I2RS WG
Internet-Draft
Intended status: Informational
Expires: April 4, 2016

D. Migault, Ed.
J. Halpern
Ericsson
S. Hares
Huawei

October 2, 2015

I2RS Environment Security Requirements
draft-mglt-i2rs-security-environment-reqs-01

Abstract

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. As a result, the requirements provided in this document are intended to provide good security practise so I2RS can be securely deployed and operated.

These security requirements are designated as environment security requirements as opposed to the protocol security requirements described in [I-D.ietf-i2rs-protocol-security-requirements]. The reason to have separate document is that protocol security requirements are intended to help the design of the I2RS protocol whether the environment requirements are rather intended for deployment or implementations.

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 April 4, 2016.

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 4
3. I2RS Plane Isolation 4
3.1. I2RS plane and management plane 4
3.2. I2RS plane and forwarding plane 5
3.3. I2RS plane and Control plane 6
3.4. Recommendations 6
4. I2RS Access Control for routing system resources 8
4.1. I2RS Access Control architecture 8
4.2. I2RS Agent Access Control policies 13
4.3. I2RS Client Access Control policies 14
4.4. Application and Access Control policies 15
5. I2RS Application Isolation 16
5.1. Robustness toward programmability 16
5.2. Application Isolation 17
5.2.1. DoS 17
5.2.2. Application Control 17
6. Security Considerations 18
7. Privacy Considerations 18
8. IANA Considerations 18
9. Acknowledgments 18
10. References 18
10.1. Normative References 18
10.2. Informative References 18
Authors' Addresses 19

1. Introduction

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. As a result, the requirements provided

in this document are intended to provide good security practise so I2RS can be securely deployed and operated.

These security requirements are designated as environment security requirements as opposed to the protocol security requirements described in [I-D.ietf-i2rs-protocol-security-requirements]. The reason to have separate document is that protocol security requirements are intended to help the design of the I2RS protocol whether the environment requirements are rather intended for deployment or implementations.

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 Access Control policies 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]. The document provides a list of environment security requirements. Motivations are placed before the requirements are announced.

2. Terminology and Acronyms

- Environment Security Requirements :

- I2RS plane : The environment the I2RS process is running on. It includes the Applications, the I2RS Client and the I2RS Agent.

- I2RS user : The user of the I2RS client software or system.

- I2RS Access Control policies: policies controlling access of the routing resources by Applications. These policies are divided into policies applied by the I2RS Client regarding Applications and policies applied by the I2RS Agent regarding I2RS Clients.

- I2RS Client Access Control policies : The Access Control policies processed by the I2RS Client.

- I2RS Agent Access Control policies : The Access Control policies processed by the I2RS Agent.

3. I2RS Plane Isolation

Isolating the I2RS plane from other network plane, such as the control plane, is foundational to the security of the I2RS environment. 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.

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 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.

The I2RS plane and the management plane both interact with several common elements on forwarding and packet processing devices. [I-D.ietf-i2rs-architecture] describes several of these interaction points such as 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, one could either control the access to these resources with northbound APIs for example. Northbound APIs are provided to limit the scope of the applications toward the routing resources. In our case, the northbound API may be provided for the I2RS applications by the I2RS Client as well as to the management plane. In case conflicting overlaps cannot be avoided, and routing resource can be accessed by both the management plane and the I2RS plane, then, they 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/issues, 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 hosted on I2RS Client belongs to the I2RS plane. These Applications are hard to remain constrained into the I2RS plane, or even to limit their scope within the I2RS 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 interactions 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 communication channel. Various level of isolation can be considered. 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, for example by using TLS or IPsec, 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 physical or logical interface. As previously, mentioned a dedicated physical interface may contribute to a higher

isolation, however logical isolation be also be considered for example by using a dedicated IP address or a dedicated port.

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 3: 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 Access Control policies 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 4: I2RS plane should be informed when a routing system resource is modified by a user outside the I2RS plane access. The notification is not expected to flood the I2RS plane. Instead, notification is expected to be provided to the I2RS components interacting, configuring or monitoring the routing system resource. The notification is at least provided by the I2RS Agent to the various I2RS Client, but additional mechanisms might eventually be required so I2RS Client can relay the notification to the I2RS applications. 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 5: 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]

4. I2RS Access Control for routing system resources

This section provides recommendations on how I2RS Access Control policies associated to the routing system resources. These policies only apply within the I2RS plane. More especially, the policies are associated to the Applications, the I2RS Clients and the I2RS Agents, with their associated identity and roles.

