**NETCONF Call Home and RESTCONF Call Home**

이 RFC 8071 문서는 **NETCONF Call Home 및 RESTCONF Call Home**을 정의하며, NETCONF 또는 RESTCONF 서버가 **클라이언트에게 보안 연결을 주도적으로 설정**할 수 있도록 지원합니다.

📌 **핵심 요약**

* **NETCONF Call Home / RESTCONF Call Home**  
  → 기존 클라이언트-서버 모델과 달리, 서버가 먼저 클라이언트에 연결을 시도하는 방식.  
  → 방화벽 내부에 있는 장치가 **외부 클라이언트와 안전하게 통신할 수 있도록 함**.
* **문서 상태**
  + IETF **Standards Track 문서**
  + **IETF 커뮤니티의 검토 및 승인**을 거쳐 공식 표준화됨.
  + RFC 7841에 따라 인터넷 표준에 대한 정보 제공.
* **저작권 및 라이선스**
  + IETF Trust 및 문서 저자가 저작권 보유.
  + **Simplified BSD License** 적용 (소스 코드 재사용 가능, 보증 없음).

이 문서는 **네트워크 장비가 클라이언트로부터의 연결을 기다리는 대신, 직접 클라이언트와 통신할 수 있도록 하는 기법**을 설명하며, 보안 및 네트워크 관리 측면에서 중요한 의미를 가집니다. 🚀

# 목차

[목차 2](#_Toc190183965)

[1. Introduction 2](#_Toc190183966)

[1.1 Motivation 3](#_Toc190183967)

[1.2 Requirements Terminology 4](#_Toc190183968)

[1.3 Applicability Statement 4](#_Toc190183969)

[1.4 Relation to RFC 4253 5](#_Toc190183970)

[1.5 The NETCONF/RESTCONF Convention 5](#_Toc190183971)

[2. Solution Overview 6](#_Toc190183972)

[3. The NETCONF or RESTCONF Client 7](#_Toc190183973)

[3.1 Client Protocol Operation 8](#_Toc190183974)

[3.2 Client Configuration Data Model 9](#_Toc190183975)

[4. The NETCONF or RESTCONF Server 9](#_Toc190183976)

[4.1 Server Protocol Operation 10](#_Toc190183977)

[4.2 Server Configuration Data Model 11](#_Toc190183978)

[5. Security Considerations 11](#_Toc190183979)

[6. IANA Considerations 14](#_Toc190183980)

[7. References 16](#_Toc190183981)

[9.1 Normative References 16](#_Toc190183982)

[9.2 Informative References 17](#_Toc190183983)

[Appendix A. Changes from RFC 4742 17](#_Toc190183984)

# 1. Introduction

이 RFC는 **NETCONF Call Home 및 RESTCONF Call Home**을 소개하며, **NETCONF/RESTCONF 서버가 클라이언트에 먼저 연결을 시도하는 방식**을 설명합니다.

**🔹 1.1 NETCONF 및 RESTCONF Call Home 개요**

✔ **NETCONF Call Home**

* **SSH (**[**RFC 6242**](#_9.1_Normative_References)**) 및 TLS (**[**RFC 7589**](#_13.1_Normative_References)**) 보안 전송을 지원**
* 기존 NETCONF 프로토콜([RFC 6241](#_13.1_Normative_References))을 따름

✔ **RESTCONF Call Home**

* **TLS(RFC 8040)만 지원**
* RESTCONF 프로토콜(RFC 8040)과 동일한 보안 방식 적용

**🔹 1.2 Call Home에서의 클라이언트/서버 역할 변화**

**✅ 기존 방식**: 클라이언트가 서버에 연결을 시도  
**✅ Call Home 방식**: 서버가 클라이언트에 먼저 연결

🚀 **변화된 점**

* **TCP 계층에서의 역할 변화**  
  → **기존**: 네트워크 장비(서버)가 TCP 서버 역할 수행  
  → **Call Home**: 네트워크 장비가 **TCP 클라이언트**로 작동하여 클라이언트와 먼저 연결
* **변하지 않는 점**
  + **SSH/TLS 보안 계층에서 서버 역할 유지**
  + **NETCONF/RESTCONF 프로토콜 계층에서도 서버 역할 유지**
  + **기존 인증서 체계 및 사용자 인증 방식 유지 가능**

**🔹 1.3 Call Home의 장점**

✅ 기존 보안 인프라를 그대로 유지하면서, 방화벽 뒤의 네트워크 장비가 **관리 서버에 안전하게 연결 가능**  
✅ 기존 **인증서 체인 및 사용자 인증 메커니즘에 영향을 주지 않음**

🔥 결론

NETCONF 및 RESTCONF Call Home은 네트워크 장비가 직접 클라이언트(관리 서버)에 안전하게 연결할 수 있도록 설계되었습니다.

