RATS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2021

E. Voit
Cisco

October 02, 2020

Trusted Path Routing
draft-voit-rats-trustworthy-path-routing-01

Abstract

There are end-users who believe encryption technologies like IPSec alone are insufficient to protect the confidentiality of their highly sensitive traffic flows. These end-users want their flows to traverse devices which have been freshly appraised and verified. This specification describes Trusted Path Routing. Trusted Path Routing protects sensitive flows as they transit a network by forwarding traffic to/from sensitive subnets across network devices recently appraised as trustworthy.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

There are end-users who believe encryption technologies like IPSec alone are insufficient to protect the confidentiality of their highly sensitive traffic flows. These customers want their highly sensitive flows to be transported over only network devices recently verified as trustworthy.

With the inclusion of TPM based cryptoprocessors into network devices, it is now possible for network providers to identify potentially compromised devices as well as potentially exploitable (or even exploited) vulnerabilities. Using this knowledge, it then becomes possible to redirect sensitive flows around these devices.

Trusted Path Routing provides a method of establishing Trusted Topologies which only include trust-verified network devices. Membership in a Trusted Topology is established and maintained via an

exchange of Stamped Passports at the link layer between peering network devices. As links to Attesting Devices are appraised as meeting at least a minimum set of formally defined Trustworthiness Levels, the links are then included as members of this Trusted Topology. Routing protocols like [I-D.ietf-lsr-flex-algo] can then used to propagate topology state throughout a network. IP Packets to and from end-user designated Sensitive Subnets are then forwarded into this Trusted Topology at each network boundary.

The specification works under the following assumptions:

- o A set of network devices supporting the TPM remote attestation profile as laid out in [RATS-Device] are connected within a network domain.

- o A routing protocol capable of maintaining multiple forwarding topologies connects these network devices.

- o One or more Verifiers continuously appraise each of network devices, and these Verifiers can return the Attestation Results back to the attesting network device.

2. Terminology

2.1. Terms

The following terms are imported from [RATS-Arch]: Attester, Evidence, Passport, Relying Party, and Verifier.

Newly defined terms for this document:

Attested Device - a device where a Verifier's most recent appraisal of Evidence has returned a Trustworthiness Vector.

Stamped Passport - a bundle of Evidence which includes at least signed Attestation Results from a Verifier, and two independent TPM quotes from an Attester.

Sensitive Subnet - an IP address range where IP packets to or from that range desire confidentially guarantees beyond those of non-identified subnets. In practice, flows to or from a Sensitive Subnet must only have their IP headers and encapsulated payloads accessible/visible only by Attested Devices supporting one or more Trustworthiness Vectors.

Transparently-Transited Device - a network device within an network domain where any packets originally passed into that network

domain are completely opaque on that network device at Layer 3 and above.

Trusted Topology - a topology which includes only Attested Devices and Transparently-Transited Devices.

Trustworthiness Level - a specific quanta of trustworthiness which can be assigned by a Verifier.

Trustworthiness Vector - a set of Trustworthiness Levels assigned during a single assessment cycle by a Verfier using Evidence and Claims related to an Attested Device. The vector is included within Attestation Results.

2.2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Protocol Independent Definitions

3.1. Trusted Path Routing Service

An end user identifies sensitive IP subnets where flows with applications using these IP subnets need enhanced privacy guarantees. Trusted Path Routing passes flows to/from these Sensitive Subnets over a Trusted Topology able to meet these guarantees. The Trusted Topology itself consists of the interconnection of network devices where each potentially transited device has passed a recent trustworthiness appraisal.

Different guarantees of end-to-end trustworthiness appraisal may be offered to network users. These guarantees are network operator specific, but might include options such as:

- all transited devices are currently boot integrity verified

- all transited devices are from a specific set of vendors and are running known software containing the latest patches

- no guarantees provided

3.2. Network Topology Assembly

To be included in a Trusted Topology, Evidence of trustworthiness is shared between network device peers (such as routers). Upon receiving and appraising this Evidence as part of link layer authentication, the network device peer decides if this link should be added as an active adjacency for the Trusted Topology.

When enough links have been successfully added, a Trusted Topology will come into existence as routing protocols flood the adjacency information across the network domain.

```
              .-------------.             .---------.
              | Compromised |             | Edge    |
 .---------.  |    Router   |             | Router  |
 | Router  |  |             |             |         |
 |         |  |        trust>-------------<no_trust |
 | no_trust>--<trust        | .--------.  |         |----Sensitive
 |         |  '-------------' |   trust>==<trust    |    Subnet
 |    trust>==================<trust   |  |         |
 '---------'                  |        |  '---------'
                              | Router |
                              '--------'
```

Figure 1: Trusted Path Topology Assembly

Traffic exchanged with Sensitive Subnets can then be forwarded into that Trusted Topology from all edges of the network domain.

