I2RS WG
Internet-Draft
Intended status: Informational
Expires: May 19, 2017

D. Migault
J. Halpern
Ericsson
S. Hares
Huawei

November 15, 2016

I2RS Environment Security Requirements
draft-ietf-i2rs-security-environment-reqs-02

Abstract

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. The I2RS protocol security requirements set the security for the communication between I2RS client and agent while the security environment requirements are rather intended for deployment or implementations features independent of the I2RS protocol. The environmental security requirements described in this document provide the good security practices to be used with the I2RS protocol so that I2RS protocol implementations can be securely deployed and operated.

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 May 19, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document provides environment security requirements for the I2RS architecture. Environment security requirements are independent of the protocol used for I2RS. The I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements] set the security for the communication between I2RS client and agent while the security environment requirements are rather intended for deployment or implementations features independent of the I2RS protocol. The environmental security requirements described in this document provide the good security practices to be used with the I2RS protocol so that I2RS protocol implementations can be securely deployed and operate. These environment security address the security considerations for I2RS protocol environment considered in the I2RS Architecture [RFC7921] in order to provide a stable and secure environment in which the dynamic programmatic interface to the routing system (I2RS) should operates.

Even though the I2RS protocol is mostly concerned with the interface between the I2RS client and the I2RS agent, the environmental security requirements must consider the entire I2RS architecture and specify where security functions may be hosted and what criteria should be met in order to address any new attack vectors exposed by deploying this architecture. Environment security for I2RS has to be considered the complete I2RS architecture and not only on the protocol interface.

This document is structured as follows:

- Section 2 describes the terminology used in this document,
- Section 3 describes how the I2RS plane can be contained or isolated from existing management plane, control plane and forwarding plane.
- the subsequent sections of the document focuses on the security within the I2RS plane including:
  - 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 includes providing a robust communication system between the components.
  - Section 5 details how I2RS keeps applications isolated from another and without affecting the I2RS components. applications may be independent, with different scopes, owned

by different tenants. In addition, the applications may modify the routing system in an automatic way.

Motivations are placed before the requirements are given.

The reader is expected to be familiar with the I2RS problem statement [RFC7920], I2RS architecture, [RFC7921], traceability requirements [RFC7922], I2RS Pub/Sub requirements [RFC7923], I2RS ephemeral state requirements [I-D.ietf-i2rs-ephemeral-state], I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements].

2. Terminology and Acronyms

- Environment Security Requirements : Security requirements specifying how the environment a protocol operates in needs to be secure. These requirements do not specify the protocol 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.

2.1. Requirements notation

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 [RFC2119].

3. I2RS Plane Isolation

Isolating the I2RS plane from other network planes (the management, forwarding plane, and control planes) is fundamental to the security of the I2RS environment. Clearly differentiating I2RS components from the rest of the network device does the following:

1. protects the I2RS components from vulnerabilities in other parts of the network, and

2. protects 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. However, the I2RS plane cannot be considered as completely isolated from other planes so the interactions between the I2RS plane and other planes should be identified and controlled. The following is a brief description of how the I2RS plane positions itself in regard to the other planes.

3.1. I2RS Plane and Management plane

The purpose of the I2RS plane is to provide a standard programmatic interface of the routing system resources to network oriented applications. The control plane and forwarding planes are related to routing protocols, and I2RS is positioned on top of the control plane and forwarding plane. The management plane is usually vendor specific and provides a broader control over the networking equipment such as system service. Given the management plane’s 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. [RFC7921] describes several of these interaction points such as the local configuration, the static system state, routing, and signaling. A routing resource may be accessed by different means (APIs, applications) and different planes which creates potential overlaps. Southbound APIs are often provided to limit the scope of the management plane’s and the I2RS plane’s interaction with the routing resources (as figure 1 shows). If there are conflicts in these overlapping southbound APIs, the conflicts should be resolved in a deterministic way.