이 방식은 기존 인증 인프라를 변경하지 않으면서도 방화벽 내부 장비의 관리 효율성을 높이는 데 기여합니다. 🚀

## 1.1 Motivation

Call Home 기능은 **네트워크 장비의 초기 배포 및 지속적인 관리**를 위해 유용합니다.  
아래는 Call Home이 필요한 주요 시나리오입니다.

**🔹 1️ Call Home을 이용한 자동 등록**

✔ 네트워크 장비가 **최초 전원 ON 시, 자동으로 관리 시스템에 등록**

**🔹 2️ 동적 IP 할당 환경**

✔ 장비가 DHCP 등을 통해 **동적으로 IP를 할당받지만, DNS 등록이 안 되는 경우**  
✔ 정적 IP 없이도 장비가 관리 서버에 먼저 연결 가능

**🔹 3️ 방화벽 및 NAT 환경**

✔ 내부 네트워크의 IP 주소가 \*\*NAT(Network Address Translation)\*\*를 통해 외부로 변환되는 경우  
✔ **외부에서 내부 네트워크로 직접 접속할 수 없을 때, Call Home을 통해 장비가 먼저 연결**

**🔹 4️ 방화벽 제한 환경**

✔ 방화벽 정책이 **외부에서 내부 네트워크로의 접근을 전면 차단**하는 경우  
✔ **Call Home을 사용하면, 장비가 내부에서 외부로 먼저 연결하여 관리 가능**

**🔹 5️ Stealth Mode (은닉 모드) 장비**

✔ 장비가 보안상의 이유로 **모든 관리 포트를 닫아둔 경우**  
✔ Call Home을 통해 **필요할 때만 보안된 연결을 생성하여 관리 가능**

**🔹 6️ 중앙 집중식 보안 관리**

✔ 관리자가 **각 네트워크 장비의 포트를 여는 것보다, 데이터센터의 특정 포트 하나만 오픈하는 것이 보안적으로 유리하다고 판단하는 경우**  
✔ **Call Home을 통해 장비가 먼저 연결을 요청하면, 중앙에서 안전하게 관리 가능**

**🔥 결론**

Call Home은 네트워크 환경이 복잡하거나 보안 제한이 강한 경우에도 **네트워크 장비의 효율적인 관리**를 가능하게 합니다.  
✅ **자동 등록, 동적 IP 환경, 방화벽/NAT 우회, 보안 강화 등 다양한 이점을 제공**합니다. 🚀

## 1.2 Requirements Terminology

이 문서에서 사용된 주요 용어들은 **RFC 2119**에서 정의된 대로 해석해야 합니다.

* **MUST / MUST NOT** → 반드시 따라야 하는 요구 사항 (강제 규칙)
* **REQUIRED / SHALL / SHALL NOT** → 필수적인 요구 사항
* **SHOULD / SHOULD NOT** → 강력히 권장하지만, 특정한 이유가 있을 경우 예외 가능
* **RECOMMENDED** → 권장사항이지만 필수는 아님
* **MAY / OPTIONAL** → 선택 사항, 구현에 따라 다를 수 있음

즉, 이 문서에서 기술된 기능들은 **RFC 2119**의 가이드라인에 따라 **강제 또는 권장 사항으로 적용됨**을 의미합니다. ✅

## 1.3 Applicability Statement

**적용 가능성 명시**

이 문서에서 설명하는 **Call Home 기술**은 **섹션 1.1**에서 제시된 네트워크 관리 시나리오에 적합합니다. 그러나, 이 기술은 **NETCONF Call Home**과 **RESTCONF Call Home**에 대해서만 정의됩니다.

**✅ 제한 사항의 이유**

각 프로토콜은 서로 다른 **보안 가정(Security Assumptions)**을 가지고 있습니다.

* **NETCONF 및 RESTCONF**는 클라이언트와 서버가 **상대방의 신원을 검증해야 한다는 요구 사항**이 있습니다.
  + NETCONF → **RFC 6241 섹션 2.2**
  + RESTCONF → **RFC 8040 섹션 2.4, 2.5**

반면, **SSH 및 TLS 기본 프로토콜**은 상대방의 신원을 **자동 검증할 필요가 없습니다.**

* SSH → **RFC 4251 섹션 9.3.4**, **RFC 4252 섹션 4**
* TLS → **RFC 5246 섹션 7.3**

따라서, **SSH/TLS 서버가 먼저 클라이언트에 접속하는 방식**(Call Home 방식)을 무분별하게 적용하면 **보안 취약점이 발생할 가능성이 큽니다.**

**⚠️ Call Home을 SSH/TLS 기반의 다른 용도로 사용하려면?**

NETCONF 및 RESTCONF 이외의 목적에 **Call Home 기술을 SSH/TLS와 함께 사용하려면**, 반드시 **철저한 보안 위험 평가(Risk Assessment)** 가 필요합니다.

