RATS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 12, 2020

E. Voit
Cisco

June 10, 2020

Trusted Path Routing
draft-voit-rats-trusted-path-routing-02

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.

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 (TPR) provides a method of establishing Trusted Topologies which only include trust-verified network devices. This specification describes a distributed variant of TPR. With this variant, 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. [I-D.ietf-lsr-flex-algo] is then used to propogate topology state throughout an IGP domain. IP Packets to and from end-user designated Sensitive Subnets are then forwarded into this Trusted Topology at each IGP boundary.

The specification works under the following assumptions:

1. All network devices supports the TPM remote attestation profile as laid out in [RATS-Device]

2. A [I-D.ietf-lsr-flex-algo] topology spans network devices within an IGP domain.

3. One or more Verifiers continuously appraise the set of network devices in the IGP domain, and the Verifiers canse return the Attestation Results back to the attesting network device.

4. 802.1x or MACSEC is used to communicate EAP credentials containing a Stamped Passport between network peers.

Beyond the distributed variant of TPR, there is another method to accomplish Trusted Path Routing. A controller-based TPR variant is described in the appendicies.

2. Terminology

2.1. Terms

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

The following terms at imported from [RFC8639]: Event Stream.

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 must only have their IP headers and encapsulated payloads accessible/visible only by Attested Devices.

Transparently-Transited Device - a network device within an IGP domain where any packets passed into that IGP domain are completely opaque 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. Distributed Trusted Path Routing

3.1. Trusted Topology

To be included in a Trusted Topology, a Stamped Passport Section 3.3 is assembled by an Attested Device. This Stamped Passport will include the most recent Verifier provided Trustworthiness Vector Section 3.2 for that Attested Device. Upon receiving and appraising this Stamped Passport as part of link layer authentication, the Relying Party decides if this link should be added to a Trusted Topology.

When enough links on enough Relying Parties have been so appraised, a Trusted Topology will now exist within an IGP domain. And traffic exchanged with Sensitive Subnets can be forwarded into that Trusted Topology from all edges of an IGP domain.

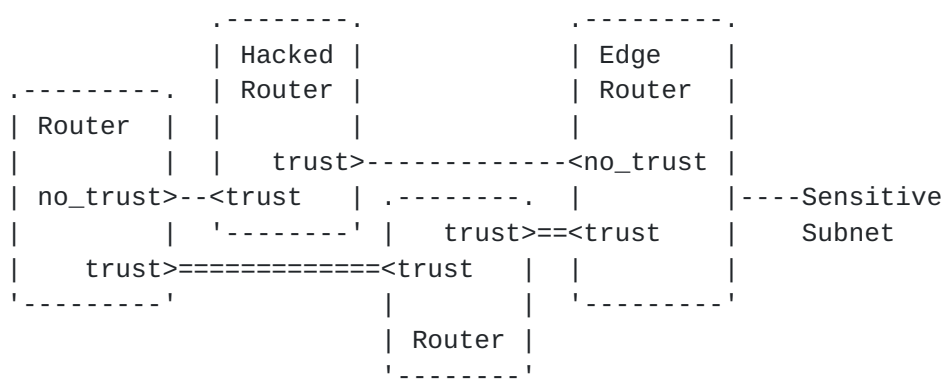

Figure 1: Distributed Trusted Path Topology Assembly

3.2. Trustworthiness Vector

For distributed TPR to operate, specific Appraisal Results need to be consistently interpreted by Relying Party network devices. The following set of Trustworthiness Levels are defined for this purpose:

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-blacklisted	A Verifier has found a file on an Attester which should not be present

More that one Trustworthiness Level may be contained within Appraisal Results. As a result, a single Trustworthiness Vector which contains a sequenced list of Trustworthiness Levels MUST be returned within the Attestation Results. The establishment of this Vector follows the following logic on the Verifier.

```
Start: TPM Quote Received, log recevied, 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, Goto 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-blacklisted) - push onto vector
   (if not evaluated, or insufficient data to conclude: take no action)

Step 6: Assemble Attestation Results, and push to Attester

End
```

3.3. Stamped Passport

Critical to the establishment and maintenance of a Trusted Topology is the Stamped Passport. Such passports are continuously exchanged between peering network devices over a link layer protocol like 802.1x. Section 3.3 provides a protocol independent process for Stamped Passport generation and evaluation. Section 3.4 later in the document binds the Stamped Passport to specific link layer protocols, YANG models, and authentication methods.