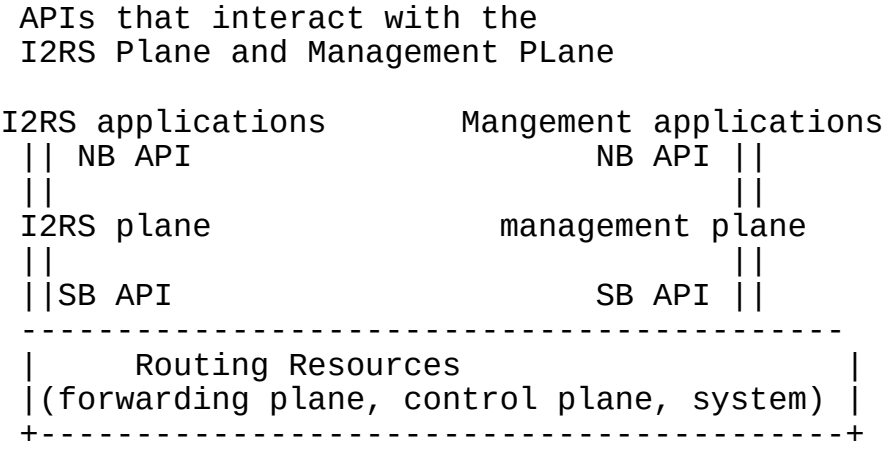


Figure 1 - North Bound (NB) APIs and South Bound (SB) APIs

The I2RS applications may be provided with a northbound API by the I2RS client (as figure 1 shows). Similarly, the management plane may provide a northbound API to management application. The northbound API from the management plane to the client application and the northbound API from the I2RS plane to I2RS applications may overlap. In case that these overlapping APIs between the have conflicts (e.g. both want to access the same routing resource), the the conflicts should be resolved in a deterministic way.

To resolve conflicts in a northbound APIs, the deterministic resolution should have clear rules about which data plane with a system takes precedence (wins during a conflict for resources). This is important to prevent attacks that attempt to drive the two systems into deadlock situation over which system has precedence (wins)

In the interactions between the I2RS plane and the applications, and the management plane and the application is it important to prevent the following things:

- the I2RS system "infecting" the management system with bad information,
- the management system "infecting" the I2RS system with bad information directly,
- the management system indirectly "infecting" the I2RS system by propagating improproper information from one system to another via I2RS.

Prevention:

The primary protection in this space is going to need to be validation rules on the data being sent/receive, notification of changes that the I2RS agent sends the client, and other access protections.

In this circumstance, we define "infecting" as inteferring with and leading into a incoherent state. The I2RS plane may update a routing resource configured by the management plane, or the reverse (the management plane updates a resource the I2RS plane has configured). Indirect "infecting", we define as as changes made by the I2RS plane that cause changes in routing protocols which add state unexpected by the management plane or the reverse (the mangement plane adding changes in routing protocols the I2RS plane does not expect).

3.2. I2RS Plane and Forwarding Plane

Applications hosted by the I2RS client belong to the I2RS plane. It is difficult to constrained these applications to the I2RS plane, or even to limit their scope within the I2RS plane. Applications using I2RS may also interact with components outside the I2RS plane. For example an application may use an I2RS client to configure the network or monitored events via an I2RS agent on a single machine, or multiple I2RS agents each on a single machine.

Applications may also communicate with multiple I2RS clients in order to have a broader view of the current and potential states of the network and the I2RS plane itself. These varied remote communication relationships between applications using the I2RS protocol to change the forwarding plane make it possible for an individual application to be an effective attack vector against the operation of the network, a router’s I2RS plane, the forwarding plane of the routing system, and other planes (management and control planes).

Prevention measures:

Systems should consider the following prevention errors:

application validation - There is little the I2RS plane can do to validate applications with which it interacts. The I2RS client passes the I2RS agent an opaque identifier for the application so that an application’s actions can be traced back to the application.

Validation against common misconfigurations or errors - One way of securing the interfaces between application, the I2RS plane, and the forwarding plane is to limit the information accepted and to limit the rate information is accepted between these three

software planes. Another method is to performing rudimentary checks on the results of any updates to the forwarding plane.

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. I2RS client configures, monitors or receives events via the I2RS agent's interaction with the routing system including the process that handling the control plane signaling protocols (BGP, ISIS, OSPF, etc.), route information databases (RIBs), and interface databases. In some situation to manage an network outage or to control traffic, the I2RS protocol may modify information in the route database or the configuration of routing process. While this is not a part of normal processing, such action that allows the network operator to bypass temporary outages or DoS attacks.