3.3. Link Appraisal

Critical to the establishment and maintenance of a Trusted Topology is the Stamped Passport. A Stamped Passport is comprised of Evidence from both an Attester and a Verifier. Stamped Passports are exchanged between adjacent network devices over a link layer protocols like 802.1x or MACSEC. As both sides of a link may need might need to appraise the other, independent Stamped Passports will often be transmitted from either side of the link. Additionally, as link layer protocols will continuously re-authenticate the link, this allows for fresh Stamped Passports to be constantly appraised by either side of the connection.

Each Stamped Passport will include the most recent Verifier provided Attestation Results, as well as the most recent TPM Quote for that Attester. Upon receiving this information as part of link layer authentication, the Relying Party Router appraises the results and decides if this link should be added to a Trusted Topology.

Figure 2 describes this flow of information using the time definitions described in [RATS-Arch], and the information flows defined in Section 7 of [RATS-Interactions].

```
     .----------.                .----------.    .---------------.
     | Attester |                | Verifier |    | Relying Party |
     |          |                |     A    |    |  / Verifier B |
     | (Router) |                |          |    |    (Router)   |
     '----------'                '----------'    '---------------'
        time(VG)                       |                 |
          |<------nonce--------------time(NS)            |
          |                            |                 |
 time(EG)(1)------Evidence------------>|                 |
          |                          time(RG)            |
          |<------Attestation Result--(2)                |
          ~                            ~                 ~
        time(VG')?                     |                 |
          ~                            ~                 ~
          |<------nonce---------------------------------(3)time(NS')
          |                            |                 |
time(EG')(4)------Stamped Pat---------------------------->|
          |                            |                (5)time(RG',RA')
                                                        (6)
                                                         ~
                                                      time(RX')
```

Figure 2: Trusted Path Timing

Specifics for each one of these information flows, including details on what happens at the items numbered (1) through (5) are described in Section 3.6.

3.4. Trustworthiness Vector

For Trusted Path Routing to operate, fresh Attestation Results need to be communicated by a Verifier back to the Attester. These Attestation Results must be encoded in a way which is known and actionable.

To support this requirement, specific levels of appraised trustworthiness have been defined. These are known as Trustworthiness Levels. It is these Trustworthiness Levels which are asserted as part of the Attestation Results by a Verifier. It is out of the scope of this document for the Verifier to provide proof or logic on how the assertion was derived.

Following are the set of available Trustworthiness Levels:

Trustworthiness Level	Definition
hw-authentic	A Verifier has appraised an Attester as having authentic hardware
fw-authentic	A Verifier has appraised an Attester as having authentic firmware
hw-verification-fail	A Verifier has appraised an Attester has failed its hardware or firmware verification
identity-verified	A Verifier has appraised and verified an Attester's unique identity
identity-fail	A Verifier has been unable to assess or verify an Attester's unique identity
boot-verified	A Verifier has appraised an Attester as Boot Integrity Verified
boot-verification-fail	A Verifier has appraised an Attester has failed its Boot Integrity verification
files-verified	A Verifier has appraised an Attester's file system, and asserts that it recognizes relevant files
file-repudiated	A Verifier has found a file on an Attester which should not be present

A quick look at the list above shows that multiple Trustworthiness Level will often be applicable at single point in time. To support this, the Attestation Results will include a single Trustworthiness Vector consisting of a set of Trustworthiness Levels. The establishment of this Trustworthiness Vector follows the following logic on the Verifier:

```
Start: TPM Quote Received, log received, or appraisal timer expired

Step 0: set Trustworthiness Vector = Null

Step 1: Is there sufficient fresh signed evidence to appraise?
   (yes) - No Action
   (no) -  Goto Step 6

Step 2: Appraise Hardware Integrity
   (if hw-verification-fail) - push onto vector, go to Step 6
   (if hw-authentic) - push onto vector
   (if fw-authentic) - push onto vector
   (if not evaluated, or insufficient data to conclude: take no action)

Step 3: Appraise attester identity
   (if identity-verified) - push onto vector
   (if identity-fail) - push onto vector
   (if not evaluated, or insufficient data to conclude: take no action)

Step 4: Appraise boot integrity
   (if boot-verified) - push onto vector
   (if boot-verification-fail) - push onto vector
   (if not evaluated, or insufficient data to conclude: take no action)

Step 5: Appraise filesystem integrity
   (if files-verified) - push onto vector
   (if file-repudiated) - push onto vector
   (if not evaluated, or insufficient data to conclude: take no action)

Step 6: Assemble Attestation Results, and push to Attester

End
```

