

Nexusプラットフォーム上のセキュアなsyslogの トラブルシューティング

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[トポロジ](#)

[背景説明](#)

[従来のSyslog](#)

[セキュアSyslog](#)

[証明書チェーン](#)

[証明書が必要な理由](#)

[セキュアなsyslog TLS/ハンドシェイク](#)

[関連コンフィギュレーション](#)

[Cisco Nexusの設定](#)

[Ubuntu Syslogサーバの設定](#)

[トラブルシューティング](#)

[目的](#)

[トポロジの説明](#)

[ステップ 1：セキュアなsyslog設定の確認](#)

[ステップ 2：トラストポイント設定の確認](#)

[ステップ 3：IP 接続性の確認](#)

[ステップ 4：TLS接続の確認](#)

[このコマンドで確認できること](#)

[正常に出力されると予想される](#)

[パケットキャプチャの相関](#)

[セキュアなSyslogメッセージ伝送の確認](#)

[テストsyslogメッセージの生成](#)

[暗号化されたTLSアプリケーションデータの確認](#)

[Syslogサーバでのメッセージ受信の確認](#)

[結論](#)

概要

このドキュメントでは、Cisco Nexusスイッチとサードパーティ製サーバ間のTransport Layer Security(TLS)上のセキュアSyslogをトラブルシューティングする方法について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- NX-OSプラットフォーム
- セキュアSyslog
- 公開キーインフラストラクチャ(PKI)の基礎知識

使用するコンポーネント

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

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

トポロジ



背景説明

従来のSyslog

従来のSyslogは、UDP（デフォルトポート514）またはTCPを介して、暗号化なしでログメッセージを送信します。通信はクリアテキストで送信されるため、ログメッセージは送信中に傍受されたり変更されたりする可能性があります。従来のsyslogでは、サーバ認証も提供されません。

セキュアSyslog

セキュアなsyslogでは、syslogメッセージの暗号化された送信を提供するために、TLS（デフォルトポートは6514）を使用したTCPが使用されます。ログメッセージが送信される前に、クライアントとサーバはTLSセッションを確立します。このセッションで、サーバは自身の証明書を提示し、クライアントは信頼できる認証局(CA)に対して証明書チェーンを検証します。TLSハンドシェイクが正常に完了すると、syslogメッセージが暗号化チャネルを介して送信されます。

従来のsyslogとは異なり、セキュアなsyslogはIP接続だけでなく、正常なTLSハンドシェイク、証明書の検証、および暗号ネゴシエーション（syslogメッセージを交換できるようになる前）にも依存します。

証明書チェーン

証明書チェーンは、サーバ証明書と信頼されたルート認証局（ルートCA）間の信頼を確立するために使用される一連の証明書です。TLSハンドシェイク中、Syslogサーバはサーバ証明書を必要な中間CA証明書とともに提示します。Cisco Nexusスイッチは、信頼されたルートCAを使用して、証明書チェーン全体を検証します。

証明書チェーンは必ずしも必要ではありません。

Syslogサーバが提示する証明書は、PKIの実装方法によって異なります。以下に、いくつかの例を示します。

自己署名証明書：サーバは自身の証明書のみを提示します。この場合、Cisco Nexusスイッチはその自己署名証明書を直接信頼する必要があります。

中間CAを使用しないCA署名付き証明書：信頼されたルートCAによって直接署名されたサーバ証明書だけが提示されます。

中間CAを使用したCA署名付き証明書：サーバは、サーバ証明書を1つ以上の中間CA証明書とともに提示します。これにより、Cisco Nexusスイッチは、信頼されたルートCAまでの完全な信頼チェーンを構築できます。

PKI設計に関係なく、Cisco Nexusスイッチは、TLSセッションを確立する前に、サーバから提示される証明書を検証する必要があります。

信頼チェーン内の必要な証明書が不足しているか、無効であるか、期限切れであるか、信頼されていないCAによって発行された場合、TLSハンドシェイクは失敗し、セキュアSyslog接続は確立

されません。

証明書が必要な理由

セキュアなSyslogは、暗号化とサーバ認証を提供するためにTLSに依存します。syslogメッセージを送信する前に、Cisco Nexusスイッチは、不正なデバイスではなく、目的のSyslogサーバと通信していることを確認する必要があります。