이 문서에서의 보안 고려 사항(Security Considerations)은 **섹션 5**에서 다루고 있습니다. 🚀

## 1.4 Relation to RFC 4253

이 문서는 **SSH Transport Layer Protocol**([RFC 4253](#_13.1_Normative_References))을 기반으로 하지만, 다음 예외가 있습니다.

**🔹 RFC 4253 섹션 4의 "클라이언트가 연결을 시작한다"는 문장 적용 불가**

* **RFC 4253 섹션 4**에서는 **SSH 클라이언트가 TCP 연결을 시작한다**고 명시되어 있습니다.
* 하지만 **Call Home 방식에서는 네트워크 장비(SSH 서버)가 TCP 연결을 시작**합니다.
  + 즉, **TCP 레벨에서는 클라이언트 역할**이지만,
  + **SSH 레벨에서는 서버 역할**을 유지합니다.

이러한 역할 반전(Role Reversal)은 기존의 SSH 프로토콜 보안 모델과 차이가 있으며, **보안 영향(Security Implications)은 섹션 5에서 논의**됩니다. 🚀

## 1.5 The NETCONF/RESTCONF Convention

이 문서에서는 **"NETCONF/RESTCONF"** 라는 용어를 사용하여 **"NETCONF 또는 RESTCONF”를** 의미하는 축약형으로 사용합니다.

🔹 **중요한 점:**

* "NETCONF/RESTCONF"라는 표기가 **NETCONF와 RESTCONF를 모두 구현해야 한다는 의미는 아닙니다.**
* 즉, **클라이언트나 서버가 반드시 두 가지 표준을 모두 지원할 필요는 없습니다.**
* 이는 단순히 **NETCONF 또는 RESTCONF를 의미하는 편리한 표기 방식**입니다. 🚀

# 2. Solution Overview

📌 **Call Home 프로토콜 계층 구조**  
아래 다이어그램은 **Call Home 기능이 작동하는 프로토콜 계층 구조**를 나타냅니다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**💡 다이어그램에서 알 수 있는 주요 내용**

1️ **TCP 연결**

* NETCONF/RESTCONF **서버**가 먼저 **TCP 연결을 시작**합니다.
* 기존 방식과 반대로, 네트워크 장비(서버)가 먼저 "Call Home"을 실행하여 클라이언트와 연결됩니다.

2️ **SSH/TLS 세션 설정**

* TCP 연결이 설정된 후, NETCONF/RESTCONF **클라이언트**가 **SSH 또는 TLS 세션을 시작**합니다.
* 이를 통해 보안이 강화된 연결이 생성됩니다.

3️ **NETCONF/RESTCONF 세션 시작**

* SSH/TLS 세션이 완료된 후, **클라이언트가 NETCONF/RESTCONF 세션을 시작**합니다.
* 이제 클라이언트는 서버와의 안전한 연결을 통해 **네트워크 장비를 구성하거나 상태를 확인**할 수 있습니다.

**🔹 핵심 요점**

✅ Call Home 방식에서는 **서버가 먼저 TCP 연결을 시작**하고, 이후의 보안 세션 및 데이터 세션은 **클라이언트가 주도**합니다.  
✅ **방화벽 뒤에 있는 네트워크 장비**도 Call Home을 사용하면 **중앙 관리 시스템과 안전하게 연결 가능**합니다.  
✅ **SSH/TLS 보안 계층이 유지**되므로 기존 인증 및 보안 체계를 유지하면서도 원격 관리가 가능합니다. 🚀

# 3. The NETCONF or RESTCONF Client

🔹 **"클라이언트(Client)"의 정의**

* RFC 6241 섹션 1.1에 정의된 바와 같이, 클라이언트는 네트워크 관리 시스템(NMS)과 같은 역할을 수행할 수 있습니다.
* Call Home 방식에서는 **클라이언트가 서버의 요청을 기다리는 역할**을 합니다.

## 3.1 Client Protocol Operation

📌 **NETCONF/RESTCONF 클라이언트의 Call Home 처리 순서**

**🔹 C1. TCP 연결 대기**

* 클라이언트는 **NETCONF/RESTCONF 서버의 TCP 연결 요청을 수신할 준비**를 해야 합니다.
* 기본적으로 **IANA에서 할당한 포트(Section 6에서 정의됨)를 사용해야 하며**, 필요 시 다른 포트로도 설정 가능.

