INCH Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2010

K. Moriarty
RSA
B. Trammell
Hitachi Europe
January 21, 2010

Transport of Real-time Inter-network Defense (RID) Messages
draft-moriarty-post-inch-rid-transport-01.txt

Abstract

Documents intended to be shared among multiple constituencies must share a common format and transport mechanism. The Incident Object Description Exchange Format (IODEF) defines a common XML format for document exchange, and Realtime Internetwork Defense (RID) defines extensions to IODEF intended for the cooperative handling of security incidents within consortia of network operators and enterprises. This document outlines the transport of IODEF and RID messages over HTTP/TLS.

Table of Contents

1. Introduction

The Incident Object Description Exchange Format (IODEF) [RFC5070] describes an XML document format for the purpose of exchanging data between Computer Security Incident Response Teams (CSIRTs) or those responsible for security incident handling for network providers (NPs). The defined document format provides an easy way for CSIRTs to exchange data in a way which can be easily parsed.

IODEF defines a message format, not a transport protocol, as the sharing of messages is assumed to be out of scope in order to allow CSIRTs to exchange and store messages in a way most suited to their established incident handling processes. However, extensions such as Real-time Inter-network Defense (RID) [I-D.moriarty-post-inch-rid] do require a specification of a transport protocol to ensure interoperability among members in a RID consortium. This document specifies the transport of RID messages within HTTP [RFC2616] Request and Response messages transported over TLS [RFC5246] (herein, HTTP/TLS). Note that any IODEF message may also be transported using this mechanism, by sending it as a RID Report message.

2. Terminology

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. Transmission of RID Messages over HTTP/TLS

This section specifies the details of the transport of RID messages over HTTP/TLS. In this arrangement, each RID server is both an HTTP/TLS server and an HTTP/TLS Client. When a RID message must be sent, the sending RID system connects to the receiving RID system and sends the message, optionally receiving a message in reply. All RID systems MUST be prepared to accept HTTP/TLS connections from any RID peer with which it communicates, in order to support callback for delayed replies (see below).

BCP 56 [RFC3205] contains a number of important considerations when using HTTP for application protocols. These include the size of the payload for the application, whether the application will use a web browser, whether the protocol should be defined on a port other than 80, and if the security provided through HTTP/TLS suits the needs of the new application.

It is acknowledged within the scope of these concerns that HTTP/TLS

is not ideally suited for RID transport, as the former is a client-server protocol and the latter a message-exchange protocol; however, the ease of implementation of RID systems over HTTP/TLS outweighs these concerns. Consistent with BCP 56, RID systems will listen for TCP connections on port [IANA NOTE: assigned port goes here]. Every RID system participating in a consortium MUST listen for HTTP/TLS connections on the assigned port.

All RID messages sent in HTTP Requests MUST be sent using the POST with a Request-URI of /; additional Request-URI paths are reserved for future use by RID.

The following table lists the allowable RID message types in an HTTP Response for a given RID message type in the Request. A RID system MUST be prepared to handle an HTTP Response of the given type(s) when sending the corresponding HTTP Request. A RID system MUST NOT send an HTTP Response containing any RID message other than the one corresponding to the one sent in the HTTP Request.

As the queries and replies in a RID message exchange may be significantly separated in time, the receiving RID system MAY return 202 Accepted, terminate the connection, and connect to the requesting RID system and sending the RID reply in an HTTP Request at a later time. This mechanism is referred to in this document as "RID callback". When performing RID callback, a responding system MUST connect to the network- and transport-layer addresses from which the original request was sent; there is no mechanism in RID for redirected callback.

While a RID system SHOULD return the reply in an HTTP Response if it is available immediately or within a generally accepted HTTP client time out (about thirty seconds), this is not mandatory, and as such RID systems MUST be prepared for a query to be met with a 202 Accepted, an empty Response body, a connection termination and a callback.

