

I2RS WG
Internet-Draft
Intended status: Informational
Expires: December 27, 2015

D. Migault, Ed.
J. Halpern
Ericsson
June 25, 2015

I2RS Security Requirements
draft-mglt-i2rs-security-requirements-00

Abstract

This document provides security requirements for the I2RS architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 27, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology and Acronyms	3
3.	I2RS Plane Isolation	3
3.1.	I2RS plane and management plane	5
3.2.	I2RS plane and forwarding plane	6
3.3.	I2RS plane and Control plane	6
3.4.	Recommendations	6
4.	I2RS Authentication and Authorization Access Policy for routing system resources	8
4.1.	I2RS AAA architecture	8
4.2.	I2RS Agent AAA	10
4.3.	I2RS Client AAA	11
4.4.	I2RS AAA Security Domain	12
4.4.1.	Available I2RS Communication Channel	12
4.4.2.	Trusted I2RS Communications Channel	13
5.	I2RS Application Isolation	14
5.1.	Robustness toward programmability	14
5.2.	Application Isolation	15
5.2.1.	DoS	15
5.2.2.	Application Control	16
6.	Privacy Considerations	16
7.	IANA Considerations	16
8.	Acknowledgments	16
9.	Informative References	17
	Authors' Addresses	17

[1.](#) Introduction

This document addresses security considerations for the I2RS architecture. It provides guidance and security principles to guarantee the stability of the I2RS architecture. This documents provides an analysis of the security issues of the I2RS architecture beyond those already listed in [[I-D.ietf-i2rs-architecture](#)].

Even though I2RS is mostly concerned by the interface between the I2RS Client and the I2RS Agent, the security recommendations must consider the entire I2RS architecture, specifying where security functions may be hosted, and what should be met so to address any new attack vectors exposed by deploying this architecture. In other words, security has to be considered globally over the complete I2RS architecture and not only on the interfaces.

I2RS architecture depicted in [[I-D.ietf-i2rs-architecture](#)] describes the I2RS components and their interactions to provide a programmatic interface for the routing system. I2RS components as well as their interactions have not yet been considered in conventional routing

systems. As such it introduces a need to interface with the routing system designated as I2RS plane in this document.

This document is built as follows. [Section 3](#) describes how the I2RS plane can be contained or isolated from existing management plane, control plane and forwarding plane. The remaining sections of the document focuses on the security within the I2RS plane. [Section 4](#) analyzes how the I2RS Authentication Authorization and Access Control (I2RS AAA) can be deployed throughout the I2RS plane in order to only grant access to the routing system resources to authorized components with the authorized privileges. This also includes providing a robust communication system between the components. Then, [Section 5](#) details how I2RS keeps applications isolated one from another and do not affect the I2RS components. Applications may be independent, with different scopes, owned by different tenants. In addition, they modify the routing system that may be in an automatic way.

The reader is expected to be familiar with the [\[I-D.ietf-i2rs-architecture\]](#).

[QUESTION: Some suggested to use system instead of plane. Which is the more appropriate terminology?]

[2. Terminology and Acronyms](#)

- I2RS plane :
- I2RS user :
- I2RS AAA :
- I2RS Client AAA :
- I2RS Agent AAA :

[3. I2RS Plane Isolation](#)

Isolating the I2RS plane from other network plane, such as the control plane, is foundational to I2RS security. Clearly differentiating I2RS components from the rest of the network protects the I2RS components from vulnerabilities in other parts of the network, and protect other systems vital to the health of the network from vulnerabilities in the I2RS plane. Separating the I2RS plane from other network control and forwarding planes is similar to the best common practice of containerizing software into modules, and defense in depth in the larger world of network security.