The composite nature of the Stamped Passport exposes multiple dimensions of an attesting router's security posture to a network peer. Specifically, using capabilities defined within [RATS-YANG]

and [stream-subscription], the following can be established about the Attester:

- its hardware-based identity,
- the Trustworthiness Vector according to its most recent Verifier appraisal,
- the amount of time which has passed since the Attester has been assigned the Trustworthiness Vector, and
- if the PCRs haven't changed, the Attester's current Trustworthiness Vector

With this information, the Relying Party peer can make nuanced decisions. For example, when the Attester's legitimate 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 passport, the Relying Party can include that link in a Trusted Topology.

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

The prerequisites for this solution to work are:

- Customer designated Sensitive Subnets and their requested Trustworthiness Vectors have been identified and associated with external interfaces to/from the edge of an IGP domain.
- A Trusted Topology such as one established by [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 devices which currently meet at least minimum Trustworthiness Vectors.
- Verifiers A and B (in the figure below) are able to verify [TPM1.2] or [TPM2.0] signatures of an Attester.
- Verifier A can establish the Trustworthiness Vector of an Attester and return a signed result to that Attester.
- An Attester can assemble a Stamped Passport for Verifier B.

- Verifier B trusts the Attestation Results and can verify signatures made by Verifier A.

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

- Traffic to a Sensitive Subnet can be passed into the Trusted Topology.

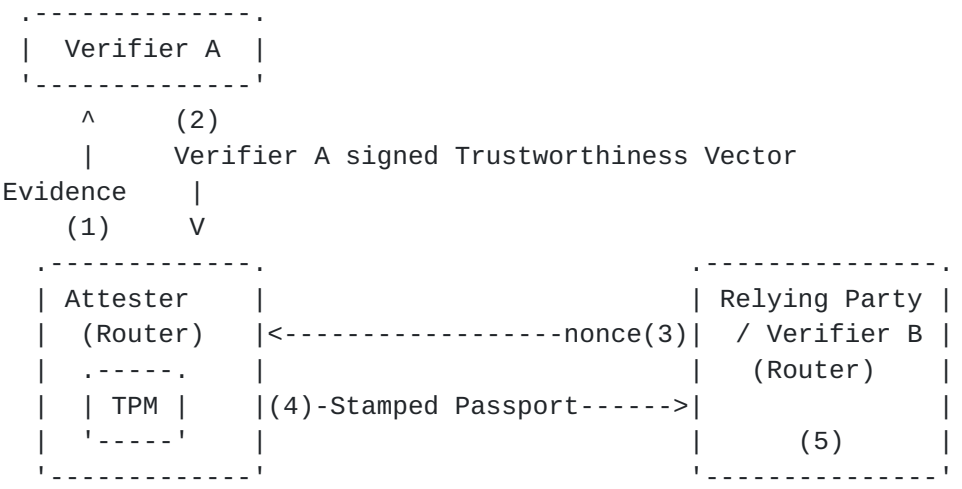

Figure 2: Stamped Passport Generation and Appraisal

In Figure 2 above, Evidence from a TPM1.2 or TPM2.0 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 match to the numbered steps of Figure 2:

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

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 signature from the TPM Quote of (1),

3. a Verifier signature across (2.1) and (2.2).

3. A nonce known to the Relying Party is received by the Attester at time(y).

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

   1. (1)

   2. (2)

   3. New signed, verifiably fresh PCR measurements at time(y), 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 (4.3) includes the nonce from (3).

   2. Verify the TPM signature within (4.2) matches the signature of (4.1).

   3. Validate the signatures of (4.1), (4.2), (4.3).

   4. Failure of (5.1), (5.2), or (5.3) means the link does not meet minimum criteria, appraise the link as having a null Trustworthiness Vector, and additionally jump to step (5.8).

   5. If selected PCR values of (1) match (4.3), then Relying Party can accept (2.1) as the link's Trustworthiness Vector.

   6. When the PCR values are different, and not much time has passed between time(x) and time(y), the Relying Party can either accept any previous Trustworthiness Vector, or attempt to acquire a new Stamped Passport. Where [stream-subscription] is used, it should only be a few seconds before a new Attestation Results should be delivered to an Attester via (2).

   7. When the PCR values are different, but there is a large time gap between time(x) and time(y), the link should be assigned a null Trustworthiness Vector.

   8. Based on the link's Trustworthiness Vector:

1. include it within any Trusted Topology which accepts that Trustworthiness Vector.

2. remove it from any Trusted Topology which does not accept that Trustworthiness Vector.

