

排除Nexus平台上的安全系统日志故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[拓扑](#)

[背景信息](#)

[传统系统日志](#)

[安全系统日志](#)

[证书链](#)

[为什么需要证书](#)

[安全系统日志TLS握手](#)

[相关配置](#)

[Cisco Nexus配置](#)

[Ubuntu系统日志服务器配置](#)

[故障排除](#)

[目标](#)

[拓扑说明](#)

[步骤1.检验安全系统日志配置](#)

[步骤2.检验信任点配置](#)

[步骤3.检验IP连通性](#)

[步骤4.检验TLS连通性](#)

[此命令验证什么](#)

[预期成功输出](#)

[数据包捕获关联](#)

[验证安全系统日志消息传输](#)

[生成测试系统日志消息](#)

[验证加密的TLS应用数据](#)

[验证系统日志服务器上的消息接收](#)

[结论](#)

简介

本文档介绍如何对Cisco Nexus交换机和第三方服务器之间的传输层安全(TLS)安全系统日志进行故障排除。

先决条件

要求

Cisco 建议您了解以下主题：

- NX-OS平台
- 安全系统日志
- 基础公共密钥基础设施(PKI)知识

使用的组件

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

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

拓扑



背景信息

传统系统日志

传统系统日志通过UDP（默认端口514）或TCP传输日志消息而不加密。由于通信以明文发送，因此日志消息可能会在传输过程中被截取或修改。传统系统日志也不提供服务器身份验证。

安全系统日志

安全系统日志使用带TLS的TCP (默认端口6514) 来提供系统日志消息的加密传输。在发送任何日志消息之前，客户端和服务端建立TLS会话，在此期间，服务器提供其证书，客户端根据受信任证书颁发机构(CA)验证证书链。成功完成TLS握手后，系统日志消息将通过加密通道传输。

与传统系统日志不同，安全系统日志不仅依赖于IP连接，还依赖成功的TLS握手、证书验证和密码协商，然后才可以交换任何系统日志消息。

证书链

证书链是在服务器证书与受信任的根证书颁发机构 (根CA) 之间建立信任所使用的证书序列。在TLS握手期间，系统日志服务器显示其服务器证书以及任何所需的中间CA证书。Cisco Nexus交换机使用其受信任的根CA验证整个证书链。

证书链并非总是必需的。

系统日志服务器提供的证书取决于其PKI的实施方式。例如：

自签名证书:服务器仅提供其自己的证书。在这种情况下，Cisco Nexus交换机必须直接信任该自签名证书。

不带中间CA的CA签名证书：服务器仅提供其服务器证书，因为它是由受信任的根CA直接签名的。

具有中间CA的CA签名证书：服务器提供其服务器证书以及一个或多个中间CA证书，允许Cisco Nexus交换机建立到受信任根CA的完整信任链。

无论PKI设计如何，Cisco Nexus交换机都必须能够在建立TLS会话之前验证服务器提供的证书。

如果信任链中的任何必需证书丢失、无效、已过期或由不受信任的CA颁发，则TLS握手会失败，并且无法建立安全系统日志连接。

为什么需要证书

安全系统日志依赖TLS来提供加密和服务端身份验证。在传输任何系统日志消息之前，Cisco Nexus交换机必须验证它是否与预期的系统日志服务器通信，而不是与未授权设备通信。

证书通过允许Nexus交换机在建立加密会话之前验证系统日志服务器的身份来提供此信任。成功验证证书链后，将创建安全TLS通道，并通过加密连接传输所有后续系统日志消息。

如果没有有效证书和受信任证书链，安全系统日志无法建立TLS会话，从而阻止安全交换日志消息

。

安全系统日志TLS握手

在传输任何系统日志消息之前，Cisco Nexus交换机和系统日志服务器必须成功建立TLS会话。在TLS握手期间，两台设备会协商TLS参数、建立共享加密密钥，并通过证书验证对系统日志服务器的身份进行身份验证。

