Workgroup: RATS Working Group
Internet-Draft:
draft-voit-rats-trustworthy-path-routing-06
Published: 6 September 2022
Intended Status: Standards Track
Expires: 10 March 2023
Authors: E. Voit, Cisco; C. Gaddam, Cisco; G. Fedorkow, Juniper; H. Birkholz, Fraunhofer SIT; M. Chen, China Mobile

Trusted Path Routing

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 for trustworthiness. 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.

By using a router's embedded TPM based cryptoprocessors in conjunction with the Remote Attestation context established by [attestation-results], a network provider can identify potentially compromised devices as well as potentially exploitable (or even exploited) vulnerabilities. Using this knowledge, it is then possible to redirect sensitive flows around these devices while other remediations are potentially considered by Network Operations.

Trusted Path Routing allows the 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 Claims, the links are then included as members of this Trusted Topology. Routing protocols are 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. This is done by an end user identifying 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 been verified as achieving a specific set of Trustworthiness Claims during its most recent trustworthiness appraisal. Interesting sets of Trustworthiness Claims might be marketed to end-users in the following ways:

* all transited devices have booted with known hardware and firmware

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

* no guarantees provided

2. Terminology

2.1. Terms

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

The following terms are impored from [attestation-results]: Trustworthiness Claim, Trustworthiness Vector, AR-augmented Evidence

Newly defined terms for this document:

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

Stamped Passport -- AR-augmented Evidence which can take two forms. First if the Attester uses a TPM2, the the Verifier Proof-of-Freshness includes the <clock>, <reset-counter>, <restart-

counter> and <safe> objects from a recent TPM2 quote made by that Attester, and the Relying Party Proof-of-Freshness is returned along with the timeticks as objects embedded within the most recent TPM quote signed by the same TPM2. Second, if the Attester uses a TPM1.2: the Verifier Proof-of-Freshness includes a global timestamp from that Verifier, and the Relying Party Proof-of-Freshness is embedded within a more recent TPM quote signed by the same TPM Attesting Environment.

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.

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. Implementation Prerequisites

The specification is a valid instance of [attestation-results]. This specification works under the following protocol and preconfiguration prerequisite assumptions:

* All Attested Devices support the TPM remote attestation profile as laid out in [RATS-Device], and include either [TPM2.0] or [TPM1.2].

* One or more Verifier A's as defined in [attestation-results] 'Interaction Model' continuously appraise each of the Attested Devices in a network domain, and these Verifiers return the Attestation Results back to each originating Attested Device.

* The Attested Devices are connected via link layer protocols such as [MACSEC] or [IEEE-802.1X].

* Each Attester can pass a Stamped Passport to a Relying Party / Verifier B as defined in [attestation-results] 'Interaction Model' within [RFC3748] over that link layer protocol.

* A Trusted Topology such as [I-D.ietf-lsr-flex-algo] exists in an IGP domain for the forwarding of Sensitive Subnet traffic. This Topology will carry traffic across a set of Attested Devices which currently meet at a defined set of Trustworthiness Vectors.

* A Relying Party is able to use mechanisms such as [I-D.ietf-lsr-flex-algo]'s affinity to include/exclude links as part of the Trusted Topology based on the appraisal of a Stamped Passport.

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

* Relying Party/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.

4. End-to-end Solution

4.1. Network Topology Assembly

To be included in a Trusted Topology, Stamped Passports are shared between Attested Devices (such as routers) as part of link layer authentication. Upon receiving and appraising the Stamped Passport during the link layer authentication phase, the Relying Party Attested Device decides if this link should be added as an active adjacency for a particular Trusted Topology. In Figure 1 below, this might be done by applying an Appraisal Policy for Attestation Results. The policy within each device might specify the evalutation of a 'hardware' claim as defined in [attestation-results], Section 2.3.4. With the appraisal, an Attesting Device be most recently appraised with the 'hardware' Trustworthiness Claim in the 'affirming' range. If Attested Device has been appraised outside that range, it would not become part of the Trustworthy Topology.

