

排除Nexus平台上的安全系統日誌故障

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[拓撲](#)

[背景資訊](#)

[傳統系統日誌](#)

[安全系統日誌](#)

[憑證鏈結](#)

[為什麼需要證書](#)

[安全系統日誌TLS握手](#)

[相關配置](#)

[Cisco Nexus配置](#)

[Ubuntu Syslog伺服器配置](#)

[疑難排解](#)

[目標](#)

[拓撲說明](#)

[步驟1.驗證安全系統日誌配置](#)

[步驟2.驗證信任點配置](#)

[步驟3.檢驗IP連通性](#)

[步驟4.檢驗TLS連通性](#)

[此命令驗證什麼](#)

[預期成功輸出](#)

[封包擷取關聯](#)

[驗證安全系統日誌消息傳輸](#)

[生成測試系統日誌消息](#)

[驗證加密的TLS應用程式資料](#)

[驗證系統日誌伺服器上的消息接收](#)

[結論](#)

簡介

本文檔介紹如何對Cisco Nexus交換機和第三方伺服器之間的傳輸層安全(TLS)安全系統日誌進行故障排除。

必要條件

需求

思科建議您瞭解以下主題：

- NX-OS平台
- 安全系統日誌
- 基本公開金鑰基礎架構(PKI)知識

採用元件

N9K1	N9K-C9336C-FX2	10.4(7)
系統日誌伺服器	Ubuntu伺服器	rsyslog 8.2112 OpenSSL 3.x

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

拓撲



背景資訊

傳統系統日誌

傳統系統日誌通過UDP（預設埠514）或不加密的TCP傳輸日誌消息。由於通訊是以明文形式傳送的，因此日誌消息可能會在傳輸過程中被攔截或修改。傳統系統日誌也不提供伺服器身份驗證。

安全系統日誌

安全系統日誌使用帶有TLS的TCP (預設埠6514) 以提供系統日誌消息的加密傳輸。在傳送任何日誌消息之前，客戶端和伺服器建立TLS會話，在此期間，伺服器呈現其證書，客戶端根據受信任的證書頒發機構(CA)驗證證書鏈。 TLS握手成功完成後，系統日誌消息將通過加密通道傳輸。

與傳統系統日誌不同，安全系統日誌不僅依賴於IP連線，還依賴成功的TLS握手、證書驗證和密碼協商，然後才可以交換任何系統日誌消息。

憑證鏈結

憑證鏈結是在伺服器憑證與受信任的根憑證授權單位 (根CA) 之間建立信任所使用的憑證序列。在TLS握手期間，系統日誌伺服器顯示其伺服器證書以及任何所需的中間CA證書。Cisco Nexus交換機使用其受信任的根CA來驗證整個證書鏈。

並非總是需要憑證鏈結。

Syslog伺服器顯示的證書取決於其PKI的實施方式。舉例來說：

自簽名證書：伺服器僅呈現其自己的憑證。在這種情況下，Cisco Nexus交換機必須直接信任該自簽名證書。

CA簽署的證書 (不含中間CA)：伺服器僅呈現其伺服器憑證，因為是由受信任的根CA直接簽署。

具有中間CA的CA簽名證書：伺服器將其伺服器證書與一個或多個中間CA證書一起顯示，允許Cisco Nexus交換機建立到受信任的根CA的完整信任鏈。

無論PKI設計如何，Cisco Nexus交換機都必須能夠在TLS會話建立之前驗證伺服器提供的證書。

如果信任鏈中的任何必需證書丟失、無效、已過期或由不受信任的CA頒發，則TLS握手將失敗，並且無法建立安全系統日誌連線。

為什麼需要證書

安全系統日誌依靠TLS提供加密和伺服器身份驗證。在傳輸任何系統日誌消息之前，Cisco Nexus交換機必須驗證它是否與預期的系統日誌伺服器通訊，而不是與未經授權的裝置通訊。

證書通過在建立加密會話之前允許Nexus交換機驗證系統日誌伺服器的身份來提供此信任。證書鏈成功驗證後，將建立安全TLS通道，所有後續的系統日誌消息將通過加密連線傳輸。

如果沒有有效證書和可信證書鏈，安全系統日誌無法建立TLS會話，從而阻止安全交換日誌消息。

安全系統日誌TLS握手

在傳輸任何系統日誌消息之前，Cisco Nexus交換機和Syslog伺服器必須成功建立TLS會話。在TLS握手期間，兩台裝置會協商TLS引數、建立共用加密金鑰，並通過證書驗證來驗證系統日誌伺服器的身份。

