                NETCONF Call Home and RESTCONF Call Home

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.

Status of This Memo

   This is an Internet Standards Track document.

   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). Further information on
   Internet Standards is available in 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
   http://www.rfc-editor.org/info/rfc8071.

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
   Network Configuration Protocol (NETCONF) [RFC6241], Secure Shell
   (SSH), and Transport Layer Security (TLS). The NETCONF protocol's
   binding to SSH is defined in [RFC6242]. The NETCONF protocol's
   binding to TLS is defined in [RFC7589].

   RESTCONF Call Home only supports TLS, the same as the RESTCONF
   protocol [RFC8040]. The RESTCONF protocol's binding to TLS is
   defined in [RFC8040].

   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
   compared to client-initiated NETCONF and RESTCONF connections. The
   one and only role reversal that occurs is at the TCP layer; that is,
   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 initially assumes the
role of the TCP client. The network element's secure transport-layer
roles (SSH server, TLS server) and its application-layer roles
(NETCONF server, RESTCONF server) all 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 support call home also.
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
ongoing management of networking elements. Here are some scenarios
enabled by call home:

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

o 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 (e.g., dynamic DNS).

o The network element may be deployed behind a firewall that
   implements Network Address Translation (NAT) for all internal
   network IP addresses.

o The network element may be deployed behind a firewall that does
   not allow any management access to the internal network.

o The network element may be configured in "stealth mode", and thus
   does not have any open ports for the management system to connect
   to.

o 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. This requirement is specified for the NETCONF protocol in
   Section 2.2 of [RFC6241], and is specified for the RESTCONF protocol
   in Sections 2.4 and 2.5 of [RFC8040].

   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 risk assessment. A risk assessment for this RFC
   is in the Security Considerations section (Section 5).

1.4. Relation to RFC 4253

   This document uses the SSH Transport Layer Protocol [RFC4253] with
   the exception that the statement "The client initiates the
   connection" made in Section 4 of RFC 4253 does not apply. Assuming
   the reference to the client means "SSH client" and the reference to
   the 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 Section 5.

1.5. The NETCONF/RESTCONF Convention

   Throughout the remainder of this document, the term "NETCONF/
   RESTCONF" is used as an abbreviation in place of the text "the
   NETCONF or the RESTCONF". The NETCONF/RESTCONF abbreviation is not
   intended to require or to imply that a client or server must
   implement both the NETCONF standard and the RESTCONF standard.

2. Solution Overview

   The diagram below illustrates call home from a protocol-layering
   perspective:
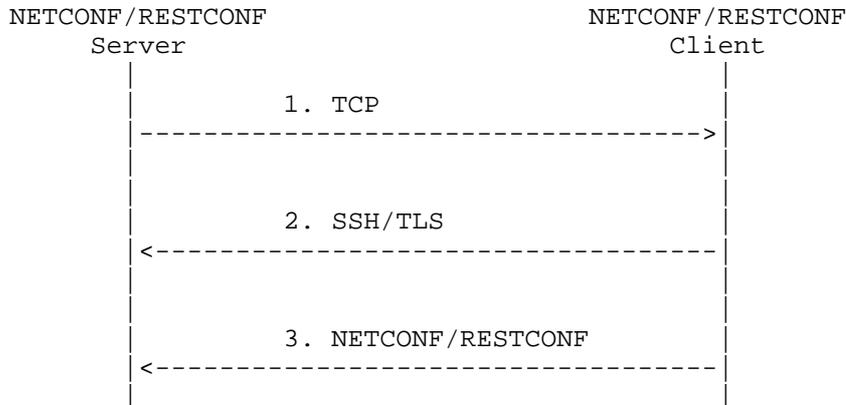
```
           NETCONF/RESTCONF                    NETCONF/RESTCONF
                Server                              Client
                  |                                    |
                  |            1. TCP                  |
                  |----------------------------------->|
                  |                                    |
                  |                                    |
                  |            2. SSH/TLS              |
                  |<-----------------------------------|
                  |                                    |
                  |                                    |
                  |         3. NETCONF/RESTCONF        |
                  |<-----------------------------------|
                  |                                    |
            Note: Arrows point from the "client" to
               the "server" at each protocol layer.
```

                 Figure 1: Call Home Sequence Diagram

   This diagram makes the following points:

   1. The NETCONF/RESTCONF server begins by initiating a TCP connection
       to the NETCONF/RESTCONF client.

   2. Using this TCP connection, the NETCONF/RESTCONF client initiates
       an SSH/TLS session to the NETCONF/RESTCONF server.

