the posture manager.

5.5.  Repository

   EPCP requires a simple administrative interface for the repository.
   The posture collection manager on the posture manager receives the
   target endpoint posture information via NETCONF [RFC6241] messages
   sent from posture collection engine on the target endpoint.  The
   posture collection manager stores this information in the repository
   linked to the identity of the target endpoint from which it was
   collected.

6.  Future Work

   This section captures ideas for future work related to EPCP that
   might be of interest to the IETF SACM WG.  These ideas are listed in
   no particular order.

   o  The [I-D.ietf-netconf-subscribed-notifications] and
      [I-D.ietf-netconf-yang-push] which have been submitted to IESG for
      publication could be leveraged for an HTTP-based subscription for
      EPCP.  Specifically, it could be used for the posture collection
      manager to continuously receive posture changes as they happen
      from the posture collection engine.  At this point, it seems like
      [I-D.ietf-netconf-restconf-notif] would be a good match to these
      requirements.  However further investigation into the
      applicability of supporting a RESTCONF server capability on to
      handle subscription requests needs to be made.  Specific questions
      which should be examined include:

* Number of endpoints which can be continuously tracked by a
  single posture collection manager.  Scalability questions to be
  considered include elements from the number of transport
  connects maintained to the volume of volume and churn of
  posture evidence which will be continuously pushed to the
  posture collection manager manager.

* Ability of the posture collection manager to establish and
  maintain a continuous state of endpoint posture during
  failures.  This includes failures/reboots on either side of the
  interface.

* Ability to support for the full set of functions described for
  NETCONF within Section 5.

o Add support endpoint types beyond workstations, servers, and
  network infrastructure devices.

o Examine the integration of [I-D.ietf-mile-xmpp-grid].

o Define a standard interface and API for interacting with the
  repository.  Requirements to consider include: creating a secure
  channel between a publisher and the repository, creating a secure
  channel between a subscriber and the repository, and the types of
  interactions that must be supported between publishers and
  subscribers to a repository.

o Define a standard interface for communications between the posture
  broker client and posture transport client(s) as well as the
  posture broker server and posture transport server(s).

o Retention of posture information on the target endpoint.

o Define an orchestrator component as well as publish/subscribe
  interface for it.

o Define an evaluator component as well as an interface for it.

o Reassess the use of MAC addresses, including market research to
  determine if MAC addresses continue to be a widely implemented
  device identifier among network tools.

7.  Acknowledgements

The authors wish to thank all of those in the TCG TNC work group who
contributed to development of the TNC ECP specification upon which
this document is based.

| Member | Organization |
|--------|--------------|
| Padma Krishnaswamy | Battelle Memorial Institute |
| Eric Fleischman | Boeing |
| Richard Hill | Boeing |
| Steven Venema | Boeing |
| Nancy Cam-Winget | Cisco Systems |
| Scott Pope | Cisco Systems |
| Max Pritikin | Cisco Systems |
| Allan Thompson | Cisco Systems |
| Nicolai Kuntze | Fraunhofer Institute for Secure Information Technology (SIT) |
| Ira McDonald | High North |
| Dr. Andreas Steffen | HSR University of Applied Sciences Rapperswil |
| Josef von Helden | Hochschule Hannover |
| James Tan | Infoblox |
| Steve Hanna (TNC-WG Co-Chair) | Juniper Networks |
| Cliff Kahn | Juniper Networks |
| Lisa Lorenzin | Juniper Networks |
| Atul Shah (TNC-WG Co-Chair) | Microsoft |
| Jon Baker | MITRE |
| Charles Schmidt | MITRE |
| Rainer Enders | NCP Engineering |
| Dick Wilkins | Phoenix Technologies |

| | |
|---|---|
| David Waltermire | NIST |
| Mike Boyle | U.S. Government |
| Emily Doll | U.S. Government |
| Jessica Fitzgerald-McKay | U.S. Government |
| Mary Lessels | U.S. Government |
| Chris Salter | U.S. Government |

Table 1: Members of the TNC Work Group that Contributed to the
Document

Special thanks also to Dan Ehrlich, Kathleen Moriarty, David Oliva
and Eric Voit for their thoughtful comments and edits.

8.  IANA Considerations

This document does not define any new IANA registries.  However, this
document does reference other documents that do define IANA
registries.  As a result, the IANA Considerations section of the
referenced documents should be consulted.

9.  Security Considerations

This Security Considerations section includes an analysis of the
attacks that may be mounted against systems that implement the EPCP
(Section 9.1) and the countermeasures that may be used to prevent or
mitigate these attacks (Section 9.2).  Overall, a substantial
reduction in cyber risk can be achieved.

9.1.  Threat Model

This section lists the attacks that can be mounted on a NEA
implementation of an EPCP environment.  The following section
(Section 9.2) describes countermeasures.

Because the EPCP describes a specific use case for NEA components,
many security considerations for these components are addressed in
more detail in the technical specifications: [RFC8412], [IF-IMC],
[RFC5793], [Server-Discovery], [RFC6876], [IF-IMV].

9.1.1.  Endpoint Attacks

   While the EPCP provides substantial improvements in endpoint
   security, endpoints can still be compromised.  For this reason, all
   parties must regard data coming from endpoints as potentially
   unreliable or even malicious.  An analogy can be drawn with human
   testimony in an investigation or trial.  Human testimony is essential
   but must be regarded with suspicion.

   o  Compromise of endpoint: A compromised endpoint may report false
      information to confuse or even provide maliciously crafted
      information with a goal of infecting others.

   o  Putting bad information in SWID directory: Even if an endpoint is
      not completely compromised, some of the software running on it may
      be unreliable or even malicious.  This software, potentially
      including the SWID generation or discovery tool, or malicious
      software pretending to be a SWID generation or discovery tool, can
      place incorrect or maliciously crafted information into the SWID
      directory.  Endpoint users may even place such information in the
      directory, whether motivated by curiosity or confusion or a desire
      to bypass restrictions on their use of the endpoint.

   o  Identity spoofing (impersonation): A compromised endpoint may
      attempt to impersonate another endpoint to gain its privileges or
      to besmirch the reputation of that other endpoint.  This is of
      particular concern when using MAC addresses to identify endpoints,
      which, while widely used in endpoint behavior monitoring and
      threat assessment tools, are easy to spoof.

