

Internet Engineering Task Force (IETF)
Request for Comments: 8241
Category: Informational
ISSN: 2070-1721

S. Hares
Huawei
D. Migault
J. Halpern
Ericsson
September 2017

Interface to the Routing System (I2RS) Security-Related Requirements

Abstract

This document presents security-related requirements for the Interface to the Routing System (I2RS) protocol, which provides a new interface to the routing system described in the I2RS architecture document (RFC 7921). The I2RS protocol is implemented by reusing portions of existing IETF protocols and adding new features to them. One such reuse is of the security features of a secure transport (e.g., Transport Layer Security (TLS), Secure SHell (SSH) Protocol, Datagram TLS (DTLS)) such as encryption, message integrity, mutual peer authentication, and anti-replay protection. The new I2RS features to consider from a security perspective are as follows: a priority mechanism to handle multi-headed write transactions, an opaque secondary identifier that identifies an application using the I2RS client, and an extremely constrained read-only non-secure transport.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8241>.

Copyright Notice

Copyright (c) 2017 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 (<https://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	3
2. Terminology and Concepts	4
2.1. Requirements Language	4
2.2. Security Terminology	4
2.3. I2RS-Specific Terminology	5
2.4. Concepts	5
3. Security Features and Protocols: Reused and New	6
3.1. Security Protocols Reused by the I2RS Protocol	6
3.2. New Features Related to Security	7
3.3. I2RS Protocol Security Requirements vs. IETF Management Protocols	8
4. Security-Related Requirements	10
4.1. I2RS Peer (Agent and Client) Identity Authentication	10
4.2. Identity Validation before Role-Based Message Actions	11
4.3. Peer Identity, Priority, and Client Redundancy	12
4.4. Multi-Channel Transport: Secure and Non-Secure	13
4.5. Management Protocol Security	15
4.6. Role-Based Data Model Security	16
4.7. Security of the Environment	17
5. IANA Considerations	17
6. Security Considerations	17
7. References	18
7.1. Normative References	18
7.2. Informative References	18
Acknowledgements	20
Authors' Addresses	20

1. Introduction

The Interface to the Routing System (I2RS) protocol provides read and write access to information and state within the routing system. An I2RS client interacts with one or more I2RS agents to collect information from network routing systems. [RFC7921] describes the architecture of this interface, and this document assumes the reader is familiar with this architecture and its definitions.

The I2RS interface is instantiated by the I2RS protocol connecting an I2RS client and an I2RS agent associated with a routing system. The I2RS protocol is implemented by reusing portions of existing IETF protocols and adding new features to them. As a reuse protocol, it can be considered a higher-layer protocol because it can be instantiated in multiple management protocols (e.g., NETCONF [RFC6241] or RESTCONF [RFC8040]) operating over a secure transport. These protocols are what provide its security.

This document is part of a suite of documents outlining requirements for the I2RS protocol, which also includes the following:

- o "An Architecture for the Interface to the Routing System" [RFC7921]
- o "I2RS Ephemeral State Requirements" [RFC8242]
- o "Interface to the Routing System (I2RS) Traceability: Framework and Information Model" (which discusses traceability) [RFC7923]
- o "Requirements for Subscription to YANG Datastores" (which highlights the publication/subscription requirements) [RFC7922]

Since the I2RS "higher-layer" protocol changes the interface to the routing systems, it is important that implementers understand the new security requirements for the environment the I2RS protocol operates in. A summary of the I2RS protocol security environment is found in the I2RS architecture [RFC7921].

I2RS reuses the secure transport protocols (TLS, SSH, DTLS) that support encryption, message integrity, peer authentication, and key distribution protocols. Optionally, implementers may utilize Authentication, Authorization, and Accounting (AAA) protocols (Radius over TLS or Diameter over TLS) to securely distribute identity information.

Section 2 highlights some of the terminology and concepts that the reader is required to be familiar with.

Section 3 provides an overview of security features and protocols being reused (Section 3.1), lists the new security features being required (Section 3.2), and explores how existing and new security features and protocols would be paired with existing IETF management protocols (Section 3.3).

The new features I2RS extends to these protocols are a priority mechanism to handle multi-headed writes, an opaque secondary identifier to allow traceability of an application utilizing a specific I2RS client to communicate with an I2RS agent, and non-secure transport constrained to be used only for read-only data, which may include publicly available data (e.g., public BGP events, public telemetry information, web service availability) and some legacy data.