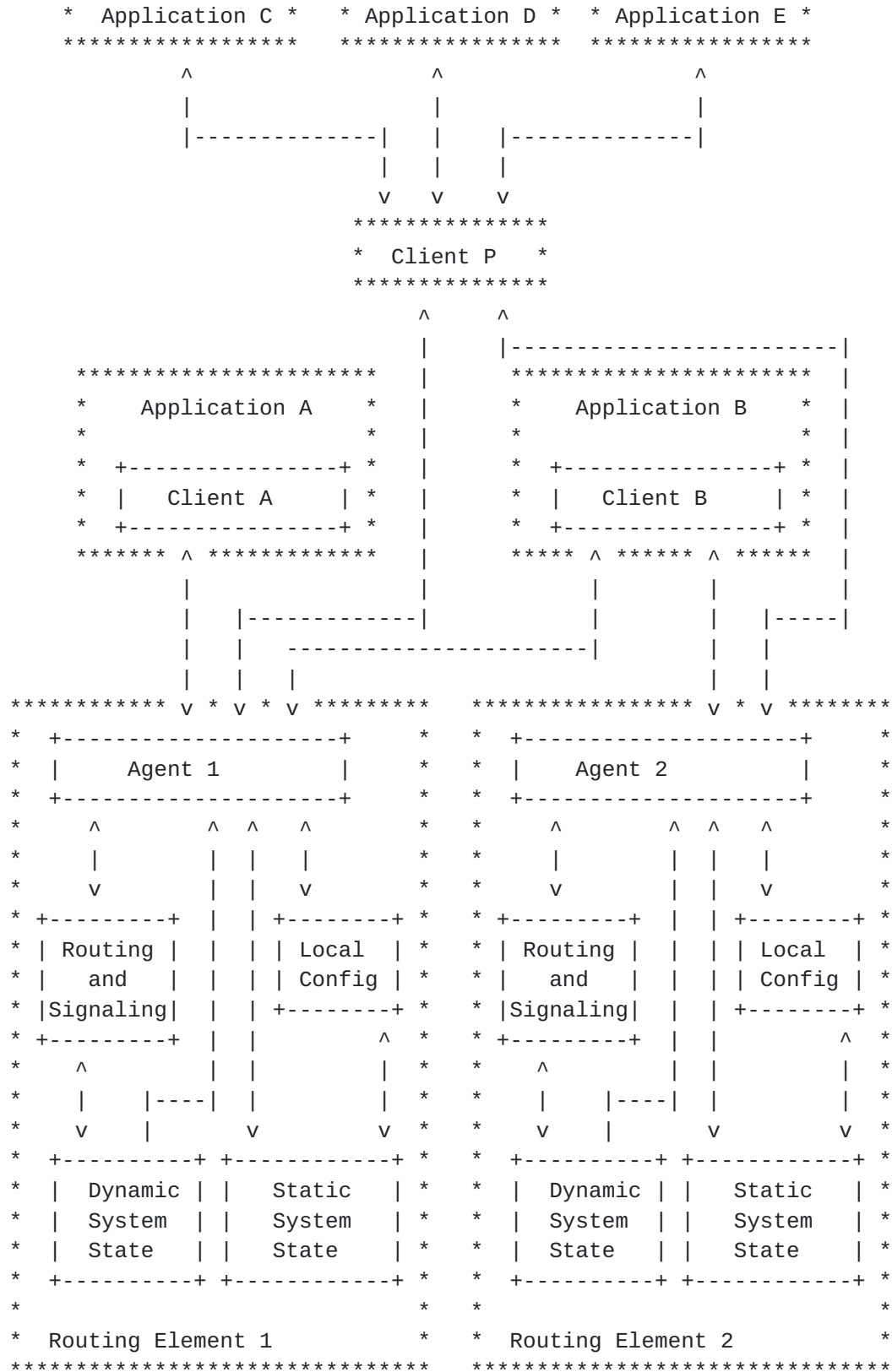


Figure 1: Architecture of I2RS clients and agents

Figure 1 (copied from [[I-D.ietf-i2rs-architecture](#)]) depicts the I2RS components that constitute the I2RS plane. More specifically, the network oriented applications, the I2RS Client, the I2RS Agent, and to some extent the routing system resources accessed by the I2RS Agent. The I2RS plane has its own goals and specificities that make it a separate plane from the others. The I2RS plane purpose is to provide a standard programmatic interface of the routing system resources to network oriented applications. Control plane and forwarding planes are related to routing protocols, and I2RS is based on top of those. The management plane is usually vendor specific, provides a broader control over the networking equipment such as system service. Given its associated privileges it is expected to be reserved to highly trusted users like network administrators.

