Network Working Group
Internet-Draft
Expires: October 18, 2004

JJ. Puig
M. Achemlal
E. Jones
D. McPherson

April 19, 2004

Generic Security Requirements for Routing Protocols
draft-puig-rpsec-generic-requirements-02

Status of this Memo

Copyright Notice

Abstract

Routing protocols are subject to threats and attacks that can harm individual users or network operations as a whole. This document describes a generic set of security requirements for routing protocols and routing systems.

Table of Contents

1. Introduction

Routing protocols are subject to threats and attacks that can harm individual users or network operations as a whole. This document describes a generic set of security requirements for routing protocols and routing systems.

Along with the "Generic Threats to Routing Protocols" document [THREATS], this work is designed to serve as a reference material for current routing protocols and routing systems analysis, for extensions design, and as a guidance for designing new, more secure, routing protocols and routing systems.

Routing protocols addressed in this document are those limited by the rpsec working group charter. This includes distance vectors protocols and link-state protocols. We are also interested in the dedicated use of such protocols for intra-domain and inter-domain routing. Host-to-routers protocols are out of scope.

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

Security terms are explained in [SEC-GLOSS].

In order to avoid confusion between user traffic forwarding and routing traffic forwarding, in this document the former is performed by "forwarders" and called "forwarding" while the latter is performed by "relays" and called "relaying".

This document is organized as follows:

- Section 2 presents general requirements to the security of routing protocols and routing systems.
- Section 3 sorts by importance threats defined in [THREATS].
- Section 4 presents routing functions and protocols components relevant to subsequent sections.
- Section 5 defines actual generic security requirements.
- Section 6 provides guidance for tackling the Byzantine problem.
- Section 7 describes techniques for routing security.
- Section 8 presents security considerations for the routing device.

- Section 9 introduces the inter-domain puzzle.

2. General Requirements

Routing protocols are responsible for distributing information about reachability to destinations attached to the network. Often, routing protocols also distribute policies associated with the properties of the path(s) to destinations.

A path is a list of successive forwarding systems through which the destination can eventually be reached.

Paths properties may be obtained from the management plane (configuration) of the routing device or from the routing protocol. When available, paths properties may have several status:

- "trusted": devices along that path are trusted to honor these properties. Possible trust paradigms are: the path is set as being trusted from the management plane, the devices are proven to belong to the same organization, it exists a verified agreement between routing devices owners, etc. Best effort packet forwarding within an intranet is a commonly "trusted" path property.

- "evaluated": properties on that path may be evaluated. This status is complementary with others. Evaluation may be achieved with the help of the peer or through a monitoring system.

- "expected": devices along that path are expected to provide these properties, but there is no certainty. Best effort packet forwarding on the Internet is a commonly "expected" path property.

- "hazardous": there are chances that these properties are available along that path, however this is dubious and these properties are not to be relied upon. Best effort packet forwarding may be a "hazardous" path property in some ad-hoc contexts. Data confidentiality on the Internet may be considered as a "hazardous" path property.

Note: It is commonly accepted that the status of a path property is the weakest status of a hop-by-hop (an elementary path) property along that path. As an instance, a path linking systems "trusted" to provide hop-by-hop data confidentiality and a single system "expected" to provide hop-by-hop data confidentiality may only be "expected" to provide data confidentiality up to the destination. However, this common acceptance is untrue in a general fashion: if a system is "trusted" to provide at least 1 minute delay, then a path through this system can still be "trusted" to have this property. Similarly, a path on which RFC 1149 encapsulation is used between some systems can still be "trusted" to provide high delay and low throughput. Thus, the path property status greatly depends on the

actual property and the way it is described.

First main requirement is:

MR(1) Correct destination reachability information (DRI) SHOULD be made available for the forwarding function.

DRI is a set of systems and paths properties associated with the destination. These systems are claimed to be the first elements of a path providing ("trusted" / "expected" / "hazardous") packet forwarding.

A "correct" DRI is such that:

- it exists at least one path to the destination from each system listed as a suitable forwarder to the destination.

- properties of such paths comply with all locally-defined mandatory policy requirements for the destination.

Note: According to this definition of DRI correctness, there is no way to be sure a DRI is correct or not when forwarding decision must be taken. Thus, from now on, we will consider that a DRI is correct when the routing device has been /convinced/ of this correctness. This /conviction/ may result from a trust measure on the way the DRI was obtained (ex: DRI signed by a business partner, static input from the management plane, etc).

We also define DRI validity. A "valid" DRI is such that:

- packets forwarded to any system of the set follow a path up to destination (assuming that range limiting features -such as TTL- within the packets allow for this).

- mandatory policy requirements for the destination are fulfilled by properties of the paths followed.

Note: According to this definition of DRI validity, there is no way to know a DRI is valid when forwarding decision is taken. There is also no certainty of even knowing afterward that a DRI was valid or not. This definition underlines the uncertain nature of any communication though it should not be taken as `petitio principii'.

MR(1.a) When DRI is unavailable or incorrect, the first main requirement means that a correct DRI SHOULD be made available, either through the use of the routing protocol or from the management plane.

MR(1.b) When DRI is available and correct, the first main requirement means that misuse of the routing protocol SHOULD NOT jeopardize DRI availability or correctness, as this would also compromise correct forwarding.

Second main requirement is:

MR(2) The forwarding decision process MUST recognize and select a correct DRI if available for the packet properties. If such a DRI is unavailable or partly incorrect, the decision process MAY investigate severed forwarding processing, according to a heuristic learnt from the management plane. Eventually, if it is decided that no forwarding will be achieved, the packet MUST be discarded or rejected according to local policy (this policy SHOULD be configurable).

MR(2.a) Packet properties analyzed by the decision process MAY include other information than destination address.

Note: the second main requirement does not preclude forwarding when full correctness or availability of DRI cannot be achieved. It also focuses more on forwarding than on routing.

Forwarding function and misuses of the routing protocol are documented in [THREATS].

Most (but not all) subsequent requirements are meant to raise the confidence that correct DRI is available when required by the forwarding decision process.

3. Threats Importance

In the [THREATS] document, threats are described according to their sources, their consequences, and eventually the behaviors -referred as "actions"- which enable sources to trigger consequences.

3.1 Threats Consequences

In an economical perspective, primary concern is about the consequences and their potentiality for damages. We will elaborate according to the following classification of consequences, sorted by importance order:

i - Usurpation. Damages cost resulting from usurpation may be extreme and may only be roughly estimated. Besides, usurpation often enables the attacker to proceed with subsequent consequences. For these reasons, usurpation is the top issue.

ii - Deception. Deception will partly result in the same damages as usurpation and is thus an important consequence.

iii - Disruption. Disruption is a significant consequence, but its range and period are usually limited and damages cost can be evaluated more accurately than for previous consequences. However, actions leading to disruption should be difficult enough to achieve so that disruption does not become a common event. Beyond a certain threshold (depending on frequency, duration, range and overall context), disruption may become more significant than usurpation or deception.

iv - Disclosure. The above consequences directly jeopardize the services expected to be provided by the routing system. Reliability and availability of the routing system is usually considered more important than confidentiality of the routing information (which is not `user data' per se and may be learnt by other means). In current protocols, it is unlikely that disclosure of routing information will lead to direct damages on routing services as a result of the information leak. In this context, concealing the services properties in order to protect against disclosure is not a priority. However, it is worth preventing against disclosure of information which would enable the attacker to trigger usurpation, deception or disruption (in-band plain text passwords are likely to be such pieces of information).

