I2RS working group
Internet-Draft
Intended status: Informational
Expires: April 2, 2017

S. Hares
Huawei
D. Migault
J. Halpern
Ericsson

September 29, 2016

I2RS Security Related Requirements
draft-ietf-i2rs-protocol-security-requirements-16

Abstract

This presents security-related requirements for the I2RS protocol which provides a new interface to the routing system described in the I2RS architecture document (RFC7921). The I2RS protocol is a re-use protocol implemented by re-using portions of existing IETF protocols and adding new features to these protocols. The I2RS protocol re-uses security features of a secure transport (E.g. TLS, SSH, DTLS) such as encryption, message integrity, mutual peer authentication, and replay protection. The new I2RS features to consider from a security perspective are: a priority mechanism to handle multi-headed write transactions, an opaque secondary identifier which identifies an application using the I2RS client, and an extremely constrained read-only non-secure transport. This document provides the detailed requirements for these security features.

Table of Contents

1. Introduction

The Interface to the Routing System (I2RS) 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 documents assumes the reader is familiar with this architecture and its definitions. Section 2 highlights some of the references the reader is required to be familiar with.

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 a re-use protocol implemented by re-using portions of existing IETF protocols, and adding new features to these protocols. As a re-use protocol, it can be considered a higher-level protocol since it can be instantiated in multiple management protocols (e.g. NETCONF [RFC6241] or RESTCONF [I-D.ietf-netconf-restconf]) operating over a secure transport. The security for the I2RS protocol comes from the management protocols operating over a a secure transport.

This document is part of the requirements for I2RS protocol which also include:

- I2RS architecture [RFC7921],
- I2RS ephemeral state requirements [I-D.ietf-i2rs-ephemeral-state],
- publication/subscription requirements [RFC7922], and
- traceability [RFC7923].

Since the I2RS "higher-level" 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. These security requirements for the I2RS environment are specified in [I-D.ietf-i2rs-security-environment-reqs]; and the summary of the I2RS protocol security environment is found in the I2RS Architecture [RFC7920].

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

Section 3 provides an overview of security features and protocols being re-used (section 3.1) and the new security features being

required (section 3.2). Section 3 also 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 insecure transport constrained to be utilized only for read-only data, which may include publically 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 by the following security features:

- peer identity authentication (section 4.1),
- peer identity validation before role-based message actions (section 4.2)
- peer identity and client redundancy (section 4.3),
- multi-channel transport requirements: Secure transport and insecure Transport (section 4.4),
- management protocol security requirements (section 4.5),
- role-based security (section 4.6),
- security environment (section 4.7)

Protocols designed to be I2RS higher-layer protocols need to fulfill these security requirements.

2. Definitions

2.1. Requirements Language

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 RFC 2119 [RFC2119].

2.2. Security Definitions

This document utilizes the definitions found in the following documents: [RFC4949] and [RFC7921]

Specifically, this document utilizes the following definitions from [RFC4949]:

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

[RFC7922] describes traceability for 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.

This document also requires that the user is familiar with the pervasive security requirements in [RFC7258].

2.3. I2RS Specific Definitions

The document utilizes the following concepts from the I2RS architecture: [RFC7921]:

- I2RS client, I2RS agent, and I2RS protocol (section 2),
- I2RS higher-layer protocol (section 7.2)
- scope: read scope, notification scope, and write scope (section 2),
- identity and scope of the identity (section 2),

- roles or security rules (section 2),
- identity and scope, and secondary identity (section 2),
- routing system/subsytem (section 2),
- I2RS assumed security environment (section 4),
- I2RS identity and authorization (section 4.1),
- I2RS authorization, scope of Authorization in I2RS client and agent (section 4.2),
- client redundancy with a single client identity (section 4.3),
- restrictions on I2RS in personal devices (section 4.4),
- communication channels and I2RS high-layer protocol (section 7.2),
- active communication versus connectivity (section 7.5),
- multi-headed control (section 7.8), and
- transaction, message, multi-message atomicity (section 7.9).

This document assumes the reader is familar with these terms.

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

I2RS component protocols

: Protocols which are re-used and combined to create the I2RS protocol.

I2RS secure-transport component protocols

: The I2RS secure transport protocols that support the I2RS higher-layer protocol.

