NETCONF Working Group
Internet-Draft
Updates: 4253 (if approved)
Intended status: Standards Track
Expires: November 13, 2015

K. Watsen
Juniper Networks

May 12, 2015

NETCONF Call Home and RESTCONF Call Home
draft-ietf-netconf-call-home-05

Abstract

This RFC presents NETCONF Call Home and RESTCONF Call Home, which enable a NETCONF or RESTCONF server to initiate a secure connection to a NETCONF or RESTCONF client respectively.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. Please note that no other RFC Editor instructions are specified anywhere else in this document.

Artwork in this document contains placeholder references for this draft. Please apply the following replacement:

- "XXXX" --> the assigned RFC value for this draft

This document contains references to other drafts in progress, both in the Normative References section, as well as in body text throughout. Please update the following references to reflect their final RFC assignments:

- draft-ietf-netconf-restconf
- draft-ietf-netconf-rfc5539bis
- draft-ietf-netconf-server-model

Artwork in this document contains placeholder values for ports pending IANA assignment from "draft-ietf-netconf-call-home". Please apply the following replacements:

- "PORT-X" --> the assigned port value for "netconf-ch-ssh"
- "PORT-Y" --> the assigned port value for "netconf-ch-tls"

- "PORT-Z" --> the assigned port value for "restconf-ch-tls"

The following two Appendix sections are to be removed prior to publication:

- Appendix A. Change Log

- Appendix B. Open Issues

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This RFC presents NETCONF Call Home and RESTCONF Call Home, which enable a NETCONF or RESTCONF server to initiate a secure connection to a NETCONF or RESTCONF client respectively.

NETCONF Call Home supports both of the secure transports used by the NETCONF protocol [RFC6241], SSH and TLS. The NETCONF protocol's binding to SSH is defined in [RFC6242]. The NETCONF protocol's binding to TLS is defined in [draft-ietf-netconf-rfc5539bis].

RESTCONF Call Home only supports TLS, same as the RESTCONF protocol [draft-ietf-netconf-restconf]. The RESTCONF protocol's binding to TLS is defined in [draft-ietf-netconf-restconf].

The SSH protocol is defined in [RFC4253]. The TLS protocol is defined in [RFC5246]. Both the SSH and TLS protocols are layered on top of the TCP protocol, which is defined in [RFC793].

Both NETCONF Call Home and RESTCONF Call Home preserve all but one of the client/server roles in their respective protocol stacks, as when

compared to standard NETCONF and RESTCONF connections. The one and only role reversal that occurs is at the TCP layer; that is, in which peer is the TCP-client and which is the TCP-server.

For example, a network element is traditionally the TCP-server however, when calling home, the network element becomes the TCP-client. The network element's secure transport layer roles (SSH-server, TLS-server) and its application layer roles (NETCONF-server, RESTCONF-server) both remain the same.

Having consistency in both the secure transport layer (SSH, TLS) and application layer (NETCONF, RESTCONF) roles conveniently enables deployed network management infrastructure to also support call home. For instance, existing certificate chains and user authentication mechanisms are unaffected by call home.

1.1. Motivation

Call home is generally useful for both the initial deployment and on-going management of networking elements. Here are some scenarios enabled by call home:

- The network element may proactively call home after being powered on for the first time in order to register itself with its management system.

- The network element may access the network in a way that dynamically assigns it an IP address, but does not register its assigned IP address to a mapping service.

- The network element may be deployed behind a firewall that implements network address translation (NAT) for all internal network IP addresses.

- The network element may be deployed behind a firewall that doesn't allow any management access to the internal network.

- The network element may be configured in "stealth mode" and thus doesn't have any open ports for the management system to connect to.

- The operator may prefer to have network elements initiate management connections, believing it is easier to secure one open-port in the data center than to have an open port on each network element in the network.

1.2. Requirements 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 RFC 2119 [RFC2119].

1.3. Applicability Statement

The techniques described in this document are suitable for network management scenarios such as the ones described in Section 1.1. However, these techniques are only defined for NETCONF Call Home and RESTCONF Call Home, as described in this document.

The reason for this restriction is that different protocols have different security assumptions. The NETCONF and RESTCONF protocols require clients and servers to verify the identity of the other party before starting the NETCONF/RESTCONF protocol (section 2.2 of [RFC6241] and sections 2.4 and 2.5 of [draft-ietf-netconf-restconf]).

This contrasts with the base SSH and TLS protocols, which do not require programmatic verification of the other party (section 9.3.4 of [RFC4251], section 4 of [RFC4252], and section 7.3 of [RFC5246]). In such circumstances, allowing the SSH/TLS server to contact the SSH/TLS client would open new vulnerabilities. Any use of call home with SSH/TLS for purposes other than NETCONF or RESTCONF will need a thorough, contextual security analysis.