Security requirements deal with prevention against the conditions of consequences. This prevention may be against the existence of threat sources or against the occurrence of threat actions (attacks).

A part of the security strategy is hardening of links and routing devices so that achieving access and subversion is significantly difficult. This part is not addressed here. Only subversion resulting from misuse of the routing protocol and actions against the routing system are studied.

We are thus primarily interested in the threat actions that result in usurpation, secondarily in those that result in deception, thirdly in disruption, lastly in disclosure.

3.2 Threats Actions

This section lists the threats actions as associated to the achievement of a specific consequence.

The following actions may result in usurpation:

- Overclaiming (source: subverted originating router)
- Misclaiming (source: subverted originating router)
- Misstatement (source: subverted relaying or forwarding devices)
- Reactions from router(s) deceived through spoofing (source: legitimate router(s))

The following actions may result in deception:

- Spoofing (source: subverted device)
- Overclaiming (source: subverted originating router)
- Misclaiming (source: subverted originating router)
- Misstatement (source: subverted relaying or forwarding devices)

The following actions may result in disruption:

- Overload (source: subverted devices)
- Interference (source: subverted devices)
- Overclaiming (source: subverted originating router)
- Misclaiming (source: subverted originating router)
- Misstatement (source: subverted relaying or forwarding devices)

- Reactions from router(s) deceived through spoofing (source: legitimate router(s))

The following actions may result in disclosure:

- Deliberate Exposure (source: subverted router)
- Sniffing (source: subverted link)
- Traffic Analysis (source: subverted link(s))
- Reactions from router(s) deceived through spoofing (source: legitimate router(s))

Lastly, Byzantine Failure is a special threat action, which occurs when at least one authorized device get subverted. Thus, many threats are also Byzantine failures. This threat is addressed in a section of its own (cf. Section 6). The Byzantine general problem resolution is limited by hypotheses which are reminded in this document.

3.3 Damages

[This part will present a rationale on the damages presented in the threats document. The point is certain damages should be issues address by hosts or specialized gateways (confidentiality of user traffic for instance), others are related to the device, others to forwarding, etc.]

4. Elements of Routing

[This will be updated according to the need expressed later in the document, or remove if unused.]

5. Security Requirements

In this section, we explore the requirements which will help in tackling the actions leading to the consequences of concern. First set of requirements addresses prevention against usurpation.

5.1 Requirements Against Overclaiming

"Overclaiming occurs when a subverted router advertises its control of some network resources, while in reality it does not, or the advertisement is not authorized" [THREATS].

Overclaiming is a threat from an originating router; it affects the data plane of the routing protocol.

Several models may be designed to counter overclaiming; these models address the delegation and the authorization of network resources ownership, control and advertisement.

Delegation allows for an entity to delegate a property in part or entirely to another entity (ex: owner of some network resources delegates ownership of a part of resources to another entity, which in turn becomes owner of this part).

Authorization allows for an entity to grant rights on network resources. An owner of some network resources grants control of resources to a controller; A controller of some network resources grants authorization of advertisement to an advertiser.

In the field, depending on the context and on the instance of the routing protocol, status of owner, controller and advertiser does not necessarily imply separate entities. The same entity may own and control the resources; the same device may have been granted control and advertisement.

Whatever the model representation, a chain of variable length involving delegation and authorization of ownership, control and advertisement exists. Overclaiming is a violation of the logic stated in the chain.

R(1.1) Integrity, data origin authenticity, validity at current date and availability of nodes of the chain of delegation and authorization of ownership, control and advertisement MUST be provided.

This expands to:

R(1.1.a) It MUST be possible to check that a routing device is currently authorized to advertise some network resources.

R(1.1.b) It MUST be possible to check that the entity which (directly or indirectly) granted the right of advertisement actually and currently controls the corresponding network resources.

R(1.1.c) It MUST be possible to check that the entity which (directly or indirectly) granted the control actually and currently owns the corresponding network resources.

R(1.1.d) It MUST be possible to check that delegation between entities is actually and currently valid.

R(1.2) Consumers and relays of DRI MUST check backward the chain of delegation and authorization of advertisement, control and ownership.

R(1.2.a) Check depth MUST be sufficient according to the context in which the routing protocol instance is in use and to the locally available information.

Requirement R(1.1.c) implies the existence of a "top level" or "root" owner. This definition MAY be limited to the scope in which the routing protocol instance is in use.

Requirement R(1.2.a) allows for using the same chain at different scales. In internal routing operations, a router will check the chain up to the routing system controller (of which it should already be aware), while in external routing operations, a router will check the entire chain or rely on the knowledge that the check was done by another edge router of the same system. It is also possible to establish several steps of "internal" and "external" routing with regard to this specific topic.

Overclaiming is thwarted by the requirement of checking that the routing device is authorized to advertise by an administrative entity which was given control of the according network resources by their owner.

Practical considerations related to these requirements are presented in Section 7.1.

Further elements regarding this topic are presented in Section 9.

5.2 Requirements Against Misclaiming

"A misclaiming threat is defined as an action where an attacker is

advertising its authorized control of some network resources in a way that is not intended by the authoritative network administrator" [THREATS].

Misclaiming is a threat from an originating router; it affects the data plane of the routing protocol.

In our approach, the authoritative network administrator is a resource controller, higher in the chain of delegation and authorization than the routing device. Misclaiming is a corruption of properties and of policies applying to the resources as intended by their controller.

R(2.1) Integrity, data origin authenticity, validity at current date and availability of the properties and policies applying to the advertised resources MUST be provided.

R(2.2) Consumers and relays of DRI MUST check that properties and policies applying to the advertised resources are effectively related to the resources and as intended by the resources controllers.

Requirements R(1.*) also apply.

Misclaiming is thwarted by the requirement of checking that the properties and policies are tied to the advertised resources and are intended by their controller.

Note: it is technically possible to bundle resources description, properties and policies in such a way that the routing device will have no choice but to advertise the resources with the correct properties and policies or not advertising authorized information at all. This may provide an interesting protection against consequences resulting from future subversion of the device, and it further propagates along the path (as far as no aggregation occurs; cf. Section 9).

Practical considerations related to these requirements are presented in Section 7.1.

5.3 Requirements Against Misstatement

Misstatement "is defined as an action whereby the attacker describes route attributes in an incorrect manner" [THREATS]. The attacker acts on attributes through deletion, insertion and substitution of data. He may also replay out-dated data.

Misstatement is a threat from subverted links and subverted relaying

or forwarding devices; it affects the data plane of the routing protocol. However, a message replay may also be considered as a control plane violation.

There is an additional difficulty in cases in which correct operation of the routing protocol requires updates of a set of attributes. This is a common situation in distance vectors protocols.

We thus define the following classification of attributes:

- Attributes intended by their originator to reach adjacent nodes unmodified: "constant, neighborhood-limited" attributes.
- Attributes intended by their originator to keep constant values and to be propagated by adjacent nodes: "constant, propagated" attributes.
- Attributes intended by their originator to reach adjacent nodes unmodified and to be propagated after an update: "updateable, propagated" attributes.

5.3.1 Constant, neighborhood-limited attributes

These attributes must reach adjacent nodes unmodified. Possible attackers are: compromised links, subverted forwarding devices, masquerading routers.

This threat results from a lack of data integrity, data origin authentication and replay protection. Protection of data between adjacent nodes, especially anti-replay, has a tendency to focus on session management and on control plane.