[QUESTION: Should we remove the text above and the figure?]

That said the I2RS plane cannot be considered as completely isolated from other planes, and interactions should be identified and controlled. Follows a brief description on how the I2RS plane positions itself in regard to the other planes. The description is indicative, and may not be exhaustive.

3.1. I2RS plane and management plane

The I2RS plane and the management plane both interact with several common elements on forwarding and packet processing devices. Figure 1 shows several of these interaction points, including the local configuration, the static system state, routing, and signalling. Because of this potential overlaps, a routing resource may be accessed by different means (APIs, applications) and different planes. To keep these overlaps under control. On the other hand, one could either control the access to these resources with northbound APIs for example, and if conflicting overlaps cannot be avoided, then conflicts should be resolved in a deterministic way. On the northbound side, there must be clear protections against the I2RS system "infecting" the management system with bad information, or the management system "infecting" the I2RS system with bad information. The primary protection in this space is going to need to be validation rules on the speed of information flow, value limits on the data presented, and other protections of this type. On the conflicting side, there should be clear rules about which plan's commands win in the case of conflict in order to prevent attacks where the two systems can be forced to deadlock.

3.2. I2RS plane and forwarding plane

Applications using I2RS are part of the I2RS plane but may also interact with other components outside the I2RS plane. A common example may be an application uses I2RS to configure the network according to security or monitored events. As these events are monitored on the forwarding plane and not the I2RS plane, the application breaks plane isolation.

In addition, applications may communicate with multiple I2RS clients; as such, any given application may have a broader view of the current and potential states of the network and the I2RS plane itself. Because of this, any individual application could be an effective attack vector against the operation of the network, the I2RS plane, or any plane with which the I2RS plane interacts. There is little the I2RS plane can do to validate applications with which it interacts, other than to provide some broad general validations against common misconfigurations or errors. As with the separation between the management plane and the I2RS plane, this should minimally take the form of limits on information accepted, limits on the rate at which information is accepted, and rudimentary checks against intentionally formed routing loops or injecting information that would cause the control plane to fail to converge. Other forms of protection may be necessary.

3.3. I2RS plane and Control plane

The network control plane consists of the processes and protocols that discover topology, advertise reachability, and determine the shortest path between any location on the network and any destination. It is not anticipated there will be any interaction between the on-the-wire signalling used by the control plane. However, in some situations the I2RS system could modify information in the local databases of the control plane. This is not normally recommended, as it can bypass the normal loop free, loop free alternate, and convergence properties of the control plane. However, if the I2RS system does directly inject information into these tables, the I2RS system should ensure that loop free routing is preserved, including loop free alternates, tunnelled interfaces, virtual overlays, and other such constructions. Any information injected into the control plane directly could cause the control plane to fail to converge, resulting in a complete network outage.

3.4. Recommendations

To isolate I2RS transactions from other planes, it is recommended that:

- REQ 1: Application-to-routing system resources communications should use an isolated network. An isolated network may be provided with various level of isolation. The highest level of isolation may be provided by using a physically isolated network. Alternatives may also consider logical isolation; for example by using VLAN. Eventually, in virtual environment that shares a common infrastructure, encryption may also be used as a way to enforce isolation.
- REQ 2: The interface (like the IP address) used by the routing element to receive I2RS transactions should be a dedicated interface.
- REQ 3: The I2RS Agent validates data to ensure injecting the information will not create a deadlock with any other system, nor will it create a routing loop, nor will it cause the control plane to fail to converge.

When the I2RS Agent performs an action on a routing element, the action is performed via process(es) associated to a system user . In a typical UNIX system, the user is designated with a user id (uid) and belong to groups designated by group ids (gid). These users are dependent of the routing element's operation system and are designated I2RS System Users. Some implementation may use a I2RS System User for the I2RS Agent that proxies the different I2RS Client, other implementations may use I2RS System User for each different I2RS Clients.

- REQ 4: I2RS Agent should have permissions separate from any other entity (for example any internal system management processes or CLI processes).