3.5. Attestation Results

As Evidence changes, a new Trustworthiness Vector needs to be returned to the Attester as Attestation Results. But this Trustworthiness Vector is not all that needs to be returned. Following is a YANG tree for all the returned objects. Each of these objects will later be used as Evidence by another Verifier which is co-resident with the Relying Party.

```
module: ietf-attestation-results-vector
  +--rw attestation-results!
     +--rw trustworthiness-vector*        identityref
     +--rw (tpm-specification-version)?
     |  +--:(TPM2.0) {taa:TPM20}?
     |  |  +--rw TPM2B_DIGEST             binary
     |  |  +--rw tpm20-pcr-bank* [TPM-hash-algo]
     |  |  |  +--rw TPM-hash-algo    identityref
     |  |  |  +--rw pcr-index*       tpm:pcr
     |  |  +--rw clock                    uint64
     |  |  +--rw reset-counter            uint32
     |  |  +--rw restart-counter          uint32
     |  |  +--rw safe                     boolean
     |  +--:(TPM1.2) {taa:TPM12}?
     |     +--rw pcr-index*               pcr
     |     +--rw tpm12-pcr-value*         binary
     |     +--rw timestamp                yang:date-and-time
     +--rw public-key-format              identityref
     +--rw public-key                     binary
     +--rw public-key-algorithm-type      identityref
     +--rw verifier-signature-key-name?   string
     +--rw verifier-key-algorithm-type    identityref
     +--rw verifier-signature             binary
```

Figure 3: Attestation Results Tree

Looking at the objects above, if the Attester has a TPM2, then the values of the TPM PCRs are included (i.e., <TPM2B_DIGEST>, <TPM2_Algo>, and <pcr-index>), as are the timing counters from the TPM (i.e., <clock>, <reset-counter>, <restart-counter>, and <safe>).

Likewise if the Attester has a TPM1.2, the TPM PCR values of the <pcr-index> and <pcr-value> are included. Timing information comes from the Verifier itself via the <timestamp> object.

For both the TPM1.2 and the TPM2, there are other Attestation Results which are sent. These are the Attester's TPM key (i.e., <public-key>, <public-key-format>, and <public-key-algorithm-type>). This key later will allow the Relying Party router to appraise a subsequent TPM Quote. It is this signature which allows the Trustworthiness Vector to be later provably associated with a recent TPM Quote.

3.6. Stamped Passport

The Attestation Results are not the only item which a Relying Party needs to consider during its appraisal. A provably recent TPM Quote from the Attester must also be included. With these two items, the

resulting Stamped Passports formats described below must be converted to CDDL and passed over EAP. If an Attester includes a TPM2, the objects are:

```
YANG structure for a TPM2 Stamped Passport
   +--ro latest-tpm-quote
   |  +--ro quote              binary
   |  +--ro quote-signature    binary
   +--ro latest-attestation-results
      +--ro trustworthiness-vector*        identityref
      +--ro TPM2B_DIGEST                   binary
      +--ro tpm20-pcr-bank* [TPM-hash-algo]
      |  +--ro TPM-hash-algo    identityref
      |  +--ro pcr-index*   tpm:pcr
      +--ro clock                          uint64
      +--ro reset-counter                  uint32
      +--ro restart-counter                uint32
      +--ro safe                           boolean
      +--ro public-key-format              identityref
      +--ro public-key                     binary
      +--ro public-key-algorithm-type      identityref
      +--ro verifier-signature-key-name?   string
      +--ro verifier-signature             binary
```

And if the Attester is a TPM1.2, the object are:

```
YANG structure for a TPM1.2 Stamped Passport
   +--ro latest-tpm-quote
   |  +--ro version* []
   |  |  +--ro major?      uint8
   |  |  +--ro minor?      uint8
   |  |  +--ro revMajor?   uint8
   |  |  +--ro revMinor?   uint8
   |  +--ro digest-value?   binary
   +--ro latest-tpm12-attestation-results
      +--ro trustworthiness-vector*        identityref
      +--ro pcr-index*                     pcr
      +--ro tpm12-pcr-value*               binary
      +--ro timestamp                      yang:date-and-time
      +--ro public-key-format              identityref
      +--ro public-key                     binary
      +--ro public-key-algorithm-type      identityref
      +--ro verifier-signature-key-name?   string
      +--ro verifier-signature             binary
```

With either of these passport formats, if the <latest-tpm-quote> is verifiably fresh, then the state of the Attester can be appraised by a network peer.