When enough links have been successfully added, the Trusted Topology will support edge-to-edge forwarding as routing protocols flood the adjacency information across the network domain.

```
               .------------.                .----------.
               | Attested   |                | Edge     |
 .----------.  | Device 'x' |                | Attested |
 | Attested |  |            |                | Device   |
 | Device   |  |            |                |          |
 |          |  |       trust>---------------<no_trust  |
 |  no_trust>--<trust       |  .----------.  |          |---Sensitive
 |          |  '------------'  |     trust>==<trust     |   Subnet
 |     trust>==================<trust     |  |          |
 '----------'                  |          |  '----------'
                               | Attested |
                               | Device   |
                               '----------'
```

Figure 1: Trusted Path Topology Assembly

As the process described above repeats over time 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. Traffic exchanged with Sensitive Subnets can then be forwarded across that Trusted Topology from all edges of the network domain. After the initial Trusted Topology establishment, new and existing devices will continue to provide incremental Stamped Passports. As each link is added/removed from the Trusted Topology, the topology will adjust itself accordingly.

Ultimately from an operator and users point of view, the delivered network will be more secure and therefore the service provided more valuable. As network operators attach great importance to the innate security of links, also delivering security for transited network and networking devices will also prove valuable.

4.2. Attestation Information Flows

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. A Stamped Passport is a valid type of AR-augmented evidence as described in [attestation-results].

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]. This figure is also a valid embodiment of the "Interaction Model" described within [attestation-results]. (Note that the Relying Party must also be an Attested Device in order to attract Sensitive Subnet traffic which may flow from the Attester.)

```
  .------------------.
  | Attester         |
  |                  |
  | (Attested Device |
  |   / Router)      |                           .------------------.
  |  .-------------. |                           | Relying Party    |
  |  | TPM based   | |                           |   / Verifier B   |
  |  | Attesting   | |           .----------.    |                  |
  |  | Environment | |           | Verifier |    | (Attested Device |
  |  '-------------' |           |     A    |    |   / Router)      |
  '------------------'           '----------'    '------------------'
        time(VG)                       |                 |
          |<------nonce--------------time(NS)            |
          |                            |                 |
 time(EG)(1)------Evidence------------>|                 |
          |                          time(RG)            |
          |<------Attestation Results-(2)                |
          ~                            ~                 ~
        time(VG')?                     |                 |
          ~                            ~                 ~
          |<------nonce---------------------------------(3)time(NS')
          |                            |                 |
time(EG')(4)------Stamped Passport---------------------->|
          |                            |   time(RG',RA')(5)
                                                        (6)
                                                         ~
                                                      time(RX')
```

Figure 2: Trusted Path Timing

To summarize Figure 2 above, Evidence about a specific Attester is generated. Some subset of this evidence will be in the form of PCR quotes which are signed by a TPM that exists as the Attester's Attesting Environment. This Evidence will be delibered to and appraised by Verifier A. Verifier A will then appraise the Attester and give it a Trustworthiness Vector. This Trustworthiness Vector is then signed by Verifier A and be returned as Attestation Results to the Attester. Later, when a request comes in from a Relying Party, the Attester assembles and returns a Stamped Passport. The Stamped Passport contains all the information necessary for Verifier B to appraise the most recent Trustworthiness Vector of the Attester. Based on the Verifier B appraisal, the link will be included or not in a Trusted Topology maintained on the Relying Party.

More details on the mechanisms used in the construction, verification, and transmitting of the Stamped Passport are listed below. These numbers match to both the numbered steps of Figure 2 and numbered steps described in Section 3 of [attestation-results]:

4.2.1. Step 1

Evidence about and Attester is generated. A portion of this Evidence will include a PCR quote signed by a TPM private LDevID key that exists within the Attester's TPM based Attesting Environment. The Attester sends a signed TPM Quote which includes PCR measurements to Verifier A at time(EG).

There are two alternatives for Verifier A to acquire this signed Evidence:

*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.

*Periodic polling of RPC <tpm20-challenge-response-attestation> or the RPC <tpm12-challenge-response-attestation> which are defined in [RATS-YANG].