**🔹 C2. TCP 연결 수락**

* 클라이언트는 들어오는 TCP 연결 요청을 수락하고, **TCP 연결을 설정**합니다.

**🔹 C3. SSH/TLS 세션 시작**

* 설정된 TCP 연결을 통해 **클라이언트는 SSH 또는 TLS 세션을 시작**합니다.
* 사용되는 포트에 따라 프로토콜이 결정됨:
  + **SSH**는 **포트 4334**에서 시작됨
  + **TLS**는 **포트 4335 또는 4336**에서 시작됨

**🔹 C4. TLS 연결 시 "peer\_allowed\_to\_send" 설정**

* 클라이언트는 **RFC 6520에서 정의된 "peer\_allowed\_to\_send"를 광고해야 함**.
* 이유: Call Home 연결에서 **서버가 Keep-Alive 메시지를 보낼 수 있도록 보장**하기 위해 필요함.

**🔹 C5. 서버 인증서 검증**

* SSH 또는 TLS 연결 설정 중, **클라이언트는 서버가 제시한 Host Key 또는 인증서를 검증해야 함**.
* 검증 방법:
  + **인증서 체인 검증**
  + **사전에 신뢰한(또는 "고정된") 값과 비교**
* **인증서가 폐기(Revoked)된 것으로 확인되면 즉시 연결을 종료해야 함**.

**🔹 C6. 인증서 경로 검증**

* **서버가 제시한 인증서가 신뢰할 수 있는 발급자(issuer)로부터 유효한 체인인지 확인해야 함**.
* 인증서에는 **사전에 알고 있던 "식별자(identifier)"가 포함되어야 함**.
* 예시:
  + 특정 인증 기관(CA)은 **X.509 인증서의 "CommonName" 필드에 장비의 고유 ID(예: 시리얼 번호)를 포함**하도록 정책을 설정할 수 있음.

**🔹 C7. SSH/TLS 연결 완료 후 클라이언트 인증**

* 클라이언트는 서버 인증이 완료된 후, **정상적인 SSH/TLS 연결을 설정**함.
* 클라이언트가 서버에 인증할 때 **사전에 저장된 해당 서버의 Host Key 또는 인증서에 연관된 자격 증명만 사용해야 함**.

**🔹 C8. NETCONF/RESTCONF 세션 시작**

* SSH 또는 TLS 세션이 설정된 후, **클라이언트는 NETCONF 또는 RESTCONF 세션을 시작**함.
* 포트 매핑:
  + **NETCONF 클라이언트 → 포트 4334 또는 4335**
  + **RESTCONF 클라이언트 → 포트 4336**

## 3.2 Client Configuration Data Model

📌 **클라이언트 설정 방법은 본 문서의 범위를 벗어남.**

* Call Home 기능 활성화, 신뢰할 수 있는 인증 기관 구성, 특정 연결을 위한 식별자 설정 등의 설정이 포함될 수 있음.
* **YANG 데이터 모델**을 사용하여 **NETCONF 및 RESTCONF 클라이언트(및 Call Home 기능)를 구성 가능**.
  + 관련 문서: **[**[**NETCONF-MODELS**](#_9.2_Informative_References)**], [**[**RESTCONF-MODELS**](#_9.2_Informative_References)**]**

**✅ 정리**

1️ 클라이언트는 **TCP 포트를 열어 서버의 Call Home 연결 요청을 수신**함.  
2️ 연결이 성립되면, **SSH/TLS 세션을 시작하고 보안 인증을 수행**함.  
3️ 서버의 **Host Key 또는 인증서를 검증**해야 하며, 인증서가 폐기된 경우 즉시 연결 종료.  
4️ SSH/TLS 연결이 완료되면, **NETCONF 또는 RESTCONF 세션을 시작**함.  
5️ **YANG 모델을 사용하여 클라이언트의 Call Home 설정을 구성할 수 있음**.

📌 **Call Home 방식의 핵심 장점**  
✅ **서버가 방화벽 뒤에 있어도 관리 가능**  
✅ **클라이언트가 주도하는 보안 연결 유지**  
✅ **기존의 인증 시스템과 연계 가능** 🚀

# 4. The NETCONF or RESTCONF Server

🔹 **"서버(Server)"의 정의**

* RFC 6241 섹션 1.1에 정의된 바와 같이, **NETCONF/RESTCONF 서버는 네트워크 요소 또는 장비**일 수 있음.
* Call Home 방식에서는 **서버가 먼저 TCP 연결을 시작**하지만, SSH/TLS 및 NETCONF/RESTCONF 프로토콜에서는 서버 역할을 유지.