3.7. Appraising the Stamped Passport

When it receives a Stamped Passport, a Verifier co-resident with the Relying Party on a network peer can make nuanced decisions about how to handle traffic coming from that link. For example, when the Attester's TPM hardware identity credentials can be verified, it might choose to accept link layer connections and forward generic Internet traffic.

Additionally, if the Attester's Trustworthiness Vector is acceptable to the Relying Party, and it hasn't been too long since the Verifier has provided a Stamped Passport, the Relying Party can include that link in a Trusted Topology.

As the process described above repeats across the set of links within a network domain, Trusted Topologies can be extended and maintained. Traffic to and from Sensitive Subnets is then identified at the edges of the network domain and passed into this Trusted Topology.

```
       .--------------.
       |  Verifier A  |
       '---------(2)--'
           ^      |
           |     Attestation Results
      Evidence    |
           |      V
        .-(1)---------.                          .---------------.
        | Attester    |                          | Relying Party |
        |  (Router)   |<--------------------nonce(3) / Verifier B |
        |  .-----.    |                          |   (Router)    |
        |  | TPM |   (4)-Stamped Passport-------->|               |
        |  '-----'    |                          |   (5) & (6)   |
        '-------------'                          '---------------'
```

Figure 4: Stamped Passport Generation and Appraisal

In Figure 4 above, Evidence from a TPM is generated and signed by that TPM. This Evidence is appraised by Verifier A, and the Attester is given a Trustworthiness Vector which is signed and returned as Attestation Results to the Attester. Later, when a request comes in from a Relying Party, the Attester assembles and returns three independently signed elements of Evidence. These three comprise the Stamped Passport which when taken together allow Verifier B to appraise and set the current Trustworthiness Vector of the Attester.

More details on the mechanisms used in the construction and verification of the Stamped Passport are listed below. These numbers match to the numbered steps of Figure 4:

1. An Attester sends a signed TPM Quote which includes PCR measurements to Verifier A at time(EG).

2. Verifier A appraises (1), then sends the following items back to that Attester as Attestation Results:

    1. the Trustworthiness Vector of an Attester,

    2. the PCR state information from the TPM Quote of (1),

    3. time information associated with the TPM Quote of (1),

    4. the Public Attestation Key which it used to validate the TPM Quote of (1), and

    5. a Verifier signature across (2.1) though (2.4).

3. At time(EG') a nonce known to the Relying Party is sent to the Attester .

4. The Attester generates and sends a Stamped Passport. This Stamped Passport includes:

    1. The Attestation Results from (2)

    2. New signed, verifiably fresh PCR measurements from time(EG'), which incorporates the nonce from (3).

5. On receipt of (4), the Relying Party makes its determination of how the Stamped Passport will impact adjacencies within a Trusted Topology. The decision process is:

    1. Verify that (4.2) includes the nonce from (3).

    2. Use a local certificate to validate the signature (4.1).

    3. Use the Attestation Results provided public key info of (2.4) to validate the signatures of (4.2).

    4. Failure of (5.1) through (5.3) means the link does not meet minimum validation criteria, therefore appraise the link as having a null Trustworthiness Vector. Jump to step (6).

    5. If all PCR values from (2.2) equal those (4.2), then Relying Party can accept (2.1) as the link's Trustworthiness Vector. Jump to step (6).

6. If the PCR state information of (2.2) doesn't equal (4.2), and not much time has passed between time(EG) and time(EG'), the Relying Party accepts any previous Trustworthiness Vector. (Note: rather than accepting, it is also viable to attempt to acquire a new Stamped Passport. Where [stream-subscription] is used, it should only be a few seconds before a new Attestation Results are delivered to an Attester via (2).)

7. When the PCR state information is different, and there is a large or uncertain time gap between time(EG) and time(EG'), the link should be assigned a null Trustworthiness Vector.

6. Take action based on Verifier B's appraised Trustworthiness Vector:

   1. Include the link within any Trusted Topology for which that Trustworthiness Vector is qualified.

   2. Remove the link from any Trusted Topology for which that Trustworthiness Vector is not qualified.

4. Implementable Solution

This section defines one set of protocols which can be used for Trusted Path Routing. The protocols include [MACSEC] or [IEEE-802.1X], ISIS [I-D.ietf-lsr-flex-algo], YANG subscriptions [RFC8639], and [RFC3748] methods. Other alternatives are also viable.