4.2.2. Step 2

Verifier A appraises the Evidence from Step 1. A portion of this appraisal process will follow the appraisal process flow described below. This appraisal process MUST be able to set at least the following set of Trustworthiness Claims from [attestation-results]: 'hardware', 'instance-identity', and 'executables'. The establishment of a Trustworthiness Vector uses the following Figure 3 logic on the Verifier:

```
Start: TPM Quote Received, log received, or appraisal timer expired
       for the the Attesting network device.

Appraisal 0: set Trustworthiness Vector = Null

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

Appraisal 2: Appraise Hardware Integrity PCRs
   if (hardware NOT "0") - push onto vector
   if (hardware NOT affirming or warning), go to End

Appraisal 3: Appraise Attesting Environment identity
   if (instance-identity <> "0") - push onto vector

Appraisal 4: Appraise executable loaded and filesystem integrity
   if (executables NOT "0") - push onto vector
   if (executables NOT affirming or warning), go to End

Appraisal 5: Appraise all remaining Trustworthiness Claims
        Independently and set as appropriate.

End
```

Figure 3: Verifier A Appraisal Flow

After the appraisal and generation of the Trustworthiness Vector, the following are assembled as the set of Attestation Results from this particular appraisal cycle:

(2.1) the Public Attestation Key which was used to validate the TPM Quote of Step 1. This is encoded by <public-key>, <public-key-format>, and <public-key-algorithm-type>.

(2.2) the appraised Trustworthiness Vector of the Attester as calculated in Figure 3

(2.3) the PCR state information from the TPM Quote of (1) plus the time information associated with the TPM Quote of (1). Specifically if the Attester has a TPM2, then the values of the TPM PCRs are included (i.e., <TPM2B_DIGEST>, <tpm20-hash-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.

(2.4) a Verifier A signature across (2.1) though (2.3). This signature is encoded by <verifier-signature>, <verifier-key-algorithm-type>, and <verifier-signature-key-name>.

Immediately subsequent to each Verifier appraisal cycle of an Attester, these Attestation Results MUST be pushed to the Attesting Router. This is done via a daatstore write to the following YANG model on the Attester. A YANG tree showing the relevant YANG objects is below. The YANG model describing each of these objects is described later in the document. Note however that although the YANG model shows the specific objects which are needed, the specific set of objects needs to be encoded in CDDL. This makes the payload going over TLS more efficient. Look for this encoding in a new version of the draft which is pending.

```
module: ietf-trustworthiness-claims
  +--rw attestation-results!
     +--rw (tpm-specification-version)?
        +--:(tpm20-attestation-results-cddl) {taa:tpm20}?
        |  +--rw tpm20-attestation-results-cddl
        |     +--rw trustworthiness-vector
        |     |  +--rw hardware?            hardware
        |     |  +--rw instance-identity?   instance-identity
        |     |  +--rw executables?         executables
        |     |  +--rw configuration?       configuration
        |     +--rw tpm20-pcr-selection* [tpm20-hash-algo]
        |     |  +--rw tpm20-hash-algo    identityref
        |     |  +--rw pcr-index*         tpm:pcr
        |     +--rw TPM2B_DIGEST                           binary
        |     +--rw clock                                  uint64
        |     +--rw reset-counter                          uint32
        |     +--rw restart-counter                        uint32
        |     +--rw safe                                   boolean
        |     +--rw attester-certificate-name
        |     |       tpm:certificate-name-ref
        |     +--rw appraisal-timestamp
        |     |       yang:date-and-time
        |     +--rw verifier-algorithm-type              identityref
        |     +--rw verifier-signature                     binary
        |     +--rw verifier-certificate-keystore-ref
        |             tpm:certificate-name-ref
        +--:(tpm12-attestation-results-cddl) {taa:tpm12}?
           +--rw tpm12-attestation-results-cddl
              +--rw trustworthiness-vector
              |  +--rw hardware?            hardware
              |  +--rw instance-identity?   instance-identity
              |  +--rw executables?         executables
              |  +--rw configuration?       configuration
              +--rw pcr-index*                             pcr
              +--rw tpm12-pcr-value*                       binary
              +--rw tpm12-hash-algo                        identityref
              +--rw TPM12-quote-timestamp
              |       yang:date-and-time
              +--rw attester-certificate-name
              |       tpm:certificate-name-ref
              +--rw appraisal-timestamp
              |       yang:date-and-time
              +--rw verifier-algorithm-type              identityref
              +--rw verifier-signature                     binary
              +--rw verifier-certificate-keystore-ref
                      tpm:certificate-name-ref
```

Figure 4: Attestation Results Tree

4.2.3. Step 3

At time(NS') some form of time-based freshness (such as a nonce or Epoch Handle [RATS-Interactions]) will be generated in a way which makes it available to the Relying Party. Soon after time(NS'), a Relying Party will make a Link Layer authentication request to an Attester via a either [MACSEC] or [IEEE-802.1X]. This connection request MUST expect the return of [RFC3748] credentials from the Attester.