9.1.2.  Network Attacks

   Generally, the network cannot be trusted.  A variety of attacks can
   be mounted using the network, including:

   o  Eavesdropping, modification, injection, replay, deletion;

   o  Traffic analysis; and

   o  Denial of service and blocking traffic.

9.1.3.  Posture Manager Attacks

   The posture manager is a critical security element and therefore
   merits considerable scrutiny.  A variety of attacks can be leveraged
   against the Posture Manager.

   o  Compromised trusted manager: A compromised posture manager or a
      malicious party that is able to impersonate a posture manager can
      incorrectly grant or deny access to endpoints, place incorrect
      information into the repository, or send malicious messages to
      endpoints.

   o  Misconfiguration of posture manager: Accidental or purposeful
      misconfiguration of a trusted posture manager can cause effects
      that are similar to those listed for compromised trusted posture
      manager.

   o  Malicious untrusted posture manager: An untrusted posture manager
      cannot mount any significant attacks because all properly
      implemented endpoints will refuse to engage in any meaningful
      dialog with such a posture manager.

9.1.4.  Repository Attacks

   The repository is also an important security element and therefore
   merits careful scrutiny.

   o  Putting bad information into trusted repository: An authorized
      repository client such as a server may be able to put incorrect
      information into a trusted repository or delete or modify
      historical information, causing incorrect decisions about endpoint
      security.  Placing maliciously crafted data in the repository
      could even lead to compromise of repository clients, if they fail
      to carefully check such data.

   o  Compromised trusted repository: A compromised trusted repository
      or a malicious untrusted repository that is able to impersonate a
      trusted repository can lead to effects similar to those listed for
      "Putting bad information into trusted repository".  Further, a
      compromised trusted repository can report different results to
      different repository clients or deny access to the repository for
      selected repository clients.

   o  Misconfiguration of trusted repository: Accidental or purposeful
      misconfiguration of a trusted repository can deny access to the
      repository or result in loss of historical data.

   o  Malicious untrusted repository: An untrusted repository cannot
      mount any significant attacks because all properly implemented
      repository clients will refuse to engage in any meaningful dialog
      with such a repository.

9.2.  Countermeasures

   This section lists the countermeasures that can be used in a NEA
   implementation of an EPCP environment.

9.2.1.  Countermeasures for Endpoint Attacks

   This profile is in and of itself a countermeasure for a compromised
   endpoint.  A primary defense for an endpoint is to run up to date
   software configured to be run as safely as possible.

   Ensuring that anti-virus signatures are up to date and that a
   firewall is configured are also protections for an endpoint that are
   supported by the current NEA specifications.

   For secure device identification and to correlate device identifiers
   if the MAC address is randomized, MAC addresses should be collected
   along with other, more secure endpoint identifiers.  Endpoints that
   have hardware cryptographic modules that are provisioned by the
   enterprise, in accordance with [IEEE-802-1ar], can protect the
   private keys used for authentication and help prevent adversaries
   from stealing credentials that can be used for impersonation.  Future
   versions of the EPCP may want to discuss in greater detail how to use
   a hardware cryptographic module, in accordance with [IEEE-802-1ar],
   to protect credentials and to protect the integrity of the code that
   executes during the bootstrap process by hashing or recording
   indicators of compromise.

9.2.2.  Countermeasures for Network Attacks

   To address network attacks, [RFC6876] includes required encryption,
   authentication, integrity protection, and replay protection.
   [Server-Discovery] also includes authorization checks to ensure that
   only authorized servers are trusted by endpoints.  Any unspecified or
   not yet specified network protocols employed in the EPCP (e.g. the
   protocol used to interface with the repository) should include
   similar protections.

   These protections reduce the scope of the network threat to traffic
   analysis and denial of service.  Countermeasures for traffic analysis
   (e.g. masking) are usually impractical but may be employed.
   Countermeasures for denial of service (e.g. detecting and blocking
   particular sources) SHOULD be used when appropriate to detect and
   block denial of service attacks.  These are routine practices in
   network security.

9.2.3.  Countermeasures for Posture Manager Attacks

   Because of the serious consequences of posture manager compromise,
   posture managers SHOULD be especially well hardened against attack
   and minimized to reduce their attack surface.  They SHOULD be
   monitored using the NEA protocols to ensure the integrity of the
   behavior and analysis data stored on the posture manager and SHOULD
   utilize an [IEEE-802-1ar]-compliant hardware cryptographic module for
   identity and/or integrity measurements of the posture manager.  They
   should be well managed to minimize vulnerabilities in the underlying
   platform and in systems upon which the posture manager depends.
   Network security measures such as firewalls or intrusion detection
   systems may be used to monitor and limit traffic to and from the
   posture manager.  Personnel with administrative access to the posture
   manager should be carefully screened and monitored to detect problems
   as soon as possible.  Posture manager administrators should not use
   password-based authentication but should instead use non-reusable
   credentials and multi-factor authentication (where available).
   Physical security measures should be employed to prevent physical
   attacks on posture managers.

   To ease detection of posture manager compromise, should it occur,
   posture manager behavior should be monitored to detect unusual
   behavior (such as a server reboot, unusual traffic patterns, or other
   odd behavior).  Endpoints should log and/or notify users and/or
   administrators when peculiar posture manager behavior is detected.
   To aid forensic investigation, permanent read-only audit logs of
   security-relevant information pertaining to posture manager
   (especially administrative actions) should be maintained.  If posture
   manager compromise is detected, the posture manager's certificate
   should be revoked and careful analysis should be performed of the
   source and impact of this compromise.  Any reusable credentials that
   may have been compromised should be reissued.