This capability to modify the routing process information carries with it the risk that the I2RS agent may alter the normal properties of the routing protocols which provide normal loop free routing in the control plane. The information configured by the I2RS agent into routing process or RIBs could cause problems, or state which is inserted or deleted from routing processes (control traffic, counters, etc.) could cause the control plan to fail to converge or fail).

Prevention measures:

The I2RS system can provide checks that during and after the problem has been resolved that loop free routing is preserved. These checks should check the preservation of all facets of routing including the following: routing with loop free alternates, tunnelled interfaces, virtual overlays, and other such constructions.

3.4. Requirements

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

SEC-ENV-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 (e.g. using vLAN). In a virtual environment that shares a common infrastructure, encryption may also be used as a way to enforce isolation. Encryption

can be added by using a secure transport required by the I2RS protocol security [I-D.ietf-i2rs-protocol-security-requirements], and sending the non-confidential I2RS data (designed for a non-secure transport) over a secure transport.

SEC-ENV-REQ 2: The interface used by the routing element to receive I2RS transactions via the I2RS protocol (e.g. the IP address) should be a dedicated physical or logical interface. As previously, mentioned a dedicated physical interface may contribute to a higher isolation. Isolation may also be achieved by using a dedicated IP address or a dedicated port.

SEC-ENV-REQ 3: An I2RS agent should have permissions separate from any other entity (for example any internal system management processes or CLI processes).

Explanation:

When the I2RS agent performs an action on a routing element, the action is performed via process(es) associated to a routing process.

For example, in a typical UNIX system, the user is designated with a user id (uid) and belongs to groups designated by group ids (gid). If such a user id (uid) and group id (gid) is the identifier for the routing processes peforming routing tasks in the control plane, then the I2RS Agent process would communicate with these routing processes. It is important that the I2RS agent has its own unique identifier and the routing processes have their own identifier so that access control can unique filter data between the processes.

SEC-ENV-REQ 4: I2RS plane should be informed when a routing system resource is modified by a user outside the I2RS plane access. Notifications from the control plane SHOULD not to flood the I2RS plane, and rate limiting (or summarization) is expected to be applied. These routing system notification MAY translated to the appropriate I2RS agent notifications, and passed to various I2RS clients via notification relays. (This requirements is also described in section 7.6 of [RFC7921] for the I2RS client, and this section extends it to the entire I2RS plane (I2RS agent, client, and application).

Explanation:

I2RS resource may be shared with the management plane and the control plane. The I2RS routing system resource management is limited to the I2RS plane. As such, update of I2RS routing system outside of the I2RS plane may remain unnoticed unless and until 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. Further an I2RS client is more likely to update an I2RS resources that has been updated by itself, then by the management plane.

SEC-ENV-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 [RFC7921]

Explanation:

A key part of the I2RS architecture is notification regarding routing system changes across the I2RS plane (I2RS client to/from I2RS agent). The security environment requirements above (SEC-ENV-REQ-03 to SEC-ENV-REQ-05) provide the assurance that the I2RS plane and the routing systems the I2RS plane attaches to remains untouched by the other planes or the I2RS plane is notificed of such changes. Section 6.3 of [RFC7921] describes how the I2RS agent within the I2RS plane interacts with forwarding plane’s local configuration, and provides the example of an overwrite policy between the I2RS plane and local configuration (instantiated in 2 Policy Knobs) that operators may wish to set. The prompt notification of any outside overwrite is key to the architecture and proper interworking of the I2RS Plane.

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, I2RS clients and I2RS agents, with their associated identity and roles.

The I2RS deployment of I2RS applications, I2RS clients, and I2RS agents can be located locally in a closed environment or distributed over open networks. The normal case for routing system management is over an open environment. Even in a 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.

[I-D.ietf-i2rs-protocol-security-requirements] defines the security requirements of the I2RS protocol between the I2RS client and the I2RS agent over a secure transport. This section focuses on I2RS access control architecture (section 5.1), access control policies of the I2RS agent (section 5.2), the I2RS client (section 5.3), and the application (section 5.4).