Section 4 provides the I2RS protocol security requirements of several security features. Protocols designed to be I2RS higher-layer protocols need to fulfill these security requirements.

2. Terminology and Concepts

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Security Terminology

This document uses the terminology found in [RFC4949] and [RFC7921]. Specifically, this document reuses the following terms from [RFC4949]:

- o access control
- o authentication
- o data confidentiality
- o data integrity
- o data privacy
- o identity
- o identifier
- o mutual authentication
- o role
- o role-based access control
- o security audit trail
- o trust

[RFC7922] describes traceability for the I2RS interface and the I2RS protocol. Traceability is not equivalent to a security audit trail or simple logging of information. A security audit trail may utilize traceability information.

2.3. I2RS-Specific Terminology

This document discusses the security of the multiple I2RS communication channels that operate over the higher-layer I2RS protocol. The higher-layer I2RS protocol combines a secure transport and I2RS contextual information, and it reuses IETF protocols and data models to create the secure transport and the contextual information driven by the I2RS data model. To describe how the I2RS higher-layer protocol combines other protocols, the following terms are used:

I2RS component protocols

Protocols that are reused and combined to create the I2RS higher-layer protocol.

I2RS secure transport component protocols (required)

Secure transport protocols that combine to support the I2RS higher-layer protocol.

I2RS management component protocols (required)

Management protocols that combine to provide the management-information context for the I@RS higher-layer protocol.

I2RS AAA component protocols (optional)

AAA protocols supporting the I2RS higher-layer protocol.

2.4. Concepts

The reader should be familiar with the pervasive security requirements in [RFC7258].

This document uses the following concepts from the I2RS architecture [RFC7921] listed below with their respective section numbers from said RFC:

- o I2RS client, agent, and protocol (Section 2)
- o I2RS higher-layer protocol (Section 7.2)

- o scope: read, notification, identity, and write (Section 2)
- o identity and secondary identity (Section 2)
- o roles or security rules (Section 2)
- o routing system/subsystem (Section 2)
- o I2RS assumed security environment (Section 4)
- o I2RS identity and authentication (Section 4.1)
- o scope of Authorization in I2RS client and agent (Section 4.2)
- o client redundancy with a single client identity (Section 4.3),
- o restrictions on I2RS in personal devices (Section 4.4)
- o communication channels and I2RS higher-layer protocol (Section 7.2)
- o active communication versus connectivity (Section 7.5)
- o multi-headed control (Section 7.8)
- o transaction, message, multi-message atomicity (Section 7.9)

3. Security Features and Protocols: Reused and New

3.1. Security Protocols Reused by the I2RS Protocol

I2RS requires a secure transport protocol and key distribution protocols. The secure transport for I2RS requires one to provide peer authentication. In addition, the features required for I2RS messages are confidentiality, authentication, and replay protection. According to [RFC8095], the secure transport protocols that support peer authentication, confidentiality, data integrity, and replay protection are the following:

1. TLS [RFC5246] over TCP or Stream Control Transmission Protocol (SCTP)
2. DTLS over UDP with replay detection and an anti-DoS stateless cookie mechanism required for the I2RS protocol and the DTLS options of record-size negotiation and conveyance of the Don't Fragment (DF) bit are set for IPv4, or no fragmentation extension headers for IPv6 to be optional in deployments are allowed by the I2RS protocol

3. HTTP over TLS (over TCP or SCTP)
4. HTTP over DTLS (with the requirements and optional features specified above in item 2)

As detailed in Section 3.3, the following protocols would need to be extended to provide confidentiality, data integrity, peer authentication, and key distribution and to run over a secure transport (TLS or DTLS):

- o IP Flow Information Export (IPFIX) over SCTP, TCP, or UDP
- o Forwarding and Control Element Separation (ForCES) Transport Mapping Layer (TML) over SCTP

The specific type of key management protocols an I2RS secure transport uses depends on the transport. Key management protocols utilized for the I2RS protocols SHOULD support automatic rotation.

An I2RS implementer may use AAA protocols over secure transport to distribute the identities for the I2RS client, I2RS agent, and role-authorization information. Two AAA protocols are as follows: Diameter [RFC6733] and Radius [RFC2865]. To provide I2RS peer identities with the best security, the AAA protocols MUST be run over a secure transport (Diameter over secure transport (TLS over TCP) [RFC6733]), Radius over a secure transport (TLS) [RFC6614]).