## 4.1 Server Protocol Operation

📌 **NETCONF/RESTCONF 서버의 Call Home 처리 순서**

**🔹 S1. TCP 연결 시작**

* **서버가 먼저 NETCONF/RESTCONF 클라이언트로 TCP 연결을 요청**함.
* **출발지 포트는 정책에 따라 설정되거나 OS에서 임의로 할당될 수 있음**.
* 서버는 **IANA에서 할당한 포트(Section 6)를 사용해야 하며**, 필요 시 다른 포트로도 설정 가능.
* IANA 할당 포트:
  + **NETCONF over SSH → 포트 4334**
  + **NETCONF over TLS → 포트 4335**
  + **RESTCONF over TLS → 포트 4336**

**🔹 S2. TCP 연결 설정**

* 클라이언트가 TCP 연결 요청을 수락하면 **TCP 연결이 확립됨**.

**🔹 S3. SSH 또는 TLS 세션 시작**

* 서버는 **설정된 TCP 연결을 통해 SSH 또는 TLS 세션을 시작**함.
* 클라이언트가 연결한 포트에 따라 프로토콜이 결정됨:
  + **포트 4334 → SSH 서버 프로토콜 사용**
  + **포트 4335 또는 4336 → TLS 서버 프로토콜 사용**

**🔹 S4. 서버 인증서 전송**

* **SSH/TLS 연결을 설정하는 과정에서 서버는 자신의 Host Key 또는 인증서를 클라이언트로 전송**함.
* **TLS 사용 시**, 서버는 **신뢰할 수 있는 루트 인증 기관(CA)까지의 모든 중간 인증서를 함께 전송해야 함**.
  + **SSH 인증서 전송 방식** → [RFC6187] Section 2.1
  + **TLS 인증서 전송 방식** → [RFC5246] Section 7.4.2

**🔹 S5. 클라이언트 인증 처리**

* **SSH/TLS 인증 과정에서 서버는 클라이언트의 인증을 수행해야 함**.
* **RESTCONF의 경우**, 일부 인증 방식은 **TLS 연결 후 수행될 수도 있음**.
* **서버는 일정 시간 내에 클라이언트가 인증을 완료하지 못하면 연결을 종료해야 함**(정책에 따름).

**🔹 S6. NETCONF/RESTCONF 세션 시작**

* SSH 또는 TLS 연결이 완료되면, **서버는 NETCONF 또는 RESTCONF 세션을 시작**함.
* 클라이언트가 연결한 포트에 따라 프로토콜이 결정됨:
  + **포트 4334 또는 4335 → NETCONF 서버 프로토콜 사용**
  + **포트 4336 → RESTCONF 서버 프로토콜 사용**

**🔹 S7. 지속적인 연결 유지**

* **Call Home 방식에서는 서버가 연결을 유지하려는 역할을 수행함**.
* **서버는 연결 유지 여부를 적극적으로 확인해야 함** → Keep-Alive 메커니즘 사용:
  + **TLS 연결의 경우**: [RFC6520]에 정의된 **HeartbeatRequest 메시지 전송**
  + **SSH 연결의 경우**: [[RFC4254](#_13.1_Normative_References)] Section 4에 따라 **SSH\_MSG\_GLOBAL\_REQUEST 메시지 전송**
    - 요청 이름을 **"keepalive@ietf.org"** 같은 존재하지 않는 값으로 설정
    - "want reply" 값을 '1'로 설정하여 응답 요구

## 4.2 Server Configuration Data Model

📌 **서버의 설정 방법은 본 문서의 범위를 벗어남.**

* Call Home 연결을 위한 **호스트명, IP 주소, 포트, 알고리즘 등**의 설정 포함 가능.
* **YANG 데이터 모델을 사용하여 NETCONF 및 RESTCONF 서버(및 Call Home 기능)를 구성 가능**.
  + 관련 문서: **[**[**NETCONF-MODELS**](#_9.2_Informative_References)**], [RESTCONF-MODELS]**

**✅ 정리**

1️ 서버는 **먼저 TCP 연결을 시작**하고, **IANA 할당 포트(4334~4336) 중 하나로 클라이언트에 연결**함.  
2️ TCP 연결이 설정되면, **서버는 SSH 또는 TLS 세션을 시작**함.  
3️ 서버는 **자신의 인증서 또는 Host Key를 클라이언트에 제공**하며, 클라이언트 인증을 요구할 수도 있음.  
4️ SSH/TLS 연결이 완료되면, **NETCONF 또는 RESTCONF 세션을 시작**함.  
5️ **서버는 연결을 유지하기 위해 주기적으로 Keep-Alive 메시지를 전송해야 함**.  
6️ **YANG 모델을 사용하여 서버의 Call Home 설정을 구성할 수 있음**.