証明書は、暗号化セッションを確立する前にNexusスイッチがSyslogサーバのIDを認証できるようにすることで、この信頼を提供します。証明書チェーンが正常に検証されると、セキュアなTLSチャネルが作成され、後続のすべてのsyslogメッセージが暗号化された接続を介して送信されます。

有効な証明書と信頼できる証明書チェーンがないと、Secure SyslogはTLSセッションを確立できず、ログメッセージを安全に交換できません。

セキュアなsyslog TLSハンドシェイク

syslogメッセージが送信される前に、Cisco NexusスイッチとSyslogサーバはTLSセッションを正常に確立する必要があります。TLSハンドシェイク中に、両方のデバイスがTLSパラメータをネゴシエートし、共有暗号化キーを確立し、証明書の検証によってSyslogサーバのIDを認証します。

TLSハンドシェイクが正常に完了した後にのみ、syslogメッセージが暗号化された接続を介して送信されます。

TLSハンドシェイクは次のフェーズで構成されます。

ステップ	説明
1. TCP接続	Cisco Nexusスイッチは、セキュアSyslogサーバ (デフォルトポート6514) へのTCP接続を確立します。
2. クライアントHello	Nexusは、サポートされているTLSバージョン、暗号スイート、およびランダムな値を送信して、TLSハンドシェイクを開始します。
3. サーバHello	Syslogサーバは、セッションに使用するTLSバージョンと暗号スイートを選択します。

ステップ	説明
4. 証明書の交換	Syslogサーバは自身のサーバ証明書と、必要に応じて中間CA証明書を提示します。
5. 証明書の検証	Nexusは、設定された信頼できるCAに対してサーバ証明書を検証します。検証には、共通名(CN)とサブジェクト代替名(SAN)を使用した証明書チェーン、有効期限、発行者、およびサーバIDの確認が含まれます。
6. キー交換	暗号化に使用される共有セッションキーを取得するために、両方のピアが暗号化情報を交換します。
7. Finishedメッセージ	各ピアは、ハンドシェイクが正常に完了し、両側が同じ暗号キーを導出したことを確認します。
8. セキュアなSyslog送信	TLSセッションが確立されると、syslogメッセージは暗号化されたTLSアプリケーションデータレコードとして送信されます。

関連コンフィギュレーション

Cisco Nexusの設定

Cisco Nexusスイッチには、セキュアSyslogの宛先、信頼できるCAを含むトラストポイント、および適切な送信元インターフェイスを設定する必要があります。

```
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サーバの設定

このドキュメントで使用するsyslogサーバは、TLSサポートが有効なsyslogを実行するUbuntuサーバです。

サーバ設定には、次の機能が含まれます。

TCPポート6514でセキュアsyslog接続をリッスンします。

TLSハンドシェイク時にTLSサーバ証明書を提示する。

サーバ証明書に関連付けられた秘密キーを使用します。

TLSハンドシェイク中に、必要な証明書チェーン（該当する場合）をCisco Nexusに提供します。受信したセキュアSyslogメッセージをローカルログファイルに保存します。

このドキュメントで使用するPKI実装は、サーバ証明書>中間CA>ルートCAで構成されます。

Cisco Nexusスイッチは、トラストポイントを介してルートCA証明書をインポートおよび信頼し、Syslogサーバによって提示される証明書チェーンを検証できるようにします。



注：証明書チェーンはすべての導入で必須というわけではありません。PKIの実装に応じて、Syslogサーバは、自己署名証明書、ルートCAによって直接署名されたサーバ証明書、または1つ以上の中間CA証明書を伴うサーバ証明書のみを提示できます。

トラブルシューティング

目的

このトラブルシューティング手順の目的は、Cisco Nexus NX-OSスイッチがSyslogサーバとのセキュアなSyslog TLS接続を正常に確立できない障害を切り分けて特定することです。

このワークフローでは、通信プロセスの各段階を検証します。最初にIP接続を検証し、次にTCP接続、TLSネゴシエーション、証明書の検証、最後にセキュアSyslogメッセージの送信を検証します。各レイヤを個別に検証することで、通信がどこで失敗したかを迅速に判断できます。

トポロジの説明

このドキュメントで使用するトポロジは、Secure Syslogクライアントとして設定されたCisco Nexusスイッチと、Secure Syslogサーバとして設定されたsyslogを実行するUbuntuサーバで構成されます。

Cisco Nexusスイッチは、ポート6514でSyslogサーバへのTCP接続を開始します。TCP接続が確立されると、両方のデバイスがTLSハンドシェイクを実行します。このプロセス中に、Syslogサーバはサーバ証明書と必要な中間CA証明書を提示します。Nexusスイッチは、トラストポイントで設定されている信頼されたルートCAに対して証明書チェーンを検証します。

TLSハンドシェイクが正常に完了すると、暗号化されたセッションが確立され、NexusスイッチはSyslogサーバへのセキュアSyslogメッセージの送信を開始します。

このドキュメントで説明するトラブルシューティングの方法論では、接続の確立と同じ通信シーケンスを使用するため、プロセスの各フェーズを個別に検証できます。

ステップ 1：セキュアなsyslog設定の確認

Cisco NexusスイッチでSecure Syslogが正しく設定されていること、および設定されたSyslogサーバがスイッチで認識されることを確認します。