3.2. New Features Related to Security

The new features are priority, an opaque secondary identifier, and a non-secure protocol for read-only data constrained to specific standard usages. The I2RS protocol allows multi-headed control by several I2RS clients. This multi-headed control is based on the assumption that the operator deploying the I2RS clients, I2RS agents, and the I2RS protocol will coordinate the read, write, and notification scope so the I2RS clients will not contend for the same write scope. However, just in case there is an unforeseen overlap of I2RS clients attempting to write a particular piece of data, the I2RS architecture [RFC7921] provides the concept of each I2RS client having a priority. The I2RS client with the highest priority will have its write succeed. This document specifies requirements for this new concept of priority (see Section 4.3).

The opaque secondary identifier identifies an application that uses communication from the I2RS client to I2RS agent to manage the routing system. The secondary identifier is opaque to the I2RS protocol. In order to protect personal privacy, the secondary identifier should not contain identifiable personal information.

The last new feature related to I2RS security is the ability to allow nonconfidential data to be transferred over a non-secure transport. It is expected that most I2RS data models will describe information that will be transferred with confidentiality. Therefore, any model that transfers data over a non-secure transport is marked. The use of a non-secure transport is optional, and an implementer **SHOULD** create knobs that allow data marked as nonconfidential to be sent over a secure transport.

Nonconfidential data can only be data with read-scope or notification-scope transmission of events. Nonconfidential data cannot have write-scope or notification-scope configuration. Examples of nonconfidential data would be the telemetry information that is publicly known (e.g., BGP route-views data or website status data) or some legacy data (e.g., interface) that cannot be transported using secure transport. The IETF I2RS data models **MUST** indicate (in the model) the specific data that is nonconfidential.

Most I2RS data models will expect that the information described in the model will be transferred with confidentiality.

3.3. I2RS Protocol Security Requirements vs. IETF Management Protocols

Figure 1 provides a partial list of the candidate management protocols. It also lists the secure transports each protocol supports. The column on the right of the table indicates whether or not the transport protocol will need I2RS security extensions.

Management Protocol =====	Transport Protocol =====	I2RS Security Extensions =====
NETCONF	TLS over TCP (*1)	None required (*2)
RESTCONF	HTTP over TLS with X.509v3 certificates, certificate validation, mutual authentication: 1) authenticated server identity, 2) authenticated client identity (*1)	None required (*2)
ForCES	TML over SCTP (*1)	Needs an extension to TML to run TML over TLS over SCTP, or DTLS with options for replay protection and anti-DoS stateless cookie mechanism. (DTLS record size negotiation and conveyance of DF bits are optional). The IPsec mechanism is not sufficient for I2RS traveling over multiple hops (router + link) (*2)
IPFIX	SCTP, TCP, UDP TLS or DTLS for secure client (*1)	Needs an extension to support TLS or DTLS with options for replay protection and anti-DoS stateless cookie mechanism. (DTLS record size negotiation and conveyance of DF bits are optional)

*1 - Key management protocols MUST support appropriate key rotation.

*2 - Identity and role authorization distributed by Diameter or Radius MUST use Diameter over TLS or Radius over TLS.

Figure 1: Candidate Management Protocols and Their Secure Transports

4. Security-Related Requirements

This section discusses security requirements based on the following security functions:

- o peer identity authentication (Section 4.1)
- o Peer Identity validation before role-based message actions (Section 4.2)
- o peer identity and client redundancy (Section 4.3)
- o multi-channel transport requirements: Secure transport and non-secure Transport (Section 4.4)
- o management protocol security requirements (Section 4.5)
- o role-based security (Section 4.6)
- o security environment (Section 4.7)

The I2RS protocol depends upon a secure transport layer for peer authentication, data integrity, confidentiality, and replay protection. The optional non-secure transport can only be used for a restricted set of data available publicly (events or information) or a select set of legacy data. Data passed over the non-secure transport channel **MUST NOT** contain any data that identifies a person.

4.1. I2RS Peer (Agent and Client) Identity Authentication

Requirements:

SEC-REQ-01: All I2RS clients and agents **MUST** have an identity and at least one unique identifier for each party in the I2RS protocol context.

SEC-REQ-02: The I2RS protocol **MUST** utilize these identifiers for mutual identification of the I2RS client and agent.

SEC-REQ-03: Identifier distribution and the loading of these identifiers into the I2RS agent and client **SHOULD** occur outside the I2RS protocol prior to the I2RS protocol establishing a connection between I2RS client and agent. AAA protocols **MAY** be used to distribute these identifiers, but other mechanism can be used.