I2RS management component protocols

The I2RS management protocol which provide the management information context.

I2RS AAA component protocols

The I2RS AAA protocols supporting the I2RS higher-layer protocol.

The I2RS higher-layer protocol requires implementation of a I2RS secure-transport component protocol and the I2RS management component protocol. The I2RS AAA component protocol is optional.

3. Security Features and Protocols: Re-used and New

3.1. Security Protocols Re-Used by the I2RS Protocol

I2RS requires a secure transport protocol and key distribution protocols. The secure transport features required by I2RS are peer authentication, confidentiality, data integrity, and replay protection for I2RS messages. According to [I-D.ietf-taps-transports], the secure transport protocols which support peer authentication, confidentiality, data integrity, and replay protection are the following:

1. TLS [RFC5246] over TCP or SCTP,

2. DTLS over UDP with replay detection and anti-DoS stateless cookie mechanism required for the I2RS protocol, and the I2RS protocol allow DTLS options of record size negotiation and and conveyance of "don't" fragment bits to be optional in deployments.

3. HTTP over TLS (over TCP or SCTP), and

4. HTTP over DTLS (with the requirements and optional features specified above in item 2).

The following protocols would need to be extended to provide confidentiality, data integrity, peer authentication, and key distribution protocols: IPFIX (over SCTP, TCP or UDP) and ForCES TML layer (over SCTP). These protocols will need extensions to run over a secure transport (TLS or DTLS) (see section 3.3 for details).

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 I2RS client and I2RS agent and role authorization information. Two AAA protocols are: Diameter [RFC6733]

and Radius [RFC2865]. To provide the best security I2RS peer identities, 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 an insecure 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 unforseen 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.

The opaque secondary identifier identifies an application which is using the I2RS client to I2RS agent communication 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 personal identifiable information.

The last new feature related to I2RS security is the ability to allow non-confidential 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 which 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 non-confidential to be sent over a secure transport.

Non-confidential data can only be read or notification scope transmission of events. Non-confidential data cannot be write scope or notification scope configuration. An example of non-confidential data is the telemetry information that is publically known (e.g. BGP route-views data or web site status data) or some legacy data (e.g. interface) which cannot be transported in secure transport. The IETF I2RS Data models MUST indicate in the data model the specific data which is non-confidential.

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

Table 1 below provides a partial list of the candidate management protocols and the secure transports each one of the support. One column in the table indicates the transport protocol will need I2RS security extensions.

Mangement Protocol	Transport Protocol	I2RS 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 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 "don't" fragment 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 to extension to support TLS or DTLS with options for replay protection and anti-DoS stateless cookie mechanism. (DTLS record size negotiation and conveyance of "don't" fragment

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.

4. Security-Related Requirements

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

- peer identity authentication (section 4.1),
- Peer Identity validation before Role-based Message Actions (section 4.2)
- peer identity and client redundancy (section 4.3),
- multi-channel transport requirements: Secure transport and insecure Transport (section 4.4),
- management protocol security requirements (section 4.5),
- role-based security (section 4.6),
- 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 insecure transport can only be used restricted set of publically data available (events or information) or a select set of legacy data. Data passed over the insecure transport channel MUST NOT contain any data which identifies a person or any "write" transactions.

4.1. I2RS Peers(agent and client) Identity Authentication

The following requirements specify the security requirements for Peer Identity Authentication for the I2RS protocol:

- SEC-REQ-01: All I2RS clients and I2RS agents MUST have an identity, and at least one unique identifier that uniquely identifies 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 I2RS agent.

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

Explanation:

These requirements specify the requirements for I2RS peer (I2RS agent and I2RS client) authentication. A secure transport (E.g. TLS) will authenticate based on these identities. The AAA protocol distributing I2RS identity information SHOULD transport its information over a secure transport.

4.2. Identity Validation Before Role-Based Message Actions

The requirements for I2RS clients with Secure Connections are the following:

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 an insecure 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 messages 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 are operating 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.

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 (section 7.8). 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 ([RFC7921], section 7.8) 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 I2RS client to I2RS agent communication. 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 is an application that allows operators to makes 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 "xzzy-splot" and the emergency secondary opaque value could be "splish-splash".