📌 **Call Home 방식의 핵심 장점**  
✅ **네트워크 요소가 방화벽 뒤에 있어도 관리 가능**  
✅ **서버가 먼저 연결을 시작함으로써 네트워크 정책을 유연하게 적용 가능**  
✅ **클라이언트가 신뢰할 수 있는 서버와만 통신하도록 보안 강화 가능** 🚀

# 5. Security Considerations

🔹 **NETCONF/RESTCONF Call Home 방식의 보안 요소**

* **기존 RFC들과의 연관성:**
  + [[RFC6242](#_13.1_Normative_References)] (NETCONF over SSH)
  + [[RFC7589](#_13.1_Normative_References)] (NETCONF over TLS)
  + [[RFC4253](#_13.1_Normative_References)] (SSH)
  + [[RFC5246](#_13.1_Normative_References)] (TLS 1.2)
  + [RFC8040] (RESTCONF)
  + 본 RFC도 **위 RFC들의 보안 권고사항을 준수해야 함**.

**📌 5.1. 일반적인 SSH/TLS 연결과의 차이점 및 위험 요소**

✅ **Call Home 방식의 특이점**

* 일반적으로 **SSH/TLS 클라이언트가 TCP 연결을 시작**하지만,  
  **Call Home 방식에서는 서버가 TCP 연결을 시작**한다.
* [**RFC4253**](#_13.1_Normative_References) **(SSH) 및** [**RFC6125**](#_13.1_Normative_References) **(TLS 서버 인증)** 표준과 충돌하는 부분이 있음.
  + 📌 **RFC4253:** "클라이언트가 연결을 시작해야 한다."
  + 📌 **RFC6125:** "클라이언트는 독립적으로 검증할 식별자를 생성해야 한다."

🔺 **위험 요소:**  
1️ **클라이언트가 서버의 인증서를 독립적으로 비교할 수 없음** → "중간자 공격" 위험  
2️ **신뢰할 수 없는 서버에 대한 연결 위험**

✅ **위험 완화 조치 (Mitigation)**

* **클라이언트는 서버의 SSH Host Key 또는 인증서를 철저히 검증해야 함.**
  + **방법 ①:** 사전에 신뢰할 수 있는 인증서 발급 기관(issuer)과의 경로 검증 (Certificate Path Validation)
  + **방법 ②:** 클라이언트가 **사전에 저장해 둔 Host Key 또는 인증서(핀닝, Pinning)와 비교**
* **서버 인증서 사용 시, 사전 정의된 ID(예: 장비 일련번호)와 비교 필수!**

**📌 5.2. X.509 인증서 사용 시 추가 고려 사항**

✅ **서버가 X.509 인증서를 제공할 때 추가적인 주의 필요!**

* 클라이언트는 **사전에 신뢰할 수 있는 발급 기관(issuer) 목록을 설정해야 함.**
* **제3자 인증기관(CA)에서 발급된 인증서는 사용하지 않는 것이 좋음.**
  + 이유: 동일한 CA에서 여러 제조사의 장비 인증서를 발급할 가능성이 있음.
* **공유 비밀번호(Shared Secret)를 사용하는 인증 방식은 더 신중해야 함!**
  + 예: 클라이언트가 공격자의 서버에 잘못된 인증 정보를 제공할 수 있음.

🔺 **문제 발생 가능 시나리오:**

* 클라이언트는 **서버 A의 일련번호(Serial Number) X를 신뢰하도록 설정**함.
* 공격자가 **동일한 Serial Number(X)를 가진 서버 B를 생성**하여 가짜 인증서를 사용.
* 클라이언트가 서버 B를 신뢰하여 **공유 비밀번호(패스워드 등)를 노출할 가능성** 존재!

