

Nexus 플랫폼의 보안 Syslog 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[토폴로지](#)

[배경 정보](#)

[기존 시스템 로그](#)

[보안 Syslog](#)

[인증서 체인](#)

[인증서가 필요한 이유](#)

[보안 Syslog TLS 핸드셰이크](#)

[관련 컨피그레이션](#)

[Cisco Nexus 컨피그레이션](#)

[Ubuntu Syslog 서버 컨피그레이션](#)

[문제 해결](#)

[목표](#)

[토폴로지 설명](#)

[1단계. 보안 Syslog 컨피그레이션 확인](#)

[2단계. 신뢰 지점 컨피그레이션 확인](#)

[3단계. IP 연결 확인](#)

[4단계. TLS 연결 확인](#)

[이 명령이 확인하는 내용](#)

[성공적인 출력이 예상됩니다.](#)

[패킷 캡처 상관관계](#)

[보안 Syslog 메시지 전송 확인](#)

[테스트 Syslog 메시지 생성](#)

[암호화된 TLS 애플리케이션 데이터 확인](#)

[Syslog 서버에서 메시지 수신 확인](#)

[결론](#)

소개

이 문서에서는 Cisco Nexus 스위치와 타사 서버 간의 TLS(Secure Syslog over Transport Layer Security) 문제를 해결하는 방법을 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- NX-OS 플랫폼
- 보안 Syslog
- 기본 PKI(Public Key Infrastructure) 지식

사용되는 구성 요소

N9K1	N9K-C9336C-FX2	10.4(7)
Syslog 서버	Ubuntu 서버	rsyslog 8.2112 OpenSSL 3.x

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

토폴로지



배경 정보

기존 시스템 로그

기존 Syslog는 암호화 없이 UDP(기본 포트 514) 또는 TCP를 통해 로그 메시지를 전송합니다. 통신

이 일반 텍스트로 전송되므로 로그 메시지는 전송 중에 가로채거나 수정할 수 있습니다. 기존 Syslog는 서버 인증도 제공하지 않습니다.

보안 Syslog

Secure Syslog는 syslog 메시지의 암호화된 전송을 제공하기 위해 TLS(기본 포트 6514)가 포함된 TCP를 사용합니다. 로그 메시지가 전송되기 전에 클라이언트와 서버는 TLS 세션을 설정하며, 이 과정에서 서버는 인증서를 제공하고 클라이언트는 신뢰할 수 있는 CA(Certificate Authority)에 대해 인증서 체인을 검증합니다. TLS 핸드셰이크가 성공적으로 완료되면 syslog 메시지가 암호화된 채널을 통해 전송됩니다.

기존 Syslog와 달리, 보안 Syslog는 IP 연결뿐만 아니라 성공적인 TLS 핸드셰이크, 인증서 검증 및 암호 협상에 따라 어떤 syslog 메시지도 교환될 수 있습니다.

인증서 체인

인증서 체인은 서버 인증서와 신뢰할 수 있는 루트 CA(Certificate Authority) 간에 신뢰를 설정하기 위해 사용되는 인증서의 시퀀스입니다. TLS 핸드셰이크 중에 Syslog 서버는 필요한 중간 CA 인증서와 함께 서버 인증서를 제공합니다. Cisco Nexus 스위치는 신뢰할 수 있는 루트 CA를 사용하여 전체 인증서 체인을 검증합니다.

인증서 체인이 항상 필요한 것은 아닙니다.

Syslog 서버에서 제공하는 인증서는 PKI가 구현된 방식에 따라 달라집니다. 예를 들면 다음과 같습니다.

자체 서명 인증서: 서버는 자체 인증서만 표시합니다. 이 경우 Cisco Nexus 스위치는 자체 서명 인증서를 직접 신뢰해야 합니다.

중간 CA 없이 CA 서명 인증서: 서버는 신뢰할 수 있는 루트 CA에서 직접 서명했기 때문에 서버 인증서만 표시합니다.

중간 CA가 있는 CA 서명 인증서: 서버는 하나 이상의 중간 CA 인증서와 함께 서버 인증서를 제공하므로 Cisco Nexus 스위치가 신뢰할 수 있는 루트 CA까지 완전한 신뢰 체인을 구축할 수 있습니다.