The following requirements CAN be addressed either:

- at the control plane level,
- by the transport subsystem (the preferred solution),
- at the data plane level.

A routing protocol design SHOULD mention at which level these requirements are fulfilled:

R(3.1) Evidence of integrity and authenticity of data exchanged between neighbors SHOULD be provided; this evidence SHOULD be dependant on data destination. When the evidence applies on data description (as opposed to applying on a per-message basis), it

SHOULD also be dependant on the resource the attributes apply to.

R(3.1.a) It SHOULD NOT be possible to impersonate a neighbor. That is: authentication of neighbors SHOULD depend on a a-priori knowledge (a public key, a shared secret, knowledge of a direct connection in a common technical room, etc). This dependency MUST be documented in the protocol design.

R(3.2) Upon reception, data integrity and authenticity SHOULD be checked. This check SHOULD also include data destination and, when the check applies directly on data description (as opposed to applying on a per-message basis), that attributes apply to appropriate resources.

R(3.3) The routing protocol SHOULD be protected against the damages resulting from data replay. This CAN be done either by preventing replays effectiveness (ex: through integrity protected sequence numbers) or by reducing replays incidence on data (ex: through lifetime limited and authenticated data).

Practical considerations related to these requirements are presented in Section 7.2.

5.3.2 Constant, propagated attributes

These attributes must be relayed but not updated. Given requirements R(3.[1-3]) above, possible remaining attackers are: subverted relaying devices.

The threat here results from a lack of attributes integrity, origin authentication and lifetime limitation.

R(3.4) Integrity, data origin authenticity and validity at current date of DRI propagated attributes MUST be provided. The evidence MUST depend on the resource the attributes apply to.

R(3.4.a) This CAN be done in such a way that deletion, insertion or substitution of attributes will invalidate the whole DRI or a set of attributes when checked.

R(3.5) Consumers and relays of DRI MUST check that constant, propagated attributes apply to the resources and are the ones intended by the entity which set them.

R(3.6) Attributes validity MUST be lifetime limited.

Requirement R(3.4) addresses protection against unauthenticated insertion and substitution, and against partial deletion of an

attribute.

However, protection against deletion further depends on how attributes and resources are related, and if relays are allowed to delete attributes: this is the scope of requirement R(3.4.a). A design may apply different treatments on attributes which "must" be propagated and on attributes which "can" be propagated or discarded along the path of relays. The latter must not invalidate DRI when deleted.

Note: when routing entities along the path are identified, it is possible to achieve accurate control against early deletion of attributes whose range is limited. This may be done through explicit identification of routing entities in attributes whose deletion would invalidate DRI, and with the mandatory presence of a stand-alone, authenticated set of attributes addressed to each particular entity. In some cases, the identified entities may be virtual (groups).

Practical considerations related to these requirements are presented in Section 7.1.

5.3.3 Updateable, propagated attributes

These attributes must be relayed and updated. Given requirements R(3.[1-3]) above, possible remaining attackers are: subverted relaying devices.

The threat here results from the absence of a verifiable history of attributes updates. In the absence of any data trace-ability, it is difficult to figure out if a misstatement occurred.

The following aims only at offering a way round the problem and input for discussion; Figure 1 presents an instance of scenario.

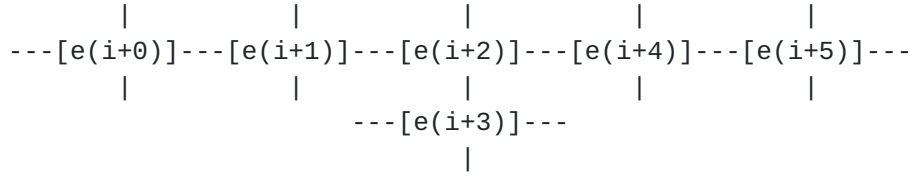

Figure 1: Updated Attributes

Suppose we consider updates as constant, propagated attributes following requirements R(3.[4-6]). These attributes, noted u(i), are authentic. A history is built from the succession of updates.

Now, if u(i) only describes an update, then e(i+2) gets the chain of updates [originated_value, ..., u(i), u(i+1)]. e(i+2) may discard u(i+1) and relay to e(i+3) the chain [originated_value, ..., u(i), u(i+2)]. That is: the chain integrity is unprotected. In order to avoid the discard, e(i) should mention that u(i) is addressed to e(i+1). The chain then becomes [originated_value, ..., {u(i) to e(i+1)}, {u(i+1) to e(i+2)}, {u(i+2) to e(i+3)}] and trivial discards can be detected. This requires of course more work for one-to-many routing protocols, building on multicast or broadcast addressing.

Other kind of discard is when a router appears several time in the path (loops): [originated_value, ..., {u(i) to e(i+1)}, {u(i+1) to e(i+2)}, {u(i+2) to e(i+3)}, {u(i+3) to e(i+2)}, {u'(i+2) to e(i+4)}, {u(i+4) to e(i+5)}]. e(i+5) can just discard sub-chain [{u(i+2) to e(i+3)}, {u(i+3) to e(i+2)}] in order to corrupt the chain. This will also not be countered if source is explicitly mentioned in updates (ex: {u(i+1) from e(i) to (e+2)}), because loops may be much more complex. A way out would be to have numbered updates.

In order to avoid cut-and-paste replays of some attributes, it may be necessary to have a chain number which is fed to the data origin authenticity and integrity function when adding an update. Similarly, routers should be able to impose an update down the chain with a way to signal an attribute freshness.

Note that in any cases, e(i+2) may forward [originated_value, ..., {u(i) to e(i+1)}, {u(i+1) to e(i+2)}, {u'(i+2) to e(i+4)}, {u(i+4) to e(i+5)}] instead of [originated_value, ..., {u(i) to e(i+1)}, {u(i+1) to e(i+2)}, {u(i+2) to e(i+3)}, {u(i+3) to e(i+2)}, {u'(i+2) to e(i+4)}, {u(i+4) to e(i+5)}] because e(i+2) owns the appropriate secret material. In doing so, e(i+2) would make misstatements as the result of being a Byzantine router, which is the topic of a later section.

5.4 Requirements Against Spoofing

"Spoofing occurs when an illegitimate device assumes the identity of a legitimate one" [THREATS].

Spoofing is possible because of a lack of combined integrity and data origin authentication. When considered an attack per se, spoofing is a threat on routing protocol control plane operations. It threatens neighbor relationship formation and state maintenance.

R(4.1) Requirements R(1.*) and R(3.[1-3]) also address spoofing.

R(4.1.a) In the context of spoofing, an emphasis SHOULD be made on the transport subsystem or on the control plane when interpreting requirements R(3.[1-3]).

It is often adequate to elect an appropriate transport subsystem which would provide functionalities against spoofing (cf. Section 7.3.1).

Requirements R(1.*) through R(4.*) aim at preventing against usurpation and deception. Following requirements address disruption and usurpation.

5.5 Requirements Against Overload

"Overload is defined as a threat action whereby attackers place excess burden on legitimate routers" [THREATS].

Overload is a threat from subverted links or devices. It may affect the data plane of the routing device, eg. as the result of link overload. It may also affect the control plane of the routing device, eg. by leading the victim router to use all its computational resources.

It is very unlikely that the design of a routing protocol will entirely prevent against this threat. However, the routing device may also include some functions which would limit the negative consequences of this threat (cf. Section 8).

At the routing protocol control plane, the following options are offered:

R(5.1) Fast rejection schemes based on tokens or cookies MAY be used. Such functionalities MAY be provided by the transport subsystem.