4.2.4. Step 4

Upon receipt of the Link Layer request from Step 3, a Stamped Passport is generated and sent to the Relying Party. The Stamped Passport MUST include the following:

(4.1) The Attestation Results from Step 2

(4.2) New signed, verifiably fresh PCR measurements based on a TPM quote at time(EG') which incorporates the freshness information known by the Relying Party from Step 3. If it is a nonce, the freshness information will have been delivered as part of the link layer connection request in Steps 3.

Stamped Passports contain following objects, defined in this document via YANG. A subsequent draft will convert the objects below into CDDL format so that the objects can efficiently be passed over EAP.

If an Attester includes a TPM2, these YANG objects are:

```
 +---n tpm20-stamped-passport
    +--ro attestation-results
    |  +--ro trustworthiness-vector
    |  |  +--ro hardware?            hardware
    |  |  +--ro instance-identity?   instance-identity
    |  |  +--ro executables?         executables
    |  |  +--ro configuration?       configuration
    |  +--ro tpm20-pcr-selection* [tpm20-hash-algo]
    |  |  +--ro tpm20-hash-algo    identityref
    |  |  +--ro pcr-index*         tpm:pcr
    |  +--ro TPM2B_DIGEST                          binary
    |  +--ro clock                                 uint64
    |  +--ro reset-counter                         uint32
    |  +--ro restart-counter                       uint32
    |  +--ro safe                                  boolean
    |  +--ro attester-certificate-name
    |  |       tpm:certificate-name-ref
    |  +--ro appraisal-timestamp
    |  |       yang:date-and-time
    |  +--ro verifier-algorithm-type               identityref
    |  +--ro verifier-signature                    binary
    |  +--ro verifier-certificate-keystore-ref
    |          tpm:certificate-name-ref
    +--ro tpm20-quote
       +--ro TPMS_QUOTE_INFO    binary
       +--ro quote-signature    binary
```

Figure 5: YANG Tree for a TPM2 Stamped Passport

Note that where a TPM2.0 is used, the PCR numbers and hash algorithms quoted in Step 1 MUST match the PCR numbers and hash algorithms quoted in this step.

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

```
 +---n tpm12-stamped-passport
    +--ro attestation-results
    |  +--ro trustworthiness-vector
    |  |  +--ro hardware?          hardware
    |  |  +--ro instance-identity?  instance-identity
    |  |  +--ro executables?       executables
    |  |  +--ro configuration?     configuration
    |  +--ro pcr-index*                          pcr
    |  +--ro tpm12-pcr-value*                    binary
    |  +--ro tpm12-hash-algo                     identityref
    |  +--ro TPM12-quote-timestamp
    |  |       yang:date-and-time
    |  +--ro attester-certificate-name
    |  |       tpm:certificate-name-ref
    |  +--ro appraisal-timestamp
    |  |       yang:date-and-time
    |  +--ro verifier-algorithm-type             identityref
    |  +--ro verifier-signature                  binary
    |  +--ro verifier-certificate-keystore-ref
    |          tpm:certificate-name-ref
    +--ro tpm12-quote
       +--ro TPM_QUOTE2?   binary
```

Figure 6: YANG Tree for a TPM1.2 Stamped Passport

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

Note that with [MACSEC] or [IEEE-802.1X], Step 3 plus Step 4 will repeat periodically independently of any subsequent iteration Steps 1 and Step 2. This allows for periodic reauthentication of the link layer in a way not bound to the updating of Verifier A's Attestation Results.

4.2.5. Step 5

Upon receipt of the Stamped Passport generated in Step 4, the Relying Party appraises this Stamped Passport as per its Appraisal Policy for Attestation Results. The result of this application will determine how the Stamped Passport will impact adjacencies within a Trusted Topology. The decision process is as follows:

(5.1) Verify that (4.2) includes the freshness context from Step 3.