Explanation:

These requirements are for I2RS peer (I2RS agent and client) authentication. A secure transport (e.g., TLS) will authenticate based on these identities, but these identities are for the I2RS management layer. A AAA protocol distributing I2RS identity information **SHOULD** transport its information over a secure transport.

4.2. Identity Validation before Role-Based Message Actions**Requirements:**

SEC-REQ-04: An I2RS agent receiving a request from an I2RS client **MUST** confirm that the I2RS client has a valid identity.

SEC-REQ-05: An I2RS client receiving an I2RS message over a secure transport **MUST** confirm that the I2RS agent has a valid identifier.

SEC-REQ-06: An I2RS agent receiving an I2RS message over a non-secure transport **MUST** confirm that the content is suitable for transfer over such a transport.

Explanation:

Each I2RS client has a scope based on its identity and the security roles (read, write, or events) associated with that identity, and that scope must be considered in processing an I2RS message sent on a communication channel. An I2RS communication channel may utilize multiple transport sessions or establish a transport session and then close the transport session. Therefore, it is important that the I2RS peers operate utilizing valid peer identities when a message is processed rather than checking if a transport session exists.

During the time period when a secure transport session is active, the I2RS agent **SHOULD** assume that the I2RS client's identity remains valid. Similarly, while a secure connection exists that included validating the I2RS agent's identity and a message is received via that connection, the I2RS client **SHOULD** assume that the I2RS agent's identity remains valid.

The definition of what constitutes a valid identity or a valid identifier **MUST** be defined by the I2RS protocol.

4.3. Peer Identity, Priority, and Client Redundancy

Requirements:

SEC-REQ-07: Each I2RS identifier **MUST** be associated with just one priority.

SEC-REQ-08: Each identifier is associated with one secondary identifier during a particular I2RS transaction (e.g., read/write sequence), but the secondary identifier may vary during the time a connection between the I2RS client and I2RS agent is active.

Explanation:

The I2RS architecture also allows multiple I2RS clients with unique identities to connect to an I2RS agent (see Section 7.8 of [RFC7921]). The I2RS deployment using multiple clients **SHOULD** coordinate this multi-headed control of I2RS agents by I2RS clients so no conflict occurs in the write scope. However, in the case of conflict on a write-scope variable, the error resolution mechanisms defined by the I2RS architecture multi-headed control (Section 7.8 of [RFC7921]) allow the I2RS agent to deterministically choose one I2RS client. The I2RS client with highest priority is given permission to write the variable, and the second client receives an error message.

A single I2RS client may be associated with multiple applications with different tasks (e.g., weekly configurations or emergency configurations). The secondary identity is an opaque value that the I2RS client passes to the I2RS agent so that this opaque value can be placed in the tracing file or event stream to identify the application using the communication from I2RS client to agent. The I2RS client is trusted to simply assert the secondary identifier.

One example of the use of the secondary identity is the situation where an operator of a network has two applications that use an I2RS client. The first application is a weekly configuration application that uses the I2RS protocol to change configurations. The second application allows operators to make emergency changes to routers in the network. Both of these applications use the same I2RS client to write to an I2RS agent. In order for traceability to determine which application (weekly configuration or emergency) wrote some configuration changes to a router, the I2RS client sends a different opaque value for each of the applications. The weekly configuration secondary opaque value could be "xzy-splot" and the emergency secondary opaque value could be "splish-splash".

A second example is if the I2RS client is used for the monitoring of critical infrastructure. The operator of a network using the I2RS client may desire I2RS client redundancy where the monitoring application with the I2RS client is deployed on two different boxes with the same I2RS client identity (see Section 4.3 of [RFC7921]). These two monitoring applications pass to the I2RS client whether the application is the primary or back-up application, and the I2RS client passes this information in the I2RS secondary identifier, as the figure below shows. The primary application's secondary identifier is "primary-monitoring", and the back-up application secondary identifier is "backup-monitoring". The I2RS tracing information will include the secondary identifier information along with the transport information in the tracing file in the agent.

```
Application A--I2RS client--Secure transport(#1)
[I2RS identity 1, secondary identifier: "primary-monitoring"]-->

Application B--I2RS client--Secure transport(#2)
[I2RS identity 1, secondary identifier: "backup-monitoring"]-->
```

Figure 2: Primary and Back-Up Application for Monitoring Identification Sent to Agent

4.4. Multi-Channel Transport: Secure and Non-Secure

Requirements:

SEC-REQ-09: The I2RS protocol **MUST** be able to transfer data over a secure transport and optionally **MAY** be able to transfer data over a non-secure transport. The default transport is a secure transport, and this secure transport is mandatory to implement in all I2RS agents and in any I2RS client that a) performs a write scope transaction that is sent to the I2RS agent or b) configures an Event Scope transaction. This secure transport is mandatory to use on any I2RS client's Write transaction or the configuration of an Event Scope transaction.