   Endpoints can reduce the threat of server compromise by minimizing
   the number of trusted posture managers, using the mechanisms
   described in [Server-Discovery].

9.2.4.  Countermeasures for Repository Attacks

   If the host for the repository is located on its own endpoint, it
   should be protected with the same measures taken to protect the
   posture manager.  In this circumstance, all messages between the
   posture manager and repository should be protected with a mature
   security protocol such as TLS or IPsec.

   The repository can aid in the detection of compromised endpoints if
   an adversary cannot tamper with its contents.  For instance, if an

endpoint reports that it does not have an application with a known
vulnerability installed, an administrator can check whether the
endpoint might be lying by querying the repository for the history of
what applications were installed on the endpoint.

To help prevent tampering with the information in the repository:

1.  Only authorized parties should have privilege to run code on the
    endpoint and to change the repository.

2.  If a separate endpoint hosts the repository, then the
    functionality of that endpoint should be limited to hosting the
    repository.  The firewall on the repository should only allow
    access to the posture manager and to any endpoint authorized for
    administration.

3.  The repository should ideally use "write once" media to archive
    the history of what was placed in the repository, to include a
    snapshot of the current status of applications on endpoints.

10.  Privacy Considerations

The EPCP specifically addresses the collection of posture data from
enterprise endpoints by an enterprise network.  As such, privacy is
not going to often arise as a concern for those deploying this
solution.

A possible exception may be the concerns a user may have when
attempting to connect a personal endpoint (such as a phone or mobile
endpoint) to an enterprise network.  The user may not want to share
certain details, such as an endpoint identifier or SWID tags, with
the enterprise.  The user can configure their NEA client to reject
requests for this information; however, it is possible that the
enterprise policy will not allow the user's endpoint to connect to
the network without providing the requested data.

An enterprise network should limit access to endpoint posture and
identification information to authorized users.

11.  References

11.1.  Informative References

   [CIS]       http://www.cisecurity.org/controls/, "CIS Critical
               Security Controls".

   [DSD]          http://www.dsd.gov.au/publications/csocprotect/
                  top_4_mitigations.htm, "Top 4 Mitigation Strategies to
                  Protect Your ICT System", November 2012.

   [ECP]          Trusted Computing Group, "TCG Trusted Network Connect
                  Endpoint Compliance Profile, Version 1.10", December 2014.

   [IEEE-802-1ar]
                  Institute of Electrical and Electronics Engineers, "IEEE
                  802.1ar", December 2009.

   [RFC5209]      Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J.
                  Tardo, "Network Endpoint Assessment (NEA): Overview and
                  Requirements", RFC 5209, DOI 10.17487/RFC5209, June 2008,
                  <https://www.rfc-editor.org/info/rfc5209>.

   [TNC]          Trusted Computing Group, "TCG Trusted Network Connect TNC
                  Architecture for Interoperability, Version 1.5", February
                  2012.

## 11.2.  Normative References

   [I-D.ietf-mile-xmpp-grid]
                  Cam-Winget, N., Appala, S., Pope, S., and P. Saint-Andre,
                  "Using XMPP for Security Information Exchange", draft-
                  ietf-mile-xmpp-grid-04 (work in progress), October 2017.

   [I-D.ietf-netconf-restconf-notif]
                  Voit, E., Rahman, R., Nilsen-Nygaard, E., Clemm, A., and
                  A. Bierman, "Dynamic subscription to YANG Events and
                  Datastores over RESTCONF", draft-ietf-netconf-restconf-
                  notif-15 (work in progress), June 2019.

   [I-D.ietf-netconf-subscribed-notifications]
                  Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and
                  A. Tripathy, "Customized Subscriptions to a Publisher's
                  Event Streams", draft-ietf-netconf-subscribed-
                  notifications-13 (work in progress), June 2018.

   [I-D.ietf-netconf-yang-push]
                  Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-
                  Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore
                  Subscription", draft-ietf-netconf-yang-push-12 (work in
                  progress), December 2017.

   [I-D.ietf-sacm-terminology]
             Waltermire, D., Montville, A., Harrington, D., and N. Cam-
             Winget, "Terminology for Security Assessment", draft-ietf-
             sacm-terminology-05 (work in progress), August 2014.

   [IF-IMC]    Trusted Computing Group, "TCG Trusted Network Connect TNC
             IF-IMC, Version 1.3", February 2013.

   [IF-IMV]    Trusted Computing Group, "TCG Trusted Network Connect TNC
             IF-IMV, Version 1.4", December 2014.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
             Requirement Levels", BCP 14, RFC 2119,
             DOI 10.17487/RFC2119, March 1997,
             <https://www.rfc-editor.org/info/rfc2119>.

   [RFC5792]   Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute
             (PA) Protocol Compatible with Trusted Network Connect
             (TNC)", RFC 5792, DOI 10.17487/RFC5792, March 2010,
             <https://www.rfc-editor.org/info/rfc5792>.

   [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, DOI 10.17487/RFC5793,
             March 2010, <https://www.rfc-editor.org/info/rfc5793>.

   [RFC6241]   Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
             and A. Bierman, Ed., "Network Configuration Protocol
             (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
             <https://www.rfc-editor.org/info/rfc6241>.

   [RFC6242]   Wasserman, M., "Using the NETCONF Protocol over Secure
             Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
             <https://www.rfc-editor.org/info/rfc6242>.

   [RFC6876]   Sangster, P., Cam-Winget, N., and J. Salowey, "A Posture
             Transport Protocol over TLS (PT-TLS)", RFC 6876,
             DOI 10.17487/RFC6876, February 2013,
             <https://www.rfc-editor.org/info/rfc6876>.

   [RFC7317]   Bierman, A. and M. Bjorklund, "A YANG Data Model for
             System Management", RFC 7317, DOI 10.17487/RFC7317, August
             2014, <https://www.rfc-editor.org/info/rfc7317>.