4.1. Prerequisites

- A Trusted Topology such as one established by ISIS exists in an IGP domain for the forwarding of Sensitive Subnet traffic. This Topology will carry traffic across a set of devices which currently meet at a defined set of Trustworthiness Vectors.

- Customer designated Sensitive Subnets and their requested Trustworthiness Vectors have been identified and associated with external interfaces to/from the edge of a network. Traffic to a Sensitive Subnet can be passed into the Trusted Topology.

- Verifiers A and B are able to verify [TPM1.2] or [TPM2.0] signatures of an Attester.

- Verifier B trusts information signed by Verifier A. Verifier B has also been pre-provisioned with certificates or public keys necessary to confirm that Stamped Passports came from Verifier A

- o Within a network, a Relying Party is able to use affinity to include/exclude links as part of the Trusted Topology based on this appraisal.

4.2. Protocol Bindings

The numbering in below matches to the steps in Figure 4.

Step (1)

There are two alternatives for Verifier A to acquires Evidence including a TPM Quote from the Attester:

- o Subscription to the <attestation> stream defined in [stream-subscription]. Note: this method is recommended as it will minimize the interval between when a PCR change is made in a TPM, and when the PCR change appraisal is incorporated within a subsequent Stamped Passport.

- o The RPCs <tpm20-challenge-response-attestation> or <tpm12-challenge-response-attestation> defined in device [RATS-YANG]

Step (2)

The delivery of these Attestation Results back to the Attester MAY be done via an operational datastore write to the YANG module <ietf-attestation-results-vector>.

Step (3)

At time(NS') a Relying Party makes a Link Layer authentication request to an Attester via a either [MACSEC] or [IEEE-802.1X]. This connection request must include [RFC3748] credentials. Specifics of the EAP mapping to the Stamped Passport is tbd.

Step (4)

Upon receipt of (3), a Stamped Passport is generated as per Section 3.6, and sent to the Relying Party. Note that with [MACSEC] or [IEEE-802.1X], steps (3) & (4) will repeat periodically independently of any subsequent iteration (1) and (2). This allows for periodic reauthentication of the link layer in a way not bound to the updating of Verifier A's Attestation Results.

Step (5)

Upon receipt of (4), the Relying Party appraises the Stamped Passport as per Section 3.6. Following are relevant mappings which replace

generic steps from Section 3.6 with specific objects available with a TPM1.2 or TPM2.0.

```
+-------------------------------------------------------------------+
| TPM2.0 - Bindings/details                                         |
+-------------------------------------------------------------------+
| (5.5): If the <TPM2B_DIGEST>, <TPML_PCR_SELECTION>, <reset-       |
| counter>, <restart-counter> and <safe> are equal between the      |
| Attestation Results and the TPM Quote at time(EG') then Relying   |
| Party can accept (2.1) as the link's Trustworthiness Vector. Jump |
| to step (6).                                                      |
|                                                                   |
| (5.6): If the <reset-counter>, <restart-counter> and <safe> are   |
| equal between the Attestation Results and the TPM Quote at        |
| time(EG'), and the <clock> object from time(EG') has not          |
| incremented by an unacceptable number of seconds since the        |
| Attestation Result, then Relying Party can accept (2.1) as the    |
| link's Trustworthiness Vector. Jump to step (6).                  |
|                                                                   |
| (5.7): Assign the link a null Trustworthiness Vector.             |
+-------------------------------------------------------------------+

+-------------------------------------------------------------------+
| TPM1.2 - Bindings/details                                         |
+-------------------------------------------------------------------+
| (5.5): If the <pcr-index>'s and <tpm12-pcr-value>'s are equal     |
| between the Attestation Results and the TPM Quote at time(EG'),   |
| then Relying Party can accept (2.1) as the link's Trustworthiness |
| Vector. Jump to step (6).                                         |
|                                                                   |
| (5.6): If the time hasn't incremented an unacceptable number of   |
| seconds from the Attestation Results <timestamp> and the system   |
| clock of the Relying Party, then Relying Party can accept (2.1)   |
| as the link's Trustworthiness Vector. Jump to step (6).           |
|                                                                   |
| (5.7): Assign the link a null Trustworthiness Vector.             |
+-------------------------------------------------------------------+
```

Step (6)

After the Trustworthiness Vector has been validated or reset, based on the link's Trustworthiness Vector, the Relying Party may adjust the link affinity of the corresponding ISIS [I-D.ietf-lsr-flex-algo] topology. ISIS will then replicate the link state across the IGP domain. Traffic will then avoid links which do not have a qualifying Trustworthiness Vector.