SEC-REQ-10: The secure transport **MUST** provide data confidentiality, data integrity, and practical replay prevention.

SEC-REQ-11: The I2RS client and I2RS agent **SHOULD** implement mechanisms that mitigate DoS attacks. This means the secure transport must support DoS prevention. For the non-secure transport, the I2RS higher-layer protocol **MUST** contain a transport management layer that considers the detection of DoS attacks and provides a warning over a secure transport channel.

SEC-REQ-12: A secure transport **MUST** be associated with a key management solution that can guarantee that only the entities having sufficient privileges can get the keys to encrypt/decrypt the sensitive data.

SEC-REQ-13: A machine-readable mechanism to indicate that a data model contains nonconfidential data **MUST** be provided. A non-secure transport **MAY** be used to publish only read-scope or notification-scope data if the associated data model indicates that the data in question is nonconfidential.

SEC-REQ-14: The I2RS protocol **MUST** be able to support multiple secure transport sessions providing protocol and data communication between an I2RS agent and client. However, a single connection between I2RS agent and client **MAY** elect to use a single secure transport session or a single non-secure transport session conforming to the requirements above.

SEC-REQ-15: Deployment configuration knobs **SHOULD** be created to allow operators to send "nonconfidential" read scope (data or event streams) over a secure transport.

SEC-REQ-16: The I2RS protocol makes use of both secure and non-secure transports, but this use **MUST NOT** be done in any way that weakens the secure transport protocol used in the I2RS protocol or other contexts that do not have this requirement for mixing secure and non-secure modes of operation.

Explanation:

The I2RS architecture defines three scopes: read, write, and notification. Non-secure data can only be used for read and notification scopes of "nonconfidential data". The configuration of ephemeral data in the I2RS agent uses write scope either for data or for configuration of event notification streams. The requirement to use secure transport for configuration prevents accidental or malevolent entities from altering the I2RS routing system through the I2RS agent.

It is anticipated that the passing of most I2RS ephemeral state operational statuses **SHOULD** be done over a secure transport.

In most circumstances, the secure transport protocol will be associated with a key management system. Most deployments of the I2RS protocol will allow for automatic key management systems. Since the data models for the I2RS protocol will control key routing functions, it is important that deployments of I2RS use automatic key management systems.

Per BCP 107 [RFC4107], while key management systems **SHOULD** be automatic, the systems **MAY** be manual in the following scenarios:

- a) The environment has limited bandwidth or high round-trip times.
- b) The information being protected has low value.
- c) The total volume of traffic over the entire lifetime of the long-term session key will be very low.
- d) The scale of the deployment is limited.

Operators deploying the I2RS protocol selecting manual key management **SHOULD** consider both short- and medium-term plans. Deploying automatic systems initially may save effort in the long term.

4.5. Management Protocol Security

Requirements:

SEC-REQ-17: In a critical infrastructure, certain data within routing elements is sensitive and read/write operations on such data **SHOULD** be controlled in order to protect its confidentiality. To achieve this, higher-layer protocols **MUST** utilize a secure transport, and they **SHOULD** provide access-control functions to protect confidentiality of the data.

SEC-REQ-18: An integrity protection mechanism for I2RS **MUST** be provided that will be able to ensure the following:

- 1) the data being protected is not modified without detection during its transportation,
- 2) the data is actually from where it is expected to come from, and
- 3) the data is not repeated from some earlier interaction the higher-layer protocol (best effort).

The I2RS higher-layer protocol operating over a secure transport provides this integrity. The I2RS higher-layer protocol operating over a non-secure transport **SHOULD** provide some way for the client receiving nonconfidential read-scoped or event-scoped data over the non-secure connection to detect when the data integrity is questionable; and in the event of a questionable data integrity, the I2RS client should disconnect the non-secure transport connection.

SEC-REQ-19: The I2RS higher-layer protocol **MUST** provide a mechanism for message traceability (requirements in [RFC7922]) that supports the tracking higher-layer functions run across secure connection or a non-secure transport.