只有在TLS握手成功完成后，系统才会通过加密连接传输系统日志消息。

TLS握手由以下阶段组成：

步骤	描述
1. TCP连接	Cisco Nexus交换机与安全系统日志服务器（默认端口6514）建立TCP连接。
2.客户端Hello	Nexus通过发送支持的TLS版本、密码套件和随机值来启动TLS握手。
3.服务器Hello	系统日志服务器选择用于会话的TLS版本和密码套件。
4.证书交换	系统日志服务器提供其服务器证书，如果需要，还提供任何中间CA证书。
5.证书验证	Nexus根据已配置的受信任CA验证服务器证书。验证包括通过公用名(CN)和主题备用名(SAN)检查证书链、过期日期、颁发者和服务器身份。
6.密钥交换	两个对等体交换加密信息，以派生用于加密的共享会话密钥。
7.已完成消息	每个对等体验证握手成功完成，并且两端获取了相同的加密密钥。
8.安全系统日志传输	建立TLS会话后，系统日志消息将作为加密TLS应用数据记录进行传输。

相关配置

Cisco Nexus配置

Cisco Nexus交换机必须配置安全系统日志目标、包含受信任CA的信任点和适当的源接口。

```
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系统日志服务器配置

本文档中使用的Syslog服务器是运行TLS支持启用的rsyslog的Ubuntu服务器。

服务器配置包括以下功能：

侦听TCP端口6514上的安全系统日志连接。
在TLS握手期间提供TLS服务器证书。
使用与服务器证书关联的私钥。
在TLS握手期间，为Cisco Nexus提供所需的证书链（如果适用）。
将收到的安全系统日志消息存储在本地日志文件中。

本文档中使用的PKI实施包括：Server Certificate > Intermediate CA > Root CA

Cisco Nexus交换机通过信任点导入和信任根CA证书，从而允许其验证系统日志服务器提供的证书链。



注意：并非每个部署都需要证书链。根据PKI实施，Syslog服务器只能提供自签名证书、由根CA直接签名的服务器证书，或附带一个或多个中间CA证书的服务器证书。

故障排除

目标

此故障排除过程的目的是隔离并识别阻止Cisco Nexus NX-OS交换机成功与Syslog服务器建立安全系统日志TLS连接的故障。

该工作流程验证通信过程的每个阶段，从IP连接开始，然后通过TCP连接、TLS协商、证书验证，最后是安全系统日志消息传输。通过独立检验每一层，可以快速确定通信故障的位置。

拓扑说明

本文档中使用的拓扑包括配置为安全系统日志客户端的Cisco Nexus交换机和配置为安全系统日志服务器的Ubuntu服务器。

Cisco Nexus交换机在端口6514上发起到系统日志服务器的TCP连接。建立TCP连接后，两台设备都会执行TLS握手。在此过程中，系统日志服务器显示其服务器证书和任何所需的中间CA证书。Nexus交换机根据在其信任点中配置的受信任根CA验证证书链。

成功完成TLS握手后，将建立加密会话，Nexus交换机开始将安全系统日志消息传输到系统日志服务器。

本文档中介绍的故障排除方法使用与连接建立相同的通信序列，允许独立验证该过程的每个阶段。

步骤1.检验安全系统日志配置

验证Cisco Nexus交换机上的安全系统日志配置是否正确，以及交换机是否识别已配置的系统日志服务器。

```
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#
```

步骤2.检验信任点配置

验证Cisco Nexus交换机信任用于签署Syslog服务器证书的CA。

验证：

预期信任点存在。

根CA证书已安装。

存在任何所需的中间CA证书。

证书未过期。

证书指纹与预期值匹配。

```
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=3BC9CD3374C4A29A6794AAAEBCFA270E3116C3D55
```

```
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#
```

如果相应的CA证书缺失或无效，则证书验证在TLS握手期间失败。

步骤3.检验IP连通性

验证Cisco Nexus交换机和安全系统日志服务器之间的第3层连接。

验证：

- 互联网控制消息协议(ICMP)可达性成功。

- 正确的域名系统(DNS)解析。
- 无路由问题。
- 无数据包丢失。