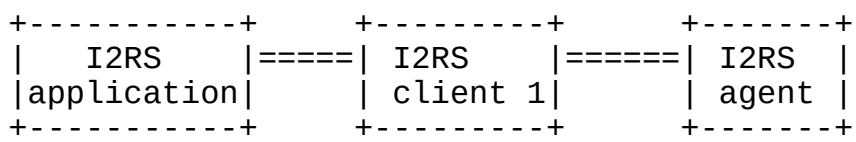
4.1. I2RS Access Control Architecture

Overview:

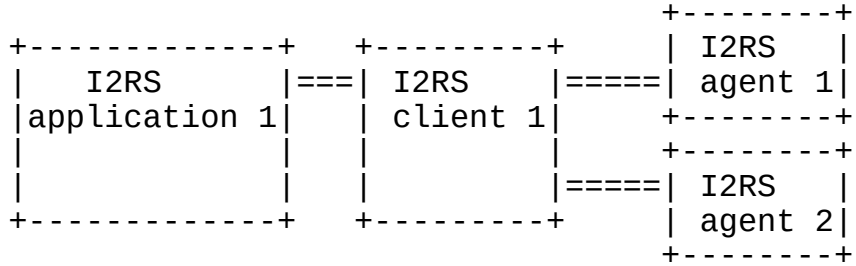
Applications access to routing system resource via numerous intermediaries nodes. The application communicates with an I2RS client. In some cases, the I2RS client is only associated to a single application attached to one or more agents (case a and case b in figure 2 below). In other cases, the I2RS client may be connected to two applications (case c in figure 2 below), or the I2RS may act as a broker (agent/client device shown in case d in the figure 2 below). 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.

The goal of the security environment requirements in this section are to control the interactions between the applications and the I2RS client, and the interactions between the I2RS client and the I2RS agent. The key challenge is that the I2RS architecture puts the I2RS Client in control of the stream of communication (application to I2RS client and I2RS client to the I2RS agent). The I2RS agent must trust the I2RS client’s actions without having an ability to verify the I2RS client’s actions.

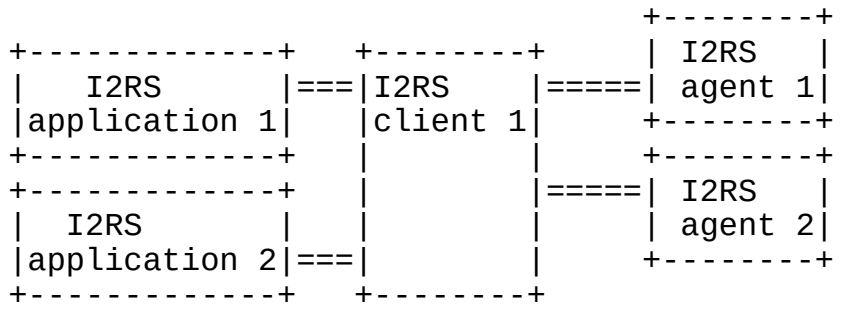
a) I2RS application-client pair talking to one I2RS agent



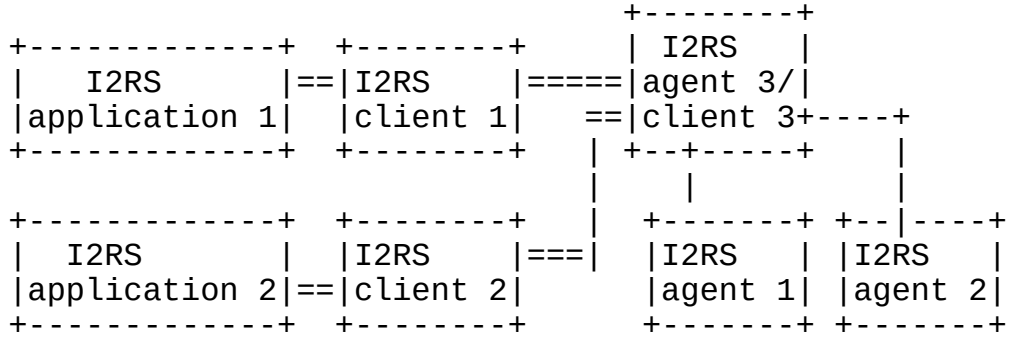
b) I2RS application client pair talking to two i2RS agents



c) two applications talk to 1 client



d) I2RS Broker (agent/client



4.1.1. access control Enforcement Scope

SEC-ENV-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. The mechanisms for the I2RS client access control are not in scope of the I2RS architecture [RFC7921], which exclusively focuses on the I2RS agent access control provided by the I2RS protocol.