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

(5.3) Verify that the hash from (4.2) matches (4.1)

(5.4) Use the identity of (2.1) to validate the signature of (4.2).

(5.5) Failure of any steps (5.1) through (5.4) means the link does not meet minimum validation criteria, therefore appraise the link as having a null Verifier B Trustworthiness Vector. Jump to Step 6.

(5.6) Compare the time(EG) TPM state to the time(EG') TPM state

*If TPM2.0

1. If the <TPM2B_DIGEST>, <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.

2. 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.)

3. Assign the link a null Trustworthiness Vector.

*If TPM1.2

1. 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).

2. 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.)

3. Assign the link a null Trustworthiness Vector.

(5.7) Assemble the Verifier B Trustworthiness Vector

1. Copy Verifier A Trustworthiness Vector to Verifier B Trustworthiness Vector

2. Prune any Trustworthiness Claims the Relying Party doesn't accept from this Verifier.

4.2.6. Step 6

After the Trustworthiness Vector has been validated or reset, based on the link's Trustworthiness Vector, the Relying Party adjusts 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-trustworthiness-claims@2021-11-03.yang
module ietf-trustworthiness-claims {
  yang-version 1.1;
  namespace
   "urn:ietf:params:xml:ns:yang:ietf-trustworthiness-claims";
  prefix tc;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-tcg-algs {
    prefix taa;
    reference
      "draft-ietf-rats-yang-tpm-charra";
  }
  import ietf-tpm-remote-attestation {
    prefix tpm;
    reference
      "draft-ietf-rats-yang-tpm-charra";
  }

  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 2021-11-03 {
    description
      "Initial version.";
    reference
      "draft-voit-rats-trustworthy-path-routing";
```

```
}


/*
 * TYPEDEF
 */

typedef trustworthiness-claim {
  type int8;
  description
    "A Verifier asserted value designed to enable a common
    understanding of a Verifier trustworthiness appraisal.  The
    value assignments for this 8 bit signed integer will follow
    these guidelines:

    Affirming: The Verifier affirms the Attester support for this
    aspect of trustworthiness
      - Values 2 to 31: A standards enumerated reason for affirming.
      - Values -2 to -32: A non-standard reason for affirming.

    Warning: The Verifier warns about this aspect of trustworthiness.
      - Values 32 to 63: A standards enumerated reason for the
        warning.
      - Values -33 to -64: A non-standard reason for the warning.

    Contraindicated: The Verifier asserts the Attester is explicitly
    untrustworthy in regard to this aspect.
      - Values 64 to 127: A standards enumerated reason for the
        contraindication.
      - Values -65 to -128: A non-standard reason for the
        contraindication.

    None: The Verifier makes no assertions about this Trustworthiness
    Claim.  The following values are reserved with the following
    meanings across all instances of trustworthiness-claim.
      - Value 0: Note: this should always be always treated
        equivalently by the Relying Party as no claim being made.
        I.e., the RP's Appraisal Policy for Attestation Results
        SHOULD NOT make any distinction between a Trustworthiness
        Claim with enumeration '0', and no Trustworthiness Claim
        being provided.
      - Value 1: The Evidence received contains unexpected elements
        which the Verifier is unable to parse.  An example might be
        that the wrong type of Evidence has been delivered.
      - Value -1: An unexpected error occurred during the Verifier's
        appraisal processing. Note: while no claim is being made, the
        Relying Party MAY make a distinction between a
        Trustworthiness Claim with enumeration '-1', and no
        Trustworthiness Claim being provided.";
}

typedef hardware {
  type trustworthiness-claim;
  description
    "A Verifier has appraised any Attester hardware and firmware
    which are able to expose fingerprints of their identity and
    running code.

    The following are specific reserved values of hardware and
    the meanings of these reserved values:

    0: No assertion

    1: The Verifer cannot parse unexpected Evidence

    -1:Verifier malfunction

    2: An Attester has passed its hardware and/or firmware
       verifications needed to demonstrate that these are
       genuine/supported.

    32:An Attester contains only genuine/supported hardware and/or
       firmware, but there are known security vulnerabilities.

    96:Attester hardware and/or firmware is recognized, but its
       trustworthiness is contraindicated.