```
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#
```

ステップ 2：トラストポイント設定の確認

Cisco Nexusスイッチが、Syslogサーバ証明書の署名に使用されるCAを信頼することを確認します。

次を確認します：

- 予期されたトラストポイントが存在します。
- ルートCA証明書がインストールされます。
- 必要な中間CA証明書が存在する。
- 証明書の有効期限が切れていません。
- 証明書フィンガープリントが予期された値と一致します。

```
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#
```

適切なCA証明書がないか無効である場合、TLSハンドシェイク中に証明書の検証が失敗します。

ステップ 3 : IP 接続性の確認

Cisco NexusスイッチとSecure Syslogサーバ間のレイヤ3接続を確認します。

次を確認します :

- インターネット制御メッセージプロトコル(ICMP)の到達可能性に成功しました。
- 正しいドメインネームシステム(DNS)解決。
- ルーティングの問題はありません。
- パケット損失はありません。

```
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接続が失敗した場合、TCP接続を確立できません。

ステップ 4：TLS接続の確認

Cisco NexusスイッチとセキュアSyslogサーバ間のエンドツーエンドTLS接続を確認します。

この手順では、セキュアSyslog接続の複数の段階を検証します。これには、次の段階が含まれます。

- TCP接続
- TLSネゴシエーション
- サーバ証明書の提示
- 証明書チェーン
- 証明書の検証
- 暗号スイートネゴシエーション

これらの操作はTLSハンドシェイクの一部として発生するため、すべての操作を1つのコマンドで確認できます。

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

このコマンドで確認できること

検証	説明
TCP接続	NexusがSyslogサーバへのTCP接続を確立できることを確認します。
TLSハンドシェイク	両方のピアがTLSセッションを正常にネゴシエートしたことを確認します。
サーバ証明書	Syslogサーバによって提示された証明書を表示します。
証明書チェーン	サーバから送信された中間CA証明書を表示します。
TLSバージョン	ネゴシエートされたTLSバージョンを表示します。


```
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 success
Extended master secret: yes
---
```

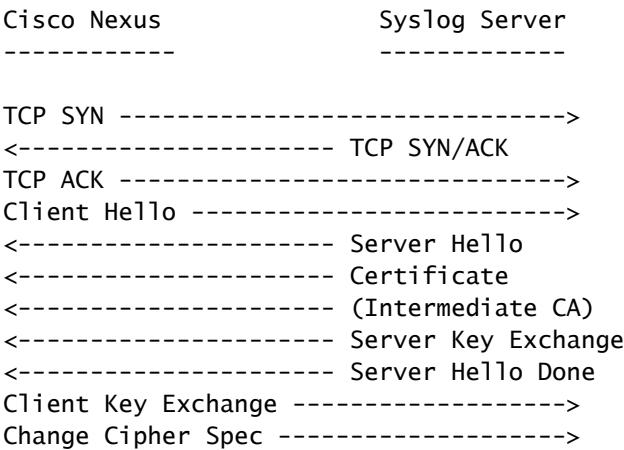
パケットキャプチャの相關

openssl s_clientの出力は、TLSハンドシェイクの各フェーズを確認するために、Ethanalyzerを使用して取得したパケットキャプチャに関連付けることができます。

tcp.stream eq 1							
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

セキュアなsyslog TLSハンドシェイクキャプチャ

TLSハンドシェイクフロー



```
Finished ----->
<----- Change Cipher Spec
<----- Finished
==== Secure TLS Session Established ====
Encrypted Syslog Messages ----->
```