PKI 설계와 상관없이 Cisco Nexus 스위치는 TLS 세션을 설정하기 전에 서버에서 제공한 인증서를 검증할 수 있어야 합니다.

신뢰 체인의 필수 인증서가 누락되었거나, 유효하지 않거나, 만료되었거나, 신뢰할 수 없는 CA에서 발급된 경우 TLS 핸드셰이크가 실패하고 보안 Syslog 연결이 설정될 수 없습니다.

인증서가 필요한 이유

보안 Syslog는 암호화 및 서버 인증을 제공하기 위해 TLS를 사용합니다. syslog 메시지가 전송되기 전에 Cisco Nexus 스위치는 권한이 없는 디바이스가 아니라 의도된 Syslog 서버와 통신하는지 확인해야 합니다.

인증서는 Nexus 스위치가 암호화된 세션을 설정하기 전에 Syslog 서버의 ID를 인증하도록 허용함으로써 이러한 신뢰를 제공합니다. 인증서 체인이 성공적으로 검증되면 보안 TLS 채널이 생성되고 모든 후속 syslog 메시지가 암호화된 연결을 통해 전송됩니다.

유효한 인증서와 신뢰할 수 있는 인증서 체인이 없으면 보안 Syslog에서 TLS 세션을 설정할 수 없으므로 로그 메시지가 안전하게 교환되지 않습니다.

보안 Syslog TLS 핸드셰이크

어떤 syslog 메시지도 전송되기 전에 Cisco Nexus 스위치와 Syslog 서버가 성공적으로 TLS 세션을 설정해야 합니다. TLS 핸드셰이크 중에 두 디바이스는 TLS 매개변수를 협상하고, 공유 암호화 키를 설정하며, 인증서 검증을 통해 Syslog 서버의 ID를 인증합니다.

TLS 핸드셰이크가 성공적으로 완료된 후에야 비로소 syslog 메시지가 암호화된 연결을 통해 전송됩니다.

TLS 핸드셰이크는 다음 단계로 구성됩니다.

단계	설명
1. TCP 연결	Cisco Nexus 스위치는 보안 Syslog 서버(기본 포트 6514)에 대한 TCP 연결을 설정합니다.
2. 클라이언트 Hello	Nexus는 지원되는 TLS 버전, 암호 그룹 및 임의 값을 전송하여 TLS 핸드셰이크를 시작합니다.
3. 서버 Hello	Syslog 서버는 세션에 사용되는 TLS 버전 및 암호 그룹을 선택합니다.
4. 인증서 교환	Syslog 서버는 서버 인증서를 제공하며, 필요한 경우 중간 CA 인증서를 제공합니다.
5. 인증서 검증	Nexus는 구성된 신뢰받는 CA에 대해 서버 인증서를 검증합니다. 검증에는

단계	설명
	CN(Common Name) 및 SAN(Subject Alternative Name)을 통해 인증서 체인, 만료 날짜, 발급자 및 서버 ID 확인이 포함됩니다.
6. 키 교환	두 피어는 암호화에 사용되는 공유 세션 키를 파생시키기 위해 암호화 정보를 교환합니다.
7. 완료된 메시지	각 피어는 핸드셰이크가 성공적으로 완료되었으며 양쪽이 동일한 암호화 키를 파생했는지 확인합니다.
8. 보안 Syslog 전송	TLS 세션이 설정되면 syslog 메시지는 암호화된 TLS 애플리케이션 데이터 레코드로 전송됩니다.

관련 컨피그레이션

Cisco Nexus 컨피그레이션

Cisco Nexus 스위치는 보안 Syslog 대상, 신뢰할 수 있는 CA를 포함하는 신뢰 지점, 적절한 소스 인터페이스로 구성해야 합니다.

```
switch(config)# crypto ca trustpoint SYSLOG-TLS-LAB-CA
switch(config-trustpoint)# crypto ca authenticate SYSLOG-TLS-LAB-CA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
<paste>
-----END CERTIFICATE-----

END OF INPUT

Do you accept this certificate? [yes/no]:y
logging server syslog.lab.local 6 secure port 6514 use-vrf default
logging source-interface Eth1/10
```

Ubuntu Syslog 서버 컨피그레이션

이 문서에 사용된 Syslog 서버는 TLS 지원이 활성화된 rsyslog를 실행하는 Ubuntu 서버입니다.

