

# Catalyst 9000スイッチでのSLP Smart Transportのトラブルシューティング

## 内容

---

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[設定](#)

[トラブルシュート](#)

[関連情報](#)

---

## はじめに

このドキュメントでは、Catalyst 9000スイッチでSmart Transportを使用して、Smart Licensing Using Policy(SLP)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

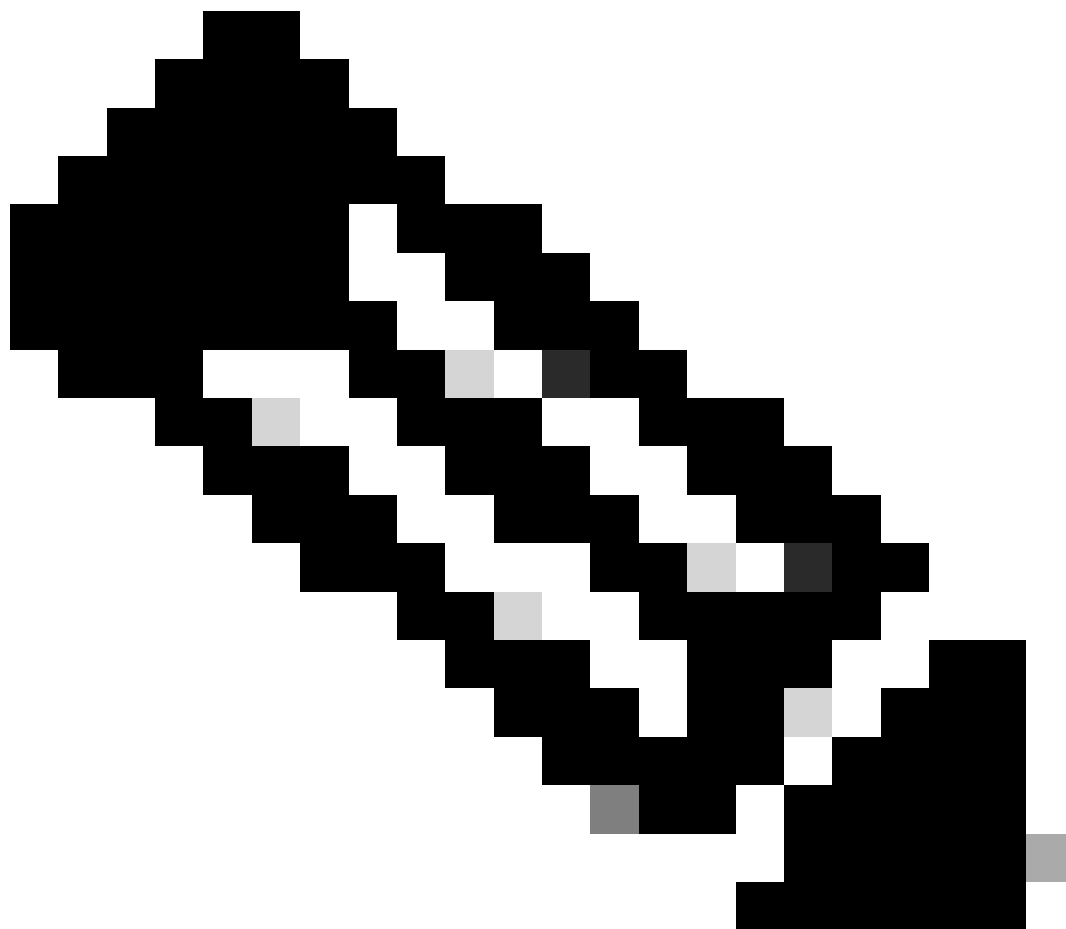
次の項目に関する知識と知識があることが推奨されます。

- Cisco IOS® XEデバイスでのポリシーを使用したスマートライセンス。

### 使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Catalyst 9200
- Catalyst 9300
- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS XE 17.9.1以降のバージョン