✅ **해결책**

* **클라이언트는 인증서 발급 기관이 제조사 전용(Unique to Manufacturer)인지 확인**해야 함.
* **제3자 CA를 사용하는 경우, 클라이언트는 추가적인 검증 절차를 도입해야 함.**

**📌 5.3. 인터넷 기반 Call Home 서버의 보안 위협**

🚨 **공격자들이 Call Home 서버를 탐지할 가능성 존재!**

* 인터넷에 노출된 Call Home 서버는 **zmap** 같은 스캐닝 도구를 통해 탐색될 수 있음.
* **SSH/TLS 서버는 성숙한 보안 기술을 사용하지만, 클라이언트 보안이 상대적으로 약할 수 있음.**
* 🔺 **보안 패치 및 업데이트 시스템이 필수!**
  + 운영자는 **취약점 발생 시 빠른 업데이트가 가능하도록 유지보수 체계를 갖춰야 함.**

**📌 5.4. 서비스 거부(DoS) 공격 위험 및 대응 방안**

🚨 **공격자가 클라이언트에 과부하를 유발할 수 있음.**

* 공격자는 **클라이언트가 인증을 처리하기 전에 연산 집약적인 작업을 수행하도록 유도 가능.**
* 예: **TLS 1.3의 ClientHello 메시지는 Key Share 값을 포함** → 연산 비용이 큼!

✅ **DoS 완화 조치 (Mitigation)**  
1️ **로그인 실패 시, 일정 횟수 초과하면 IP 차단 (Rate Limiting & Blacklisting)**  
2️ **비정상적인 접속 요청 탐지 및 차단 (Intrusion Detection & Prevention Systems, IDS/IPS)**  
3️ **암호화 연산 부하가 높은 알고리즘을 제한적으로 사용**  
4️ **SSH 및 TLS의 보안 패치 적용 필수!**

**📌 5.5. RESTCONF 사용 시 HTTP 인증 주의사항**

🚨 **HTTP 기반의 인증 방식은 추가적인 보안 위험이 있음.**

* 특히 **Basic 인증([RFC7617]) 및 Digest 인증([RFC7616])은 보안 취약점이 존재!**
  + **패스워드를 평문(Plaintext)으로 전송할 가능성**이 높음.
* **공격자가 중간에서 HTTP 요청을 가로채거나 변조할 가능성 존재!**

✅ **해결책**  
1️ **TLS를 항상 사용하여 트래픽을 암호화**  
2️ **OAuth 2.0 또는 mTLS (Mutual TLS) 같은 강력한 인증 방식 적용**  
3️ **RFC 문서의 HTTP 인증 관련 보안 고려 사항을 반드시 준수**

**✅ 결론 (Summary)**

**Call Home 방식은 기존 SSH/TLS 클라이언트-서버 모델과 반대되는 구조이므로, 보안에 대한 추가적인 고려가 필요함!**

📌 **위험 요소 & 대응 방안 정리**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

🚀 **Call Home 보안을 강화하는 방법:**  
✅ **서버 인증서의 신뢰성을 철저히 검증 (Pinned Certificates, Unique Issuers)**  
✅ **인터넷 노출 최소화 및 방화벽 규칙 강화**  
✅ **SSH/TLS 클라이언트 보안 업데이트 체계 수립**  
✅ **DoS 공격 방어책 적용 (Rate Limiting, IDS, 보안 알고리즘 최적화)**  
✅ **RESTCONF의 경우 OAuth 2.0 등의 강력한 인증 방식 사용**

📌 **적절한 보안 조치를 적용하면 Call Home 기능은 안전한 원격 네트워크 관리를 가능하게 함!** 🚀

# 6. IANA Considerations

✅ **NETCONF 및 RESTCONF Call Home을 위한 기본 포트가 IANA에 등록됨.**  
✅ **IANA(User Ports Range)에서 할당된 공식 포트 번호:**

📌 IANA 등록된 포트 정보

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**📌 포트 등록 정보 (IANA 등록 템플릿)**

**1️ NETCONF Call Home (SSH)**

* **서비스 이름:** netconf-ch-ssh
* **포트 번호:** 4334
* **전송 프로토콜:** TCP
* **설명:** NETCONF Call Home (SSH)
* **담당자:** IESG <iesg@ietf.org>
* **연락처:** IETF Chair <chair@ietf.org>
* **참조 문서:** RFC 8071

**2️ NETCONF Call Home (TLS)**

* **서비스 이름:** netconf-ch-tls
* **포트 번호:** 4335
* **전송 프로토콜:** TCP
* **설명:** NETCONF Call Home (TLS)
* **담당자:** IESG <iesg@ietf.org>
* **연락처:** IETF Chair <chair@ietf.org>
* **참조 문서:** RFC 8071

**3️ RESTCONF Call Home (TLS)**

* **서비스 이름:** restconf-ch-tls
* **포트 번호:** 4336
* **전송 프로토콜:** TCP
* **설명:** RESTCONF Call Home (TLS)
* **담당자:** IESG <iesg@ietf.org>
* **연락처:** IETF Chair <chair@ietf.org>
* **참조 문서:** RFC 8071