3.4. Passport Protocol Bindings

This section provides details of how a Stamped Passport described in Section 3.3 interacts with link layer protocols like [MACSEC] or [IEEE-802.1X], YANG subscriptions [RFC8639], and [RFC3748] methods. Additional linkages to the YANG module defined in Section 3.5 are described.

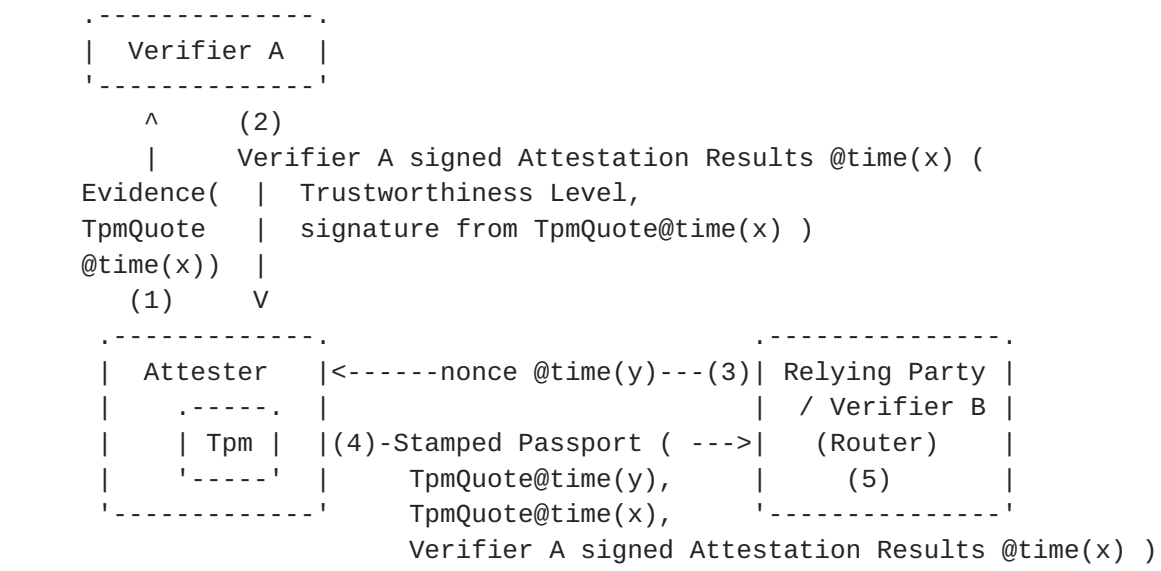

Figure 3: Details of Stamped Passport Generation

Figure 3 above expands upon the previously described Figure 2. The numbering in both figures is the same.

Step (1)

Verifier A acquires Evidence including a TPM Quote from the attester via [RATS-YANG] and/or [stream-subscription].

Step (2)

As the Evidence changes, Verifier A evaluates the totality of the Evidence received. Verifier A then sets the Trustworthiness Vector of the Attester. Subsequently it sends back a signed Attestation Result which includes the Trustworthiness Vector and the signature sent as part of (1) from the Attester. It is this signature which

allows the Trustworthiness Vector to be later provably associated with a recent TPM Quote.

The delivery of Attestation Results back to the Attester can be done via a YANG operational datastore write of the following objects:

```
  +--rw attestation-results! {passport}?
     +--rw trustworthiness-vector*        identityref
     +--rw timestamp                      yang:date-and-time
     +--rw tpmt-signature?                binary
     +--rw verifier-signature?            binary
     +--rw verifier-signature-key-name?   binary
```

Figure 4: Attestation Results Tree

Step (3)

At time(y) a Relying Party makes a Link Layer connection request to an Attester via a protocol such as [MACSEC] or [IEEE-802.1X]. This connection request must include [RFC3748] credentials. Specifics of the EAP credentials are TBD. If there is no central distribution of time via [I-D.birkholz-rats-tuda] a nonce must be included to ensure freshness of a response.

This step can 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 (4)

Upon receipt of (3), a Stamped Passport is generated as per Section 3.3, and sent to the Relying Party.

Step (5)

Upon receipt of (4), the Relying Party verifies the Stamped Passport as per Section 3.3. Most often, the relevant PCR values at time(x) will be the same as the PCR values at time(y). In this case, the Relying Party can simply accept the Trustworthiness Vector assigned by the Verifier A. When the PCR values are different, and not much time has passed between time(x) and time(y), the Relying Party can either accept the previous Trustworthiness Vector, or attempt another EAP request in a few seconds as new Attestation Results are delivered by Step (2). When there is a large time gap between time(x) and time(y) and the PCR values are different, the Attester should be given a blank Trustworthiness Vector.