只有在TLS握手成功完成後，系統日誌消息才會通過加密連線傳輸。

TLS握手包含以下階段：

步驟	說明
1. TCP連線	Cisco Nexus交換機建立與安全Syslog伺服器（預設埠6514）的TCP連線。
2.客戶端Hello	Nexus通過傳送支援的TLS版本、密碼套件和隨機值來啟動TLS握手。
3.伺服器Hello	系統日誌伺服器選擇用於會話的TLS版本和密碼套件。
4.證書交換	系統日誌伺服器顯示其伺服器證書，如果需要，顯示任何中間CA證書。
5.證書驗證	Nexus根據已配置的受信任CA驗證伺服器證書。驗證包括通過通用名稱(CN)和主體替代名稱(SAN)檢查證書鏈、過期日期、頒發者和伺服器標識。
6.金鑰交換	兩個對等體都會交換加密資訊，以派生用於加密的共用會話金鑰。
7.已完成消息	每個對等體驗證握手是否成功完成，以及雙方是否派生相同的加密金鑰。
8.安全系統日誌傳輸	建立TLS會話後，系統日誌消息將作為加密的TLS應用資料記錄進行傳輸。

相關配置

Cisco Nexus配置

Cisco Nexus交換機必須配置安全系統日誌目標、包含受信任CA的信任點和適當的源介面。

```
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伺服器配置

本文檔中使用的Syslog伺服器是運行rsyslog並啟用TLS支援的Ubuntu伺服器。

伺服器配置包括以下功能：

偵聽TCP埠6514上的安全系統日誌連線。
在TLS握手期間呈現TLS伺服器證書。
使用與伺服器證書關聯的私鑰。
在TLS握手期間，為Cisco Nexus提供所需的證書鏈（如果適用）。
將收到的安全系統日誌消息儲存在本地日誌檔案中。

本文檔中使用的PKI實現包括：Server Certificate > Intermediate CA > Root CA

Cisco Nexus交換機通過信任點匯入並信任根CA證書，從而使其能夠驗證Syslog伺服器提供的證書鏈。



附註：並非每個部署都需要證書鏈。根據PKI的實現，系統日誌伺服器只能呈現自簽名證書、由根CA直接簽名的伺服器證書，或附帶一個或多個中間CA證書的伺服器證書。

疑難排解

目標

此故障排除過程的目的是隔離和識別防止Cisco Nexus NX-OS交換機成功與Syslog伺服器建立安全

Syslog TLS連線的故障。

該工作流程驗證通訊過程的每個階段，從IP連線開始，通過TCP連線、TLS協商、證書驗證以及最終的Secure Syslog消息傳輸進行操作。通過獨立驗證每一層，可以快速確定通訊失敗的位置。

拓撲說明

本文檔中使用的拓撲包括配置為安全系統日誌客戶端的Cisco Nexus交換機以及配置為安全系統日誌伺服器的Ubuntu伺服器。

Cisco Nexus交換機在埠6514上發起到系統日誌伺服器的TCP連線。建立TCP連線後，兩台裝置都會執行TLS握手。在此過程中，系統日誌伺服器會顯示其伺服器證書和任何所需的中間CA證書。Nexus交換機根據在其信任點中配置的受信任根CA來驗證證書鏈。

成功完成TLS握手後，將建立加密會話，並且Nexus交換機開始將安全系統日誌消息傳輸到Syslog伺服器。

本文檔中介紹的故障排除方法使用與連線建立相同的通訊序列，允許獨立驗證該過程的每個階段。

步驟1.驗證安全系統日誌配置

確認安全系統日誌在Cisco Nexus交換機上配置正確，並且交換機可識別已配置的系統日誌伺服器。

```
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#
```

步驟2.驗證信任點配置

驗證Cisco Nexus交換機是否信任用於簽署Syslog伺服器證書的CA。

驗證:

預期的Trustpoint存在。

根CA證書已安裝。

存在任何所需的中間CA證書。

證書未過期。

證書指紋與預期值匹配。

```
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#
```

如果相應的CA證書缺失或無效，則在TLS握手期間證書驗證將失敗。

步驟3.檢驗IP連通性

驗證Cisco Nexus交換機和安全系統日誌伺服器之間的第3層連線。

驗證:

- 成功的網際網路控制訊息通訊協定(ICMP)可達性。
- 正確的域名系統(DNS)解析。
- 無路由問題。
- 無資料包丟失。