注：シスコの他のプラットフォームでこれらの機能を有効にするために使用されるコマンドについては、該当するコンフィギュレーション ガイドを参照してください。

---

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、クリアな（デフォルト）設定で作業を開始しています。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。

## 背景説明

Smart Licensing Using Policyは、Smart Licensingの拡張版で、ネットワークの運用を中断させないライセンスソリューションを提供することを主な目的として設計されています。システムの停止を伴うことなく、購入および使用するハードウェアおよびソフトウェアライセンスに対応したコンプライアンス関係を確立します。

スマートライセンスレポートを正しく機能させるために、Catalyst 9000スイッチはCisco Smart Software Manager(CSSM)に接続してライセンスの使用状況をレポートします。CSSMは、すべ

てのシスコソフトウェアライセンスを管理し、使用状況を確認するための中央集中型プラットフォームとして機能します。これにより、将来のライセンスのニーズを計画するのに役立ちます。

Catalyst 9000スイッチをCSSMに接続するトポロジは複数ありますが、このドキュメントでは、スイッチがCSSMに直接接続されている特定のトポロジを中心に説明します。つまり、スイッチは、インターネット上でホストされているCSSMに到達して接続を確立できる必要があります。

この直接接続トポロジには、スマートトランスポートとCall Homeの2つの転送オプションがあり、スマートトランスポートが推奨される方法です。さらに、Smart TransportではHTTPSプロキシの使用がサポートされており、CSSMとのSmart Licensing通信を処理する特定のVRFを選択できます。

スマートトランスポートを使用する場合、Catalyst 9000スイッチは、HTTPSメッセージ内でJavaScript Object Notation(JSON)形式のCSSMとライセンス使用情報を交換します。この情報はRUMレポートと呼ばれ、スイッチによって送信され、CSSMからの応答はACKと呼ばれます。

このトポロジの最小レポート頻度は1日に制限されます。つまり、製品インスタンスが1日に送信するRUMレポートは1つだけであり、特定のライセンスに対して過剰な数のレポートが生成および送信されるのを防ぎます。これにより、RUMレポートのオーバープロダクションによるメモリ関連の問題やシステムの速度低下を解決できます。

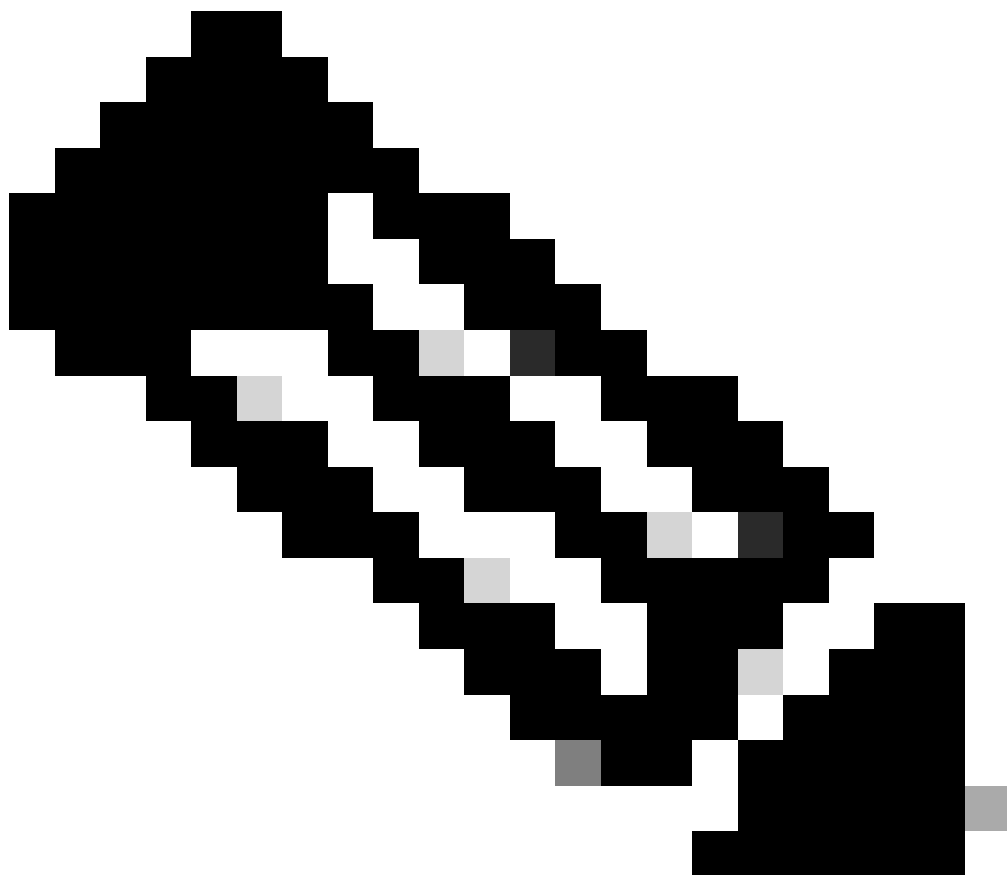
必要な場合は、特権EXECモードで`license smart sync`コマンドを使用して、このスロットリング制限を上書きできます