Explanation:

This architecture 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.

4.1.2. Notification Requirements

SEC-ENV-REQ 7: When an access request to a routing resource is refused by one party (the I2RS client or the I2RS agent), the requester (e.g the application) as well as all intermediaries should indicate the reason the access has not been granted, and which entity rejected the request.

Explanation:

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.

SEC-REQ-07 indicates the I2RS agent may reject the request because the I2RS client is not an authorized I2RS client or have enough privileges to perform the requested transaction (read, write, start notifications or logging). The I2RS client should be notified of the reason the I2RS agent rejected the transaction due to a lack of authorization or priviledges, and the I2RS client should return a message to the application indicating the I2RS agent reject the transacation with an indication of this reason. Similarly, if the I2RS client does not grant the access to the application, the I2RS client should also inform the application. An example of an error message could be, "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 the following interactions:

- interactions between the application and the I2RS client,
- interactions between the I2RS client and the I2RS agent (security requirements are described by [I-D.ietf-i2rs-protocol-security-requirements]),
- and interactions between the I2RS agent and the I2RS routing system (forwarding plane, control plane, and routing plane).

4.1.3. Trust

SEC-ENV-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 (e.g. TLS) in order to reduce additional vector of attacks.

SEC-ENV-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.

Explanation:

The participants in the I2RS Plane (I2RS client, I2RS agent, and I2RS application) exchange critical information, and to be effective the communication should be trusted and free from security attacks.

The I2RS client or the I2RS agent should be able to reject messages over a communication channel that can be easily hijacked, like a clear text UDP channel.

4.1.4. Sharing access control Information

For the I2RS client:

SEC-ENV-REQ 10: The I2RS client MAY request information on its I2RS access control subset policies from the I2RS agent or cache requests that have been rejected by the I2RS agent to limit forwarding unnecessary queries to the I2RS agent.

SEC-ENV-REQ 11: The I2RS client MAY support receiving notifications when its I2RS access control subset policies have been updated by the I2RS agent.

Similarly, for the applications:

SEC-ENV-REQ 12: The applications MAY request information on its I2RS access control subset policies in order to limit forwarding unnecessary queries to the I2RS client.

SEC-ENV-REQ 13: The applications MAY subscribe to a service that provides notification when its I2RS access control subset policies have been updated.

For both the application and the client:

SEC-ENV-REQ 14: The I2RS access control should explicitly specify accesses that are granted. More specifically, anything not explicitly granted should be denied (default rule).

Explanation:

In order to limit the number of access requests that result in an error, each application or I2RS client can 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. One example, may be caching transaction requests that have been rejected.

I2RS access control should be appropriately be balanced between the I2RS client and the I2RS agent. It remains relatively easy to avoid the complete disclosure of the access control policies of the I2RS agent. Relative disclosure of access control policies may allow the leaking confidential information in case of misconfiguration. It is important to balance the level of trust of the I2RS client and the necessity of distributing the enforcement of the access control policies.

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. 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 benefits and problems with synchronizing its access control policies with the I2RS agent to proxy request validation versus simply passing the access request to the I2RS agent.

- 2) A single I2RS client connects to multiple applications or acts as a broker for many applications:
  In the case the I2RS agent has a single I2RS client attached. This may end up in a situation where the I2RS client is being delegated by the I2RS agent to enforce access control policies. In such as case, the I2RS agent may grant the I2RS client with high priviledges and blidngly trust the I2RS client without enforcing access control policies on what the I2RS client can do. Such a situation must be avoided as it could be used by malicious applications for a priviledge escalation by compromising the I2RS client. In this situation, the application uses the I2RS client to perform some action on behalf of the application that it normally does not have the priviledges to perform. In order to mitigate such attack, the I2RS client that connects to multiple applications or operates as a broker is expected to host application with an equivalent level of privileges.

4.1.5. Sharing Access Control in Groups of I2RS Clients and Agents

Overview:

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.

Requirements:

SEC-ENV-REQ 15: I2RS clients should be distributed and act as brokers for applications that share roughly similar permissions.