I2RS resource may be shared with the management plane and the control plane. It is hardly possible to prevent interactions between the planes. I2RS routing system resource management is limited to the I2RS plane. As such, update of I2RS routing system outside of the I2RS plane may be remain unnoticed unless explicitly notified to the I2RS plane. Such notification is expected to trigger synchronization of the I2RS resource state within each I2RS component. This guarantees that I2RS resource are maintained in a coherent state among the I2RS plane. In addition, depending on the I2RS resource that is updated as well as the origin of the modification performed, the I2RS Authentication Authorization and Access Control policies (I2RS AAA) may be impacted. More especially, a I2RS Client is more likely to update an I2RS resources that has been updated by itself, then by the management plane for example.

- REQ 5: I2RS plane should be informed when a routing system resource is modified by a user outside the I2RS plane access. This is designated as "I2RS resource modified out of I2RS plane". This requirements is also described in section 7.6 of [[I-D.ietf-i2rs-architecture](#)] for the I2RS Client. This document extends the requirement to the I2RS plane, in case future evolution of the I2RS plane.
- REQ 6: I2RS plane should define an "I2RS plane overwrite policy". Such policy defines how an I2RS is able to update and overwrite a resource set by a user outside the I2RS plane. Such hierarchy has been described in [section 6.3](#) and 7.8 of [[I-D.ietf-i2rs-architecture](#)]
- REQ 7: I2RS AAA policies should be updated upon receipt of a "I2RS resource modified out of I2RS plane".

[4.](#) I2RS Authentication and Authorization Access Policy for routing system resources

This section details the I2RS Authentication and Authorization Access Policy (I2RS AAA) associated to the routing system resources. These policies only apply within the I2RS plane for I2RS users.

[4.1.](#) I2RS AAA architecture

Applications access to routing system resource via numerous intermediaries nodes. The application communicates with an I2RS Client. In some cases, the I2RS Client is only associated to a single application, but the I2RS Client may also act as a broker. The I2RS Client, then, communicates with the I2RS Agent that may eventually access the resource.

The I2RS Client broker approach provides scalability to the I2RS architecture as it avoids that each Application be registered to the I2RS Agent. Similarly, the I2RS AAA should be able to scale numerous applications.

- REQ 8: I2RS AAA should be performed through the whole I2RS plane. I2RS AAA should not be enforced by the I2RS Agent only within the routing element. Instead, the I2RS Client should enforce the I2RS Client AAA against applications and the I2RS Agent should enforce the I2RS Agent AAA against the I2RS Client. Note that I2RS Client AAA is not in the scope of the I2RS architecture [[I-D.ietf-i2rs-architecture](#)], which exclusively focuses on the I2RS Agent AAA.

This results in a layered and hierarchical I2RS AAA. An application will be able to access a routing system resource only if both the I2RS Client is granted access by the I2RS Agent AAA and the application is granted access by the I2RS Client AAA.

REQ 9: In case I2RS Client AAA or I2RS Agent AAA does not grant the access to a routing system resource, the Application should be able to define the I2RS AAA that generated this reject, as well as the reason. More specifically, the I2RS Agent may reject the request based on the I2RS Client privileges, and the I2RS Client should return a message to the application, indicating the I2RS Client does not have enough privileges. Similarly, if the I2RS Client does not grant the access to the application, the I2RS Client should also inform the application. Note that although multiple reject may occur, only the first reject should be mandatory.

In order to limit the number of access request that result in an error, each component should be able to retrieve the global I2RS AAA policies that applies to it. This subset of rules is designated as the "I2RS AAA component's subset policies". As they are subject to changes, a dynamic synchronization mechanism should be provided.

REQ 10: The I2RS Client should be able to request for its I2RS AAA Agent subset policies to the I2RS Agent AAA, so to limit forwarding unnecessary queries to the I2RS Agent.

REQ 11: The I2RS Client should be able to be notified when its I2RS AAA Agent subset policies have been updated.

Similarly, for the application

REQ 12: The Application may be able to request for its I2RS AAA Client subset policies, so to limit forwarding unnecessary queries to the I2RS Client.