RID systems accepting a callback message in an HTTP Request MUST return 202 Accepted.

The following table lists the allowable request/response pairs for RID.

Request RID type	Callback	Result	Response RID type
TraceRequest		200	RequestAuthorization
TraceRequest		200	Result
TraceRequest		202	[empty]
RequestAuthorization	X	202	[empty]
Result	X	202	[empty]
Investigation		200	Result
Investigation		202	[empty]
Report	X	202	[empty]
IncidentQuery		200	Report
IncidentQuery		202	[empty]

For security purposes, RID systems SHOULD NOT return 3xx Redirect response codes, and MUST NOT follow any 3xx Redirect. When a RID System's address changes, contact point information within the consortium must be updated out of band.

If a RID system receives an improper RID message in an HTTP Request, it MUST return an appropriate 4xx Client Error result code to the requesting RID system. If a RID system cannot process a RID message received in an HTTP Request due to an error on its own side, it MUST return an appropriate 5xx Server Error result code to the requesting RID system.

Note that HTTP provides no mechanism for signaling to a server than a response body is improper. If an RID system receives and improper RID message in an HTTP Response, or cannot process a RID message received in an HTTP Response due to an error on its own side, it MUST log the error and present it to the RID system administrator for handling; the error logging format is an implementation detail and is considered out of scope for this specification.

RID systems MUST support and SHOULD use HTTP/1.1 persistent connections as described in [RFC2616] to minimize TCP connection setup overhead. RID systems MUST support chunked transfer encoding on the HTTP server side to allow the implementation of clients that do not need to precalculate message sizes before constructing HTTP headers.

RID systems MUST use TLS for confidentiality, identification, and strong mutual authentication as in [RFC2818]; see Section 4 below for details.

4. Security Considerations

All security considerations of related documents MUST be considered, especially the Incident Object Description Exchange Format (IODEF) [RFC5070] and Real-time Inter-network Defense (RID) [I-D.moriarty-post-inch-rid]. The transport described herein is built on the foundation of these documents; the security considerations contained therein are incorporated by reference.

For transport confidentiality, identification, and authentication, TLS with mutual authentication MUST be used to secure the HTTP connection as in [RFC2818]. Each RID consortium SHOULD use a trusted public key infrastructure (PKI) to manage identities dor RID systems participating in TLS connections. At minimum, each RID system MUST trust a set of X.509 Issuer identities ("Certificate Authorities") to authenticate RID system peers with which it is willing to exchange information, and/or a specific white list of X.509 Subject identities of RID system peers directly.

RID systems SHOULD additionally verify the fully qualified domain name (FQDN) of a connected RID system peer against the presented Subject identity. The fully qualified domain name used to identify a RID system may be stored either in a subjectAltName extension of type dNSName, or in the most specific Common Name field of the Subject identity of the RID system's X.509 certificate. Internationalized FQDNs MUST be encoded using Punycode [RFC3492]. If both a Common Name and subjectAltName FQDN are present, the subjectAltName is to be given preference.

5. IANA Considerations

Consistent with BCP 56 [RFC3205], since RID over HTTP/TLS is a substantially new service, and should be controlled at the consortium member network's border differently than HTTP/TLS, it requires a new port number. IANA has assigned port [IANA NOTE: assign port number here] to RID over HTTP/TLS.

6. References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3205] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, February 2002.

[RFC3492] Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[I-D.moriarty-post-inch-rid] Moriarty, K., "Real-time Inter-network Defense", draft-moriarty-post-inch-rid-09 (work in progress), July 2009.

Authors' Addresses

Kathleen M. Moriarty
RSA, The Security Division of EMC
174 Middlesex Turnpike
Bedford 01730
United States

Email: Moriarty_Kathleen@EMC.com

Brian H. Trammell
Hitachi Europe
c/o ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Phone: +41 44 632 70 13
Email: brian.trammell@hitachi-eu.com