Note that the deployment of Applications, I2RS Client and I2RS Agent in a closed environment, should not be considered by default as a secure environment. Even for closed environment access control policies should be carefully defined to be able to, in the future to carefully extend the I2RS plane to remote Applications or remote I2RS Clients. As a result, this section always consider the case Applications and I2RS Client can be located locally, in a closed environment or distributed over open networks.

Although [I-D.ietf-i2rs-protocol-security-requirements] provides security requirements of the transport and protocol between the I2RS Client and the I2RS Agent, this section is mostly focused on access control.

4.1. I2RS Access Control 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 Access Control should be able to scale numerous applications.

REQ 6: I2RS Access Control should be performed through the whole I2RS plane. It should not be enforced by the I2RS Agent only within the routing element. Instead, the I2RS Client should enforce the I2RS Client Access Control against Applications and the I2RS Agent should enforce the I2RS Agent Access Control against the I2RS Clients. Note that I2RS Client

Access Control is not in the scope of the I2RS architecture [I-D.ietf-i2rs-architecture], which exclusively focuses on the I2RS Agent Access Control.

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

REQ 7: When an access request to a routing resource is refused by one party (the I2RS Client or the I2RS Agent), the initiator of the request (e.g the Application) as well as all intermediaries should indicate the reason the access has not been granted as well as the entity that has rejected the request.

REQ 8: In order to provide coherent Access Control policies enforced by multiple parties (e.g. the I2RS Client or the I2RS Agent), theses parties should trust each others, and communication between them should also be trusted, - that is should not introduce additional vector of attacks.

In case the I2RS Client Access Control or the I2RS Agent Access Control does not grant access to a routing system resource, the Application should be able to determine whether its request has been rejected by the I2RS Client or the I2RS Agent as well as the reason that caused the reject. More specifically, the I2RS Agent may reject the request because, for example, the I2RS Client is not an authorized I2RS Client, or because the I2RS Client does not not have enough privileges. The I2RS Client should be notified of the reason that caused the reject by the I2RS Agent, and The I2RS Client should return a message to the Application, indicating the I2RS Client is not authorized or 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. The error message returned should be for example: "Read failure: you do not have the read permission", "Write failure: you do not have write permission" or "Write failure: resource accessed by someone else". This requirement has been written in a generic manner as it concerns various interactions: interactions between the application and the I2RS Client, interactions between the I2RS Client and the I2RS Agent. In the latest case, the requirement is part of the protocol security requirements addressed by [I-D.ietf-i2rs-protocol-security-requirements].

Although [I-D.ietf-i2rs-protocol-security-requirements] is focused on transport security requirements between the I2RS Client and the I2RS

Agent, the similar requirements may apply between the Application and the I2RS Client for a remote Application.

REQ 9: I2RS Client or I2RS Agent SHOULD also be able to refuse a communication with an Application or an I2RS Client when the communication channel does not fulfill enough security requirements. For example, the it should be able to reject messages over a communication channel that can be easily hijacked, like a clear text UDP channel.

In order to limit the number of access request that result in an error, each Application or I2RS Client may be able to retrieve the I2RS Access Control policies that applies to it. This subset of rules is designated as the "Individual I2RS Access Control policies". As these policies are subject to changes, a dynamic synchronization mechanism should be provided. However, such mechanism may be implemented with different level of completeness and dynamicity of the Individual I2RS Access Control policies. Caching requests that have been rejected may be one such variant. It remains relatively easy to implement and may avoid the complete disclosure of the Access Control policies of the I2RS Agent. In fact the relative disclosure of Access Control policies may leak confidential information in case of misconfiguration and should be balanced with the level of trust of the I2RS Client and the necessity of distributing the enforcement of the Access Control policies.

REQ 10: The I2RS Client may be able to request for its I2RS Access Control subset policies to the I2RS Agent or cache requests that have been rejected by the I2RS Agent to limit forwarding unnecessary queries to the I2RS Agent.

REQ 11: The I2RS Client may be able to be notified when its I2RS Access Control subset policies have been updated by the I2RS Agent.

Similarly, for the Applications

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

REQ 13: The Applications may be able to subscribe a service that provides notification when its I2RS Access Control subset policies have been updated.