REQ 13: The Application may be able to subscribe a service that provides notification when its I2RS AAA Client subset policies have been updated.

I2RS AAA should be appropriately be balanced between the I2RS Client and the I2RS Agent which can be illustrated by two extreme cases:

- 1) I2RS Clients are dedicated to a single Application: In this case, it is likely that I2RS AAA is enforced only by the I2RS Agent AAA, as the I2RS Client is likely to accept all access request of the application. However, it is recommended that even in this case, I2RS Client AAA is not based on an "Allow

anything from application" policy, but instead the I2RS Client specifies accesses that are enabled. In addition, the I2RS Client may sync its associated I2RS Agent AAA with the I2RS Agent to limit the number of refused access requests being sent to the I2RS Agent. The I2RS Client is expected to balance pro and cons between sync the I2RS Agent AAA and simply guessing the access request to the I2RS Agent.

- 2) A single I2RS Client acts as a broker for all Applications: In the case the I2RS Agent has a single I2RS Client. Such architecture results in I2RS Client with high privileges, as it sums the privileges of all applications. As end-to-end authentication is not provided between the Application and the I2RS Agent, if the I2RS Client becomes corrupted, it is possible for the malicious application escalates its privileges and make the I2RS Client perform some action on behalf of the application with more privileges. This would not have been possible with end-to-end authentication. In order to mitigate such attack, the I2RS Client that acts as a broker is expected to host application with an equivalent level of privileges.

REQ 14: The I2RS AAA should explicitly specify accesses that are granted. More specifically, anything not explicitly granted -- the default rule-- should be denied.

In order to keep the I2RS AAA architecture as distributed as possible,

REQ 15: I2RS Client should be distributed and act as brokers for applications that share roughly similar permissions. This avoids ending with over privileges I2RS Client compared to hosted applications and thus discourages applications to perform privilege escalation within an I2RS Client.

REQ 16: I2RS Agent should be avoided being granted over privileged regarding to their authorized I2RS Client. I2RS Agent should be shared by I2RS Client with roughly similar permissions.

[4.2.](#) I2RS Agent AAA

The I2RS Agent AAA restricts the routing system resource access to authorized components. Possible access policies may be none, read or write. The component represents the one originating the access request. The origin of the query is always an application. However, the I2RS Agent may not be able to authenticate the application. Instead, the I2RS Client may act as a broker. Similarly, multiple I2RS Agents may be used, and different access privilege may be provided depending on the I2RS Agent used. As a result, the origin

of the query may be represented in multiple ways, and each way may be associated to a specific AAA.

REQ 17: I2RS Agent AAA may use various ways to represent the origin of the access request of a routing system resource. However, representation of the origin should be based on information that can be authenticated. The I2RS Client, optionally the I2RS Agent in case of multiple I2RS Agents go into this category. On the hand, unless some additional means for authentication have been provided, the secondary identity used to tag the application as defined in [\[I-D.ietf-i2rs-architecture\]](#) should not be considered.

The I2RS Agent AAA may evolve over time as resource may also be updated outside the I2RS plane. Similarly, a given resource may be accessed by multiple I2RS users within the I2RS plane. Although this is considered as an error, depending on the I2RS Client that performed the update, the I2RS may accept or refuse to overwrite the routing system resource.

REQ 18: Each routing system resource updated by a I2RS Agent should keep track of the component that performed the last update.

REQ 19: the I2RS Agent should have a "I2RS Agent overwrite Policy" that indicates how the originating components can be prioritized. This requirements is also described in [section 7.6](#) of [\[I-D.ietf-i2rs-architecture\]](#)

[4.3.](#) I2RS Client AAA

The I2RS Client AAA works similarly to the I2RS Agent AAA. The main difference is that components are applications. As a result,

REQ 20: The I2RS Client should be able to authenticate its application.

In case, no authentication mechanisms have being provided between the I2RS Client and the application, then I2RS Client may not act as broker, and be instead dedicated to a single application. In this case, although this is not recommended, the I2RS AAA is only enforced by the I2RS Agent AAA. The I2RS Client may only sync and cache the I2RS Agent AAA associated to its application, in order to limit the access requests to the I2RS Agent. The I2RS Client is expected to balance pro and cons between synchronization of the I2RS Agent AAA, or simply sending the request to the I2RS Agent.