A second example is if the I2RS client is used for monitoring of critical infrastructure. The operator of a network using the I2RS client may desire I2RS client redundancy where the monitoring application wth the I2RS client is deployed on two different boxes with the same I2RS client identity (see [RFC7921] section 4.3) 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 identitifier as the figure below shows. The primary applications secondary identifier is "primary-monitoring", and the backup 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.

Example 2: Primary and Backup Application for Monitoring Identification sent to 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 1

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

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 (MTI) in all I2RS agents, and in any I2RS client which: a) performs a Write scope transaction which is sent to the I2RS agent or b): configures an Event Scope transaction. This secure transport is mandatory to use (MTU) 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 protocol SHOULD implement mechanisms that mitigate DoS attacks. For the secure transport, this means the secure transport must support DoS prevention. For the insecure transport protocol, 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 non-confidential 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 that data is non-confidential.

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 an I2RS client. However, a single I2RS agent to I2RS client connection MAY elect to use a single secure transport session or a single non-secure transport session conforming the requirements above.

SEC-REQ-15: Deployment configuration knobs SHOULD be created to allow operators to send "non-confidential" Read scope (data or Event streams) over a secure transport.

SEC-REQ-16: The I2RS protocol makes use of both secure and insecure 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 insecure modes of operation.

Explanation:

The I2RS architecture defines three scopes: read, write, and notification scope. Insecure data can only be used for read scope and notification scope of "non-confidential data". The configuration of ephemeral data in the I2RS agent uses either write scope for data or write scope 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 status 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 BCP107 [RFC4107] while key management system 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 over 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 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 an insecure transport SHOULD provide some way for the client receiving non-confidential read-scoped or event-scoped data over the insecure connection to detect when the data integrity is questionable; and in the event of a questionable data integrity the I2RS client should disconnect the insecure 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 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 (NACM [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 non-confidential data over an insecure 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 insecure transport. Contextual checks on specific non-confidential data sent over a insecure connection may indicate the data integrity is questionable.

4.6. Role-Based Data Model Security

The I2RS Architecture [RFC7921] specifies access control by "role" where role is a method of making access control more manageable by creating a grouping of users so that access control can be specified for a role rather than for each of the individuals. Therefore, 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 I2RS agent as the I2RS architecture [RFC7921] describes.

Sec-REQ-22: If an I2RS agents or an I2RS 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:

I2RS higher-layer uses management protocol E.g. NETCONF, RESTCONF) to pass messages in order to enact I2RS transactions. Role Security must secure data (sensitivity 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 which runs over UDP ([RFC6347]), DDCP ([RFC6238]), and SCTP ([RFC5764]).

Please note the security of the application to I2RS client connection 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 managemen 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. The environmental security requirements are found in: [I-D.ietf-i2rs-security-environment-reqs].

5. Security Considerations

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

6. IANA Considerations

This draft is requirements, and does not request anything of IANA.

7. Acknowledgement

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.

8. References

8.1. Normative References

[I-D.ietf-i2rs-security-environment-reqs] Migault, D., Halpern, J., and S. Hares, "I2RS Environment Security Requirements", draft-ietf-i2rs-security-environment-reqs-01 (work in progress), April 2016.

[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>.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <http://www.rfc-editor.org/info/rfc4107>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://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, <http://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, <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>.

8.2. Informative References

[I-D.ietf-i2rs-ephemeral-state] Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-18 (work in progress), September 2016.

[I-D.ietf-netconf-restconf] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-17 (work in progress), September 2016.

[I-D.ietf-taps-transports] Fairhurst, G., Trammell, B., and M. Kuehlewind, "Services provided by IETF transport protocols and congestion control mechanisms", draft-ietf-taps-transports-11 (work in progress), July 2016.

[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, <http://www.rfc-editor.org/info/rfc2865>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc5764>.

[RFC6238] M'Raihi, D., Machani, S., Pei, M., and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm", RFC 6238, DOI 10.17487/RFC6238, May 2011, <http://www.rfc-editor.org/info/rfc6238>.

[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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc6733>.

[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>.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC HAP 2N2
Canada

Email: daniel.migault@ericsson.com

Joel Halpern
Ericsson
US

Email: joel.halpern@ericsson.com