서버 컨피그레이션에는 다음 기능이 포함됩니다.

TCP 포트 6514에서 보안 Syslog 연결을 수신합니다.
TLS 핸드셰이크 중에 TLS 서버 인증서를 제공합니다.
서버 인증서와 연결된 개인 키를 사용합니다.
TLS 핸드셰이크 중에 Cisco Nexus에 필요한 인증서 체인(해당하는 경우)을 제공합니다.
수신된 보안 Syslog 메시지를 로컬 로그 파일에 저장합니다.

이 문서에서 사용되는 PKI 구현은 다음과 같이 구성됩니다. Server Certificate(서버 인증서) > Intermediate CA(중간 CA) > Root CA(루트 CA)

Cisco Nexus 스위치는 신뢰 지점을 통해 루트 CA 인증서를 가져오고 신뢰하여 Syslog 서버에서 제공하는 인증서 체인을 검증할 수 있습니다.



참고: 인증서 체인은 모든 구축에 반드시 필요한 것은 아닙니다. PKI 구현에 따라 Syslog 서버는 자체 서명 인증서, 루트 CA가 직접 서명한 서버 인증서 또는 하나 이상의 중간 CA 인증서가 포함된 서버 인증서만 표시할 수 있습니다.

문제 해결

목표

이 트러블슈팅 절차의 목적은 Cisco Nexus NX-OS 스위치가 Syslog 서버와의 보안 Syslog TLS 연결을 성공적으로 설정하지 못하게 하는 장애를 격리하고 식별하는 것입니다.

이 워크플로는 IP 연결로 시작하여 TCP 연결, TLS 협상, 인증서 검증 및 마지막으로 보안 Syslog 메시지 전송을 통해 진행되는 통신 프로세스의 각 단계를 검증합니다. 각 레이어를 독립적으로 검증함으로써 통신이 실패한 위치를 빠르게 판단할 수 있다.

토폴로지 설명

이 문서에 사용된 토폴로지는 보안 Syslog 클라이언트로 구성된 Cisco Nexus 스위치와 보안 Syslog 서버로 구성된 rsyslog를 실행하는 Ubuntu 서버로 구성됩니다.

Cisco Nexus 스위치는 포트 6514에서 Syslog 서버에 대한 TCP 연결을 시작합니다. TCP 연결이 설정되면 두 디바이스 모두 TLS 핸드셰이크를 수행합니다. 이 과정에서 Syslog 서버는 서버 인증서 및 필요한 중간 CA 인증서를 제공합니다. Nexus 스위치는 신뢰 지점에 구성된 신뢰할 수 있는 루트 CA에 대해 인증서 체인을 검증합니다.

TLS 핸드셰이크가 성공적으로 완료되면 암호화된 세션이 설정되고 Nexus 스위치가 보안 Syslog 메시지를 Syslog 서버로 전송하기 시작합니다.

이 문서에 제시된 트러블슈팅 방법론은 연결 설정과 동일한 통신 시퀀스를 사용하므로 프로세스의 각 단계가 독립적으로 검증될 수 있습니다.

1단계. 보안 Syslog 컨피그레이션 확인

Cisco Nexus 스위치에 Secure Syslog가 올바르게 구성되어 있고 스위치에서 구성된 Syslog 서버를 인식하는지 확인합니다.

```
switch# show logging server
Logging server: enabled.          >>>>>>> Secure Syslog is enabled.
{syslog.lab.local}               >>>>>>> The correct destination (IP address or Fully Qualified Domain
server status: No errors found   >>>>>>> No configuration errors are reported.
server severity: notifications
server facility: local7
server VRF: default
server port: 6514                >>>>>>> The destination port is 6514.
server transport: secure         >>>>>>> The transport is secure.
switch#
```

2단계. 신뢰 지점 컨피그레이션 확인

Cisco Nexus 스위치가 Syslog 서버 인증서 서명에 사용되는 CA를 신뢰하는지 확인합니다.

