**YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)**

✅ **YANG은 NETCONF에서 사용할 데이터 모델링 언어**  
✅ **NETCONF 프로토콜을 통한 구성(Configuration) 및 상태(State) 데이터 모델링, 원격 프로시저 호출(RPC), 알림(Notification) 지원**

**1️ 문서 개요 (Abstract)**

🔹 **YANG** = **NETCONF에서 사용되는 데이터 모델링 언어**  
🔹 지원 기능:

* **구성 데이터 (Configuration Data) 모델링**
* **상태 데이터 (State Data) 모델링**
* **NETCONF RPC (Remote Procedure Call) 지원**
* **NETCONF Notification 지원**

📌 **YANG은 네트워크 장비 및 서비스를 정의하는 데이터 모델을 만들기 위해 설계된 언어임!**

**2️ 문서 상태 (Status of This Memo)**

🔹 **RFC 6020은 "Internet Standards Track" 문서**  
🔹 **IETF에서 공식 표준으로 채택**  
🔹 **IETF 커뮤니티의 합의를 통해 승인되었으며, 공개 검토를 거쳐 출판됨**

📌 **인터넷 표준 관련 세부 정보는 RFC 5741에서 확인 가능**

**3️ 저작권 및 라이선스 (Copyright Notice)**

🔹 **2010년 IETF Trust 및 저자가 저작권 보유**  
🔹 **IETF 문서 라이선스 규정을 따름 (BCP 78 및 IETF Trust Legal Provisions 적용)**  
🔹 **해당 문서의 코드 컴포넌트(Code Components)는 "Simplified BSD License" 하에 제공됨**

📌 **RFC 문서는 자유롭게 참조 가능하지만, 특정 변경 및 파생 작업에는 제한이 있을 수 있음**

**🔹 결론 (Summary)**

📌 **YANG은 NETCONF와 함께 사용되는 표준 데이터 모델링 언어**  
📌 **IETF 공식 표준이며, 네트워크 장비 및 서비스의 설정/상태를 모델링하는 데 필수적**  
📌 **라이선스 조항을 준수하여 자유롭게 사용 가능하지만, 일부 제한 사항 존재**

🚀 **NETCONF를 활용한 네트워크 자동화 및 관리에서는 YANG이 필수적인 요소!**

# 목차

[목차 2](#_Toc190187325)

[1. Introduction 2](#_Toc190187326)

[2. Keywords 3](#_Toc190187327)

[3. Terminology 4](#_Toc190187328)

[3.1 Client Protocol Operation 4](#_Toc190187329)

[3.2 Client Configuration Data Model 6](#_Toc190187330)

[4. The NETCONF or RESTCONF Server 6](#_Toc190187331)

[4.1 Server Protocol Operation 6](#_Toc190187332)

[4.2 Server Configuration Data Model 8](#_Toc190187333)

[5. Security Considerations 8](#_Toc190187334)

[6. IANA Considerations 11](#_Toc190187335)

[7. References 13](#_Toc190187336)

[9.1 Normative References 13](#_Toc190187337)

[9.2 Informative References 14](#_Toc190187338)

[Appendix A. Changes from RFC 4742 14](#_Toc190187339)

# 1. Introduction

**📌 YANG 소개 (Introduction)**

✅ **YANG = NETCONF을 위한 데이터 모델링 언어**  
✅ **구성(Configuration) 데이터, 상태(State) 데이터, RPC, 알림(Notification) 모델링 지원**  
✅ **XML 형식으로 데이터 모델 표현**

**1️ YANG이란?**

🔹 **YANG (Yet Another Next Generation) = NETCONF을 위한 데이터 모델링 언어**  
🔹 **NETCONF과 함께 사용되어 네트워크 장비 및 서비스의 데이터를 구조화하고 관리**  
🔹 **지원 기능:**

* **구성 데이터 (Configuration Data) 모델링**
* **상태 데이터 (State Data) 모델링**
* **NETCONF 원격 프로시저 호출 (Remote Procedure Call, RPC) 모델링**
* **NETCONF 알림 (Notification) 모델링**