   [RFC7589]  Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the
              NETCONF Protocol over Transport Layer Security (TLS) with
              Mutual X.509 Authentication", RFC 7589,
              DOI 10.17487/RFC7589, June 2015,
              <https://www.rfc-editor.org/info/rfc7589>.

   [RFC7950]  Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
              RFC 7950, DOI 10.17487/RFC7950, August 2016,
              <https://www.rfc-editor.org/info/rfc7950>.

   [RFC8040]  Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
              Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
              <https://www.rfc-editor.org/info/rfc8040>.

   [RFC8412]  Schmidt, C., Haynes, D., Coffin, C., Waltermire, D., and
              J. Fitzgerald-McKay, "Software Inventory Message and
              Attributes (SWIMA) for PA-TNC", RFC 8412,
              DOI 10.17487/RFC8412, July 2018,
              <https://www.rfc-editor.org/info/rfc8412>.

   [Server-Discovery]
              Trusted Computing Group, "DRAFT: TCG Trusted Network
              Connect PDP Discovery and Validation, Version 1.0",
              October 2015.

   [SWID]     "Information technology--Software asset management--Part
              2: Software identification tag", ISO/IEC 9899:1999, 2009.

Appendix A.  Rationale for an EPCP Solution

A.1.  Preventative Posture Assessments

   The value of continuous endpoint posture assessment is well
   established.  Security experts have identified asset management and
   vulnerability remediation as a critical step for preventing
   intrusions.  Application whitelisting, patching applications and
   operating systems, and using the latest versions of applications top
   the Defense Signals Directorate's "Top 4 Mitigations to Protect Your
   ICT System".  [DSD] "Inventory of Authorized and Unauthorized
   Endpoints", "Inventory of Authorized and Unauthorized Software", and
   "Continuous Vulnerability Assessment and Remediation" are Controls 1,
   2, and 3, respectively, of the CIS Controls [CIS].  While there are
   commercially available solutions that attempt to address these
   security controls, these solutions do not run on all types of
   endpoints; consistently interoperate with other tools that could make
   use of the data collected; collect posture information from all types
   of endpoints in a consistent, standardized schema; or require vetted,

standardized protocols that have been evaluated by the international
community for cryptographic soundness.

As is true of most solutions offered today, the solution found in the
EPCP does not attempt to solve the lying endpoint problem, or detect
infected endpoints; rather, it focuses on ensuring that healthy
endpoints remain healthy by keeping software up-to-date and patched.

A.2.  All Network-Connected Endpoints are Endpoints

As defined by [I-D.ietf-sacm-terminology], an endpoint is any
physical or virtual computing endpoint that can be connected to a
network.  Posture assessment against policy is equally, if not more,
important for continuously connected endpoints, such as enterprise
workstations and infrastructure endpoints, as it is for sporadically
connected endpoints.  Continuously connected endpoints are just as
likely to fall out of compliance with policy, and a standardized
posture assessment method is necessary to ensure they can be properly
handled.

A.3.  All Endpoints on the Network Must be Uniquely Identified

Many administrators struggle to identify what endpoints are connected
to the network at any given time.  By requiring a standardized method
of endpoint identity, the EPCP will enable administrators to answer
the basic question, "What is on my network?"  In
[I-D.ietf-sacm-terminology], SACM defines this set of endpoints on
the network as the SACM domain.  Unique endpoint identification also
enables the comparison of current and past endpoint posture
assessments, by allowing administrators to correlate assessments from
the same endpoint.  This makes it easier to flag suspicious changes
in endpoint posture for manual or automatic review, and helps to
swiftly identify malicious changes to endpoint applications.

A.4.  Standardized Data Models

Meeting EPCP best practices requires the use of standardized data
models for the exchange of posture information.  This helps to ensure
that the posture information sent from endpoints to the repository
can be easily stored, due to their known format, and shared with
authorized endpoints and users.

Posture information must be sent over standardized protocols to
ensure the confidentiality and authenticity of this data while in
transit.  Implementations of the EPCP include [RFC6876] and [RFC6241]
for communication between the target endpoint and the posture
manager.  These protocols allow networks that implement this solution
to collect large amounts of posture information from an endpoint to

make decisions about that endpoint's compliance with some policy.
The EPCP offers a solution for all endpoints already connected to the
network.  Periodic assessments and automated reporting of changes to
endpoint posture allow for instantaneous identification of connected
endpoints that are no longer compliant to some policy.

A.5.  Posture Information Must Be Stored

   Posture information must be stored by the repository and must be
   exposed to an interface at the posture manager.  Standard data models
   enable standard queries from an interface exposed to an administrator
   at the posture manager console.  A repository must retain any current
   posture information retrieved from the target endpoint and store it
   indexed by the unique identifier for the endpoint.  Any posture
   collection manager specified by this profile must be able to
   ascertain from its corresponding posture collection engine whether
   the posture information is up to date.  An interface on the posture
   manager must support a request to obtain up-to-date information when
   an endpoint is connected.  This interface must also support the
   ability to make a standard set of queries about the posture
   information stored by the repository.  In the future, some forms of
   posture information might be retained at the endpoint.  The interface
   on the posture manager must accommodate the ability to make a request
   to the corresponding posture collection engine about the posture of
   the target endpoint.  Standard data models and protocols also enable
   the security of posture assessment results.  By storing these results
   indexed under the endpoint's unique identification, secure storage
   itself enables endpoint posture information correlation, and ensures
   that the enterprise's repositories always offer the freshest, most
   up-to-date view of the enterprise's endpoint posture information
   possible.

A.6.  Posture Information Can Be Shared

   By exposing posture information using a standard interface and API,
   other security and operational components have a high level of
   insight into the enterprise's endpoints and the software installed on
   them.  This will support innovation in the areas of asset management,
   vulnerability scanning, and administrative interfaces, as any
   authorized infrastructure endpoint can interact with the posture
   information.