확인:

필요한 신뢰 지점이 있습니다.
루트 CA 인증서가 설치됩니다.
필요한 중간 CA 인증서가 모두 있습니다.
인증서가 만료되지 않았습니다.
인증서 지문이 예상 값과 일치합니다.

```
switch# show crypto ca certificates
Trustpoint: SYSLOG-TLS-LAB-CA
CA certificate 0:
subject=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Int
issuer=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root
serial=4959A9F5A8D174553E0FB4FA01A21DDC2C085F12
```

```
notBefore=Jun 30 20:47:09 2026 GMT
notAfter=Jun 29 20:47:09 2031 GMT
SHA1 Fingerprint=24:8A:3A:C0:7C:61:53:3B:BE:A7:BB:51:12:FD:7E:B3:D7:AB:44:4C
purposes: sslserver sslclient
```

CA certificate 1:

```
subject=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root
issuer=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root
serial=3BC9CD3374C4A29A6794AAAEBA270E3116C3D55
notBefore=Jun 30 20:43:04 2026 GMT
notAfter=Jun 27 20:43:04 2036 GMT
SHA1 Fingerprint=C8:AC:81:89:AA:A5:89:13:CA:01:5C:A9:90:15:3E:78:22:DD:34:A5
purposes: sslserver sslclient
```

switch#

적절한 CA 인증서가 없거나 유효하지 않은 경우 TLS 핸드셰이크 중에 인증서 검증이 실패합니다.

3단계. IP 연결 확인

Cisco Nexus 스위치와 보안 Syslog 서버 간의 레이어 3 연결을 확인합니다.

확인:

- ICMP(Internet Control Message Protocol) 연결 성공
- 올바른 DNS(Domain Name System) 확인
- 라우팅 문제가 없습니다.
- 패킷 손실이 없습니다.

```
switch# ping syslog.lab.local
PING syslog.lab.local (192.168.100.10): 56 data bytes
64 bytes from 192.168.100.10: icmp_seq=0 ttl=63 time=0.672 ms
64 bytes from 192.168.100.10: icmp_seq=1 ttl=63 time=0.331 ms
64 bytes from 192.168.100.10: icmp_seq=2 ttl=63 time=0.295 ms
64 bytes from 192.168.100.10: icmp_seq=3 ttl=63 time=0.286 ms
64 bytes from 192.168.100.10: icmp_seq=4 ttl=63 time=0.285 ms

--- syslog.lab.local ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.285/0.373/0.672 ms
switch#
```

IP 연결이 실패하면 TCP 연결을 설정할 수 없습니다.

4단계. TLS 연결 확인

Cisco Nexus 스위치와 보안 Syslog 서버 간의 엔드 투 엔드 TLS 연결을 확인합니다.

이 단계에서는 다음을 포함하여 보안 Syslog 연결의 여러 단계를 검증합니다.

- TCP 연결
- TLS 협상
- 서버 인증서 프레젠테이션
- 인증서 체인
- 인증서 검증
- 암호 그룹 협상

이러한 작업은 TLS 핸드셰이크의 일부로 수행되므로 단일 명령으로 모두 확인할 수 있습니다.

```
switch# run bash sudo ip netns exec default openssl s_client -connect 192.168.100.10:6514 -CAfile /boot
```

이 명령이 확인하는 내용

검증	설명
TCP 연결	Nexus에서 Syslog 서버에 대한 TCP 연결을 설정할 수 있는지 확인합니다.
TLS 핸드셰이크	두 피어가 TLS 세션을 성공적으로 협상하는지 확인합니다.
서버 인증서	Syslog 서버에서 제공하는 인증서를 표시합니다.
인증서 체인	서버에서 보낸 중간 CA 인증서를 표시합니다.
TLS 버전	협상된 TLS 버전을 표시합니다.
암호 그룹	협상된 암호화 알고리즘을 표시합니다.
인증서 검증	-CAfile 옵션을 사용하면 지정된 신뢰받는 CA에 대해 인증서 체인을 확인합니다.