R(5.2) Above requirements regarding authentication of neighbors messages may result in limiting routing protocol data plane overload but at the expense of computational checks at the control plane. A design SHOULD consider and document opportunities of overloads resulting from protection against usurpation and deception.

R(5.3) The routing protocol operation SHOULD NOT suppose the full-time availability of protocols whose correct behavior depend on forwarding service (eg. live authentication through EAP).

R(5.4) The routing protocol design SHOULD limit the amount of traffic needed for correct operation. This is greatly dependant on the context the protocol addresses. This also implies slow recovery on

reboots.

Requirements R(5.[1-2]) suppose that authorized neighbors messages can be authenticated, which helps rejecting attackers solicitations. Further cautions are nonetheless required against neighbors acting in a Byzantine manner.

5.6 Requirements Against Interference

"Interference is a threat action where an attacker uses a subverted link or router to inhibit the exchanges by legitimate routers" [THREATS].

Interference is a general threat which can be perpetrated through:

- Noise addition
- Packets replay
- Denial of forwarding
- Denial of receipts
- Delay of responses
- Break of synchronization
- Slow down of exchanges
- Flapping

Noise addition, where it affects integrity of routing exchanges, is addressed by requirements against misstatements R(3.*). Where it affects the link layer or other traffic, the nature of the threat changes to break of synchronization, overload, etc.

Packets replay is addressed by requirements against misstatements R(3.*).

Denial of forwarding (of routing protocol packets) cannot be countered. However, it can be detected if the emitter expects a receipts, though it is difficult (yet not impossible) in such a case to make the difference with a denial of receipt (cf. R(6.1.*) below).

Denial of receipts can be detected; even if it is difficult to figure out the cause of the threat.

R(6.1.a) An implementation SHOULD revise the level of confidence (trust or stability) associated with paths getting through a neighbor with which it has been detected occurrence of denial of forwarding or denial of receipt.

R(6.1.b) The protocol design MAY affect the way these data are represented, and allow for signaling / sharing stability / trust information.

R(6.1.c) Neighbors are likely to keep states associated with each-others. Kind of keep alive / heart beat messages facility MAY periodically states that "All systems are functional, Everything is going extremely well." in order to set a threshold for triggering state hygiene functions and confidence revision (Note: a manual reboot should not be considered as a state hygiene process).

Delaying responses beyond a certain threshold is likely to break neighboring relationship, because a routing protocol implementation should time out a neighboring relationship beyond such a threshold (cf. breaks of synchronization, R(6.2)). Behind the threshold, delaying may result in the same consequences as a slow down of exchanges (cf. R(6.3)).

There are no way of preventing against breaks of synchronization from subverted links or routers. However:

R(6.2) Protocol design MUST take into account possible breaks of synchronization, even when the threat may only be accidental and improbable. State hygiene and computation of confidence level SHOULD be affected by the detection of such breaks.

Note: Occurrence of "break of synchronization" events may be regarded as a random variable whose evaluations (possibly biased) of central moments affect the level of confidence associated with the relationship.

Slow down of exchanges may be subjective. It is likely to affect pace to convergence, but slow down may also be a 'natural event' when traffic or processing is high, when queues are filling up, etc.

R(6.3) "Reactivity" of neighbors, possibly with knowledge of the traffic load on the link, MAY be a variable of the heuristic function which computes confidence associated with a neighbor or a DRI.

Flapping can be addressed by the routing protocol through the use of heuristics as for previous threats; however:

R(6.4) Part of flapping occurrences SHOULD be thwarted through enabling of the routing protocol to achieve atomic updates (as an instance, a removal and a subsequent addition which are related to overlapping fields of piece of DRI SHOULD be committed as an atomic update).

With the general threat of interference, the routing process is deemed to make choices based on a heuristic evaluation of the confidence associated with a particular neighbor or DRI. Requirements R(6.[1-3]) DO NOT prevent against threats actions, but aim at evaluating the cost of trust as associated to a link with a neighbor. The device may then allocate resources, invalidate DRIs, etc., according to the confidence measure. This is not conditions prevention; this is consequences limitation.

Last sections address requirements against disclosure.

5.7 Requirements Against Deliberate Exposure

"Deliberate Exposure occurs when an attacker takes control of a router and intentionally releases routing information directly to devices that, otherwise, should not receive the exposed information" [THREATS].

Deliberate exposure is an information leak about the routing system. Yet, it is unclear to which extend it affects the routing protocol. If neighbors take the exposure into account, then it turns to actually be a spoofing threat, and actual consequence is deception.

There is no way a local instance of the routing protocol may protect against this action if the attacker achieves full control of the device.

This threat may be limited by hardening access to the router, enforcing privilege separations, validating through external devices on the link, etc. This is not directly related to the routing protocol.

5.8 Requirements Against Sniffing

"Sniffing is an action whereby attackers monitor and/or record the routing exchanges between authorized routers. Attackers can use subverted links to sniff for routing information" [THREATS].

As mentioned in the threat document, confidentiality is not generally a design goal of routing protocols. However, confidentiality may be desirable when collecting votes (Byzantine participants may observe others votes and set their alignment so that majority is impossible

or lead to future consequences; on the other hand, clear text communications may also help detecting failures).

R(8.1) A routing protocol design process SHOULD investigate the needs for confidentiality. Conclusions from this process MAY be documented.

R(8.2) A routing protocol CAN optionally provide confidentiality. This SHOULD be implemented on the transport subsystem unless otherwise justified (eg. it is also possible to provide optional and partial confidentiality at the data plane level, or to conceal only a subset of messages).

R(8.3) When confidentiality is in scope, deployment, scalability and performance issues related to it's use SHOULD be studied and the conclusions documented.

5.9 Requirements Against Traffic Analysis

"Traffic analysis is an action whereby attackers gain routing information by analyzing the characteristics of the data traffic on a subverted link" [THREATS].

Even if the confidentiality of the routing traffic is activated, the attacker may access some routing information by analyzing the characteristics of data traffic.

Protections against traffic analysis include traffic flow confidentiality (TFC) (inter-times padding, data padding & compression, generation of dummy packets) and anonymity. Currently, these functionalities are scarcely used on the Internet and often oppose provision of quality of service.

Protecting only the routing protocol against traffic analysis is insufficient because analysis of user traffic will also leak information about the topology and the policies.

R(9.1) When user traffic is protected against traffic analysis, the routing protocol operations SHOULD investigate the use of a TFC & anonymity enabled transport subsystem shared with user traffic. Design of the routing protocol SHOULD be independent of this operational consideration, unless goal of the protocol is to set up the traffic flow concealing and 'anonymizing' network used by the transport subsystem.

R(9.2) When TFC & anonymity are among the design goals of the routing protocol, their effects on performance and correct operations of the routing system MUST be documented.

6. Living with Byzantine Failures

6.1 The Byzantine Problem

"A node with a Byzantine failure may corrupt messages, forge messages, delay messages, or send conflicting messages to different nodes" [BYZANTINE]. These faults may arise from routers which have been subverted by an attacker or which have faulty hardware or software [THREATS].

Byzantine resistance includes detection of Byzantine failures, Byzantine detection and Byzantine robustness, where the two latter are not necessarily correlated. Next section gives a thorough description of these forms of resistance.

The following main requirements aim at helping in the design of a Byzantine resistant routing protocol:

MR(3.1.a) Local instance of the protocol SHOULD NOT rely on correct operation of any particular neighbor.