1.4. Update to RFC 4253

This document updates the SSH Transport Layer Protocol [RFC4253] only in removing the statement "The client initiates the connection" made in Section 4 (Connection Setup). Assuming the reference to client means "SSH client" and the reference to connection means "TCP connection", this statement doesn't hold true in call home, where the network element is the SSH server and yet still initiates the TCP connection. Security implications related to this change are discussed in Security Considerations (Section 4).

2. The NETCONF or RESTCONF Client

2.1. Protocol Operation

C1 The NETCONF/RESTCONF client listens for TCP connection requests from NETCONF/RESTCONF servers. The client SHOULD listen for connections on the IANA-assigned ports defined in section Section 5, but MAY be configured to use a non-default port.

C2 The NETCONF/RESTCONF client accepts an incoming TCP connection request and a TCP connection is established.

C3 Using this TCP connection, the NETCONF/RESTCONF client MUST immediately start either the SSH-client [RFC4253] or the TLS-client [RFC5246] protocol. For example, assuming the use of the IANA-assigned ports, the SSH-client protocol is started when the connection is accepted on port PORT-X and the TLS-client protocol is started when the connection is accepted on either port PORT-Y or PORT-Z.

C4 If using TLS, the NETCONF/RESTCONF client MUST advertise "peer_allowed_to_send", as defined by [RFC6520]. This is required so NETCONF/RESTCONF servers can depend on it being there for call home connections, when keep-alives are needed the most.

C5 As part of establishing an SSH or TLS connection, the NETCONF/RESTCONF client MUST validate both the server's credentials and identity (e.g., SSH host key, TLS certificate). The client first validates the server's credentials, either via previously trusted validation data (e.g., pinning, using the entire host key or certificate as the lookup key) or via certificate path validation to a preconfigured trust anchor. The identity MAY then be derived from the pinned data or extracted from the server's certificate (e.g., SubjectAltName). Finally, the client MUST validate the server's identity is one that it was expecting to connect to it.

C6 Once the SSH or TLS connection is established, the NETCONF/RESTCONF client MUST immediately start using either the NETCONF-client [RFC6241] or RESTCONF-client [draft-ietf-netconf-restconf] protocol. Assuming the use of the IANA-assigned ports, the NETCONF-client protocol is started when the connection is accepted on either port PORT-X or PORT-Y and the RESTCONF-client protocol is started when the connection is accepted on port PORT-Z.

2.2. Configuration Data Model

How to configure a NETCONF or RESTCONF client is outside the scope of this document. This includes configuration that might be used to enable listening for call home connections, configuring trust anchors, and/or pinning the identities for expected connections.

3. The NETCONF or RESTCONF Server

3.1. Protocol Operation

S1 The NETCONF/RESTCONF server initiates a TCP connection request to the NETCONF/RESTCONF client. The server SHOULD connect to one of the IANA-assigned ports defined in section Section 5, but MAY be configured to use a non-default port. Using the IANA-assigned ports, the server connects to port PORT-X for NETCONF over SSH, port PORT-Y for NETCONF over TLS, and port PORT-Z for RESTCONF over TLS.

S2 The TCP connection request is accepted and a TCP connection is established.

S3 Using this TCP connection, the NETCONF/RESTCONF server MUST immediately start using either the SSH-server [RFC4253] or the TLS-server [RFC5246] protocol, depending on how it is configured. For example, assuming the use of the IANA-assigned ports, the SSH-server protocol is used after connecting to the remote port PORT-X and the TLS-server protocol is used after connecting to one of the remote ports PORT-Y or PORT-Z.

S4 Once the SSH or TLS connection is established, the NETCONF/ RESTCONF server MUST immediately start using either the NETCONF-server [RFC6241] or RESTCONF-server [draft-ietf-netconf-restconf] protocol, depending on how it is configured. Assuming the use of the IANA-assigned ports, the NETCONF-server protocol is used after connecting to remote port PORT-X or PORT-Y, and the RESTCONF-server protocol is used after connecting to remote port PORT-Z.

S5 If a persistent connection is desired, the NETCONF/RESTCONF server, as the connection initiator, SHOULD actively test the aliveness of the connection using a keep-alive mechanism. For TLS based connections, the NETCONF/RESTCONF server SHOULD send HeartbeatRequest messages, as defined by [RFC6520]. For SSH based connections, per section 4 of [RFC4254], the NETCONF/ RESTCONF server SHOULD send a SSH_MSG_GLOBAL_REQUEST message with the purposely nonexistent "request name" value "keepalive@ietf.org" and the "want reply" value set to '1'.

3.2. Configuration Data Model

How to configure a NETCONF or RESTCONF server is outside the scope of this document. This includes configuration that might be used to specify hostnames, IP addresses, ports, algorithms, or other relevant parameters.

A YANG [RFC6020] model for configuring both NETCONF and RESTCONF servers, including call home, is provided in [draft-ietf-netconf-server-model].

4. Security Considerations

The security considerations described in [RFC6242] and [draft-ietf-netconf-rfc5539bis], and by extension [RFC4253], [RFC5246], and [draft-ietf-netconf-restconf] apply here as well.