5. YANG Module

This YANG module imports modules from [RATS-YANG], [crypto-types] and [RFC6021].

```
<CODE BEGINS> ietf-attestation-results-vector@2020-09-17.yang
module ietf-attestation-results-vector {
  yang-version 1.1;
  namespace
     "urn:ietf:params:xml:ns:yang:ietf-attestation-results-vector";
  prefix arv;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-tpm-remote-attestation {
    prefix tpm;
    reference
      "draft-ietf-rats-yang-tpm-charra";
  }
  import ietf-crypto-types {
    prefix ct;
    reference
      "RFC XXXX: Common YANG Data Types for Cryptography
      (currently draft-ietf-netconf-crypto-types)";
  }
  import ietf-tcg-algs {
    prefix taa;
  }

  organization "IETF";
  contact
    "WG Web:   <http://tools.ietf.org/wg/rats/>
     WG List:  <mailto:rats@ietf.org>

     Editor:   Eric Voit
               <mailto:evoit@cisco.com>";

  description
    "This module contains conceptual YANG specifications for
    subscribing to attestation streams being generated from TPM chips.

    Copyright (c) 2020 IETF Trust and the persons identified as authors
    of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or without
    modification, is permitted pursuant to, and subject to the license
    terms contained in, the Simplified BSD License set forth in Section
```

```
    4.c of the IETF Trust's Legal Provisions Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX; see the RFC
    itself for full legal notices.";

  revision 2020-09-17 {
    description
      "Initial version.";
    reference
      "draft-voit-rats-trustworthy-path-routing";
  }


  /*
   * IDENTITIES
   */

  identity trustworthiness-level {
    description
      "Base identity for a Verifier that uses its Appraisal Policy for
      Evidence to establish a trustworthiness level.";
  }

  identity trustworthiness-pass {
    description
      "A trustworthiness-level which successfully meets an Appraisal
      Policy for Evidence.";
  }

  identity trustworthiness-fail {
    description
      "A trustworthiness-level which hit Appraisal Policy for Evidence
      necessary to fail an evaluation.  Note: this failure might or
      might not consider whether sufficient Evidence has been provided.
      In other words having insufficient evidence might not drive the
      setting of this failing trustworthiness-level.";
  }

  identity boot-verified {
    base trustworthiness-pass;
    description
      "A Verifier has appraised an Attester as Boot Integrity
      Verified.";
  }

  identity boot-verification-fail {
    base trustworthiness-fail;
```

```
    description
      "A Verifier has appraised an Attester has failed its Boot
      Integrity verification.";
  }

  identity hw-authentic {
    base trustworthiness-pass;
    description
      "A Verifier has appraised an Attester as having authentic
      hardware.";
  }

  identity fw-authentic {
    base trustworthiness-pass;
    description
      "A Verifier has appraised an Attester as having authentic
      firmware.";
  }

  identity hw-verification-fail {
    base trustworthiness-fail;
    description
      "A Verifier has appraised an Attester has failed its hardware or
      firmware verification.";
  }
  identity identity-verified {
    base trustworthiness-pass;
    description
      "A Verifier has appraised and verified an Attester's unique
      identity.";
  }

  identity identity-fail {
    base trustworthiness-fail;
    description
      "A Verifier has been unable to assess or verify an Attester's
      unique identity";
  }

  identity files-verified {
    base trustworthiness-pass;
    description
      "A Verifier has appraised an Attester's file system, and asserts
      that it recognizes relevant files.";
  }

  identity file-repudiated {
    base trustworthiness-fail;
```

```
    description
      "A Verifier has found a file on an Attester which should not be
      present.";
  }



  grouping TPM20-unsigned-internals {
    description
      "The unsigned extract of a TPM2 Quote.";
    leaf TPM2B_DIGEST {
      mandatory true;
      type binary;
      description
        "A hash of the latest PCR values (and the hash algorithm used)
        which have been returned from a Verifier for the selected PCRs
        identified within TPML_PCR_SELECTION.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
        TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.12.1";
    }
    list tpm20-pcr-bank {
        min-elements 1;
        key "TPM-hash-algo";
        description
          "Specifies the list of PCRs and Hash Algorithms used for the
          latest returned TPM2B_DIGEST. Identifying
          this object simplifies Stamped Passport troubleshooting if the
          same PCRs and Hash algorithms are not used when attempting to
          correlate independent TPM2B_DIGESTs.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.9.7";
      leaf TPM-hash-algo {
        type identityref {
          base taa:hash;
        }
        description
          "The hash scheme actively being used to hash a PCRs.";
      }
      leaf-list pcr-index {
        type tpm:pcr;
        min-elements 1;