성공적인 출력이 예상됩니다.

```

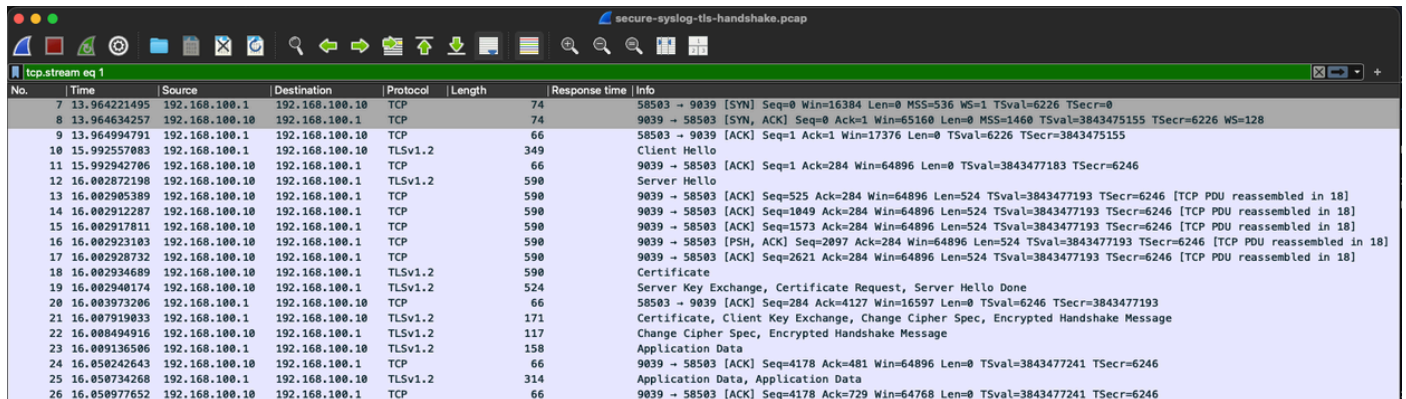
switch# run bash sudo ip netns exec default openssl s_client -connect 192.168.100.10:6514 -CAfile /boot
CONNECTED(00000003) >>>> this confirms that the TCP connection has been established.
Can't use SSL_get_servername
depth=2 C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root CA
verify return:1
depth=1 C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
verify return:1
depth=0 C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = syslog.lab.local
verify return:1
---
Certificate chain >>>>>>>>>
0 s:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = syslog.lab.local,
i:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
1 s:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
i:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root CA,
---
Server certificate
-----BEGIN CERTIFICATE-----
<snip>
-----END CERTIFICATE-----
subject=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = syslog.lab.local

issuer=C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
---
Acceptable client certificate CA names
C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root CA, email=
Client Certificate Types: RSA sign, ECDSA sign
Requested Signature Algorithms: RSA+SHA256:RSA-PSS+SHA256:RSA-PSS+SHA256:ECDSA+SHA256:Ed25519:RSA+SHA384
Shared Requested Signature Algorithms: RSA+SHA256:RSA-PSS+SHA256:RSA-PSS+SHA256:ECDSA+SHA256:Ed25519:RSA+SHA384
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 4177 bytes and written 392 bytes
Verification: OK >>>>>>>
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 4096 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
Protocol : TLSv1.2. >>>>>>>>> The negotiated protocol
Cipher : ECDHE-RSA-AES256-GCM-SHA384 >>>>>>>>> The negotiated cipher
Session-ID: E67D3C91AB57212C16C2A240668CBC5BA1720BC5DA6AD69C69B209C90E828483
Session-ID-ctx:
Master-Key: 5C45FC4ACE18657DBCDC0946A68A9BE480ECB0CF24313C56C3067EA9D23D68EB5090A4DFE36D4E4CB72BA282D0A
PSK identity: None
PSK identity hint: None
SRP username: None
Start Time: 1782867300
Timeout : 7200 (sec)
Verify return code: 0 (ok). >>>>>>>>> indicates that the certificate chain was successful
Extended master secret: yes
---
switch#

```

패킷 캡처 상관관계

Openssl s_client 출력은 TLS 핸드셰이크의 각 단계를 확인하기 위해 Ethanalalyzer를 사용하여 얻은 패킷 캡처와 상관관계가 있을 수 있습니다.