📌 **YANG은 네트워크 장비 및 서비스의 설정, 모니터링, 운영 자동화를 지원하는 필수적인 요소!**

**2️ YANG과 NETCONF의 관계**

🔹 **YANG은 NETCONF 프로토콜의 "Operations Layer"와 "Content Layer"를 모델링하는 역할**  
🔹 **NETCONF 프로토콜과 함께 동작하며, 네트워크 장비의 설정 및 데이터를 관리하는 데 사용됨**

📌 **NETCONF 구성 프로토콜 (**[**RFC 4741**](#_13.1_Normative_References)**, Section 1.1)에서 NETCONF의 동작 방식과 관계 정의**

**3️ YANG 문서에서 다루는 내용**

📌 **이 문서(RFC 6020)는 다음을 설명:**  
🔹 **YANG의 문법(Syntax) 및 의미(Semantics)**  
🔹 **YANG 모듈에서 정의된 데이터 모델이 XML로 표현되는 방식**  
🔹 **NETCONF 작업을 통해 YANG 데이터 조작하는 방법**

**🔹 결론 (Summary)**

📌 **YANG은 NETCONF과 함께 사용되는 데이터 모델링 언어**  
📌 **네트워크 장비의 설정 및 상태 데이터를 표현하고, NETCONF을 통해 이를 조작할 수 있도록 설계됨**  
📌 **XML 기반 표현을 사용하며, RFC 6020에서는 YANG의 문법과 활용 방법을 설명함**

🚀 **네트워크 자동화 및 구성 관리를 위해 YANG + NETCONF는 필수적인 조합!**

# 2. Keywords

이 문서에서 사용되는 특정 키워드는 \*\*RFC 2119 (BCP 14)\*\*에 따라 해석되어야 합니다.

✅ **필수 요구사항 (강제적 규칙)**

* **MUST / REQUIRED / SHALL** → 반드시 준수해야 함
* **MUST NOT / SHALL NOT** → 절대 허용되지 않음

✅ **권장 사항 (비강제적 규칙)**

* **SHOULD / RECOMMENDED** → 강력히 권장되지만 특정 상황에서 예외 가능
* **SHOULD NOT / NOT RECOMMENDED** → 특별한 이유가 없는 한 사용하지 않아야 함

✅ **선택 사항 (자율적 규칙)**

* **MAY / OPTIONAL** → 선택적 사항, 필요에 따라 사용 가능

📌 **즉, 이 문서에서 위 키워드는 명확한 기술적 요구사항을 정의하는 데 사용됨!**

# 3. Terminology

**📌 YANG에서의 주요 용어들**

* **anyxml**: XML 데이터를 포함할 수 있는 데이터 노드로, 데이터의 구조를 모를 때 사용됩니다.
* **augment**: 기존의 스키마 노드에 새로운 노드를 추가하는 방법입니다.
* **base type**: 파생된 타입의 기본 타입으로, 내장 타입 또는 다른 파생 타입이 될 수 있습니다.
* **built-in type**: YANG에서 정의된 내장 데이터 타입 (예: uint32, string).
* **choice**: 여러 대체 중 하나만 유효한 스키마 노드입니다.
* **configuration data**: 시스템을 초기 상태에서 현재 상태로 변환하기 위해 필요한 데이터를 말합니다.
* **conformance**: 디바이스가 데이터 모델을 얼마나 정확히 따르는지를 나타내는 측정 기준입니다.
* **container**: 데이터 트리에서 하나의 인스턴스만 존재하는 내부 데이터 노드로, 값을 가지지 않지만 자식 노드들을 가질 수 있습니다.
* **data definition statement**: 데이터 노드를 정의하는 명령어로, 다양한 타입이 포함될 수 있습니다.
* **data model**: 데이터가 어떻게 표현되고 접근되는지를 설명하는 모델입니다.
* **data node**: 데이터 트리에서 인스턴스화할 수 있는 스키마 트리의 노드입니다.
* **data tree**: 디바이스에서 설정 및 상태 데이터를 표현하는 트리 구조입니다.
* **derived type**: 내장 타입 또는 다른 파생 타입에서 파생된 데이터 타입입니다.
* **device deviation**: 디바이스가 데이터 모델을 충실히 구현하지 않는 경우입니다.
* **extension**: YANG 명령어에 비-YANG 의미를 부여하는 확장 기능입니다.
* **feature**: 데이터 모델의 일부를 선택적으로 사용할 수 있게 하는 메커니즘입니다.
* **grouping**: 여러 스키마 노드를 재사용 가능하게 정의한 세트입니다.
* **identifier**: YANG 항목들을 이름으로 구별하는 방법입니다.
* **instance identifier**: 데이터 트리에서 특정 노드를 식별하는 방법입니다.
* **interior node**: 리프 노드를 제외한 트리 구조의 노드입니다.
* **leaf**: 하나의 인스턴스만 존재하는 데이터 노드로, 값을 가지며 자식 노드는 없습니다.
* **leaf-list**: 여러 인스턴스를 가지는 값들을 정의하는 데이터 노드입니다.
* **list**: 여러 인스턴스를 가질 수 있는 내부 데이터 노드입니다.
* **module**: YANG 모듈은 NETCONF 기반 작업을 위한 노드 계층을 정의합니다.
* **RPC**: NETCONF 프로토콜에서 원격 프로시저 호출을 의미합니다.
* **RPC operation**: NETCONF 프로토콜 내에서 특정한 원격 프로시저 호출을 의미합니다.
* **schema node**: 스키마 트리 내에서 하나의 노드를 나타냅니다.
* **schema tree**: 모듈 내에서 정의된 노드 계층입니다.
* **state data**: 읽기 전용 상태 정보나 수집된 통계 등의 설정 외 추가 데이터입니다.
* **submodule**: 다른 모듈에 파생된 타입, 그룹화, 데이터 노드 등을 추가하는 부분 모듈 정의입니다.
* **top-level data node**: 모듈이나 서브모듈 문서와 직결된 데이터 노드입니다.
* **uses**: 그룹화된 스키마 노드 세트를 인스턴스화하고 이를 특정 요구에 맞게 조정하는 명령어입니다.