MR(3.1.b) Operations associated with a particular neighbor SHOULD always apply a least privilege policy.

MR(3.1.c) Only traffic source and destination SHOULD be considered trustworthy.

MR(3.2) Messages MUST be authenticated when sent and checked for their authenticity when received (cf. also R(3.[1-3]). Use of cryptography simplifies the Byzantine problem.

6.2 Byzantine General Requirements

Classical hypotheses for Byzantine failure resolution are:

- devices are fully connected,
- the decision that must be agreed upon is binary (yes/no),
- the network is synchronous,
- strictly less than a third of the devices are faulty.

Under these hypotheses, a distributed algorithm requires as many rounds as the number of faults to be tolerated plus one.

Further information about distributed agreement can be found in

[CONSENSUS]. In the following, we will only focus on what makes the problem tractable in IP networks.

The ability to send messages to all participants simultaneously allow for simulation of both full connectivity and synchronization. The fact that routing information is not a agreeable binary decision has little consequences because agreement is not an absolute requirement; see Section 6.5 and [BYZANTINE].

6.3 Detection of the Occurence of a Byzantine Failure

The protocol algorithm may detect incoherences within the correlated routing information upon algorithm termination, abnormal attractive cycles within routes computations, etc. These events may be symptoms of a Byzantine failure occurring. More trivial evidences of a possible Byzantine failure is when agreement, termination or validity of the consensus cannot be achieved.

R(10.1) It SHOULD be possible to derive from a routing protocol design a set of coherence and sanity checks. The routing protocol documentation SHOULD mention directions when incoherence occurs, and describes reactions which are of direct impact on the protocol operation.

6.4 Byzantine Detection

Byzantine detection is much more accurate than just detecting a Byzantine failure and consists in the ability to find out which participants are subverted. A part of inherent risk of Byzantine detection is that when the number of traitors grow past a limit, it may be difficult for a device to figure out which group is subverted. Sometimes, the considered device may be itself -or conclude it is itself- faulty.

R(11.1) When Byzantine detection is achieved, automatic responses MAY be triggered in order to prevent Byzantine nodes from damaging operation of the routing protocol.

R(11.1.a) Automatic responses following a Byzantine detection MUST NOT prevent subverted devices from participating again when they cease to behave incorrectly.

R(11.1.b) Automatic responses following a Byzantine detection MUST NOT deceive non-faulty neighbors in concluding that responding devices are Byzantine nodes.

Possible automatic responses that may be investigated are the

simulation of a link shutdown, setup of adequate policies, quarantine cell. Collaborative approach between detectors to limit the influence of some subverted devices may be quite hazardous.

Either at the database maintenance level or at the forwarding function level, the following SHOULD be configurable when dealing with a detected subverted device:

R(11.2.a) "Detour": Allow or deny forwarding along an alternate route (if available), possibly on a path of "lower quality" (much many hops, long delay, etc). The routing protocol instance MAY also seek actively after an alternate path.

R(11.2.b) "Send & Hope": Allow forwarding to the subverted device anyway or,

R(11.2.c) "Discard": Treat destination as unreachable.

Eventually, note that sharing symmetric material for partial authentication between more than two devices would make Byzantine detection impossible to achieve in most cases (and so would do the absence of any authentication mechanism).

6.5 Byzantine Robustness

Purpose of Byzantine robustness, in the general problem context, is for any given device to achieve algorithm termination, agreement and -naturally- validity. This does not imply Byzantine detection.

However, in the routing context, what matters really is DRI correctness (cf. Section 2):

R(12.1) Routing protocols do NOT REQUIRE to achieve agreement.

R(12.2) Routing protocols do NOT REQUIRE to terminate; in fact, it is generally expected that they will not terminate during normal operation.

Some routing protocols address scenarii in which reachability is more important than policies and attributes associated with the destination. In such scenarii, Byzantine robustness aims at protecting reachability. This manages opportunities for "severed configurations" in which some policy requirements for a traffic could not be enforced though reachability is still possible / probable (Remember that what is often expected on the Internet is a high probability of packet delivery).

7. Security Techniques for Routing

7.1 Techniques when Originating

When originating, security requirements have a tendency to focus on the data plane. Indeed, data will further get relayed through the network, out of originator's catch. Security mechanisms addressing the control side will have no control on the way data are eventually propagated. Moreover, believing that other devices will relay the information unmodified is naive. As an instance, mere aggregation may be a threat against policies applying to resources.

As a consequence, it is important to know whether the originated information is authentic or not. In order to allow for this, information may be considered as a kind of 'record', composed of sections of the kind:

- Network resources description
- Related policies set by resources' controller.
- Information Lifetime
- Integrity and data-origin authenticity evidence of information, provided by the controller of the resources.

The division presented in Section 5.1 between the controller and the advertiser then allows for granting the advertising device with a very limited control on what is advertised; this is an interesting protection against potential damages resulting from possible advertiser's subversion. It also enables the controller with setting an authenticated registry of authorized advertisers. The concept of an authorization chain linking ownership, control and advertisement is nonetheless necessary in order to build confidence between neighbors from different organizations. An issue with this kind of model is the need for a definition (or furthermore: a specification and an allocation scheme) of identities.

Obviously, on a large scale, this kind of data protection requires public key operations, regardless of the actual technology eventually used (authorization tokens, digital signature). There are quite a lot of drawbacks associated with cryptography in general and with public key cryptography in particular.

Where these drawbacks affect devices, an increase amount of memory is needed for buffering cryptographic information and for caching (cf. next paragraph). Besides, public key operations are also quite CPU consuming. A performance study SHOULD be pursued when designing a

routing protocol using cryptography; threats opened because of crypto processing SHOULD NOT nullify the interest of tackling routing threats which would result in comparable consequences (eg. disruption). A performance study often requires hypotheses on the underlaying hardware, which is somewhat restricting but necessary.

Where these drawbacks concern the overall architecture, they involve deployment, administration and public information reachability issues. Regarding this latter topic, in-band or stand-alone channels are necessary for the provision of public data, for revocation and for key roll-over. A routing protocol may find itself in a dead-end if such a channel is needed for authenticity check of data which are necessary to enable access to the ad-hoc channel. This is a tricky point, which claims for a distributed caching mechanism. Caching is all the more important when scalability is a significant issue and when centralization of data creates bottleneck; on the other hand, the whole architecture is less reactive in case revocation or key roll-overs are required, even though soft key transitions should not be necessary in this context.

Further in this direction, neighbors' public material may be kept in non-volatile storage for recovery. There may be no routes available in order to retrieve this material after a reboot, though in-band provisioning within the routing protocol is also a possibility.

Hence, security from the originating part is the big problem of routing security. In effect, a trade-off must be found between performance (sensibility to denial of service) and heavy cryptographic protection, public material reachability and it's synchronization.

As for setting up channels for public material diffusion: this requires an expensive investment in architecture. Consequently, temptation may be high to rely on other architectures, especially when large scalability is in scope: DNS and the routing and forwarding system itself are the architectures which (almost) succeeded in scaling, but care must be taken not to misuse or overload these. Whatever the path taken by an architecture specification, it's resistance against trivial denial of services must be evaluated.

Requirements related to this section are R(1.*), R(2.*), R(3.[4-6]).

All cryptographic material MUST have their lifetime limited, and both evaluated in terms of time and in terms of amount of data.