No.	Time	Source	Destination	Protocol	Length	Response time	Info
7	13.964221495	192.168.100.1	192.168.100.10	TCP	74		58503 → 9039 [SYN] Seq=0 Win=16384 Len=0 MSS=536 WS=1 TSval=6226 TSecr=0
8	13.964634257	192.168.100.10	192.168.100.1	TCP	74		9039 → 58503 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 TSval=3843475155 TSecr=6226 WS=128
9	13.964994791	192.168.100.1	192.168.100.10	TCP	66		58503 → 9039 [ACK] Seq=1 Ack=1 Win=17376 Len=0 TSval=6226 TSecr=3843475155
10	15.992557083	192.168.100.1	192.168.100.10	TLSv1.2	349		Client Hello
11	15.992942706	192.168.100.10	192.168.100.1	TCP	66		9039 → 58503 [ACK] Seq=1 Ack=284 Win=64896 Len=0 TSval=3843477183 TSecr=6246
12	16.002872198	192.168.100.10	192.168.100.1	TLSv1.2	590		Server Hello
13	16.002905389	192.168.100.10	192.168.100.1	TCP	590		9039 → 58503 [ACK] Seq=525 Ack=284 Win=64896 Len=524 TSval=3843477193 TSecr=6246 [TCP PDU reassembled in 18]
14	16.002912287	192.168.100.10	192.168.100.1	TCP	590		9039 → 58503 [ACK] Seq=1049 Ack=284 Win=64896 Len=524 TSval=3843477193 TSecr=6246 [TCP PDU reassembled in 18]
15	16.002917811	192.168.100.10	192.168.100.1	TCP	590		9039 → 58503 [ACK] Seq=1573 Ack=284 Win=64896 Len=524 TSval=3843477193 TSecr=6246 [TCP PDU reassembled in 18]
16	16.002923103	192.168.100.10	192.168.100.1	TCP	590		9039 → 58503 [PSH, ACK] Seq=2097 Ack=284 Win=64896 Len=524 TSval=3843477193 TSecr=6246 [TCP PDU reassembled in 18]
17	16.002928732	192.168.100.10	192.168.100.1	TCP	590		9039 → 58503 [ACK] Seq=2621 Ack=284 Win=64896 Len=524 TSval=3843477193 TSecr=6246 [TCP PDU reassembled in 18]
18	16.002934689	192.168.100.10	192.168.100.1	TLSv1.2	590		Certificate
19	16.002940174	192.168.100.10	192.168.100.1	TLSv1.2	524		Server Key Exchange, Certificate Request, Server Hello Done
20	16.003973206	192.168.100.1	192.168.100.10	TCP	66		58503 → 9039 [ACK] Seq=284 Ack=4127 Win=16597 Len=0 TSval=6246 TSecr=3843477193
21	16.007919033	192.168.100.1	192.168.100.10	TLSv1.2	171		Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
22	16.008494916	192.168.100.10	192.168.100.1	TLSv1.2	117		Change Cipher Spec, Encrypted Handshake Message
23	16.009136506	192.168.100.1	192.168.100.10	TLSv1.2	158		Application Data
24	16.050242643	192.168.100.10	192.168.100.1	TCP	66		9039 → 58503 [ACK] Seq=4178 Ack=481 Win=64896 Len=0 TSval=3843477241 TSecr=6246
25	16.050734268	192.168.100.1	192.168.100.10	TLSv1.2	314		Application Data, Application Data
26	16.050977652	192.168.100.10	192.168.100.1	TCP	66		9039 → 58503 [ACK] Seq=4178 Ack=729 Win=64768 Len=0 TSval=3843477241 TSecr=6246

보안 Syslog TLS 핸드셰이크 캡처

TLS 핸드셰이크 흐름

```
Cisco Nexus                               Syslog Server
-----
TCP SYN ----->
<----- TCP SYN/ACK
TCP ACK ----->
Client Hello ----->
<----- Server Hello
<----- Certificate
<----- (Intermediate CA)
<----- Server Key Exchange
<----- Server Hello Done
Client Key Exchange ----->
Change Cipher Spec ----->
Finished ----->
<----- Change Cipher Spec
<----- Finished
==== Secure TLS Session Established ====
Encrypted Syslog Messages ----->
```

TLS 핸드셰이크가 성공적으로 완료되어야 syslog 메시지를 교환할 수 있습니다. 핸드셰이크가 실패하면 보안 Syslog 트래픽이 생성되지 않습니다.