TLSハンドシェイクが正常に完了しなければ、syslogメッセージを交換できません。ハンドシェイクが失敗すると、セキュアSyslogトラフィックは生成されません。

この検証を実行するには、Cisco Nexusスイッチに対して2つのCLIセッションを開きます。最初のセッションで、次のコマンドを使用してTLS接続を開始します。

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

同時に、2番目のCLIセッションで、TLS交換をモニタするためにEthanalyzerキャプチャを開始します。

```
switch# ethanalyzer 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
```

パケットキャプチャでは、TCP 3ウェイハンドシェイク、Client Hello、Server Hello、Certificate、Key Exchange、Finishedメッセージを含む完全なTLSハンドシェイクシーケンスと、ハンドシェイクが正常に完了した場合にセキュアSyslogメッセージを送信する暗号化TLSアプリケーションデータレコードを表示する必要があります。

セキュアなSyslogメッセージ伝送の確認

TLSセッションが確立された後、セキュアなsyslogメッセージがCisco Nexusスイッチからsyslogサーバに正常に送信されることを確認します。

TLSハンドシェイクが成功すると、暗号化チャネルが作成されたことを確認できますが、syslogメッセージが生成または配信されていることを保証するものではありません。

この手順では、テストメッセージを生成し、その送信と受信を確認することによって、セキュリティで保護されたsyslogワークフロー全体を検証します。

テストsyslogメッセージの生成

```
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
```

このコマンドは、既存のTLSセッションを介して設定済みのセキュアsyslogサーバに送信する必要があるローカルsyslogメッセージを生成します。

暗号化されたTLSアプリケーションデータの確認

(前のセクションで説明したように) Ethalyzerキャプチャがすでに実行されている場合は、テストメッセージの生成後に新しいTLSアプリケーションデータパケットが送信されることを確認します。

```
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 Application Dataが存在することは、syslogメッセージが暗号化されたTLSセッションを介して送信されていることを示します。ペイロードは暗号化されているため、Ethalyzerはsyslogメ

ッセージの内容を復号化できません。



注:TLSハンドシェイクが正常に完了すると、すべてのsyslogメッセージは暗号化されたTLSアプリケーションデータレコード内にカプセル化されます。これらの内容は、パケットキャプチャでは表示されません。

Syslogサーバでのメッセージ受信の確認

Syslogサーバで、メッセージが正常に受信され、宛先ログファイルに書き込まれたことを確認します。

```
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
```

生成されたテストメッセージの存在により、次のことが確認されます。

Cisco Nexusスイッチがsyslogメッセージを正常に生成しました。
メッセージは暗号化されたTLSセッション経由で送信されました。
Syslogサーバはメッセージの復号化と処理に成功しました。
セキュアなSyslog設定がエンドツーエンドで正しく動作している。

EthalyzerにTLSアプリケーションデータが表示されれば、TLSセッションが暗号化されたアプリケーショントラフィックをアクティブに伝送していることを確認できます。ただし、セキュアSyslogが正常に機能していることは、生成されたSyslogメッセージがSyslogサーバで正常に受信されたことを示す決定的な証拠です。

結論

Cisco Nexusスイッチ上のセキュアSyslogは、IPおよびTCP接続、TLSネゴシエーション、証明書の検証、および暗号化されたメッセージ配信が成功するかどうかで決まります。各レイヤを個別に検証すると、障害ポイントを特定し、接続、TLS、証明書、およびメッセージ送信の問題を区別するのに役立ちます。TLSハンドシェイクが成功すると、暗号化されたセッションが確立されたことが確認され、Syslogサーバで生成されたテストメッセージを受信すると、エンドツーエンドのセキュアSyslog動作が確認されます。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。