   3. Using this SSH/TLS session, the NETCONF/RESTCONF client initiates
       a NETCONF/RESTCONF session to the NETCONF/RESTCONF server.

3. The NETCONF or RESTCONF Client

   The term "client" is defined in [RFC6241], Section 1.1. In the
   context of network management, the NETCONF/RESTCONF client might be a
   network management system.

3.1. Client Protocol Operation

   C1 The NETCONF/RESTCONF client listens for TCP connection requests
       from NETCONF/RESTCONF servers. The client MUST support accepting
       TCP connections on the IANA-assigned ports defined in Section 6,
       but MAY be configured to listen to a different 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 starts
     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 4334 and the TLS client protocol is started when
     the connection is accepted on either port 4335 or port 4336.

C4   When using TLS, the NETCONF/RESTCONF client MUST advertise
     "peer_allowed_to_send", as defined by [RFC6520]. This is
     required so that 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 the server's presented host key or
     certificate. This validation MAY be accomplished by certificate
     path validation or by comparing the host key or certificate to a
     previously trusted or "pinned" value. If a certificate is
     presented and it contains revocation-checking information, the
     NETCONF/RESTCONF client SHOULD check the revocation status of the
     certificate. If it is determined that a certificate has been
     revoked, the client MUST immediately close the connection.

C6   If certificate path validation is used, the NETCONF/RESTCONF
     client MUST ensure that the presented certificate has a valid
     chain of trust to a preconfigured issuer certificate, and that
     the presented certificate encodes an "identifier" [RFC6125] that
     the client was aware of before the connection attempt. How
     identifiers are encoded in certificates MAY be determined by a
     policy associated with the certificate's issuer. For instance, a
     given issuer may be known to only sign IDevID certificates
     [Std-802.1AR-2009] having a unique identifier (e.g., a serial
     number) in the X.509 certificate's "CommonName" field.

C7   After the server's host key or certificate is validated, the SSH
     or TLS protocol proceeds as normal to establish an SSH or TLS
     connection. When performing client authentication with the
     NETCONF/RESTCONF server, the NETCONF/RESTCONF client MUST only
     use credentials that it had previously associated for the
     NETCONF/RESTCONF server's presented host key or server
     certificate.

   C8   Once the SSH or TLS connection is established, the NETCONF/
        RESTCONF client starts either the NETCONF client [RFC6241] or
        RESTCONF client [RFC8040] protocol. Assuming the use of the
        IANA-assigned ports, the NETCONF client protocol is started when
        the connection is accepted on either port 4334 or port 4335 and
        the RESTCONF client protocol is started when the connection is
        accepted on port 4336.

3.2. Client Configuration Data Model

   How a NETCONF or RESTCONF client is configured is outside the scope
   of this document. For instance, such a configuration might be used
   to enable listening for call home connections, configuring trusted
   certificate issuers, or configuring identifiers for expected
   connections. That said, YANG [RFC7950] data modules for configuring
   NETCONF and RESTCONF clients, including call home, are provided in
   [NETCONF-MODELS] and [RESTCONF-MODELS].

4. The NETCONF or RESTCONF Server

   The term "server" is defined in [RFC6241], Section 1.1. In the
   context of network management, the NETCONF/RESTCONF server might be a
   network element or a device.

4.1. Server Protocol Operation

   S1   The NETCONF/RESTCONF server initiates a TCP connection request to
        the NETCONF/RESTCONF client. The source port may be per local
        policy or randomly assigned by the operating system. The server
        MUST support connecting to one of the IANA-assigned ports defined
        in Section 6, but MAY be configured to connect to a different
        port. Using the IANA-assigned ports, the server connects to port
        4334 for NETCONF over SSH, port 4335 for NETCONF over TLS, and
        port 4336 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 starts
        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 4334 and the
        TLS server protocol is used after connecting to either remote
        port 4335 or remote port 4336.

   S4 As part of establishing the SSH or TLS connection, the NETCONF/
       RESTCONF server will send its host key or certificate to the
       client. If a certificate is sent, the server MUST also send all
       intermediate certificates leading up to a well-known and trusted
       issuer. How to send a list of certificates is defined for SSH in
       [RFC6187], Section 2.1, and for TLS in [RFC5246], Section 7.4.2.