Public keys strength is a matter of context: in inter-domain operations, one may expect that public material will not change very

often, and then such a material should be significantly strong. Locally, the rate of public material updates may depend on administrator's decision; he alone evaluates the risks for the network and the administrative cost. In a conference, people may build a ephemeral network by exchanging public material on an direct IR link before roaming and participating in ad-hoc routing through wireless links; public material is such a case would only be used a few hours and may be kept voluntarily weak.

7.2 Techniques when Relaying

According to the distinction made in Section 1, the following concerns relays but not forwarders.

When relaying, security requirements have a tendency to focus on the control plane. Relaying security is that of entities communicating in a direct fashion (and perhaps interactively) over the transport subsystem. In such a situation, we're concerned with:

- Integrity: data integrity between neighbors is an obvious requirement. Note that error detection and correction codes are not integrity evidences. Means to achieve integrity are signed-hash and keyed-hash. Data integrity is always closely related to authenticity.

- Authenticity: the above feature is of no use without authentication of the information producer. Authenticating correctly the messages sent from neighbors is one of the most important security requirement. Authentication techniques that can be considered are: digital signature, keyed hash.

- Anti-replay: comes here mainly for protection against active attacks from subverted Links, though this feature will also provide protection against 'natural' packets duplication. Note that underlying layers may provide an unauthenticated anti-replay feature, which would be of no use from a security point of view unless it gets also authenticated. Authentication of routing exchanges sequence numbers may bring this kind of protection to the protocol.

Other features include confidentiality and traffic flow confidentiality, which are generally out of scope in routing protocols (cf. R(8.*) and R(9.*)).

Main differences with origin-based security practices presented in the previous section include:

- message oriented protection (as opposed to data protection),

- messages are addressed (to one or many peers),
- messages are limited in time through anti-replay techniques (as opposed to limited lifetime),
- neighbors may use symmetric cryptography.

The above characteristics may be implemented by the routing protocol, or by the transport subsystem. In this latter case, a specification MUST document which security properties are provided by the transport subsystem, which are provided by the routing protocol and, eventually, how they interact. Note that transport subsystems may experience evolutions; as a trivial instance, one may design a routing protocol which will run on wire Ethernet (802.3) with the hypothesis that physical and logical access to layer 2 infrastructure is under control. Such an hypothesis may no longer be suitable on wireless Ethernet (802.11).

Further protection may include range limiting features, enabled by the use of special addresses (link-local, limited broadcast, multicast) or of counter based schemes (TTL). Most of these features are provided by adequate transport subsystems.

Specific issues for communications between neighbors include:

- Address protection: sometimes extra care is needed against transport subsystem's address spoofing, even though an identity has been defined at an upper layer. Address protection requires inclusion of the address in the integrity and authenticity evidence computation. [AH] may be seen as an instance of a protocol with built-in address spoofing protection.
- In 'one to many' communication contexts, sharing symmetric material opens opportunities for damages resulting from subverted insiders.
- Interactivity involves managing sessions and keeping states associated with neighbors. For the sake of state hygiene, reactivity of neighbors SHOULD be evaluated. This calls for setting delays threshold, using keep alive / heart beat mechanisms and explicitly tearing sessions down.
- Participants are vulnerable to direct computational harassment, against which DOS mitigation mechanisms are necessary. These include puzzles, cookies, tokens chains.

Note: There had been several discussions on the use of a token based fast rejection scheme, which could be embedded on interfaces of the

devices. Such a scheme would protect against a category of denials of service in which malign traffic gets in at a high rate. The management of such a scheme may require a stand-alone protocol and raises issues when neighbors communicate through several interfaces.

Requirements related to this section are R(1.*), R(3.[1-3]) and R(4.*). Section 5.3.3 is also related to this section.

When possible, methods to derive a symmetric key from public exponents should be used, given that the symmetric cryptography operations considered are less computationally expensive. Caution should be taken if the number of devices sharing the same symmetric key is greater than two.

Limiting keys lifetime and refreshing them is good cryptographic hygiene. Therefore, a mechanism to roll-over keys is REQUIRED both for public keys and for session keys; Public keys roll-over may not require a soft transition, while refreshing session keys may require to move from the old key to the new one with no session interruption. Lifetime MUST be evaluated both in terms of time and of amount of data.

7.3 Security of the Functional Parts

The threats document [THREATS] introduces a set of functions commonly shared by routing protocols: the transport subsystem, the neighbor state maintenance function and the database maintenance function.

Each of these functions may contain inner security weaknesses and simultaneously a potential for providing adequate security services for the interest of operation of the whole system.

In the following sections, the security related parts of these functions are explored.

7.3.1 The Transport Subsystem

"The routing protocol transmits messages to its neighbors using some underlying protocol. For example, OSPF uses IP, while other protocols may run over TCP" [THREATS].

One may design a routing protocol independent -to a certain extent- from a specific transport subsystem, by requiring the availability of a minimal set of capabilities from this subsystem.

Yet, relevant, specific capabilities of a transport subsystem should be exploited by a routing protocol. An adequate transport subsystem provides capabilities which would be cumbersome if included in the

routing protocol itself and have been -ideally- thoroughly tested. This is a net gain in complexity, even though at the expense of added complexity on protocol interactions and addresses resolution mechanisms.

FR(T.1) A routing protocol specification SHOULD document which capabilities of the transport subsystem are exploited by the routing protocol.

FR(T.2) Where issues may arise from interactions between the transport subsystem and the routing protocol, the specification MUST mention these issues (The "Security Considerations" section may be the appropriate place for IETF/IRTF documents).

The transport subsystem may already provide the following properties:

- Neighbors discovery and maintenance: A given Transport Subsystem technology may provide a way to discover and communicate with adjacent devices participating in the routing domain (neighbors). This is a critical property.

- Range limitation: the subsystem may provide a way to limit propagation of messages outside a certain range and in the same way limit intrusions from outsiders in the neighborhood. This may be achieved either through the use of an appropriate layer (likely, link layer), through special addresses (limited broadcast, multicast, link-local, site-link, etc.), through conditions expressed on TTL (see also [BTSH]). This provides a limited access control to neighborhood (yet, there are ways around these limitations: VLAN frames hopping, tunneling).

- Separate control channel: if the underlying technology provides separated channels for control traffic and user data traffic, this may help against DOS against the routing protocol. Such control channels may be provided via the same Link Layer infrastructure, or perhaps via a distinct network.

- Integrity: While the Transport Subsystem chosen by the routing protocol designer may provide error detection code, this does not provide data integrity from a security point of view. The Transport Subsystem may also provide data integrity which will still be useless from a security perspective if the secret material used by the data integrity service cannot be tied to the routing protocol participant identity.

- Authenticity: if the underlying layer both provides authenticity and integrity, many routing threats may be thwarted. Further investigations are required though, among which are studies of

resistance to replay, performance, Byzantine detection and robustness, etc. In such a case, the documentation of the routing protocol MUST state which security properties are provided by the Transport Layer, which are provided by the routing protocol design and eventually how they interact (cf. FR(T.2)).

- o Address spoofing protection: the subsystem is protected against address spoofing if integrity and authenticity evidence covers also the address.

7.3.2 The Neighbor State Maintenance

"Neighboring relationship formation is the first step for topology determination. For this reason, routing protocols may need to maintain state information. Each routing protocol may use a different mechanism for determining its neighbors in the routing topology. Some protocols have distinct exchanges through which they establish neighboring relationships, e.g., Hello exchanges in OSPF" [THREATS].

[TBD] Cookies ? Damping ?