A.7.  Enterprise Asset Posture Information Belongs to the Enterprise

   Owners and administrators must have complete control of posture
   information, policy, and endpoint mitigation.  Standardized data
   models, protocols and interfaces help to ensure that this posture
   information is not locked in proprietary databases, but is made

available to its owners.  This enables administrators to develop as
nuanced a policy as necessary to keep their networks secure.  Of
course, there may be exceptions to this such as the case with
privacy-related information (e.g., personally identifiable
information).

Appendix B.  EPCP Supported Use Cases and Non-Supported Use Cases

B.1.  Supported Use Cases

   The following sections describe the different use cases supported by
   the EPCP.

B.1.1.  Hardware Asset Management

   Using the administrative interface on the posture manager, an
   authorized user can learn:

   o  what endpoints are connected to the network at any given time; and

   o  what SWID tags were reported for the endpoints.

   The ability to answer these questions offers a standards-based
   approach to asset management, which is a vital part of enterprise
   processes such as compliance report generation for the Federal
   Information Security Modernization Act (FISMA), Payment Card Industry
   Data Security Standard (PCI DSS), Health Insurance Portability and
   Accountability Act (HIPAA), etc.

B.1.2.  Software Asset Management

   The administrative interface on the posture manager provides the
   ability for authorized users and infrastructure to know which
   software is installed on which endpoints on the enterprise's network.
   This allows the enterprise to answer questions about what software is
   installed to determine if it is licensed or prohibited.  This
   information can also drive other use cases such as:

   o  vulnerability management: knowing what software is installed
      supports the ability to determine which endpoints contain
      vulnerable software and need to be patched.

   o  configuration management: knowing which security controls need to
      be applied to harden installed software and better protect
      endpoints.

B.1.3.  Vulnerability Management

   The administrative interface also provides the ability for authorized
   users or infrastructure to locate endpoints running software for
   which vulnerabilities have been announced.  Because of

   1.  the unique IDs assigned to each endpoint; and

   2.  the rich application data provided in the endpoints' posture
       information,

   the repository can be queried to find all endpoints running a
   vulnerable application.  Endpoints suspected of being vulnerable can
   be addressed by the administrator or flagged for further scrutiny.

B.1.4.  Threat Detection and Analysis

   The repository's standardized API allows authorized infrastructure
   endpoints and software to search endpoint posture assessment
   information for evidence that an endpoint's software inventory has
   changed, and can make endpoint software inventory data available to
   other endpoints.  This automates security data sharing in a way that
   expedites the correlation of relevant network data, allowing
   administrators and infrastructure endpoints to identify odd endpoint
   behavior and configuration using secure, standards-based data models
   and protocols.

B.2.  Non-Supported Use Cases

   Several use cases, including but not limited to these, are not
   covered by the EPCP:

   o  Gathering non-standardized types of posture information: The EPCP
      does not prevent administrators from collecting posture
      information in proprietary formats from the endpoint; however it
      does not set requirements for doing so.

   o  Solving the lying endpoint problem: The EPCP does not address the
      lying endpoint problem; the Profile makes no assertions that it
      can catch an endpoint that is, either maliciously or accidentally,
      reporting false posture information to the posture manager.
      However, other solutions may be able to use the posture
      information collected using the capabilities described in this
      profile to catch an endpoint in a lie.  For example, a sensor may
      be able to compare the posture information it has collected on an
      endpoint's activity on the network to what the endpoint reported
      to the server and flag discrepancies.  However, these capabilities
      are not described in this profile.

Appendix C.  Endpoint Posture Collection Profile Examples

   The following subsections provide examples of the EPCP as implemented
   using components from the NEA architecture.

C.1.  Continuous Posture Assessment of an Endpoint

```
              Endpoint                  Posture Manager
              +---------------+         +---------------+
              |               |         |               |
              | +----------+  |         | +----------+  |
              | | SWID     |  |         | | SWID     |  |
              | | Posture  |  |         | | Posture  |  |
              | | Collector|  |         | | Validator|  |
              | +----------+  |         | +----------+  |
              |      |        |         |      |        |
              |      |  IF-IMC |        |      |  IF-IMV |
              |      |        |         |      |        |
              | +----------+  |         | +----------+  |
              | | PB Client|  |         | | PB Server|  |
              | +----------+  |         | +----------+  |
              |      |        |         |      |        |
              |      |        |         |      |        |
              |      |        |         |      |        |
              | +----------+  |         | +----------+  |
              | | PT Client|  | <------> | | PT Server|  |
              | +----------+  |         | +----------+  |
              |               | PT-TLS  |               |
              |               |         |               |
              +---------------+         +---------------+
```
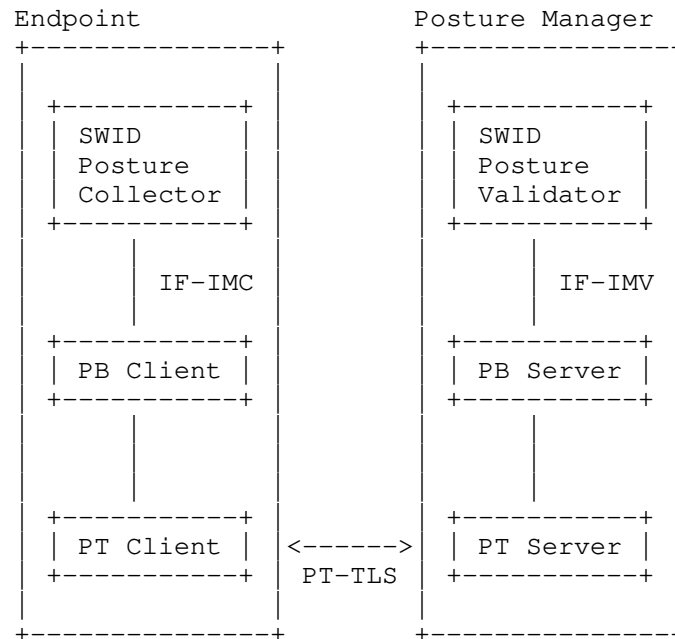
            Figure 4: Continuous Posture Assessment of an Endpoint