이 확인을 수행하려면 Cisco Nexus 스위치에 대한 CLI 세션 2개를 엽니다. 첫 번째 세션에서 다음 명령을 사용하여 TLS 연결을 시작합니다.

```
switch# run bash sudo ip netns exec default openssl s_client -connect 192.168.100.10:6514 -CAfile /boot
```

동시에 두 번째 CLI 세션에서 TLS 교환을 모니터링하기 위해 Ethalyzer 캡처를 시작합니다.

```
switch# ethalyzer local interface inband display-filter "tcp.port==6514" limit-captured-frames 0
Capturing on 'ps-inb'
3 2026-07-01 00:55:00.194836329 192.168.100.1 → 192.168.100.10 TCP 74 18553 → 6514 [SYN] Seq=0 Win=42340 Len=
4 2026-07-01 00:55:00.195053907 192.168.100.10 → 192.168.100.1 TCP 74 6514 → 18553 [SYN, ACK] Seq=0 Ack=1 Win=
2103467794 WS=128
5 2026-07-01 00:55:00.195110969 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=1 Ack=1 Win=4300
6 2026-07-01 00:55:00.195361105 192.168.100.1 → 192.168.100.10 TLSv1 353 Client Hello
7 2026-07-01 00:55:00.195478134 192.168.100.10 → 192.168.100.1 TCP 66 6514 → 18553 [ACK] Seq=1 Ack=288 Win=64
8 2026-07-01 00:55:00.204176209 192.168.100.10 → 192.168.100.1 TLSv1.2 1514 Server Hello
9 2026-07-01 00:55:00.204200237 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=288 Ack=1449 Win
10 2026-07-01 00:55:00.204207757 192.168.100.10 → 192.168.100.1 TCP 1514 6514 → 18553 [ACK] Seq=1449 Ack=288
P segment of a reassembled PDU]
11 2026-07-01 00:55:00.204219906 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=288 Ack=2897 Wi
12 2026-07-01 00:55:00.204225215 192.168.100.10 → 192.168.100.1 TLSv1.2 1296 Certificate, Server Key Exchange
13 2026-07-01 00:55:00.204234417 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=288 Ack=4127 Wi
14 2026-07-01 00:55:00.206544194 192.168.100.1 → 192.168.100.10 TLSv1.2 171 Certificate, Client Key Exchange,
15 2026-07-01 00:55:00.206818190 192.168.100.10 → 192.168.100.1 TLSv1.2 117 Change Cipher Spec, Encrypted Har
14 16 2026-07-01 00:55:00.249090950 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=393 Ack=417
17 60 2026-07-01 00:56:10.555097655 192.168.100.1 → 192.168.100.10 TCP 66 18553 → 6514 [ACK] Seq=394 Ack=417
61 2026-07-01 00:56:10.887874467 192.168.100.1 → 192.168.100.10 TLSv1.2 226 Application Data
19 62 2026-07-01 00:56:10.888066832 192.168.100.10 → 192.168.100.1 TCP 66 6514 → 60177 [ACK] Seq=1 Ack=161 W
19 65 2026-07-01 00:56:13.891148284 192.168.100.1 → 192.168.100.10 TLSv1.2 189 Application Data
21 66 2026-07-01 00:56:13.891341818 192.168.100.10 → 192.168.100.1 TCP 66 6514 → 60177 [ACK] Seq=1 Ack=284 W
```

패킷 캡처는 TCP 3-way 핸드셰이크, Client Hello, Server Hello, Certificate, Key Exchange, Finished 메시지를 포함한 전체 TLS 핸드셰이크 시퀀스를 표시해야 하며, 핸드셰이크가 성공적으로 완료되면 Secure Syslog 메시지를 전달하는 암호화된 TLS Application Data 레코드가 표시됩니다.

보안 Syslog 메시지 전송 확인

TLS 세션이 설정된 후 Secure Syslog 메시지가 Cisco Nexus 스위치에서 Syslog 서버로 성공적으로 전송되는지 확인합니다.

TLS 핸드셰이크에 성공하면 암호화된 채널이 생성되었음을 확인합니다. 그러나 syslog 메시지가 생성되거나 전달된다고 보장하지는 않습니다.

이 단계에서는 테스트 메시지를 생성하고 전송 및 수신을 확인하여 완전한 보안 Syslog 워크플로를 검증합니다.