7.3.3 The Database Maintenance

"Routing protocols exchange network topology and reachability information. The routers collect this information in routing databases with varying detail. The maintenance of these databases is a significant portion of the function of a routing protocol" [THREATS].

From a local perspective, and with a selfish point of view, database maintenance is what really matters for a particular device.

For this reason, resources should be 'flagged' according to trust, stability, quality... scales.

Coherence of information may be checked actively (with probes) and passively (observation of user traffic). In ad-hoc contexts, database may also be fed reactively.

7.3.3.1 Fail-back Procedures

When detecting obvious routing misbehavior which result from misuse of the routing protocol, but when sources responsible for this misbehavior cannot be identified (no Byzantine detection), fail-back procedures may be attempted, based on previous recorded states, fail-safe states or heuristics on the routing information and on trust. Degradation of the service should often be better than no

service at all, thus the device may adjust local route costs information when such events occur. The routing protocol design may document guidelines and requirements on such procedures.

Network management must be able to install unalterable (static) routes to allow debugging network problems without interference from routing protocols.

8. Local Security

8.1 Active Participation to Security

Topics presented within this section may not be directly tied to the protocol design. However, it addresses several local considerations that are requirements for a secure operation of the routing protocol and of the device it is running on.

8.1.1 Checking

A routing device may be configured to run extra checks on the routing state, like checking databases against previous information. Some active tests may also be triggered: sending source routed ICMP packets, etc. Such tests may also involve the neighbors. High caution should be taken regarding implementation of such features and they should not jeopardize the routing protocol mechanisms.

8.1.2 Reporting

A set of error messages may be designed in order to report detection of failures to other participants. Locally, a set of auditable events MUST be defined.

8.1.2.1 Auditable Events

The following events should be audited:

1. Authentication failure

2. Required public information (keys, authority) is not available

3. Errors reported by forwarders

4. Detection of a Byzantine event

5. Detection of a rebooting peer

[TBD] The above has nothing to do with routing. Or has-it ? Should the protocol automate detect and act according to the detection of these events ?

8.1.3 Reacting

8.1.3.1 Filtering

Upon detection of subverted devices, a process may enforce security procedures such as ingress filtering or participant exclusion.

A routing device MAY be set to drop/reject routing messages if these are incorrect with current configuration of the network, e.g. if they do not belong to the correct range of the IGP, etc.

Note that this protection is topological and partial. Extreme care should be taken not to jeopardize correct behavior of the protocol.

8.1.3.2 Correcting

[TBD] Correcting wrong / malicious routing info; investigate correction resulting from the above procedures.

8.2 Local Resources Considerations

Even though this document addresses routing protocols, these cannot operate without a platform of hardware and software to support them. All the resources belonging to this platform form what is generally referred to as a router. Thus, routers comprise all local resources of a routing daemon participating in a routing session.

This section will first highlight critical underlying components and their security issues regarding Denial of Service (DoS) vulnerabilities and then suggest suitable routing protocols' requirements addressing these issues.

8.2.1 Denial of Service Attacks

The Computer Emergency Response Team (CERT) defines in [DOS] Denial of Service attacks as being explicit attempts by attackers to prevent legitimate users of a service from using that service. Denial of Service attacks can be launched against a target for the mere purpose of preventing the victim from using a resource or can be a component of a greater attack that may ultimately aim at stealing information.

A modern router is a complex system made of several hardware and software components that interact in the effort to serve the general purpose of routing as defined in Section 2. All of these components are finite resources and therefore intrinsically prone to Denial of Service. The impact of Denial of Service attacks on certain local resources can be critical for the routing protocols running on them.

8.2.2 Hardware Resources

Almost every hardware component in a router is essential to the correct functioning of the local instances of the various routing protocols that run on it, for example - trivially speaking - without power no packets will be routed. Among others buffers/queues and CPU cycles are two of the less obvious resources that are critical for

routing protocols.

8.2.2.1 Buffers/Queues

Buffers are widely used in hardware to store information that needs to be aggregated or delayed before being consumed. In general once a buffer is full every subsequent object that needs to be stored in that queue will simply be discarded. Depending on what messages are discarded, the consequences of dropping information for routing protocols can vary from negligible to critical.

Since all messages exchanged between participants to a routing session need to reach the control-plane, the queues and buffers that support this link are critical for routing protocols. Often people are deceived by thinking that the throughput of a switching fabric is roughly the amount of bandwidth needed to launch a DoS attack against a given router; in reality, routers have smaller bandwidth links toward the control plane. The goal of an attacker could be easier in terms of resources, if he/she were to attempt to exhaust the buffers and queues on the link to the control plane with bogus control plane packets rather than trying to congest the resources serving the switching fabric. The goal of such attacks would be to cause queues and buffers to drop legitimate routing messages together with bogus ones.

8.2.2.2 CPU Cycles

Processors units, and in particular Network Processors (NPs), are a valuable resource that can perform predetermined sets of operations during a single cycle. Generally speaking, CPU cycles are a finite resource that is shared among many different processes, some of these being instances of routing protocols. As a consequence of congestion, and from an oversimplified point of view, some processes may be put "on hold" until more CPU cycles are available, or every process may be "starved a bit". Both scenarios may cause great damage to interactive processes. In particular routing protocols' instances may enter critical states where a timely reaction to an event is necessary but not available.

In general the more a CPU serves an heterogeneous pool of processes, the more easy it will be for an attacker (or a faulty router) to find a single service/process that will exhaust a significant portion of the available CPU cycles, denying service to other processes, such as routing.

8.2.2.3 Buffer/Queues and CPU Cycles Requirements

Routing messages SHOULD be identifiable as coming from legitimate

participants in their routing session before being directed towards the control-plane.

If any rate limiting mechanism is intended by the routing protocol to mitigate congestion of control-plane links, said solution MUST be designed ensuring that an attacker cannot directly exploit it in the attempt to block a legitimate routing peer from exchanging routing messages.

8.2.2.4 Bandwidth

Routing protocols are based on the exchange of information between the participants to a session over network links. A link's bandwidth is finite critical resource that, if starved, can lead to Denial of Service attacks on the routing protocols. If a link is not malfunctioning, and neglecting transmission errors, then DoS attacks on a link's bandwidth can only take place at the link's ends. A router may receive an aggregate of traffic higher than it can be forwarded by a given output interface, or a receiving router may not be capable of handling the current load of traffic incoming on a given interface due to an internal scheduling priority problem or because it entered a critical or unknown state.

8.2.2.4.1 General Mitigation Techniques

Some mitigation techniques can be deployed to limit the exhaustion of bandwidth between two routing peers; two current examples are: ingress filtering, as described in [FILTERING], and solutions that relay on Quality of Service mandating that the highest priority and availability be assigned to routing messages.

8.2.2.5 Bandwidth Requirements

Routing protocols MUST be designed to easily inter-work with lower layers Quality of Service mechanisms.

8.2.3 Logic (Software) Resources

Similarly to hardware resources, logic resources can be finite and therefore exhausted thus affording attackers with the possibility of launching Denial of Service attacks. Databases are critical resources for every routing protocol and they may contain information about link-state, direct neighbors, active peers, external routes database, etc...

Routing databases have a maximum number of entries that can be stored in them and this is generally not defined by the routing protocols. This upper bound can be set by an administrator through a

configuration parameter or can be restricted only by the hardware memory available to the routing platform. Either way, when this limit is approaching, for any of the databases maintained by a routing protocol, some action must be taken.