I2RS Access Control should be appropriately be balanced between the I2RS Client and the I2RS Agent. I2RS Access Control should not

solely rely only on the I2RS Client or the I2RS Agent as illustrated below:

- 1) I2RS Clients are dedicated to a single Application: In this case, it is likely that I2RS Access Control is enforced only by the I2RS Agent, 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 Access Control 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 Access Control policies 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 its access control policies with the I2RS Agent 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 Access Control should explicitly specify accesses that are granted. More specifically, anything not explicitly granted -- the default rule-- should be denied.

In addition to distribute the I2RS Access Control policies between I2RS Clients and I2RS Agents, I2RS Access Control policies can also be distributed within a set of I2RS Clients or a set of I2RS Agents.

REQ 15: I2RS Clients 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 Agents should be avoided being granted over privileges regarding to their authorized I2RS Client. I2RS Agent should be shared by I2RS Client with roughly similar permissions. More explicitly, an I2RS Agent shared between I2RS Clients

that are only provided read access to the routing system resources does not need to perform any write access, and so should not be provided these accesses. Suppose an I2RS Client requires write access to the resources. It is not recommended to grant the I2RS Agent the write access in order to satisfy a unique I2RS Client. Instead, the I2RS Client that requires write access should be connected to a I2RS Agent that is already shared by I2RS Client that requires a write access.

Access Control policies enforcement should be monitored in order to detect violation of the policies or detect an attack. Access Control policies enforcement may not be performed by the I2RS Client or the I2RS Agent as violation may require a more global view of the I2RS Access Control policies. As a result, consistency check and mitigation may instead be performed by the management plane. However, I2RS Clients and I2RS Agents play a central role.

REQ 17: I2RS Client and I2RS Agent should be able to log the various transaction they perform, as well as suspicious activities. These logs should be collected regularly and analyzed by functions that may be out of the I2RS plane.

Access Control policies should be implemented so that they remain manageable in short and longer term. This means the way they are managed today should be address future deployment and use of I2RS.

REQ 18: Access Control should be managed in an automated way, that is granting or revoking an Application should not involve manual configuration over the I2RS plane - like all the I2RS Clients.

REQ 19: Access Control should be scalable when the number of Application grows as well as when the number of I2RS Client increases. A typical implementation of a local I2RS Client Access Control policies may result in creating manually a system user associated to each Application. Such an approach is likely not to scale when the number of Applications increases or the number of I2RS Client increases.

REQ 20: Access Control should be dynamically managed and easy to be updated. Although the number of I2RS Clients is expected to be lower than the number of Application, as I2RS Agent provide access to the routing resource, it is of primary importance that an access can be granted or revoke in an efficient way.

REQ 21: I2RS Clients and I2RS Agents should be uniquely identified in the network to enable centralized management of the I2RS Access Control policies.

4.2. I2RS Agent Access Control policies

The I2RS Agent Access Control restricts the routing system resource access to authorized identities - possible access policies may be none, read or write. The initiator of an access request to a routing resource is always an Application. However, it remains challenging for the I2RS Agent to establish its access control policies based on the application that initiates the request. First, when an I2RS Client acts as a broker, the I2RS Agent may not be able to authenticate the Application. In that sense, the I2RS Agent relies on the capability of the I2RS Client to authenticate the Applications and apply the appropriated I2RS Client Access Control. Then, an I2RS Agent may not uniquely identify a piece of software implementing an I2RS Client. In fact, an I2RS Client may be provided multiple identities which can be associated to different roles or privileges. The I2RS Client is left responsible for using them appropriately according to the Application. Finally, each I2RS Client may contact various I2RS Agent with different privileges and Access Control policies.

This section provides recommendations on the I2RS Agent Access Control policies to keep I2RS Access Control coherent within the I2RS plane.

REQ 22: I2RS Agent Access Control policies should be primarily based on the I2RS Clients as described in [I-D.ietf-i2rs-architecture].

REQ 23: I2RS Agent Access Control policies may be based on the Application. In this case the identity of the Application MUST be authenticated by the I2RS Agent, and the secondary identity used to tag the application as defined in [I-D.ietf-i2rs-architecture] should be considered cautiously. The tag may be used associated only to an authenticated I2RS Client that is known to authenticate its Application.

The I2RS Agent Access Control policies 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 24: The I2RS Agent should know which identity (most likely system user) performed the latest update of the routing resource. This is true for an identity inside and outside the I2RS plane, so the I2RS Agent can appropriately perform an update according to the priorities associated to the requesting identity and the identity that last updated the resource. On an environment perspective, the I2RS Agent MUST be aware when the resource has been modified outside the I2RS plane, as well as its priority associated towards the I2RS plane. Similar requirements exist for identities within the I2RS plane, but belongs to the protocol security requirements.