## 設定

スマートトランスポートを使用してSLPを設定するには、次の手順に従います。

1. トランスポートタイプとしてSmartを設定し、デフォルトURLを使用します。

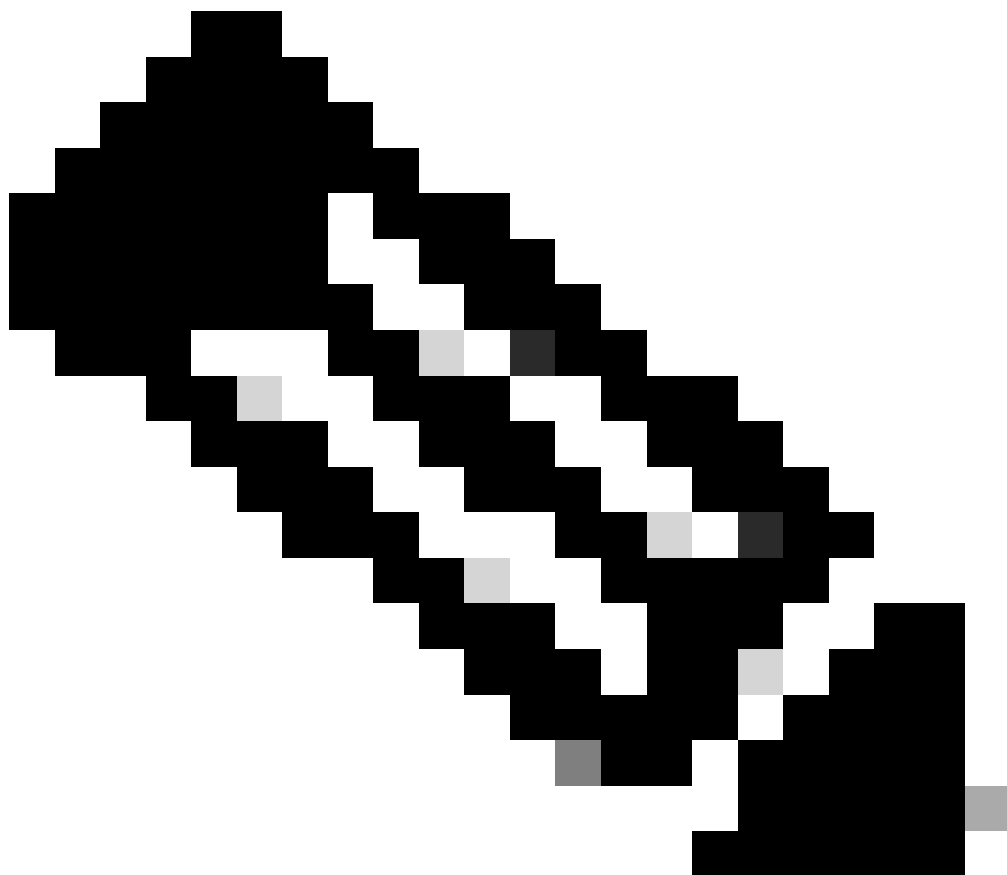
```
Switch#configure terminal
Switch(config)#license smart transport smart
Switch(config)#license smart url default
```