   S5 Establishing an SSH or TLS session requires server authentication
       of client credentials in all cases except with RESTCONF, where
       some client authentication schemes occur after the secure
       transport connection (TLS) has been established. If transport-
       level (SSH or TLS) client authentication is required, and the
       client is unable to successfully authenticate itself to the
       server in an amount of time defined by local policy, the server
       MUST close the connection.

   S6 Once the SSH or TLS connection is established, the NETCONF/
       RESTCONF server starts either the NETCONF server [RFC6241] or
       RESTCONF server [RFC8040] 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
       4334 or remote port 4335, and the RESTCONF server protocol is
       used after connecting to remote port 4336.

   S7 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 server SHOULD
       send an SSH_MSG_GLOBAL_REQUEST message with a purposely
       nonexistent "request name" value (e.g., keepalive@ietf.org) and
       the "want reply" value set to '1'.

4.2. Server Configuration Data Model

   How a NETCONF or RESTCONF server is configured 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. That said, YANG [RFC7950] data modules for configuring
   NETCONF and RESTCONF servers, including call home, are provided in
   [NETCONF-MODELS] and [RESTCONF-MODELS].

5. Security Considerations

   The security considerations described in [RFC6242] and [RFC7589], and
   by extension [RFC4253], [RFC5246], and [RFC8040] 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."

   Risks associated with these variances are centered around server
   authentication and the inability for clients to compare an
   independently constructed reference identifier to one presented by
   the server. To mitigate against these risks, this RFC requires that
   the NETCONF/RESTCONF client validate the server's SSH host key or
   certificate, by certificate path validation to a preconfigured issuer
   certificate, or by comparing the host key or certificate to a
   previously trusted or "pinned" value. Furthermore, when a
   certificate is used, this RFC requires that the client be able to
   match an identifier encoded in the presented certificate with an
   identifier the client was preconfigured to expect (e.g., a serial
   number).

   For cases when the NETCONF/RESTCONF server presents an X.509
   certificate, NETCONF/RESTCONF clients should ensure that the
   preconfigured issuer certificate used for certificate path validation
   is unique to the manufacturer of the server. That is, the
   certificate should not belong to a third-party certificate authority
   that might issue certificates for more than one manufacturer. This
   is especially important when a client authentication mechanism
   passing a shared secret (e.g., a password) to the server is used.
   Not doing so could otherwise lead to a case where the client sends
   the shared secret to another server that happens to have the same
   identity (e.g., a serial number) as the server the client was
   configured to expect.

   Considerations not associated with server authentication follow next.

   Internet-facing hosts running NETCONF Call Home or RESTCONF Call Home
   will be fingerprinted via scanning tools such as "zmap" [zmap]. Both
   SSH and TLS provide many ways in which a host can be fingerprinted.
   SSH and TLS servers are fairly mature and able to withstand attacks,
   but SSH and TLS clients may not be as robust. Implementers and
   deployments need to ensure that software update mechanisms are
   provided so that vulnerabilities can be fixed in a timely fashion.

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
possess a valid key. For instance, in TLS 1.3 [TLS1.3], the
ClientHello message contains a Key Share value based on an expensive
asymmetric key operation. Common precautions mitigating DoS attacks
are recommended, such as temporarily blacklisting the source address
after a set number of unsuccessful login attempts.

When using call home with the RESTCONF protocol, special care is
required when using some HTTP authentication schemes, especially the
Basic [RFC7617] and Digest [RFC7616] schemes, which convey a shared
secret (e.g., a password). Implementers and deployments should be
sure to review the Security Considerations section in the RFC for any
HTTP client authentication scheme used.

6. IANA Considerations

   IANA has assigned three TCP port numbers in the "User Ports" 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
   Port Number: 4334
   Transport Protocol(s): TCP
   Description: NETCONF Call Home (SSH)
   Assignee: IESG <iesg@ietf.org>
   Contact: IETF Chair <chair@ietf.org>
   Reference: RFC 8071

   Service Name: netconf-ch-tls
   Port Number: 4335
   Transport Protocol(s): TCP
   Description: NETCONF Call Home (TLS)
   Assignee: IESG <iesg@ietf.org>
   Contact: IETF Chair <chair@ietf.org>
   Reference: RFC 8071

   Service Name: restconf-ch-tls
   Port Number: 4336
   Transport Protocol(s): TCP
   Description: RESTCONF Call Home (TLS)
   Assignee: IESG <iesg@ietf.org>
   Contact: IETF Chair <chair@ietf.org>
   Reference: RFC 8071

7. References

7.1. Normative References

   [RFC793]   Postel, J., "Transmission Control Protocol", STD 7,
               RFC 793, DOI 10.17487/RFC0793, September 1981,
               <http://www.rfc-editor.org/info/rfc793>.