8.2.3.1 Logic (Software) Requirements

Routing protocols MUST mandate verification of every piece of information that can be verified before committing it to any underlying database.

Every piece of information that cannot be verified by the routing protocol immediately MUST be marked as temporary and means should be provided, by the routing protocol itself, to keep track of these entries, verify and discard them as soon as possible.

Every piece of information that cannot be verified by the routing protocol MUST be installed in the apposite database with the minimum time to live compatible with its function.

Routing protocols MUST provide mechanisms for routing platforms' databases, in overflow state, to discard information that will cause minimum possible disruption to the routing session.

Routing protocols SHOULD be designed as to incorporate feed-back solutions from databases approaching overflow state so that mitigative actions can be taken.

Routing protocols SHOULD be designed with the concept of graceful degradation in mind in order to better survive in case any of the underlying databases approaches or enters overflow state.

9. Inter-Domain Routing Issues

9.1 Legitimacy

An important issue in inter-domain routing is legitimacy for claiming network resources. In fact, this is where confidence edifice starts. Requirements R(1.*) are related to this topic, though they do not address some decisions.

Parts of these decisions regard routes specialization.

Hierarchical addressing is necessary in order to aggregate entries in local forwarding tables; this reduces tables size and improve general performances, even though this may threaten performances on a specific path. When preferred (eg. for confidentiality reasons), some specific routes may appear in the table. A problem with hierarchical addressing is that, when used as such in the routing protocol, it may generate resources masking. This is especially obvious with operations like aggregations of destinations or removal of a specific destination: both these operations will result in the generic entry taking over the specific one.

These operations may be considered as a violation of ownership, though it is also unclear whether a shorter prefix ownership should -administratively speaking- involve authority on a corresponding longer prefix.

On the other hand, if care is taken within the routing protocol to protect specific routes against overclaiming resulting from aggregations or removal, then this involves extra architecture requirements and more bandwidth get consumed in routing protocol exchanges.

Besides, this will not prevent forwarding tables from aggregating or removing entries, and this kind of decorrelation between forwarding and routing information may not be desirable, even though loose relation between forwarding tables and routing information is common.

Another part of the problem is public information reachability.

When public material may help in establishing right to claim resources, availability of the required material is problematic. Section 7.1 presents this in further details. With regard to public cryptography, it should be clear that a light paradigm (authorizations ?) would better fit in most cases, though third parties also appear to be a necessity at this point.

9.2 Policies

Policy propagation within a routing protocol which operates between administrative routing domains, exterior gateway protocols, is very difficult. This particular area of security is fraught with difficulties making it next to impossible to actually secure policy across multiple administrative domains.

Since each administrative domain can add policies to a given route, anyone can essentially insert any policy. Even if a full history of policies is available, the question: "Who's policy are we honoring ?" has to be answered. The originator's policy ? Or the AS we received the route from ? Or the AS that currently has the route ? Or some other AS ?

9.3 Coherence

Where domains are multi-homed, should operations of the edge routers be coherent ? In a nutshell: should a domain be considered as a standalone, non-schizophrenic, entity ? Note that coherence does not preclude edge routers from behaving differently.

9.4 Confidentiality

As was mentioned several times previously, confidentiality is usually not a design goal of routing protocols. In inter-domain operations, enabling confidentiality would require finding a balance between information that is required to be publicly available and information whose concealing is desirable. May be a possible path is not to care about concealing destination info, but about policies. Yet, the value of a route without knowledge of according policies is certainly dubious.

9.5 Agreements involving operators

Secure EGPs operations will require kind of agreements between the involved parties. Though operators may achieve these agreements on a case by case basis, this is unlikely to be effective in the field. Emergence of trusted third parties upon which would rely the diffusion of public key material and relations to prefix ownership would fit better.

Another question is whether these pieces of information must be tied with public information related to the system ownership, such as the organization name. This may lead to specific routing policies or abuses that would introduce more complexity.

A limited set of fields composing a signed tuple could include an

identity (WRT to RP), address(es), public key, authorization on prefixes and adequate lifetimes.

Access control also imply agreements: who's granted right to participate to the protocol ?

10. Security Considerations

This entire draft RFC is security related. Specifically it addresses security of routing protocols and routing systems as associated with requirements to those protocols and systems. In a larger context, this work builds upon the recognition of the IETF community that signaling and control/management planes of networked devices need strengthening. Routing protocols and routing systems can be considered part of that signaling and control plane, may be the most important. However, to date, these protocols and systems have largely remained unprotected and opened to malicious attacks. This document discusses routing protocol and routing system security requirements as we know them today and lays the foundation for the design of new, more secure, routing protocols and systems.

Normative References

[AH] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998, <http://www.ietf.org/rfc/rfc2402.txt>.

[DAMPING] Villamizar, C., Chandra, R. and R. Govindan, "BGP Route Flap Damping", RFC 2439, November 1998, <http://www.ietf.org/rfc/rfc2439.txt>.

[FILTERING] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000, <http://www.ietf.org/rfc/rfc2827.txt>.

[KEYWORDS] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

[SEC-GLOSS] Shirey, R., "Internet Security Glossary", RFC 2828, May 2000, <http://www.ietf.org/rfc/rfc2828.txt>.

Informative References

[BTSH] Vijay, G., Heasley, J. and D. Meyer, "The BGP TTL Security Hack (BTSH)", Internet Draft; version 02, May 2003, <http://www.ietf.org/internet-drafts/draft-gill-btsh-02.txt>.

[BYZANTINE] Perlman, R., "Network Layer Protocols with Byzantine Robustness", , August 1988, <http://www.vendian.org/mncharity/dir3/perlman_thesis/>.

[CONSENSUS] Coulouris, G., Kindberg, T. and J. Dollimore, "Distributed Systems: Concepts and Design", Addison Wesley ISBN - 0201619180, 2000 September.

[DOS] CERT, "Denial of Service Attacks", June 2001, <http://www.cert.org/tech_tips/denial_of_service.html>.

[SMITH] Smith, R. and al., "Securing Distance-Vector Routing Protocols", Symposium on Network and Distributed System Security , February 1997, <http://www.isoc.org/isoc/conferences/ndss/97/smith_sl.pdf>.

[THREATS] Barbir, A., Murphy, S. and Y. Yang, "Generic Threats to Routing Protocols", Internet Draft; version 06, April 2004, <http://www.ietf.org/internet-drafts/draft-ietf-rpsec-routing-threats-06.txt>.

Authors' Addresses

Jean-Jacques Puig
CNRS / UMR 5157 (Samovar) / Piece A-108
9, Rue Charles Fourier
Evry 91011
France

Phone: +33 1 60 76 44 65
Fax: +33 1 60 76 47 11
EMail: jean-jacques.puig@int-evry.fr
URI: http://www-lor.int-evry.fr/~puig/

Mohammed Achemlal
France Telecom R & D

EMail: mohammed.achemlal@francetelecom.com

Emanuele Jones
Alcatel Canada - R&I - Security group
600 March Road
Kanata, ON K2K 2E6
Canada

Phone: +1 613 784 5977
Fax: +1 613 784 8944
EMail: emanuele.jones@alcatel.com

Danny McPherson
Arbor Networks

EMail: danny@arbor.net

Appendix A. Revision History

A.1 Changes from draft-puig-rpsec-generic-requirements-01

TOC tweaking. Phrasing simplifications. Development of the requirements.

A.2 Changes from draft-puig-rpsec-generic-requirements-00

Full TOC change.

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.