테스트 Syslog 메시지 생성

```
switch# logit test1-FromN9K
```

```
2026 Jul 1 01:04:04 switch %$ VDC-1 %$ %LOCAL0-2-SYSTEM_MSG: systest from 0: test1-FromN9K - vsh.bin[15
```

```
switch# logit test2-FromN9K
```

```
2026 Jul 1 01:12:44 switch %$ VDC-1 %$ %LOCAL0-2-SYSTEM_MSG: systest from 0: test2-FromN9K - vsh.bin[15
```

이 명령은 기존 TLS 세션을 통해 구성된 보안 Syslog 서버에 전송해야 하는 로컬 syslog 메시지를 생성합니다.

암호화된 TLS 애플리케이션 데이터 확인

이전 섹션에서 설명한 대로 Ethalyzer 캡처가 이미 실행 중인 경우 테스트 메시지가 생성된 후 새 TLS 애플리케이션 데이터 패킷이 전송되는지 확인합니다.

```
switch# ethalyzer local interface inband display-filter "tcp.port==6514" limit-captured-frames 0
```

```
25 483 2026-07-01 01:05:33.547471509 192.168.100.1 → 192.168.100.10 TLSv1.2 199 Application Data
```

```
27 484 2026-07-01 01:05:33.547728284 192.168.100.10 → 192.168.100.1 TCP 66 6514 → 60177 [ACK] Seq=1 Ack=707
```

```
27 489 2026-07-01 01:05:40.582487840 192.168.100.1 → 192.168.100.10 TLSv1.2 198 Application Data
```

```
29 490 2026-07-01 01:05:40.582705431 192.168.100.10 → 192.168.100.1 TCP 66 6514 → 60177 [ACK] Seq=1 Ack=839
```

TLS 애플리케이션 데이터가 있으면 syslog 메시지가 암호화된 TLS 세션을 통해 전송되고 있음을 나타냅니다. 페이로드가 암호화되므로 Ethalyzer는 syslog 메시지의 내용을 디코딩할 수 없습니다.



참고: TLS 핸드셰이크가 성공적으로 완료되면 모든 syslog 메시지가 암호화된 TLS 애플리케이션 데이터 레코드 내에 캡슐화됩니다. 패킷 캡처에는 내용이 표시되지 않습니다.

Syslog 서버에서 메시지 수신 확인

Syslog 서버에서 메시지가 성공적으로 수신되어 대상 로그 파일에 기록되었는지 확인합니다.

```
calo@calo-ubuntu22 ~ % sudo tail -20 /var/log/remote.log 8:03:45
```

```
Jun 30 20:04:36 192.168.100.1 : 2026 Jul 1 01:05:40 UTC: %LOCAL0-2-SYSTEM_MSG: systest from 0: test1-Fr
Jun 30 20:11:41 192.168.100.1 : 2026 Jul 1 01:12:45 UTC: %LOCAL0-2-SYSTEM_MSG: systest from 0: test2-Fr
```

생성된 테스트 메시지가 있으면 다음을 확인할 수 있습니다.

Cisco Nexus 스위치에서 syslog 메시지를 생성했습니다.

메시지는 암호화된 TLS 세션을 통해 전송되었습니다.

Syslog 서버가 메시지를 성공적으로 해독하고 처리했습니다.

보안 Syslog 컨피그레이션이 엔드 투 엔드로 올바르게 작동하고 있습니다.

Ethanalyzer의 TLS 애플리케이션 데이터 표시에서는 TLS 세션이 암호화된 애플리케이션 트래픽을 적극적으로 전달한다는 것을 확인합니다. 그러나 Secure Syslog가 올바르게 작동한다는 확실한 증거는 Syslog 서버에서 생성된 syslog 메시지를 성공적으로 수신하는 것입니다.

결론

Cisco Nexus 스위치의 보안 Syslog는 성공적인 IP 및 TCP 연결, TLS 협상, 인증서 검증, 암호화된 메시지 전달에 따라 달라집니다. 각 레이어를 독립적으로 확인하면 장애 지점을 격리하고 연결, TLS, 인증서 및 메시지 전송 문제를 구분할 수 있습니다. 성공적인 TLS 핸드셰이크는 암호화된 세션이 설정되었음을 확인하는 한편 Syslog 서버에서 생성된 테스트 메시지를 수신하면 엔드 투 엔드 보안 Syslog 작업을 확인합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.