Based on the link's Trustworthiness Vector, the Relying Party may adjust the link affinity of the corresponding [I-D.ietf-lsr-flex-algo] topology.

3.5. YANG Module

This YANG module imports modules from [RATS-YANG] and [RFC8639].

```
<CODE BEGINS> ietf-rats-attestation-results-vector@2020-06-03.yang
module ietf-rats-attestation-results-vector {
  yang-version 1.1;
  namespace
     "urn:ietf:params:xml:ns:yang:ietf-rats-attestation-results-vector";
  prefix arv;

  import ietf-yang-types {
    prefix yang;
  }

  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 2020-06-03 {
    description
      "Initial version.";
    reference
      "draft-voit-rats-trusted-path-routing";
```

```
  }


  /*
   * IDENTITIES
   */

  identity trustworthiness-level {
    description
      "Base identity for a verifier assessed trustworthiness level.";
  }

  identity trustworthiness-pass {
    description
      "Identity for a verifier assessed trustworthiness pass.";
  }

  identity trustworthiness-fail {
    description
      "Base identity for a verifier assessed trustworthiness fail.";
  }

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

  identity boot-verification-fail {
    base trustworthiness-fail;
    description
      "A Verifier has assessed an Attester has failed its Boot Integrity
      verification.";
  }

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

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

  identity hw-verification-fail {
```

```
    base trustworthiness-fail;
    description
      "A Verifier has assessed an Attester has failed its hardware or
      firmware verification.";
  }
  identity identity-verified {
    base trustworthiness-pass;
    description
      "A Verifier has assessed 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 assessed an Attester's file system, and asserts that
      it recognizes relevant files.";
  }

  identity file-blacklisted {
    base trustworthiness-fail;
    description
      "A Verifier has found a file on an Attester which should not be
      present.";
  }

  /*
   * DATA NODES
   */

  container attestation-results {
    presence
      "An attestation Verifier has appraised the security posture of the
      device, and returned the results within this container.";
    description
      "Containes the latest Verifier appraisal of an Attester.";
    leaf-list trustworthiness-vector {
      type identityref {
        base trustworthiness-level;
      }
      ordered-by system;
      description
```

```
        "One or more Trustworthiness Levels assigned which expose the
        Verifiers evaluation of the Evidence associated with the
        'tpmt-signature'.";
    }
    leaf timestamp {
      type yang:date-and-time;
      mandatory true;
      description
        "The timestamp of the Verifier's appraisal.";
    }
    leaf tpmt-signature {
      type binary;
      description
        "Must match a recent tpmt-signature sent in a notification to
        a Verifier.  This allows correlation of the Attestation Results to
        a recent PCR change.";
    }
    leaf verifier-signature {
      type binary;
      mandatory true;
      description
        "Signature of the Verifier across all the current objects in the
        attestation-results container.";
    }
    leaf verifier-signature-key-name {
      type binary;
      description
        "Name of the key the Verifier used to sign the results.";
    }
  }
}
<CODE ENDS>
```

4. Security Considerations

Successful attacks on an IGP domain Verifier has the potential of affecting traffic on the Trusted Topology.

For Distributed 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.

5. References

5.1. Normative References

[RATS-Arch]
"Remote Attestation Procedures Architecture", July 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", January 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>.

[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/>.

5.2. Informative References

[I-D.birkholz-rats-tuda]
Fuchs, A., Birkholz, H., McDonald, I., and C. Bormann, "Time-Based Uni-Directional Attestation", draft-birkholz-rats-tuda-02 (work in progress), March 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-09 (work in progress), May 2020.

[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-07 (work in progress), April 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]
Fedorkow, G., Voit, E., and J. Fitzgerald-McKay, "Network Device Remote Integrity Verification", n.d., <https://tools.ietf.org/html/draft-ietf-rats-tpm-based-network-device-attest-00>.

[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://tools.ietf.org/html/draft-birkholz-rats-network-device-subscription-00>.

Appendix A. Centralized Trusted Path Routing

Trusted Path Routing does not require integration with Routing protocols as is done with Distributed Trusted Path Routing. It is also possible for a Controller to choose a path through a network. This architural alternative is called Centralized Trusted Path Routing.

With Centralized Trusted Path Routing, trusted end-to-end paths are pre-assigned through a network provider domain. Along these paths, Evidence of potentially transited components has been assessed. Each path is guaranteed to only include devices which achieve at least a minimum set of a formally defined Trustworthiness Levels.