C.1.1.  Change on Endpoint Triggers Posture Assessment

   A new application is installed on the endpoint, and the SWID
   directory is updated.  This triggers an update from the SWID posture
   collector to the SWID posture validator.  The message is sent down
   the NEA stack, encapsulated by NEA protocols until it is sent by the
   posture transport client to the posture transport server.  The
   posture transport server then forwards it up through the stack, where
   the layers of encapsulation are removed until the SWID Message
   arrives at the SWID posture validator.

```
                Endpoint                      Posture Manager
                +--------------+              +--------------+
                |              |              |              |
                | +----------+ |              | +----------+ |
                | | SWID     | |              | | SWID     | |
                | | Posture  | |              | | Posture  | |
                | | Collector| |              | | Validator| |
                | +----------+ |              | +----------+ |
                |      |       |  SWID Message |      |       |
                |      | IF-IMC|  for PA-TNC   |      | IF-IMV|
                |      |       |              |      |       |
                | +----------+ |              | +----------+ |
                | | PB Client| |              | | PB Server| |
                | +----------+ |              | +----------+ |
                |      |       |              |      |       |
                |      |       |  PB-TNC {SWID |      |       |
                |      |       |  Message for  |      |       |
                |      |       |  PA-TNC}      |      |       |
                |      |       |              |      |       |
                | +----------+ |              | +----------+ |
                | | PT Client| | <-------------->| | PT Server| |
                | +----------+ |  PT-TLS {PB-TNC| +----------+ |
                |              |  {SWID Message |              |
                +--------------+  for PA-TNC}}  +--------------+
```
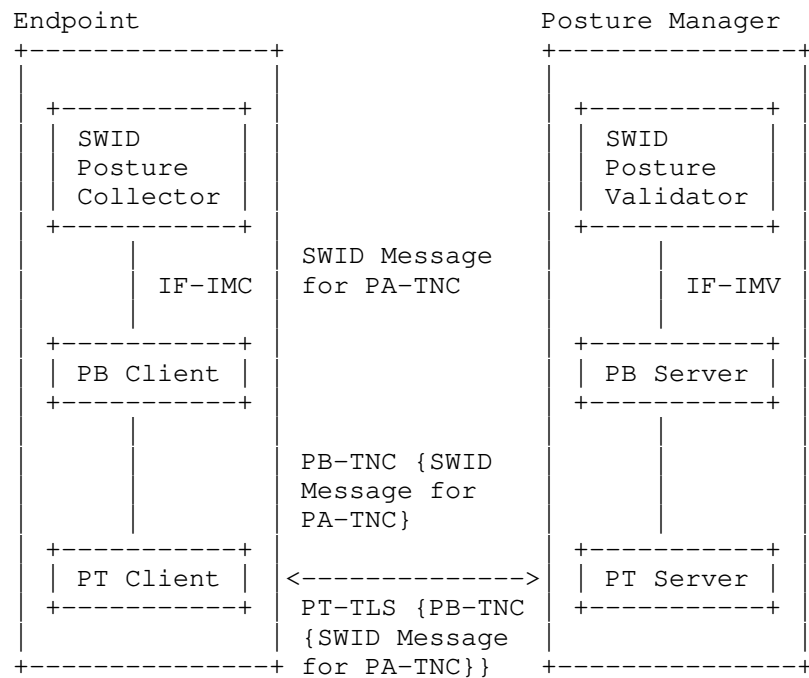
                Figure 5: Compliance Protocol Encapsulation

   The SWID posture validator stores the new tag information in the
   repository.  If the tag indicates that the endpoint is compliant to
   the policy, then the process is complete until the next time an
   update is needed (either because policy states that the endpoint must
   submit posture assessment results periodically or because an
   install/uninstall/update on the endpoint triggers a posture
   assessment).

```
              Endpoint                    Posture Manager
              +--------------+            +--------------+
              |              |            |              |
              | +----------+ |            | +----------+ |
              | | SWID     | |            | | SWID     |-|-+
              | | Posture  | |            | | Posture  | | |
              | | Collector| |            | | Validator| | |
              | +----------+ |            | +----------+ | |
              |      |       |            |      |       | |        Repository
              |      | IF-IMC |           |      | IF-IMV | |        +--------+
              |      |       |            |      |       | |        |        |
              | +----------+ |            | +----------+ | |        |        |
              | | PB Client| |            | | PB Server| | +---->|  |        |
              | +----------+ |            | +----------+ | |        |        |
              |      |       |            |      |       | |        +--------+
              |      |       |            |      |       | |
              |      |       |            |      |       | |
              | +----------+ |            | +----------+ | |
              | | PT Client| | <------>|  | | PT Server| | |
              | +----------+ |  PT-TLS    | +----------+ | |
              |              |            |              | |
              +--------------+            +--------------+
```