SEC-ENV-REQ 16: I2RS agents should be avoided granting extra privileges 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, so the I2RS client should not be provided these accesses.

SEC-ENV-REQ 17: I2RS client and I2RS agent should be able to trace [RFC7922] 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.

Explanation:

This restriction for distributed I2RS clients to act as brokers only for applications with roughly the same priviledges avoids the I2RS client extra priviledges compared to hosted applications, and discourages applications from perform privilege escalation within an I2RS client. For example, 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.

The I2RS agent can trace transactions that an I2RS client requests it to perform, and to link this to the application via the secondary opaque identifier to the application. This information is placed in a tracing log which is retrieved by management processes. If a particular application is granted a level of priviledges it should not have, then this tracing mechanism may detect this security intrusion after the instrusion has occurred.

4.1.6. Managing Access Control Policy

Access control policies should be implemented so that the policies remain manageable in short and longer term deployments of the I2RS protocol and the I2RS plane.

Requirements:

SEC-ENV-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 (I2RS client, I2RS agent, and application).

Explanation:

Granting or configuring an application with new policy should not require manual configuration of I2RS clients, I2RS agents, or other applications.

SEC-ENV-REQ 19: Access control should be scalable when the number of application grows as well as when the number of I2RS client increases.

Explanation:

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

SEC-ENV-REQ 20: Access control should be dynamically managed and easily updated.

Explanation:

Although the numberof 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.

SEC-ENV-REQ 21: I2RS clients and I2RS agents should be uniquely identified in the network to enable centralized management of the I2RS access control policies.

Explanation:

Centralized management of the access control policies of an I2RS plane with network that hosts several I2RS applications, clients and agents requires that each devices can be identified.

4.2. I2RS Agent Access Control Policies

Overview:

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.

Second, 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.

Third, each I2RS client may contact various I2RS agent with different privileges and access control policies.

4.2.1. I2RS Agent Access Control

This section provides recommendations on the I2RS agent access control policies to keep I2RS access control coherent within the I2RS plane.

Requirements:

SEC-ENV-REQ 22: I2RS agent access control policies should be primarily based on the I2RS clients as described in [RFC7921].

SEC-ENV-REQ 23: I2RS agent access control policies MAY be based on the application if the application identity has been authenticated by the I2RS client and passed via the secondary identity to the I2RS agent.

SEC-ENV-REQ 24: The I2RS agent should know which identity (E.g. 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.

SEC-ENV-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 [RFC7921]. Similar requirements exist

for components within the I2RS plane, but this is within the scope of the I2RS protocol security requirements [I-D.ietf-i2rs-protocol-security-requirements].

Explanation:

If the I2RS application is authenticated to the I2RS client, and the I2RS client is authenticated to the I2RS agent, and the I2RS client uses the opaque secondary identifier to pass an authenticated identifier to the I2RS agent, this may be used for access control. However, caution should be taken when using this chain of authentication since the secondary identifier is intended in the I2RS protocol for tracing.

From the environment perspective the I2RS agent MUST be aware when the resource has been modified outside the I2RS plane by another plane (management, control, or forwarding). The prioritization between the different planes should set a deterministic policy that allows the collision of two planes (I2RS plane and another plane) to be resolved via an overwrite policy in the I2RS agent.

Similar requirements exist for knowledge about identities within the I2RS plane which modify things in the routing system, but this is within the scope of the I2RS protocol's requirements for ephemeral state [I-D.ietf-i2rs-ephemeral-state] and security requirements [I-D.ietf-i2rs-protocol-security-requirements].

4.2.2. I2RS Client Access Control Policies

Overview:

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.

Requirements:

REQ 26: I2RS client should authenticate its applications. If the I2RS client acts as a broker and supports multiple applications, it should authenticate each application.

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 immediately and transparently 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.

Explanation:

If no authentication mechanisms have being provided between the I2RS client and the application, then I2RS client must be dedicated to a single application. By doing so, application authentication relies on the I2RS authentication mechanisms between the I2RS client and the I2RS agent.

If an I2RS client has multiple identities that are associated with different privileges for accessing an I2RS agent(s), the I2RS client access control policies should specify the I2RS client identity with the access control policy.

4.2.3. Application and Access Control Policies

Overview

Applications do 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.