注：スマートトランスポートにVRFを使用する場合は、license smart vrf  
を設定する必要があります。

- 
2. DNS解決およびHTTPクライアント接続用のDNSサーバと送信元インターフェイスを設定します。

```
Switch(config)#ip domain lookup
Switch(config)#ip name-server 10.31.104.74
Switch(config)#ip domain name cisco.com
Switch(config)#ip domain lookup source-interface Vlan10
```

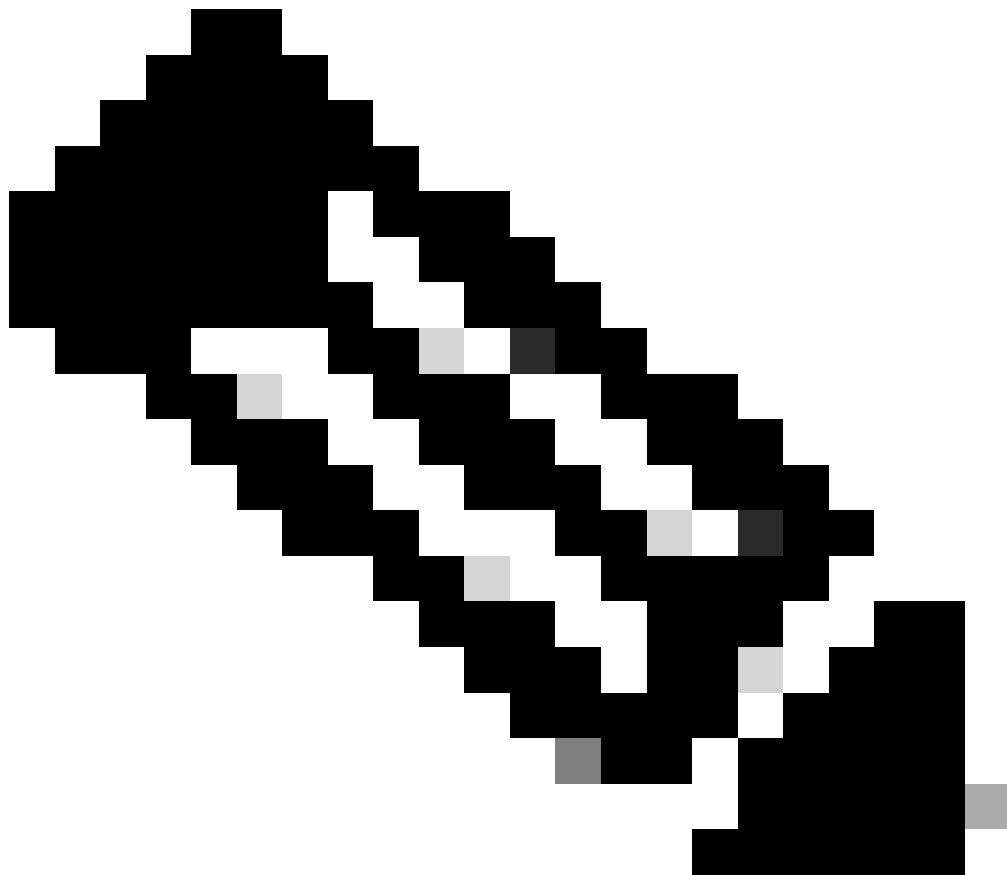


注：スマートトランスポートにVRFを使用する場合、これらのコマンドのVRFバリエーションを使用する必要があります。

---

### 3. 必要に応じて、HTTPSプロキシを設定します。

```
Switch(config)#license smart proxy address 192.168.217.105  
Switch(config)#license smart proxy port 80
```