                Figure 6: Storing SWIDs in the Repository

   If the endpoint has fallen out of compliance with a policy, the
   posture manager can alert the administrator via the posture manager's
   administrative interface.  The administrator can then take steps to
   address the problem.  If the administrator has already established a
   policy for automatically addressing this problem, that policy will be
   followed.

```
                                                             (")
                                                             __|__
                                                       +-->|
                  Endpoint                Posture Manager  |    / \
                  +---------------+        +---------------+ |
                  |               |        |               | |
                  | +-----------+ |        | +-----------+ | |
                  | | SWID      | |        | | SWID      |-|-+
                  | | Posture   | |        | | Posture   | |
                  | | Collector | |        | | Validator | |
                  | +-----------+ |        | +-----------+ |            Repository
                  |      |        |        |      |        |            +--------+
                  |      | IF-IMC |        |      | IF-IMV |            |        |
                  |      |        |        |      |        |            |        |
                  | +-----------+ |        | +-----------+ |            |        |
                  | | PB Client | |        | | PB Server | |            +--------+
                  | +-----------+ |        | +-----------+ |
                  |      |        |        |      |        |
                  |      |        |        |      |        |
                  |      |        |        |      |        |
                  | +-----------+ |        | +-----------+ |
                  | | PT Client | | <------>| | PT Server | |
                  | +-----------+ | |      | +-----------+ |
                  |               | | PT-TLS |               |
                  +---------------+        +---------------+
```
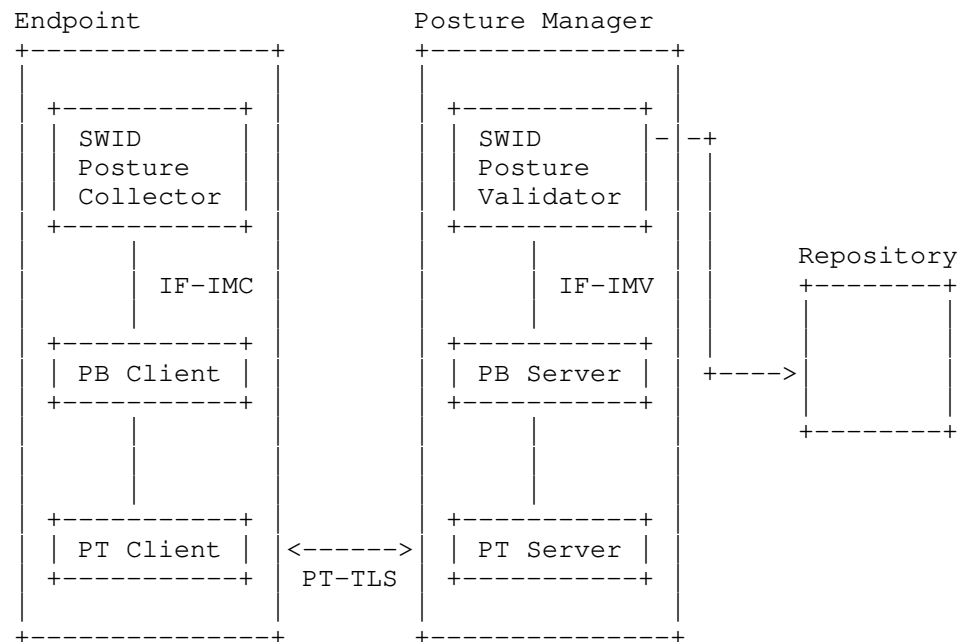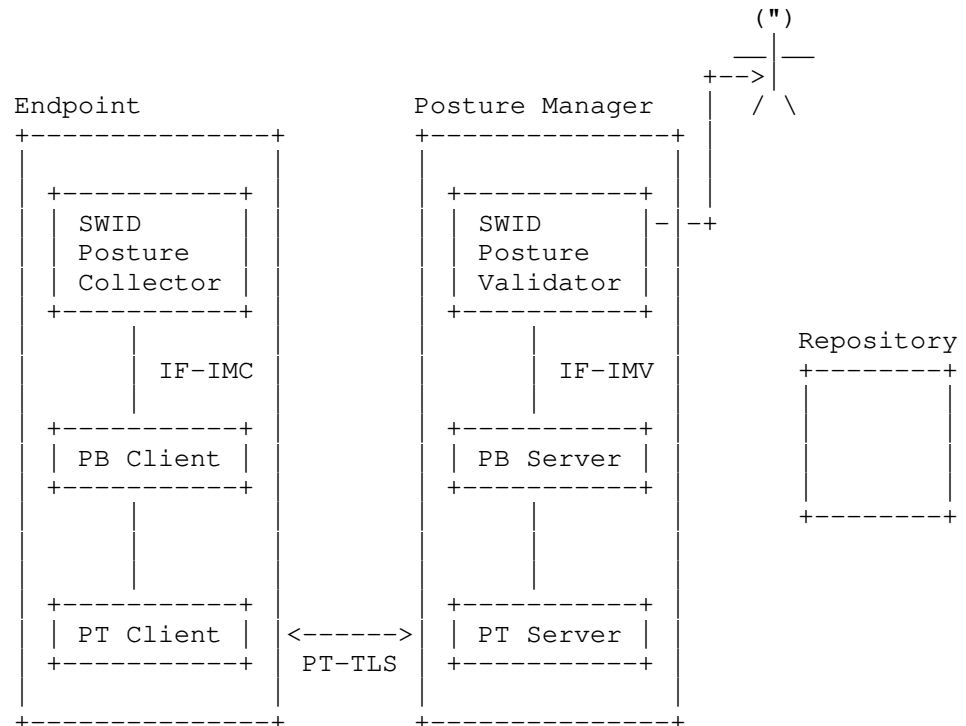
Figure 7: Server Alerts Network Admin