This RFC deviates from standard SSH and TLS usage by having the SSH/TLS server initiate the underlying TCP connection. This reversal is incongruous with [RFC4253], which says "the client initiates the connection" and also [RFC6125], which says "the client MUST construct a list of acceptable reference identifiers, and MUST do so independently of the identifiers presented by the service." To account for these variances, this RFC requires that the NETCONF/RESTCONF client be able to 1) validate the SSH host key via pinning and/or PKI, 2) extract an identity from the server's SSH host-key or TLS certificate, and 3) validate that the identity is one that the client was expecting to connect to it.

An attacker could launch a denial of service (DoS) attack on the NETCONF/RESTCONF client by having it perform computationally expensive operations, before deducing that the attacker doesn't posses a valid key. This is no different than any secured service and all common precautions apply (e.g., blacklisting the source address after a set number of unsuccessful login attempts).

5. IANA Considerations

This RFC requests that IANA assigns three TCP port numbers in the "Registered Port Numbers" range with the service names "netconf-ch-ssh", "netconf-ch-tls", and "restconf-ch-tls". These ports will be the default ports for NETCONF Call Home and RESTCONF Call Home protocols. Below is the registration template following the rules in [RFC6335].

```
   Service Name:           netconf-ch-ssh
   Transport Protocol(s):  TCP
   Assignee:               IESG <iesg@ietf.org>
   Contact:                IETF Chair <chair@ietf.org>
   Description:            NETCONF Call Home (SSH)
   Reference:              RFC XXXX
   Port Number:            PORT-X

   Service Name:           netconf-ch-tls
   Transport Protocol(s):  TCP
   Assignee:               IESG <iesg@ietf.org>
   Contact:                IETF Chair <chair@ietf.org>
   Description:            NETCONF Call Home (TLS)
   Reference:              RFC XXXX
   Port Number:            PORT-Y

   Service Name:           restconf-ch-tls
   Transport Protocol(s):  TCP
   Assignee:               IESG <iesg@ietf.org>
   Contact:                IETF Chair <chair@ietf.org>
   Description:            RESTCONF Call Home (TLS)
   Reference:              RFC XXXX
   Port Number:            PORT-Z
```

6. Acknowledgements

The author would like to thank the following for lively discussions on list and in the halls (ordered by last name): Andy Bierman, Martin Bjorklund, Mehmet Ersue, Wes Hardaker, Stephen Hanna, David Harrington, Jeffrey Hutzelman, Radek Krejci, Alan Luchuk, Mouse, Russ Mundy, Tom Petch, Juergen Schoenwaelder, Peter Saint-Andre, Joe Touch, Hannes Tschofenig, Sean Turner, and Bert Wijnen.

7. References

7.1. Normative References

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

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

[RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.

[RFC4254] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Connection Protocol", RFC 4254, January 2006.

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

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, February 2012.

[RFC793] Postel, J., "TRANSMISSION CONTROL PROTOCOL", STD 7, September 1981, <https://www.ietf.org/rfc/rfc793.txt>.

[draft-ietf-netconf-restconf] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ieft-netconf-restconf-04 (work in progress), 2014, <https://tools.ietf.org/html/draft-ietf-netconf-restconf>.

[draft-ietf-netconf-rfc5539bis]
Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", draft-ietf-netconf-rfc5539bis-10 (work in progress), April 2015, <https://tools.ietf.org/html/draft-ietf-netconf-rfc5539bis>.

7.2. Informative References

[draft-ietf-netconf-server-model]
Watsen, K. and J. Schoenwaelder, "NETCONF Server Configuration Model", 2014, <http://tools.ietf.org/html/draft-ietf-netconf-server-model>.

Appendix A. Change Log

A.1. 00 to 01

- The term "TCP connection" is now used throughout.
- The terms "network element" and "management system" are now only used in the Motivation section.
- Restructured doc a little to create an Introduction section.
- Fixed reference in Applicability Statement so it would work equally well for SSH and TLS.
- Fixed reported odd wording and three references.

A.2. 01 to 02

- Added call home support for the RESTCONF protocol.
- Fixed paragraph 3 of Security Considerations to equally apply to the TLS protocol.

A.3. 02 to 03

- Tried to improve readability (issue #6)
- Removed "FIXME" in section 1.3 (issue #7)
- Added RFC Editor notes (issue #8)
- Removed "TCP session" term (issue #9)
- Improved language for usage of IANA-assigned ports (issue #10)

A.4. 03 to 04

- Replaced "verify credentials" with "verify identity" (issue #11)

A.5. 04 to 05

- Applied many suggestions from WGLC
- Removed essay like "Server Identification and Verification" section
- Added text about keep-alives

- Added Configuration Data Model section for client protocol

- Improved Security Considerations section

Appendix B. Open Issues

All issues with this draft are tracked using GitHub issues. Please see: https://github.com/netconf-wg/call-home/issues to see currently opened issues.

Author's Address

Kent Watsen
Juniper Networks

EMail: kwatsen@juniper.net