Requirements:

SEC-ENV-REQ 28: applications SHOULD be uniquely identified by their associated I2RS clients

Explanation:

Different application may use different methods (or multiple methods) to communicate with its associated I2RS client, and each application may not use the same form of an application identifier. However, the I2RS client must obtain an identifier for each application. One method for this identification can be a system user id.

SEC-ENV-REQ 29: Each application SHOULD be associated to a restricted number of I2RS client

Explanation:

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. This does not prevent an I2RS client to host a large number of applications with the same levels of trust.

SEC-ENV-REQ 30: An application SHOULD be provided means and methods to contact their associated I2RS client.

Explanation:

It is obvious when an I2RS client belongs to the application as part of a module or a library that the application can communicate with a I2RS client. Similarly, if the application runs into a dedicated system with a I2RS client, it is obvious which I2RS client the application should contact. If the application connects to the I2RS client remotely, the application needs some to be able to retrieve the necessary information to contact its associated I2RS client (e.g. an IP address or a FQDN).

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

Overview

I2RS provides a programmatic interface in and out of the Internet routing system which provides the following advantages for security

- the use of automation reduces configuration errors
- the programmatic interface enables fast network reconfiguration and agility in adapting to network attacks,
- monitoring facilities to detect and configuration to mitigate a network attack.

Programmability allows applications to flexible control which may cause problems due to:

- applications which belong to different tenants with different objectives,

- applications which lack coordination resulting in unstable routing configurations such as oscillations between network configurations, and creation of loops for example. For example, one application may monitor a state and change to positive, and a second application performs the reverse operation (turns it negative). This fluctuation can cause a routing system to become unstable.

The I2RS plane requires data and application isolation to prevent such situations to happen. However, to guarantee the network stability constant monitoring and error detection are recommended to be activated.

Requirement:

SEC-ENV-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.

Explanation:

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

Overview:

Requirements for robustness to DoS attacks have been addressed in the communication channel section [RFC7921]. This section focuses on requirements for application isolation that help prevent DoS.

Requirements:

SEC-ENV-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 the broker performs resource controls for the hosted applications.

SEC-ENV-REQ 33: I2RS agent does not make response redirection possible unless the redirection is previously validated and agreed by the destination.

SEC-ENV-REQ 34: avoid the use of underlying protocols that are not robust to reflection attacks.

Explanation:

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. These attacks on the I2RS agent may involve an application (requesting through an I2RS Client) heavy computation by the I2RS agent in order to block operations like disk access.

Some DoS attacks may attack the I2RS Client’s reception of notification and monitoring data stream over the network. Other DoS attacks may focus on the application directly by performing reflection attacks. Such an attack could be performed by first detecting an application is related to monitoring the RIB or changing the RIB.

Reflection-based DoS may be performed at various levels utilizing UDP at the service to redirect data to a specific repository.

5.2.2. Application Logic Control

Overview

Requirements for application control have been addressed in the I2RS plane isolation as well as in the trusted Communication Channel sections. This section examines how application logic must be design to ensure application isolation.

Requirements:

SEC-ENV-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.

Explanation:

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.

6. Security Considerations

The whole document is about security requirements for the I2RS environment. To protect personal privacy, any identifier (I2RS application identifier, I2RS client identifier, or I2RS agent identifier) should not contain personal identifiable information.

7. IANA Considerations

No IANA considerations for this requirements.

8. 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, and Linda Dunbar.

9. References

9.1. 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>.

[RFC7920] Atlas, A., Ed., Nadeau, T., Ed., and D. Ward, "Problem Statement for the Interface to the Routing System", RFC 7920, DOI 10.17487/RFC7920, June 2016, <http://www.rfc-editor.org/info/rfc7920>.

[RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <http://www.rfc-editor.org/info/rfc7921>.

[RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", RFC 7922, DOI 10.17487/RFC7922, June 2016, <http://www.rfc-editor.org/info/rfc7922>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <http://www.rfc-editor.org/info/rfc7923>.

[I-D.ietf-i2rs-protocol-security-requirements] Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-ietf-i2rs-protocol-security-requirements-17 (work in progress), September 2016.

9.2. Informative References

[I-D.ietf-i2rs-ephemeral-state] Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-22 (work in progress), November 2016.

Authors' Addresses

Daniel Migault
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