    97:A Verifier does not recognize an Attester's hardware or
       firmware, but it should be recognized.";
}

typedef instance-identity {
  type trustworthiness-claim;
  description
    "A Verifier has appraised an Attesting Environment's unique
    identity based upon private key signed Evidence which can be
    correlated to a unique instantiated instance of the Attester.
    (Note: this Trustworthiness Claim should only be generated if
    the Verifier actually expects to recognize the unique identity
    of the Attester.)

    The following are specific reserved values of instance-identity
    and the meanings of these reserved values:

    0: No assertion

    1: The Verifer cannot parse unexpected Evidence

    -1:Verifier malfunction
    2: The Attesting Environment is recognized, and the associated
       instance of the Attester is not known to be compromised.

    96:The Attesting Environment is recognized, and but its unique
       private key indicates a device which is not trustworthy.

    97:The Attesting Environment is not recognized; however the
       Verifier believes it should be.";
}

typedef executables {
  type trustworthiness-claim;
  description
    "A Verifier has appraised and evaluated relevant runtime files,
    scripts, and/or other objects which have been loaded into the
    Target environment's memory.

    The following are specific reserved values of executables and
    the meanings of these reserved values:

    0: No assertion

    1: The Verifer cannot parse unexpected Evidence

    -1:Verifier malfunction

    2: Only a recognized genuine set of approved executables,
       scripts, files, and/or objects have been loaded during
       and after the boot process.

    3: Only a recognized genuine set of approved executables have
       been loaded during the boot process.

    32:Only a recognized genuine set of executables, scripts, files,
       and/or objects have been loaded.  However the Verifier cannot
       vouch for a subset of these due to known bugs or other known
       vulnerabilities.

    33:Runtime memory includes executables, scripts, files, and/or
       objects which are not recognized.

    96:Runtime memory includes executables, scripts, files, and/or
       object which are contraindicated.

    99:Cryptographic validation of the Evidence has failed.";
}

typedef configuration {
  type trustworthiness-claim;
```

```
    description
      "A Verifier has appraised an Attester's configuration, and is
      able to make conclusions regarding the exposure of known
      vulnerabilities.

      The following are specific reserved values of configuration and
      the meanings of these reserved values:

      0: No assertion

      1: The Verifer cannot parse unexpected Evidence

      -1:Verifier malfunction

      2: The configuration is a known and approved config

      3: The configuration includes or exposes no known vulnerabilities

      32:The configuration includes or exposes known vulnerabilities

      64:The configuration is unsupportable as it exposes unacceptable
         security vulnerabilities.";
  }


  /*
   * GROUPINGS
   */

  grouping trustworthiness-vector {
    description
      "Allows the inclusion of a Trustworthiness Vector into
      other constructs.";
    container trustworthiness-vector {
      description
        "One or more Trustworthiness Claims assigned which expose the
        Verifiers evaluation of the Evidence associated with the
        AIK which signed as associated TPM Quote.";
      leaf hardware {
        type hardware;
        description
          "An 8 bit signed integter encoded per the typedef.";
      }
      leaf instance-identity {
        type instance-identity;
        description
          "An 8 bit signed integter encoded per the typedef.";
      }
      leaf executables {
        type executables;
        description
          "An 8 bit signed integter encoded per the typedef.";
      }
      leaf configuration {
        type configuration;
        description
          "An 8 bit signed integter encoded per the typedef.";
      }
    }
  }

  grouping verifier-evidence {
    description
      "Evidence generated by the Verifier.";
    leaf appraisal-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.";
    }
    leaf verifier-algorithm-type {
      type identityref {
        base taa:asymmetric;
      }
      mandatory true;
      description
        "Platform asymmetric algorithm used in the Verifier signature
        process.";
    }
    leaf verifier-signature {
      type binary;
      mandatory true;
      description
        "Signature of the Verifier across all the current objects in
        the attestation-results container except for 'verifier-
        signature' and 'verifier-certificate-keystore-ref'.
        This assumes CDDL encoding of the objects in the current
        order of this YANG model.";
    }
    leaf verifier-certificate-keystore-ref {
      type tpm:certificate-name-ref;
      mandatory true;
      description
        "A reference to a specific certificate to an asymmetric key
        in the Keystore for the Verifier which can be used to validate
        the 'verifier-signature'. Note that the
        'name' reference must be globally unique so that it can be
        read by the Relying Party in a way which identifies a
        specific Verifier.";
    }
  }