注：プロキシサーバは、そのIPアドレスまたはホスト名を使用して設定できます。

- 
4. 仮想アカウントでトークンを生成します。この手順を実行するには、[このドキュメント](#)に記載されているプロセスに従ってください。
  5. スイッチに信頼コードをインストールします。

```
Switch#license smart trust idtoken NGFkODgzMGUtZmNkMS00NTRjLWI5MjUtYjI0YWYzZjU1ZGQzLTE3NDAYNjU5%OA  
[OK]
```

CSSMを使用して信頼が正常に確立されると、次のようなsyslogが表示されます。

```
*Jan 24 23:19:05.144: %SMART_LIC-6-TRUST_INSTALL_SUCCESS: A new licensing trust code was successfu
```

また、`show license status`は信頼コードのインストール時間を示します。

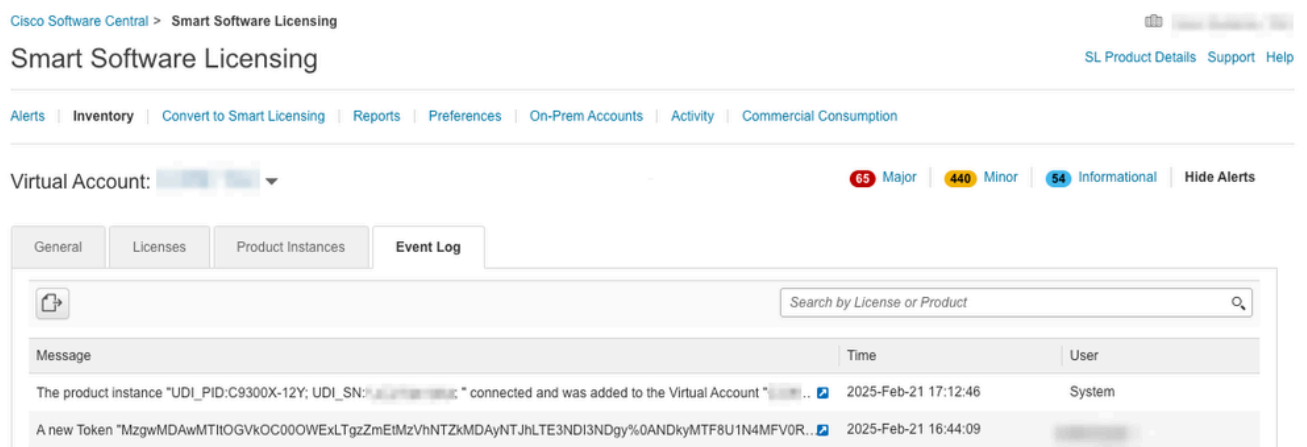
```
<#root>
```

```
Switch#show license status | i Trust
```

```
Trust Code Installed: Jan 24 23:19:05 2025 UTC
```

```
<--- Trust code was installed
```

最後に、仮想アカウントイベントログに、スイッチが追加されたことが示されます。



デバイスを示す仮想アカウントイベントログ。

## トラブルシューティング

CSSMによる信頼の確立が失敗すると、失敗の理由を示すsyslogが表示されます。

\*Jan 24 15:17:46.341: %SMART\_LIC-3-COMM\_FAILED: Communications failure with the Cisco Smart Software Ma

通信障害の考えられる原因には、次のものがあります。

- サーバホスト名/ドメイン名を解決できません：スマートトランスポートURLまたはプロキシサーバのホスト名がDNSによって解決されませんでした。DNSサーバの名前解決の設定と到達可能性を検証します。
- 接続がタイムアウトしました：接続を試行しましたが、応答がありませんでした。パケットキャプチャを使用してCSSMとのHTTPS接続が確立されていることを確認し、

プロキシサーバやファイアウォールなどのデバイスが接続をブロックしているかどうかを確認します。プロキシサーバを使用している場合は、正しいポートが使用されていることを確認します。