C.2.  Administrator Searches for Vulnerable Endpoints

   An announcement is made that a particular version of a piece of
   software has a vulnerability.  The administrator uses the
   administrative interface on the server to search the repository for
   endpoints that reported the SWID tag for the vulnerable software.

```
                                                       (")
                                                      __|__
                                                 +-->|
        Endpoint                 Posture Manager  |    / \
        +--------------+         +--------------+  |
        |              |         |              |  |
        |  +---------+ |         |  +---------+ |  |
        |  | SWID    | |         |  | SWID    |-|-+
        |  | Posture | |         |  | Posture | |
        |  | Collector| |        |  | Validator| |
        |  +---------+ |         |  +---------+ |       Repository
        |      |       |         |      |       |      +--------+
        |      | IF-IMC |        |      | IF-IMV |      |        |
        |      |       |         |      |       |      |        |
        |  +---------+ |         |  +---------+ |      |        |
        |  | PB Client| |        |  | PB Server| |----->|        |
        |  +---------+ |         |  +---------+ |      |        |
        |      |       |         |      |       |      +--------+
        |      |       |         |      |       |
        |      |       |         |      |       |
        |  +---------+ |         |  +---------+ |
        |  | PT Client| |<------>|  | PT Server| |
        |  +---------+ | PT-TLS  |  +---------+ |
        |              |         |              |
        +--------------+         +--------------+
```
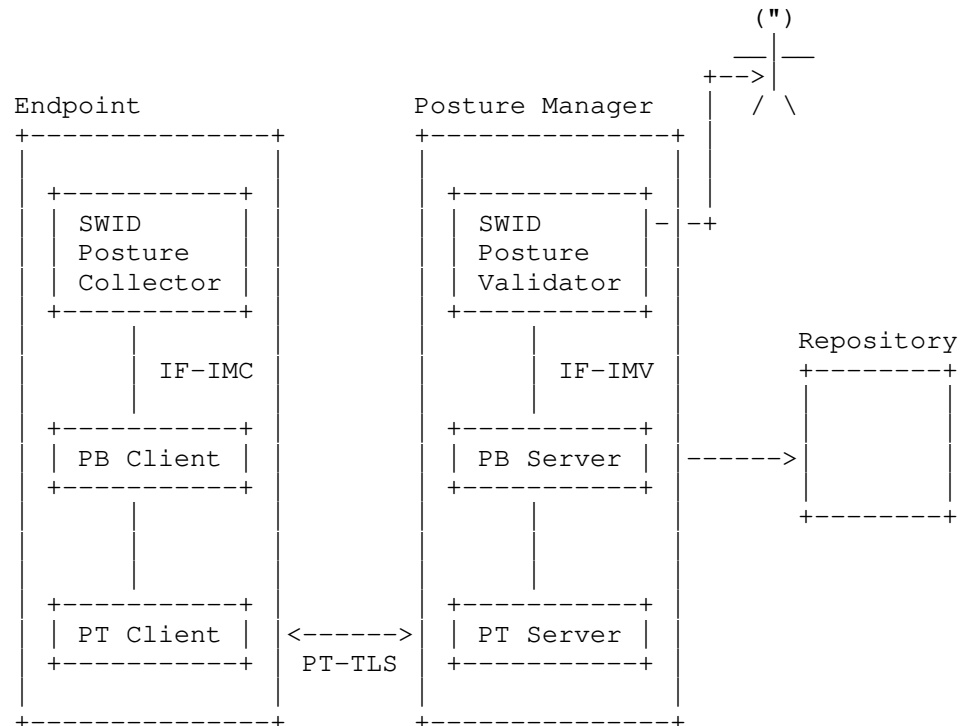
              Figure 8: Admin Searches for Vulnerable Endpoints

   The repository returns a list of entries in the matching the
   administrator's search.  The administrator can then address the
   vulnerable endpoints by taking some follow-up action such as removing
   it from the network, quarantining it, or updating the vulnerable
   software.

Appendix D.  Change Log

D.1.  -04 to -05

   Updated the diagram so the Evaluator and Repository are "current
   work".

   Clarified how the Posture Collection Engine can push data, respond to
   queries, and establish secure transport connectivity for fulfilling
   subscriptions.

   Expanded on the future work around leveraging NETCONF, RESTCONF, and
   YANG Push for network devices.

   Documented the need to reassess MAC addresses as a device identifier.

   Made various typographical and editorial changes.

D.2.  -03 to -04

   Addressed various comments from the SACM WG.

   Refactored the document to better focus it on the communications
   between endpoints and the posture manager and the best practices for
   EPCP implementations.

   Made other editorial changes and improved consistency throughout the
   document.

D.3.  -02 to -03

   Addressed various comments from the SACM WG.

   Added a reference to TCG ECP 1.0.

   Removed text in the "SWID Posture Validator" section that states it
   performs evaluation.  This was removed because it contradicts the
   posture manager not performing any evaluations.

   Expanded the "Provisioning" section of the "EPCP Transactions"
   section to include examples of endpoint identifiers and the need to
   provision endpoints with components and data models.

   Combined text for the capabilities of the Administrative Interface
   and API.

   Removed superfluous and introductory text from the "Security
   Considerations" section.

   Renamed section "Vulnerability Searches" to Vulnerability
   Management".

   Changed I-D category to BCP.

   Changed references to the NETMOD architecture to the NETCONF
   architecture because NETCONF represents the management protocol
   whereas NETMOD is focused on the definition of data models.

   Addressed various editorial suggestions.

D.4.  -01 to -02

   Addressed various comments from the SACM WG.

   Added a section for the collection of posture information from
   network devices using standards from the NETMOD WG.

   Updated EPCP component diagrams so they were not specific to a NEA-
   based implementation.

   Updated EPCP NEA example diagrams to reflect all the components in
   the NEA architecture.

D.5.  -00 to -01

   There are no textual changes associated with this revision.  This
   revision simply reflects a resubmission of the document so that it
   remains in active status.

D.6.  -01 to -02

   Added references to the Software Inventory Message and Attributes
   (SWIMA) for PA-TNC I-D.

   Replaced references to PC-TNC with IF-IMC.

   Removed erroneous hyphens from a couple of section titles.

   Made a few minor editorial changes.

D.7.  -02 to -00

   Draft adopted by IETF SACM WG.

D.8.  -00 to -01

   Significant edits to up-level the draft to describe SACM collection
   over multiple different protocols.

   Replaced references to SANS with CIS.

   Made other minor editorial changes.

Authors' Addresses

Danny Haynes
The MITRE Corporation
202 Burlington Road
Bedford, MA  01730
USA


Email: dhaynes@mitre.org



Jessica Fitzgerald-McKay
Department of Defense
9800 Savage Road
Ft. Meade, Maryland
USA


Email: jmfitz2@nsa.gov



Lisa Lorenzin
Pulse Secure
2700 Zanker Rd., Suite 200
San Jose, CA  95134
US


Email: llorenzin@pulsesecure.net