  grouping tpm20-cddl-attestation-results {
    description
      "Elements combined into a CDDL representation for TPM2.0.";
    uses trustworthiness-vector;
    list tpm20-pcr-selection {
      key "tpm20-hash-algo";
      description
        "Specifies the list of PCRs and Hash Algorithms used by the
        Verifier.";
      reference
        "https://www.trustedcomputinggroup.org/wp-content/uploads/
         TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.9.7";
      uses tpm:tpm20-hash-algo;
      leaf-list pcr-index {
        type tpm:pcr;
        description
          "The numbers of the PCRs associated with the TPM2B_DIGEST.";
      }
    }
    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";
    }
    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";
    }
    leaf attester-certificate-name {
      mandatory true;
      description
        "The Attester is associated with these results.";
      type tpm:certificate-name-ref;
    }
    uses verifier-evidence;
  }

  grouping tpm12-cddl-attestation-results {
    description
      "Elements combined into a CDDL representation for TPM1.2.";
    uses trustworthiness-vector;
    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";
    }
    uses tpm:tpm12-hash-algo {
      refine "tpm12-hash-algo" {
        mandatory true;
      }
    }
    leaf TPM12-quote-timestamp {
      type yang:date-and-time;
      mandatory true;
      description
        "The timestamp for when the indicator of freshness (such as a
        nonce) was generated.  This is the indicator of freshness
        which was used in the generation of the TPM1.2 quote.  This
        timestamp can be used by a Relying Party to determine the
        freshness of the attestation results.";
    }
    leaf attester-certificate-name {
      mandatory true;
      description
        "The Attester is associated with these results.";
      type tpm:certificate-name-ref;
    }
    uses verifier-evidence;
  }

  /*
   * NOTIFICATIONS
   */

  notification tpm20-stamped-passport {
    description
      "The augmentation of the most recent Attestation Results
      delivered from a Verifier with a TPM2.0 Quote.";
    container attestation-results {
      description
        "The latest Verifier delivered Attestation Results.";
      uses tpm20-cddl-attestation-results;
    }
    container tpm20-quote {
      description
        "The TPM2.0 quote delivered in response to a connectivity
        request.";
      leaf TPMS_QUOTE_INFO {
        type binary;
        mandatory true;
        description
          "A hash of the latest PCR values (and the hash algorithm
           used) which have been returned from a Verifier for the
           selected PCRs and Hash Algorithms.";
        reference
          "https://www.trustedcomputinggroup.org/wp-content/uploads/
           TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 10.12.1";
      }
      leaf quote-signature {
        type binary;
        mandatory true;
        description
          "Quote signature returned by TPM Quote.  The signature was
           generated using the key associated with the
           certificate 'name'.";
        reference
          "https://www.trustedcomputinggroup.org/wp-content/uploads/
           TPM-Rev-2.0-Part-2-Structures-01.38.pdf  Section 11.2.1";
      }
    }
  }

  notification tpm12-stamped-passport {
    description
      "The augmentation of the most recent Attestation Results
      delivered from a Verifier with a TPM1.2 Quote.";
    container attestation-results {
      description
        "The latest Verifier delivered Attestation Results.";
      uses tpm12-cddl-attestation-results;
    }
    container tpm12-quote {
      description
        "The TPM1.2 quote delivered in response to a connectivity
        request.";
      leaf TPM_QUOTE2 {
        type binary;
        description
          "Result of a TPM1.2 Quote2 operation. This includes PCRs,
           signatures, locality, the provided nonce and other data which
           can be further parsed to appraise the Attester.";
        reference
          "TPM1.2 commands rev116 July 2007, Section 16.5";
      }
    }
  }