**📌 결론 (Summary)**

🔹 **IANA가 공식적으로 Call Home 기능을 위한 포트를 지정함.**  
🔹 **기본적으로 IANA 할당 포트(4334, 4335, 4336)를 사용해야 하지만, 필요하면 다른 포트로 변경 가능.**  
🔹 **IANA 등록 사항은 RFC 8071을 기반으로 관리됨.**

📌 **NETCONF 및 RESTCONF Call Home을 설정할 때, 해당 포트를 방화벽 및 네트워크 정책에서 허용해야 함!** 🚀

# 7. References

## 9.1 Normative References

[RFC 793](https://datatracker.ietf.org/doc/html/rfc793) Transmission Control Protocol

[RFC 2119](https://datatracker.ietf.org/doc/html/rfc2119) Key words for use in RFCs to Indicate Requirement Levels

[RFC 4251](https://datatracker.ietf.org/doc/html/rfc4251) The Secure Shell (SSH) Protocol Architecture

[RFC 4252](https://datatracker.ietf.org/doc/html/rfc4252) The Secure Shell (SSH) Authentication Protocol

[RFC 4253](https://datatracker.ietf.org/doc/html/rfc4253) The Secure Shell (SSH) Transport Layer Protocol

[RFC 4254](https://datatracker.ietf.org/doc/html/rfc4254) The Secure Shell (SSH) Connection Protocol

[RFC 5246](https://datatracker.ietf.org/doc/html/rfc5246) The Transport Layer Security (TLS) Protocol Version 1.2

[RFC 6125](https://datatracker.ietf.org/doc/html/rfc6125) 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 6187](https://datatracker.ietf.org/doc/html/rfc6187) X.509v3 Certificates for Secure Shell Authentication

[RFC 6241](https://datatracker.ietf.org/doc/html/rfc6241) Network Configuration Protocol (NETCONF)

[RFC 6242](https://datatracker.ietf.org/doc/html/rfc6242) Using the NETCONF Protocol over Secure Shell (SSH)

[RFC 6335](https://datatracker.ietf.org/doc/html/rfc6335) Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry

[RFC 6520](https://datatracker.ietf.org/doc/html/rfc6520) Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension

[RFC 7589](https://datatracker.ietf.org/doc/html/rfc7589) Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication

[RFC 8040](https://datatracker.ietf.org/doc/html/rfc8040) RESTCONF Protocol

## 9.2 Informative References

[NETCONF-MODELS](https://datatracker.ietf.org/doc/html/draft-ietf-netconf-netconf-client-server-01) NETCONF Client and Server Models draft-ietf-netconf-netconf-client-server-01

[RESTCONF-MODELS](https://datatracker.ietf.org/doc/html/draft-ietf-netconf-restconf-client-server-01) RESTCONF Client and Server Models draft-ietf-netconf-restconf-client-server-01

[RFC7616](https://datatracker.ietf.org/doc/html/rfc7616) HTTP Digest Access Authentication

[RFC7617](https://datatracker.ietf.org/doc/html/rfc7617) The 'Basic' HTTP Authentication Scheme

[RFC7950](https://datatracker.ietf.org/doc/html/rfc7950) The YANG 1.1 Data Modeling Language

[TLS1.3](https://datatracker.ietf.org/doc/html/draft-ietf-tls-tls13-18) The Transport Layer Security (TLS) Protocol Version 1.3 draft-ietf-tls-tls13-18

# Appendix A. Changes from RFC 4742

이 문서는 **RFC 4742**와 비교하여 다음과 같은 주요 변경 사항을 포함함:

📌 **보안 강화**

* 새로운 **청크 프레이밍 메커니즘**(chunked framing mechanism) 도입하여 **EOM 프레이밍의 보안 문제 해결**.
* \*\*보안 고려 사항(Security Considerations)\*\*을 확장하고, **EOM 관련 문제**에 대한 설명 추가.

📌 **기술적 개선**

* \*\*청크 인코딩(chunked encoding)\*\*의 예시 추가 및 줄바꿈 위치 강조.
* **NETCONF 사용자명(username) 처리 관련 요구 사항**을 **RFC 6241**에 맞춰 수정.
* **"client/server" 및 "manager/agent"** 용어를 각각 \*\*"SSH client/server" 및 "NETCONF client/server"\*\*로 일관되게 변경.
* **"command" 또는 "message" 대신 "operation"** 용어를 일관되게 사용.

📌 **기타 정리**

* **RFC 4742에서 보고된 오류(errata) 반영**. (공식 오류 목록은 [RFC 에디터 웹사이트](http://www.rfc-editor.org)에서 확인 가능)