次の手順と検証コマンドに従って、Smart Transportを使用したSLPをトラブルシューティングします。

## 1. スマートライセンスのイベントログを検証します。

```
<#root>
```

```
Switch#
```

```
show license eventlog 1
```

```
**** Event Log ****
```

```
2025-01-24 13:58:23.900 UTC SAEVT_INIT_START version="5.5.29_rel/114"
2025-01-24 13:58:23.922 UTC SAEVT_INIT_CRYPT0 success="False" error="Crypto Initialization has not
2025-01-24 13:58:23.922 UTC SAEVT_HA_EVENT eventType="SmartAgentEvtHArmRegister"
2025-01-24 13:58:27.620 UTC SAEVT_READY
2025-01-24 13:58:27.621 UTC SAEVT_ENABLED
2025-01-24 13:58:27.665 UTC SAEVT_EXPORT_FLAG exportAllowed="False"
2025-01-24 13:58:27.732 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.742 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.742 UTC SAEVT_TAG_AUTHORIZED count="1" entitlementTag="regid.2017-05.com.cisco
2025-01-24 13:58:27.744 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.744 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_LICENS
2025-01-24 13:58:27.763 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.767 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.767 UTC SAEVT_TAG_AUTHORIZED count="1" entitlementTag="regid.2017-05.com.cisco
2025-01-24 13:58:27.767 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_SYSDAT
2025-01-24 13:58:27.768 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_LICENS
2025-01-24 13:58:30.425 UTC SAEVT_HA_EVENT eventType="SmartAgentEvtHArmInitialize"
2025-01-24 13:58:30.431 UTC SAEVT_HA_CHASSIS_ROLE udi="PID:C9300-48UN,SN:<SN>"
2025-01-24 13:58:30.431 UTC SAEVT_HA_EVENT eventType="SmartAgentEvtHArmRegister"
2025-01-24 13:58:37.975 UTC SAEVT_HA_ROLE udi="PID:C9300-48UN,SN:<SN>" haRole="Active"
2025-01-24 13:58:38.048 UTC SAEVT_HA_CHASSIS_ROLE udi="PID:C9300-48UN,SN:<SN>" haRole="Active"
2025-01-24 13:58:38.048 UTC SAEVT_HA_ROLE udi="PID:C9300-48UN,SN:<SN>" haRole="Active"
2025-01-24 13:58:38.062 UTC SAEVT_INIT_CONFIG_READ_BEGIN
2025-01-24 13:58:40.884 UTC SAEVT_HOSTNAME_CHANGE
2025-01-24 13:58:41.734 UTC SAEVT_HA_EVENT eventType="SmartAgentSetNVPairs"
2025-01-24 13:58:42.408 UTC SAEVT_INIT_CONFIG_READ_DONE
2025-01-24 13:58:42.531 UTC SAEVT_PLATFORM eventSource="INFRA_SL" eventName="INFRA_SL_EVLOG_OIR_AD
2025-01-24 13:58:42.531 UTC SAEVT_HA_CONFIG
2025-01-24 13:58:42.531 UTC SAEVT_HA_UDI udi="PID:C9300-48UN,SN:<SN>" haRole="Active"
2025-01-24 13:58:42.732 UTC SAEVT_LICENSE_USAGE count="0" type="destroy" entitlementTag="regid.201
2025-01-24 13:58:42.744 UTC SAEVT_LICENSE_USAGE count="0" type="destroy" entitlementTag="regid.201
2025-01-24 13:58:43.140 UTC SAEVT_INIT_SYSTEM_INIT
2025-01-24 13:58:44.143 UTC SAEVT_INIT_CRYPT0 success="False" error="Crypto Initialization has not
2025-01-24 13:59:14.143 UTC SAEVT_INIT_CRYPT0 success="True"
2025-01-24 13:59:14.144 UTC SAEVT_COMM_RESTORED
2025-01-24 13:59:14.176 UTC SAEVT_INIT_COMPLETE
2025-01-24 14:00:14.145 UTC SAEVT_PRIVACY_CHANGED enabled="True"
2025-01-24 14:00:27.432 UTC
```

```
SAEVT_UTILITY_REPORT_START
```

```
2025-01-24 15:17:46.341 UTC
```



```
SAEVT_COMM_FAIL error="Connection timed out". <--- Connection timed out
2025-01-24 15:35:22.627 UTC SAEVT_COMM_RESTORED <--- Communication with CSSM restored
```