```
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#
```

如果IP連線失敗，則無法建立TCP連線。

步驟4.檢驗TLS連通性

驗證Cisco Nexus交換機和安全系統日誌伺服器之間的端到端TLS連線。

此步驟驗證安全系統日誌連線的多個階段，包括：

- TCP連線
- TLS協商
- 伺服器證書演示
- 憑證鏈結
- 證書驗證
- 密碼套件協商

由於這些操作作為TLS握手的一部分發生，因此可以使用單個命令驗證所有這些操作。

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

此命令驗證什麼

驗證	說明
TCP連線	驗證Nexus是否可以建立到Syslog伺服器的TCP連線。
TLS握手	確認兩個對等體成功協商TLS會話。

驗證	說明
伺服器證書	顯示Syslog伺服器提供的證書。
憑證鏈結	顯示伺服器傳送的任何中間CA證書。
TLS版本	顯示協商的TLS版本。
密碼套件	顯示協商的加密演算法。
證書驗證	使用-CAfile選項時，根據指定的受信任CA驗證證書鏈。

預期成功輸出

```
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, OU = DCRS, CN = Cisco Lab Root CA
1 s:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
2 s:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Intermediate CA
3 s:C = US, ST = California, L = San Jose, O = "Cisco Systems, Inc.", OU = DCRS, CN = Cisco Lab Root CA, 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, OU = DCRS, CN = Cisco Lab Root CA, em
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#
```

封包擷取關聯

openssl s_client輸出可以與使用Ethanalyzer獲得的封包擷取相關聯，以便驗證TLS握手的每個階段。

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=3843477193 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

安全系統日誌TLS握手捕獲

TLS握手流程

```
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握手必須成功完成，才能交換任何系統日誌消息。如果握手失敗，則絕對不會生成安全系統日誌流量。

若要執行此驗證，請開啟與Cisco Nexus交換器的兩個CLI作業階段。在第一個會話中，使用以下命令啟動TLS連線：

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

同時，在第二個CLI會話中，啟動Ethanalyzer捕獲以監控TLS交換：

```

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

```

資料包捕獲必須顯示完整的TLS握手序列，包括TCP三次握手、客戶端Hello、伺服器Hello、證書、金鑰交換、完成消息，如果握手成功完成，則加密的TLS應用資料記錄中攜帶安全系統日誌消息。

驗證安全系統日誌消息傳輸

驗證TLS會話建立後，是否已將安全系統日誌消息從Cisco Nexus交換機成功傳輸到Syslog伺服器。

成功的TLS握手確認已建立加密通道；但是，它不能保證系統日誌消息正在生成或傳遞。

此步驟通過生成測試消息並確認其傳輸和接收來驗證完整的安全系統日誌工作流。

生成測試系統日誌消息

```
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
```

此命令將生成本地系統日誌消息，該消息必須通過現有TLS會話傳輸到已配置的安全系統日誌伺服器。

驗證加密的TLS應用程式資料

如果Ethanalyzer捕獲已在運行（如前一節所述），請驗證在生成測試消息後是否傳輸新的TLS應用程式資料包。

```
switch# ethanalyzer 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應用程式資料的存在表示系統日誌消息正在通過加密的TLS會話傳輸。由於負載已加密，Ethanalyzer無法解碼系統日誌消息的內容。



附註：TLS握手成功完成後，所有系統日誌消息都將封裝在加密的TLS應用資料記錄中。其內容在資料包捕獲中不可見。

驗證系統日誌伺服器上的消息接收

在Syslog伺服器上，驗證消息是否已成功接收並寫入目標日誌檔案。

```
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
```

生成的測試報文的出現可確認：

Cisco Nexus交換機成功生成了系統日誌消息。
郵件是通過加密的TLS會話傳輸的。
系統日誌伺服器已成功解密並處理該消息。
安全系統日誌配置端到端運行正常。

Ethalyzer中出現TLS應用程式資料證實TLS會話正在主動傳輸加密的應用程式流量。但是，安全系統日誌運行正常的確切證據是在系統日誌伺服器上成功接收生成的系統日誌消息。

結論

Cisco Nexus交換機上的安全系統日誌取決於成功的IP和TCP連線、TLS協商、證書驗證和加密消息傳遞。獨立驗證每一層有助於隔離故障點，並區分連通性、TLS、證書和消息傳輸問題。成功的TLS握手確認已建立加密會話，而接收系統日誌伺服器上生成的測試消息則確認端到端安全系統日誌操作。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。