4.4. I2RS AAA Security Domain

I2RS Client AAA and I2RS Agent AAA are respectively enforced within the I2RS Client and the I2RS Agent. I2RS AAA enforcement should not remain local, and should be also enforced through the network communications. More specifically:

- 1) Communication should remain available at any time, and it should be robust to potential attacks, or misbehaviors.
- 2) Components' operation requests should be guaranteed to have been properly authorized by the I2RS AAA policies.

4.4.1. Available I2RS Communication Channel

Communication is considered available if and only if all components are available as well as the communication channel itself. In order to maintain it available here are the considered aspects:

- 1) Make communication robust to DoS by design
- 2) Provide active ways to mitigate an DoS attack
- 3) Limit damages when a DoS event occurs

Protocols used to communicate between components should not provide means that would result in a component's resource exhaustion.

At the transport layer, when possible, protocols that do not implement any mechanisms to check the origin reachability should be avoided (like UDP). Instead, if possible, protocols like TCP or SCTP with origin reachability verification should be preferred. Anti DoS mechanisms should also be considered at other layers including the application layer. More specifically, it should be avoided to perform actions that generate heavy computation on a component. At least the component should be able to post-pone and re-schedule the action. Similarly, DoS by amplification should be avoided, and special attention should be given to small access request that generate massive network traffic without any control. An example of asymmetric dialogue could be the subscription of information streams like prefix announcement from OSPF. In addition, some service may also provide the ability to redirect these streams to a third party. In the case of information stream, registration by an I2RS Client may provide the possibility to redirect the stream on a shared directory, so it can be accessed by multiple I2RS Clients, while not flooding the network. In this case, special attention should be provided so the shared directory can agree based on its available resources the

service subscription by the I2RS Client. Otherwise, the shared directory may become overloaded.

REQ 21: Communication protocols should be robust against DoS attacks. DoS should be considered at multiple layers, in the design of the communication protocol. Engaged resources should be agreed by the component using these resources.

Components should be able to control the computing resource they allocate to each other components, or each actions. Based on available resource, requests should be differed, or returned an error.

REQ 22: I2RS Client and I2RS Agent should implement mechanisms to mitigate DoS attacks.

One alternative way to mitigate a DoS attack or event is to limit the damages when resource exhaustion happens. This can be done by appropriately group or ungroup applications. For example, critical applications may not share their I2RS Client with multiple other Applications. This limits the probability of I2RS Client failure for the critical application. Similarly, I2RS Agent may also be selective regarding their I2RS Client as well as to the scope of their routing system resources. In fact some, some I2RS Client may be less trusted than others and some routing system resource access may be more sensitive than the others. Note that trust of an I2RS Client is orthogonal to authentication and rather involves, for example, the quality of the hosted Applications.

REQ 23: Application, I2RS Client and I2RS Agent should be distributed among the I2RS Plane to minimize the impact of a failure.

Even though this should be considered, it does not address the high availability issue. In order to reduce the impact of a single I2RS Client failure, remote applications may load balance their access request against multiple I2RS Clients. Non remote I2RS Client or I2RS Agent are bound the system hosting the application or to the routing element. This makes high availability be provided by the system, and thus implementation dependent.

REQ 24: I2RS Client should provide resilient and high availability for the hosted applications.

4.4.2. Trusted I2RS Communications Channel

This section addresses the authorization and trust of the Communication Channel. The main purpose of this section is to provide guidance to avoid identity usurpation. More specifically,

resource access request should only be issued and responded by the expected components and concern the expected resource only.

REQ 25: Communications between the different components should be mutually authenticated.

REQ 26: Communications between components should be protected against replay attacks.

Within the I2RS AAA Security Domain, information exchanged between the I2RS Client and the I2RS Agent or the application and the I2RS Client may leak information about the application goal, as well as its internal logic. As a result, it is recommended to isolate components communications.

REQ 27: Communications between components should avoid leaking information about internal logic, and thus, it is recommended to encrypt them.