  /*
   * DATA NODES
```

```
     */

    container attestation-results {
      presence
        "Indicates that Verifier has appraised the security posture of
        the Attester, and returned the results within this container.";
      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.";

      choice tpm-specification-version {
        description
          "Identifies the cryptoprocessor API set which drove the
          Attestation Results.";
        case tpm20-attestation-results-cddl {
          if-feature "taa:tpm20";
          description
            "Attestation Results which are returned from the
             evaluation of Evidence which includes a TPM2 quote.";
          container tpm20-attestation-results-cddl {
            description
              "Attestation Results which are returned from the
               evaluation of Evidence which includes a TPM2 quote.";
            uses tpm20-cddl-attestation-results;
          }
        }
        case tpm12-attestation-results-cddl {
          if-feature "taa:tpm12";
          description
            "Attestation Results which are returned from the
             evaluation of Evidence which includes a TPM1.2 quote.";
          container tpm12-attestation-results-cddl {
            description
              "Attestation Results which are returned from the
               evaluation of Evidence which includes a TPM1.2 quote.";
            uses tpm12-cddl-attestation-results;
          }
        }
      }
    }
  }
  <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 4 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

[attestation-results] "Attestation Results for Connectivity", 2 December 2021, <https://datatracker.ietf.org/doc/draft-ietf-rats-ar4si/>.

[crypto-types] "Common YANG Data Types for Cryptography", 17 December 2021, <https://datatracker.ietf.org/doc/draft-ietf-netconf-crypto-types/>.

[RATS-Arch] "Remote Attestation Procedures Architecture", 8 February 2022, <https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/>.

[RATS-YANG] "A YANG Data Model for Challenge-Response-based Remote Attestation Procedures using TPMs", 28 February 2022, <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>.

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

[TPM2.0] TCG, "TPM 2.0 Library Specification", 15 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", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-18, 25 October 2021, <https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-18.txt>.

[IEEE-802.1X] Parsons, G., "802.1AE: MAC Security (MACsec)", 1 January 2020, <https://standards.ieee.org/standard/802_1X-2010.html>.

[MACSEC] Seaman, M., "802.1AE: MAC Security (MACsec)", 1 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", 26 January 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-rats-reference-interaction-models>.

[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", 16 October 2021, <https://datatracker.ietf.org/doc/draft-ietf-rats-network-device-subscription/>.

Appendix A. Acknowledgements

Peter Psenak, Shwetha Bhandari, Adwaith Gautham, Annu Singh, Sujal Sheth, Nancy Cam Winget, and Ned Smith.

Appendix B. Change Log

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

v05-v06

* No changes

v04-v05

* Tweaks to text

* Added text which was morphed from that provided by Meiling

v03-v04

* YANG model updated in concert with draft-voit-rats-attestation-results as Trustworthiness Claim values are added.

v03-v04

* Moved in concert with draft-voit-rats-attestation-results as Trustworthiness Claims became 8 bit signed integers.

v02-v03

* Integrated [attestation-results] as prerequisite context.

* Totally rearranged content. But there were not meaningful process changes.

* Redid YANG model, and highlighted CDDL needs.

v01-v02

* Minor tweaks such as renaming and removal of a few trustworthiness-claims

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

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

* Introduced the Trustworthiness Vector

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

* Move all FlexAlgo terminology to allow passport definition to be more generic.

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

* Added text freshness mechanisms, and articulated configured subscription support.

* Minor YANG model clarifications.

* 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?

Format of the reference to the 'verifier-certificate-name' based on WG desire to include more information in the Stamped Passport. Also we need to make sure that the keystore referenced names are globally unique, else we will need to include a node name in the object set.

(3) Encoding of objects in CDDL. A Verifier will want to sign encoded objects rather than YANG structures. It is most efficient to encode the Attestation Results once on the Verifier, and push these down via a YANG model to the Attester.

Authors' Addresses

Eric Voit
Cisco Systems, Inc.
8135 Maple Lawn Blvd
Fulton, Maryland 20759
United States of America

Email: evoit@cisco.com

Chennakesava Reddy Gaddam
Cisco Systems, Inc.
Cessna Business Park, Kadubeesanahalli
Bangalore 560103
Karnataka
India

Email: chgaddam@cisco.com

Guy C. Fedorkow
Juniper Networks
10 Technology Park Drive
Westford, Massachusetts 01886
United States of America

Email: gfedorkow@juniper.net

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
64295 Darmstadt
Germany

Email: henk.birkholz@sit.fraunhofer.de

Meiling Chen
China Mobile

Email: chenmeiling@chinamobile.com