   [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>.

   [RFC4251]   Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH)
               Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251,
               January 2006, <http://www.rfc-editor.org/info/rfc4251>.

   [RFC4252]   Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH)
               Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252,
               January 2006, <http://www.rfc-editor.org/info/rfc4252>.

   [RFC4253]   Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH)
               Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253,
               January 2006, <http://www.rfc-editor.org/info/rfc4253>.

   [RFC4254]   Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH)
               Connection Protocol", RFC 4254, DOI 10.17487/RFC4254,
               January 2006, <http://www.rfc-editor.org/info/rfc4254>.

   [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>.

   [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, DOI 10.17487/RFC6125, March
               2011, <http://www.rfc-editor.org/info/rfc6125>.

   [RFC6187]   Igoe, K. and D. Stebila, "X.509v3 Certificates for Secure
               Shell Authentication", RFC 6187, DOI 10.17487/RFC6187,
               March 2011, <http://www.rfc-editor.org/info/rfc6187>.

   [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>.

   [RFC6242]   Wasserman, M., "Using the NETCONF Protocol over Secure
               Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
               <http://www.rfc-editor.org/info/rfc6242>.

   [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, DOI 10.17487/RFC6335, August 2011,
               <http://www.rfc-editor.org/info/rfc6335>.

   [RFC6520]   Seggelmann, R., Tuexen, M., and M. Williams, "Transport
               Layer Security (TLS) and Datagram Transport Layer Security
               (DTLS) Heartbeat Extension", RFC 6520,
               DOI 10.17487/RFC6520, February 2012,
               <http://www.rfc-editor.org/info/rfc6520>.

   [RFC7589]   Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the
               NETCONF Protocol over Transport Layer Security (TLS) with
               Mutual X.509 Authentication", RFC 7589,
               DOI 10.17487/RFC7589, June 2015,
               <http://www.rfc-editor.org/info/rfc7589>.

   [RFC8040]   Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF
               Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,
               <http://www.rfc-editor.org/info/rfc8040>.

7.2. Informative References

   [NETCONF-MODELS]
               Watsen, K., Wu, G., and J. Schoenwaelder, "NETCONF Client
               and Server Models", Work in Progress, draft-ietf-netconf-
               netconf-client-server-01, November 2016.

   [RESTCONF-MODELS]
               Watsen, K. and J. Schoenwaelder, "RESTCONF Client and
               Server Models", Work in Progress draft-ietf-netconf-
               restconf-client-server-01, November 2016.

   [RFC7616]   Shekh-Yusef, R., Ed., Ahrens, D., and S. Bremer, "HTTP
               Digest Access Authentication", RFC 7616,
               DOI 10.17487/RFC7616, September 2015,
               <http://www.rfc-editor.org/info/rfc7616>.

   [RFC7617]   Reschke, J., "The 'Basic' HTTP Authentication Scheme",
               RFC 7617, DOI 10.17487/RFC7617, September 2015,
               <http://www.rfc-editor.org/info/rfc7617>.

   [RFC7950]   Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
               RFC 7950, DOI 10.17487/RFC7950, August 2016,
               <http://www.rfc-editor.org/info/rfc7950>.

   [Std-802.1AR-2009]
               IEEE, "IEEE Standard for Local and metropolitan area
               networks - Secure Device Identity", IEEE Std 802.1AR-2009,
               DOI 10.1109/IEEESTD.2009.5367679, December 2009,
               <http://standards.ieee.org/findstds/
               standard/802.1AR-2009.html>.

   [TLS1.3]   Rescorla, E., "The Transport Layer Security (TLS) Protocol
               Version 1.3", Work in Progress, draft-ietf-tls-tls13-18,
               October 2016.

   [zmap]   Durumeric, Z., Wustrow, E., and J. Halderman, "ZMap: Fast
               Internet-Wide Scanning and its Security Applications",
               22nd Usenix Security Symposium, August 2013,
               <https://zmap.io/paper.html>.

Acknowledgements

Author's Address

   Kent Watsen
   Juniper Networks

   Email: kwatsen@juniper.net