REQ 25: the I2RS Agent should have a "I2RS Agent overwrite Policy" that indicates how identities can be prioritized. This requirements is also described in section 7.6 of [I-D.ietf-i2rs-architecture]. Similar requirements exist for components within the I2RS plane, but belongs to the protocol security requirements.

4.3. I2RS Client Access Control policies

The I2RS Client Access Control policies are responsible for authenticating the application managing the privileges for the applications, and enforcing access control to resources by the applications. As a result,

REQ 26: I2RS Client should authenticate its applications. If the I2RS Client acts as a broker and supports multiple Applications, it should authenticate each of them. Authentication of the application may used GSSAPI, Secure RPC mechanisms.

REQ 27: I2RS Client should define Access Control policies associated to each applications. An access to a routing resource by an Application should not be forwarded by the I2RS Client based on the I2RS Agent Access Control policies. The I2RS Client should first check whether the Application has sufficient privileges, and if so send an access request to the I2RS Agent. When an I2RS Client has multiple identities that are associated with different privileges. The I2RS Client Access Control policies should specify the associated I2RS Client's identities, especially, when the I2RS Agent Access Control policies are changed for a given I2RS Client's identity.

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. By doing so, application authentication may rely on the I2RS authentication

mechanisms between the I2RS Client and the I2RS Agent. On the other hand, although this is not recommended, the I2RS Access Control policies is only enforced by the I2RS Agent.

4.4. Application and Access Control policies

Application does not enforce access control policies. Instead these are enforced by the I2RS Clients and the I2RS Agents. This section provides recommendations for Applications in order to ease I2RS Access Control by the I2RS Client and the I2RS Agent.

As multiple ways may be used for an Application to communicate with its associated I2RS Client, it is not expected that all Applications use the same conventional identifier format across the I2RS plane. However, if all Applications are running on a dedicated system sharing an I2RS Client, it is expected each Application may uniquely identified, for example using different system users.

REQ 28: Applications SHOULD be uniquely identified by their associated I2RS Clients

The I2RS Client provides access to resource on its behalf and this access should only be granted for trusted applications, or Applications with an similar level of trust. On the other hand, this does not prevent an I2RS Client to host a large number of Applications. Similarly, an Application may also require to access multiple I2RS Clients depending on the resource to be accessed. As I2RS Client are restricted for a subset of Applications,

REQ 29: Each Application SHOULD be associated to a restricted number of I2RS Client

REQ 30: Application SHOULD be provided means and methods to contact their associated I2RS Client. If the I2RS Client belongs to the Application (as a module or a library for example), or when the Application runs into a dedicated system (like a container) with a I2RS Client, it is obvious which I2RS Client the Application is associated to. On the other hand, Applications may also remotely access the I2RS Client. In this case, the Application is expected to be provided some means to be able to retrieve the necessary information to contact its associated I2RS Client. The IP address may not be appropriated in case renumbering occurs within the network or in case the traffic from Applications should be shared between multiple instances of a given I2RS Client. In this case a FQDN may be preferred.

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 31: The I2RS Agents should monitor constantly parts of the system for which I2RS Clients or Applications have provided requests. It should also be able to detect I2RS Clients or Applications that lead the routing system in an unstable state. Monitoring consists at least in logging events and eventually provide notifications or alerts to the management plane in case, something has been detected. The management

plane is in charge of collecting the logs, the notifications and eventually to consider the appropriated actions. A typical action may be the update of I2RS Access Control policies for example or re-configuring routing elements.

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 32: 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 33: I2RS Agent does not make response redirection possible unless the redirection is previously validated and agreed by the destination.

REQ 34: 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 35: 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. Security Considerations

The whole document is about security.

7. Privacy Considerations

8. IANA Considerations

9. Acknowledgments

A number of people provided a significant amount of helping comments and reviews. Among them the authors would like to thank Russ White, Russ Housley, Thomas Nadeau, Juergen Schoenwaelder, Jeffrey Haas, Alia Atlas, Linda Dunbar

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

10.2. 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.

[I-D.ietf-i2rs-protocol-security-requirements]
Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-ietf-i2rs-protocol-security-requirements-01 (work in progress), September 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

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com