Explanation:

Most carriers do not want a router's configuration and data-flow statistics to be known by hackers or their competitors. While carriers may share peering information, most carriers do not share configuration and traffic statistics. To achieve this, the I2RS higher-layer protocol (e.g., NETCONF) requires access control (NETCONF Access Control Model [RFC6536]) for sensitive data needs to be provided; and the confidentiality protection on such data during transportation needs to be enforced.

Integrity of data is important even if the I2RS protocol is sending nonconfidential data over a non-secure connection. The ability to trace I2RS protocol messages that enact I2RS transactions provides a minimal aid to helping operators check how messages enact transactions on a secure or non-secure transport. Contextual checks on specific nonconfidential data sent over a non-secure connection may indicate the data has been modified.

4.6. Role-Based Data Model Security

In order to make access control more manageable, the I2RS architecture [RFC7921] specifies a "role" to categorize users into a group (rather than handling them individually) for access-control purposes (role-based access control). Therefore, an I2RS role specifies the access control for a group as being read, write, or notification.

SEC-REQ-20: The rules around what I2RS security role is permitted to access and manipulate what information over a secure transport (which protects the data in transit) **SHOULD** ensure that data of any level of sensitivity is reasonably protected from being observed by those without permission to view it, so that privacy requirements are met.

SEC-REQ-21: Role security **MUST** work when multiple transport connections are being used between the I2RS client and agent as the I2RS architecture [RFC7921] describes.

Sec-REQ-22: If an I2RS agent or client is tightly correlated with a person, then the I2RS protocol and data models **SHOULD** provide additional security that protects the person's privacy.

Explanation:

An I2RS higher-layer protocol uses a management protocol (e.g., NETCONF, RESTCONF) to pass messages in order to enact I2RS transactions. Role security must secure data (sensitive and normal data) in a router even when it is operating over multiple connections at the same time. NETCONF can run over TLS (over TCP or SCTP) or SSH. RESTCONF runs over HTTP over a secure transport (TLS). SCTP [RFC4960] provides security for multiple streams plus end-to-end transport of data. Some I2RS functions may wish to operate over DTLS [RFC6347], which runs over UDP ([RFC768]) and SCTP ([RFC5764]).

Please note the security of the connection between application and I2RS client is outside of the I2RS protocol or I2RS interface.

While I2RS clients are expected to be related to network devices and not individual people, if an I2RS client ran on a person's phone, then privacy protection to anonymize any data relating to a person's identity or location would be needed.

A variety of forms of management may set policy on roles: "operator-applied knobs", roles that restrict personal access, data models with specific "privacy roles", and access filters.

4.7. Security of the Environment

The security for the implementation of a protocol also considers the protocol environment. Implementers should review the summary of the I2RS security environment in [RFC7921].

5. IANA Considerations

This document does not require any IANA actions.

6. Security Considerations

This is a document about security requirements for the I2RS protocol and data models. Security considerations for the I2RS protocol include both the protocol and the security environment.

7. References

7.1. 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <<https://www.rfc-editor.org/info/rfc4107>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc7923>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

7.2. Informative References

- [RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.

- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <<https://www.rfc-editor.org/info/rfc2865>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <<https://www.rfc-editor.org/info/rfc5764>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, DOI 10.17487/RFC6614, May 2012, <<https://www.rfc-editor.org/info/rfc6614>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<https://www.rfc-editor.org/info/rfc6733>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [RFC8095] Fairhurst, G., Ed., Trammell, B., Ed., and M. Kuehlewind, Ed., "Services Provided by IETF Transport Protocols and Congestion Control Mechanisms", RFC 8095, DOI 10.17487/RFC8095, March 2017, <<https://www.rfc-editor.org/info/rfc8095>>.
- [RFC8242] Haas, J. and S. Hares, "Interface to the Routing System (I2RS) Ephemeral State Requirements", RFC 8242, DOI 10.17487/RFC8242, September 2017, <<http://www.rfc-editor.org/info/rfc8242>>.

Acknowledgements

The authors would like to thank Wes George, Ahmed Abro, Qin Wu, Eric Yu, Joel Halpern, Scott Brim, Nancy Cam-Winget, Dacheng Zhang, Alia Atlas, and Jeff Haas for their contributions to the I2RS security requirements discussion and this document. The authors would like to thank Bob Moskowitz, Kathleen Moriarty, Stephen Farrell, Radia Perlman, Alvaro Retana, Ben Campbell, and Alissa Cooper for their review of these requirements.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America

Email: shares@ndzh.com

Daniel Migault
Ericsson
8275 Trans Canada Route
Saint Laurent, QC H4S
Canada

Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson
United States of America

Email: joel.halpern@ericsson.com