When an incident occurs, one should be able to understand the reasons in order to prevent it to happen again.

REQ 28: Log and monitoring facilities should be provided and made available for forensic investigation.

5. I2RS Application Isolation

A key aspect of the I2RS architecture is the network oriented application. As these application are supposed to be independent, controlled by independent and various tenants. In addition to independent logic, these applications may be malicious. Then, these applications introduce also programmability which results in fast network settings.

The I2RS architecture should remain robust to these applications and make sure an application does not impact the other applications. This section discusses both security aspects related to programmability as well as application isolation in the I2RS architecture.

5.1. Robustness toward programmability

I2RS provides a programmatic interface in and out of the Internet routing system. This feature, in addition to the global network view provided by the centralized architecture comes with a few advantages in term of security.

The use of automation reduces configuration errors. In addition, this interface enables fast network reconfiguration. Agility provides a key advantage in term of deployment as side effect configuration may be easily addressed. Finally, it also provides facilities to monitor and mitigate an attack when the network is under attack.

On the other hand programmability also comes with a few drawbacks. First, applications can belong to multiple tenants with different objectives. This absence of coordination may result in unstable routing configurations such as oscillations between network configurations, and creation of loops for example. A typical example would be an application monitoring a state and changing its state. If another application performs the reverse operation, the routing system may become unstable. Data and application isolation is expected to prevent such situations to happen, however, to guarantee the network stability, constant monitoring and error detection are recommended to be activated.

REQ 29: I2RS should monitor constantly parts of the system for which clients have requested notification. It should also be able to detect components that lead the routing system in an unstable state.

5.2. Application Isolation

5.2.1. DoS

Requirements for robustness to Dos Attacks have been addressed in the Communication channel section [[I-D.ietf-i2rs-architecture](#)].

The I2RS interface is used by application to interact with the routing states. As the I2RS Agent is shared between multiple applications, one application can prevent an application by performing DoS or DDoS attacks on the I2RS Agent or on the network. DoS attack targeting the I2RS Agent would consist in providing requests that keep the I2RS Agent busy for a long time. This may involve heavy computation by the I2RS Agent for example to blocking operations like disk access. In addition, DoS attacks targeting the network may use specific commands like monitoring stream over the network. Then, DoS attack may be also targeting the application directly by performing reflection attacks. Such an attack could be performed by indicating the target application as the target for some information like the listing of the RIB. Reflection may be performed at various levels and can be based on the use of UDP or at the service level like redirection of information to a specific repository.

- REQ 30: In order to prevent DoS, it is recommended the I2RS Agent controls the resources allocated to each I2RS Clients. I2RS Client that acts as broker may not be protected as efficiently against these attacks unless they perform resource controls themselves of their hosted applications.
- REQ 31: I2RS Agent does not make response redirection possible unless the redirection is previously validated and agreed by the destination.
- REQ 32: avoid the use of underlying protocols that are not robust to reflection attacks.

5.2.2. Application Control

Requirements for Application Control have been addressed in the I2RS plane isolation as well as in the trusted Communication Channel sections.

Applications use the I2RS interface in order to update the routing system. These updates may be driven by behavior on the forwarding plane or any external behaviors. In this case, correlating observation to the I2RS traffic may enable to derive the application logic. Once the application logic has been derived, a malicious application may generate traffic or any event in the network in order to activate the alternate application.

- REQ 33: Application logic should remain opaque to external listeners. Application logic may be partly hidden by encrypting the communication between the I2RS Client and the I2RS Agent. Additional ways to obfuscate the communications may involve sending random messages of various sizes. Such strategies have to be balanced with network load. Note that I2RS Client broker are more likely to hide the application logic compared to I2RS Client associated to a single application.

6. Privacy Considerations

7. IANA Considerations

8. Acknowledgments

We would like to thanks Russ White for its review and editorial contributions.

9. Informative References

[I-D.ietf-i2rs-architecture]

Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", [draft-ietf-i2rs-architecture-09](#) (work in progress), March 2015.

Authors' Addresses

Daniel Migault (editor)
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson

Email: Joel.Halpern@ericsson.com