        description
          "Defines what TPM2 Banks are available.  A bank is a set
          of PCRs which are extended using a particular hash
          algorithm.";
      }
```

```
    }
    leaf clock {
      mandatory true;
      type uint64;
      description
        "Clock is a monotonically increasing counter that advances
         whenever power is applied to a TPM2. The value of Clock is
         incremented each millisecond.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.11.2";
    }
    leaf reset-counter {
      mandatory true;
      type uint32;
      description
        "This counter increments on each TPM Reset.  The most common
        TPM Reset would be due to a hardware power cycle.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.11.3";
    }
    leaf restart-counter {
      mandatory true;
      type uint32;
      description
        "This counter shall increment by one for each TPM Restart or
        TPM Resume. The restartCount shall be reset to zero on a TPM
        Reset.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.11.4";
    }
    leaf safe {
      mandatory true;
      type boolean;
      description
        "This parameter is set to YES when the value reported in Clock
        is guaranteed to be unique for the current Owner. It is set to
        NO when the value of Clock may have been reported in a previous
        attestation or access.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
        TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.11.5";
    }
  }

  grouping TPM12-unsigned-internals-extended {
```

```
    description
      "The unsigned extract of a TPM12 Quote, with extra content from
       the Verifier specific to a TPM12.";
    uses tpm:tpm12-pcr-selection;
    leaf-list tpm12-pcr-value {
      type binary;
      description
        "The list of TPM_PCRVALUEs from each PCR selected in sequence
        of tpm12-pcr-selection.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf
         Section 10.9.7";
    }
    leaf timestamp {
      type yang:date-and-time;
      mandatory true;
      description
        "The timestamp of the Verifier's appraisal.  This can be used by
        a Relying Party to determine the freshness of the attestation
        results.";
    }
  }

  /*
   * DATA NODES
   */

  container attestation-results {
    presence
      "Indicates that Verifier has appraised the security posture of the
      Attester, and returned the results within this container.  If the
      Attester believes this information is no longer fresh, this
      container should automatically be deleted.";
    description
      "Retains the most recent Attestation Results for this Attester.
      It must only be written by a Verfier which is to be trusted by a
      Relying Party.";
    leaf-list trustworthiness-vector {
      type identityref {
        base trustworthiness-level;
      }
      ordered-by system;
      description
        "One or more Trustworthiness Levels assigned which expose the
        Verifier's evaluation of the Evidence associated with the
        'tpmt-signature'.";
    }
```

```
    choice tpm-specification-version {
      description
        "Identifies the cryptoprocessor API set which drove the
         Attestation Results.";
      case TPM2.0 {
        if-feature "taa:TPM20";
        description
          "The Attestation Results are from a TPM2.";
        uses TPM20-unsigned-internals;
      }
      case TPM1.2 {
        if-feature "taa:TPM12";
        description
          "The most recent Attestation Results from a TPM1.2.";
        uses TPM12-unsigned-internals-extended;
      }
    }
    uses ct:public-key-grouping {
      description
        "In order to avoid having to provision AIK certificates on a
         Relying Party network device, it is possible to send the AIK
         public key as from the Verifier as part of the passport. This
         is safe because the key is signed by the Verifier (hence
         vouching for its validity.) The two objects within this group
         allow the Verifier to include this information as part of the
         Attestation Results.";
    }
    leaf public-key-algorithm-type {
      mandatory true;
      type identityref {
        base taa:asymmetric;
      }
      description
        "Indicates what kind of algorithm is used with the Attester's
        Public Key Value.";
    }
    leaf verifier-signature-key-name {
      type string;
      description
        "Name of the key the Verifier used to sign the results.";
    }
    leaf verifier-key-algorithm-type {
      mandatory true;
      type identityref {
        base taa:asymmetric;
      }
      description
        "Indicates what kind of algorithm was used for the
```

```
        'verifier-signature'.";
    }
    leaf verifier-signature {
      type binary;
      mandatory true;
      description
        "Signature of the Verifier across all the other objects within
        the attestation-results container.  The signature will assume
        the sequence of objects as defined in the YANG model schema.";
    }
  }
}
<CODE ENDS>
```