In this alternative, a controller-based Verifier ensures communications with Sensitive Subnets traverses a Trusted Topology within the controller's routing domain. To do this, the Verifier continuously acquires Evidence about each potentially transited device. This access is done via the context established within [RATS-Device]. The controller then appraises the Evidence and decides on a Trustworthiness Vector for each device. The controller then identifies end-to-end path(s) which avoid any devices which are unable to meet the minimum Trustworthiness Levels. Finally, the controller provisions network policy so that flows to and from Sensitive Subnet to use just these end-to-end paths.

Evidence passed to the Verifier which are used to establish a device's Trustworthiness Vector will include but is not limited to:

- An Attester's security measurements being extended into [TPM1.2] or [TPM2.0] compliant Platform Configuration Registers (PCR).

- An Attester's current PCR measurements.

The prerequisites for this solution are:

1. Customer designated Sensitive Subnet ranges and their acceptable Trustworthiness Vectors have been identified and associated with external interfaces to/from the edge of a routing domain.

2. A Verifier which can continuously acquire Evidence and appraise the Trustworthiness Levels of all network devices within the routing domain.

3. A Verifier which continuously optimizes a set of network paths/ tunnels. These paths must traverse only Attested Devices or Transparently-Transited Devices while on their way to an egress interface for a routing Domain.

4. A Verifier which can provision and maintain the set of Sensitive Subnets associated with specific network paths/tunnels.

Figure 5 provides a network diagram of where these four sit within a network topology.

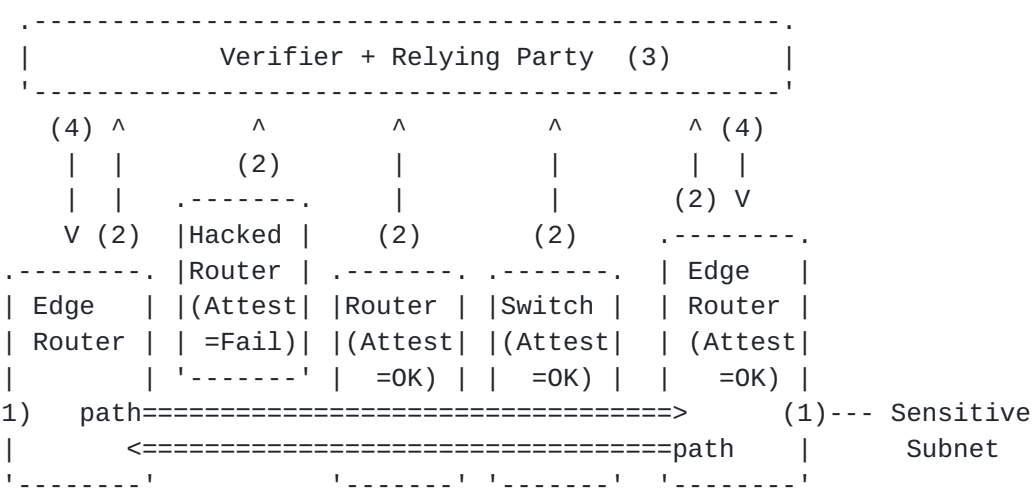

Figure 5: Centralized Trusted Path Routing

The feature functionality describing how to achieve (1) - (4) are outside the scope of this specification. The reasoning is that each of these can be accomplished via technologies specified elsewhere. For example, in step (4), it is possible for a Verifier to provision each ingress device with the set of Sensitive Subnets for which traffic would be placed into a specific [I-D.ietf-idr-segment-routing-te-policy] tunnel. As another example, consider prerequisite (2): network devices can stream changes in Evidence to a Verifier by establishing an [RFC8639] subscription to the <attestation> Event Stream as described in [stream-subscription].

Appendix B. Acknowledgements

Shwetha Bhandari, Henk Birkholz, Chennakesava Reddy Gaddam, Sujal Sheth, Peter Psenak, Nancy Cam Winget, Ned Smith, Guy Fedorkow, Liang Xia.

Appendix C. Change Log

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

v01-v02

- Extracted the attestation stream, and placed into draft-birkholz-rats-network-device-subscription
- Introduced the Trustworthiness Vector

v00-v01

- Move all FlexAlgo terminology to Section 3.4. This allows Section 3.3 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 D. Open Questions

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.

Author's Address

Eric Voit
Cisco Systems, Inc.

Email: evoit@cisco.com