```
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连接失败，则无法建立TCP连接。

步骤4.检验TLS连通性

验证Cisco Nexus交换机和安全系统日志服务器之间的端到端TLS连接。

此步骤验证安全系统日志连接的多个阶段，包括：

- TCP连接
- TLS协商
- 服务器证书演示
- 证书链
- 证书验证
- 密码套件协商

由于这些操作作为TLS握手的一部分而发生，因此所有这些操作都可以通过单个命令进行验证。

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

此命令验证什么

验证	描述
TCP连接	验证Nexus是否可以建立到系统日志服务器的TCP连接。
TLS握手	确认两个对等体成功协商TLS会话。
服务器证书	显示系统日志服务器提供的证书。
证书链	显示服务器发送的任何中间CA证书。
TLS版本	显示协商的TLS版本。
密码套件	显示协商的加密算法。
证书验证	使用-CAfile选项时，根据指定的受信任CA验证证书链。

预期成功输出

```
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,
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:RS
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 success
Extended master secret: yes
---
switch#
```

数据包捕获关联

openssl s_client输出可以与使用Ethanalyzer获取的数据包捕获相关联，以验证TLS握手的每个阶段。

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	192.168.100.1	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	192.168.100.1	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

安全系统日志TLS握手捕获

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握手必须成功完成，才能交换任何系统日志消息。如果握手失败，则不会生成安全系统日志流量。

要执行此验证，请打开与Cisco Nexus交换机的两个CLI会话。在第一个会话中，使用以下命令启动TLS连接：

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

同时，在第二个CLI会话中，启动Ethanalyzer捕获以监控TLS交换：

```

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
```

数据包捕获必须显示完整的TLS握手序列，包括TCP三次握手、客户端Hello、服务器Hello、证书、密钥交换、完成消息，如果握手成功完成，则加密的TLS应用数据记录中携带安全系统日志消息。

验证安全系统日志消息传输

在建立TLS会话后，验证安全系统日志消息是否已成功从Cisco Nexus交换机传输到Syslog服务器。

成功的TLS握手确认已创建加密通道；但是，它不保证系统日志消息正在生成或传送。

此步骤通过生成测试消息并确认其传输和接收来验证完整的安全系统日志工作流程。

生成测试系统日志消息

```
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
```

此命令会生成本地系统日志消息，该消息必须通过现有TLS会话传输到已配置的安全系统日志服务器。

验证加密的TLS应用数据

如果Ethanalyzer捕获已在运行（如上一节所述），请验证在生成测试消息后是否传输新的TLS应用数据包。

```
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应用数据的存在表示系统日志消息正在通过加密的TLS会话传输。由于负载已加密，Ethanalyzer无法解码系统日志消息的内容。



注意：成功完成TLS握手后，所有系统日志消息都将封装在加密的TLS应用数据记录中。其内容在数据包捕获中不可见。

验证系统日志服务器上的消息接收

在Syslog服务器上，验证消息是否已成功接收并写入目标日志文件。

```
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
```

生成的测试消息证实：

Cisco Nexus交换机成功生成了系统日志消息。

邮件通过加密的TLS会话传输。

系统日志服务器成功解密并处理了消息。

安全系统日志配置端到端运行正常。

TLS应用数据在Ethanalyzer中的出现证实TLS会话正在主动传输加密的应用流量。但是，安全系统日志功能正常的确凿证据是在系统日志服务器上成功接收生成的系统日志消息。

结论

Cisco Nexus交换机上的安全系统日志取决于成功的IP和TCP连接、TLS协商、证书验证和加密消息传输。独立验证每一层有助于隔离故障点，区分连通性、TLS、证书和消息传输问题。成功的TLS握手确认已建立加密会话，而系统日志服务器上所生成的测试消息则确认端到端安全系统日志操作。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。