6. Security Considerations

Verifiers are limited to the Evidence available for appraisal from a Router. Although the state of the art is improving, some exploits may not be visible via Evidence.

Only security measurements which are placed into PCRs are capable of being exposed via TPM Quote at time(EG').

Successful attacks on an Verifier have the potential of affecting traffic on the Trusted Topology.

For Trusted Path Routing, links which are part of the FlexAlgo are visible across the entire IGP domain. Therefore a compromised device will know when it is being bypassed.

Access control for the objects in Figure 3 should be tightly controlled so that it becomes difficult for the Stamped Passport to become a denial of service vector.

7. References

7.1. Normative References

[crypto-types]
"Common YANG Data Types for Cryptography", May 2020, <https://datatracker.ietf.org/doc/draft-ietf-netconf-crypto-types/>.

[RATS-Arch]
"Remote Attestation Procedures Architecture", March 2020, <https://tools.ietf.org/html/draft-ietf-rats-architecture-02>.

[RATS-YANG]
"A YANG Data Model for Challenge-Response-based Remote Attestation Procedures using TPMs", June 2020, <https://datatracker.ietf.org/doc/draft-ietf-rats-yang-tpm-charra/>.

[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>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <https://www.rfc-editor.org/info/rfc6021>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[TPM1.2] TCG, ., "TPM 1.2 Main Specification", October 2003, <https://trustedcomputinggroup.org/resource/tpm-main-specification/>.

[TPM2.0] TCG, ., "TPM 2.0 Library Specification", March 2013, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

7.2. Informative References

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-11 (work in progress), September 2020.

[IEEE-802.1X]
Parsons, G., "802.1AE: MAC Security (MACsec)", January 2020, <https://standards.ieee.org/standard/802_1X-2010.html>.

[MACSEC] Seaman, M., "802.1AE: MAC Security (MACsec)", January 2006, <https://1.ieee802.org/security/802-1ae/>.

[RATS-Device]
"Network Device Remote Integrity Verification", n.d., <https://datatracker.ietf.org/doc/draft-ietf-rats-tpm-based-network-device-attest>.

[RATS-Interactions]
"Reference Interaction Models for Remote Attestation Procedures", June 2020, <https://ietf-rats.github.io/draft-birkholz-rats-reference-interaction-model/draft-birkholz-rats-reference-interaction-model.html#section-7>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/info/rfc3748>.

[stream-subscription]
"Attestation Event Stream Subscription", June 2020, <https://datatracker.ietf.org/doc/draft-birkholz-rats-network-device-subscription>.

Appendix A. Acknowledgements

Shwetha Bhandari, Henk Birkholz, Chennakesava Reddy Gaddam, Sujal Sheth, Peter Psenak, Adwaith Gautham, Annu Singh, Nancy Cam Winget, Ned Smith, and Guy Fedorkow.

Appendix B. Change Log

[THIS SECTION TO BE REMOVED BY THE RFC EDITOR.]

v00-v01

- Minor tweaks

v02-v00 of draft-voit-rats-trustworthy-path-routing-00

- file rename was due to an IETF tool submission glitch

- The Attester's AIK is included within the Stamped Passport. This eliminates the need to provision to AIK certificate on the Relying Party.

- Removed Centralized variant

- Added timing diagram, and moved content around to match

v01-v02 of draft-voit-rats-trusted-path-routing

- o Extracted the attestation stream, and placed into draft-birkholz-rats-network-device-subscription

- o Introduced the Trustworthiness Vector

v00-v01 of draft-voit-rats-trusted-path-routing

- o Move all FlexAlgo terminology to Section 4.2. This allows Section 3.6 to be more generic.

- o Edited Figure 1 so that (4) points to the egress router.

- o Added text freshness mechanisms, and articulated configured subscription support.

- o Minor YANG model clarifications.

- o Added a few open questions which Frank thinks interesting to work.

Appendix C. Open Questions

(1) When there is no available Trusted Topology?

Do we need functional requirements on how to handle traffic to/from Sensitive Subnets when no Trusted Topology exists between IGP edges? The network typically can make this unnecessary. For example it is possible to construct a local IPSec tunnel to make untrusted devices appear as Transparently-Transited Devices. This way Secure Subnets could be tunneled between FlexAlgo nodes where an end-to-end path doesn't currently exist. However there still is a corner case where all IGP egress points are not considered sufficiently trustworthy.

(2) Extension of the Stamped Passport?

We might move to 'verifier-certificate' and 'verifier-certificate-name' based on WG desire to include more information in the Stamped Passport. The format used could be extracted from ietf-keystore.yang, grouping keystore-grouping.

Author's Address

Eric Voit
Cisco Systems, Inc.

Email: evoit@cisco.com