💡 **YANG**의 각 용어들은 데이터 모델링과 관련된 핵심 개념들을 명확히 정의하여 네트워크 구성과 상태 데이터의 표현 및 조작을 지원합니다.

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

[RFC 3023](https://datatracker.ietf.org/doc/html/rfc3023) XML Media Types

[RFC 3629](https://datatracker.ietf.org/doc/html/rfc3629) UTF-8, a transformation format of ISO 10646

[RFC 3688](https://datatracker.ietf.org/doc/html/rfc3688) The IETF XML Registry

[RFC 3986](https://datatracker.ietf.org/doc/html/rfc3986) Uniform Resource Identifier (URI): Generic Syntax

[RFC 4648](https://datatracker.ietf.org/doc/html/rfc4648) The Base16, Base32, and Base64 Data Encodings

[RFC 4741](https://datatracker.ietf.org/doc/html/rfc4741) NETCONF Configuration Protocol

[RFC 5226](https://datatracker.ietf.org/doc/html/rfc5226) Guidelines for Writing an IANA Considerations Section in RFCs

[RFC 5234](https://datatracker.ietf.org/doc/html/rfc5234) Augmented BNF for Syntax Specifications: ABNF

[RFC 5277](https://datatracker.ietf.org/doc/html/rfc5277) NETCONF Event Notifications

[XML-NAMES](https://www.w3.org/TR/2009/REC-xml-names-20091208/) Namespaces in XML 1.0 (Third Edition)

[XPATH](https://www.w3.org/TR/1999/REC-xpath-19991116/) XML Path Language (XPath) Version 1.0

[XSD-TYPES](https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/) XML Schema Part 2: Datatypes Second Edition

## 9.2 Informative References

[RFC 2578](https://datatracker.ietf.org/doc/html/rfc2578) Structure of Management Information Version 2 (SMIv2)

[RFC 2579](https://datatracker.ietf.org/doc/html/rfc2579) Textual Conventions for SMIv2

[RFC 3780](https://datatracker.ietf.org/doc/html/rfc3780) SMIng - Next Generation Structure of Management Information

[RFC 4844](https://datatracker.ietf.org/doc/html/rfc4844) The RFC Series and RFC Editor

[XPATH2.0](https://www.w3.org/TR/xpath20/) XML Path Language (XPath) 2.0 (Second Edition)

[XSLT](https://www.w3.org/TR/1999/REC-xslt-19991116) XSL Transformations (XSLT) Version 1.0

# 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)