2. pingを送信し、Telnet接続を試行して、CSSMとの接続を検証します。これにより、名前解決が実行されていること、およびスイッチとCSSMの間に管理ブロックが存在しないことを確認できます。

```
<#root>
```

```
Switch#
```

```
show ip interface brief | exclude unassigned
```

| Interface | IP-Address    | OK? | Method | Status | Protocol |
|-----------|---------------|-----|--------|--------|----------|
| Vlan10    | 10.31.121.118 |     |        |        |          |
| YES DHCP  | up            |     | up     |        |          |

```
Switch#
```

```
ping smartreceiver.cisco.com source Vlan10
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to X.X.X.X, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5)
```

```
, round-trip min/avg/max = 364/365/368 ms
```

```
Switch#
```

```
telnet smartreceiver.cisco.com 443 /ipv4 /source-interface vlan10
```

```
Trying X.X.X.X, 80 ...
```

```
Open
```

```
[Connection to X.X.X.X closed by foreign host]
```

また、プロキシサーバを使用している場合は、プロキシサーバのIPアドレスとポートを使用して同じコマンドを実行できます。

```
<#root>
```

```
Switch#
```

```
ping 192.168.217.105
```

Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.217.105, timeout is 2 seconds:  
!!!!

**Success rate is 100 percent (5/5)**

, round-trip min/avg/max = 364/365/368 ms

Switch#

**telnet 192.168.217.105 80 /ipv4 /source-interface vlan10**

Trying 192.168.217.105, 80 ...

**Open**

[Connection to 192.168.217.105 closed by foreign host]

### 3. RUMレポートが送信されていることと、応答があるかどうかを検証します。

<#root>

Switch#

**show license history message**

Message History (oldest to newest):

=====

Trust Establishment:

**REQUEST: Jan 24 23:18:59 2025 UTC <--- RUM report was sent**

{"request":{"header":{"request\_type":"ID\_TOKEN\_TRUST","sudi":{"udi\_pid":"C9300-48UN",

**RESPONSE: Jan 24 23:19:05 2025 UTC <--- Response from CSSM was received**

{"signature":{"type":null,"value":null,"piid":null,"cert\_sn":null},"response":{"header":{"vers

Usage Reporting:

No past history

Result Polling:

No past history

Authorization Request:

No past history

Authorization Return:

No past history

Trust Sync:

No past history

Import Message History (oldest to newest):

=====

Import POLICY:

No past Import history

Import AUTH:

No past Import history

Import TRUST CODE:

Received on Jan 24 23:19:05 2025 UTC

<TRUST\_CODE>

Import RUM ACK:

No past Import history

Import CONVERSION ACK:

No past Import history

Import ACCOUNT INFO:

Last policy received on Jan 24 23:19:05 2025 UTC

<ACCOUNT\_INFO>

Switch#

**show license tech support | sec Trust**

Trust Establishment:

Attempts: Total=1,

**Success=1**

, Fail=0 Ongoing Failure: Overall=0 Communication=0

**Last Response: OK on Jan 24 23:19:05 2025 UTC <--- Trust establishment succeeded**

Failure Reason:

Last Success Time: Jan 24 23:19:05 2025 UTC

Last Failure Time:

Trust Acknowledgement:

Attempts: Total=0, Success=0, Fail=0 Ongoing Failure: Overall=0 Communication=0

Last Response:

Failure Reason:

Last Success Time:

Last Failure Time:

Trust Sync:

Attempts: Total=1,

**Success=1,**

Fail=0 Ongoing Failure: Overall=0 Communication=0

**Last Response: OK on Jan 24 23:19:50 2025 UTC**

Failure Reason:

Last Success Time: Jan 24 23:19:50 2025 UTC

Last Failure Time:

Trusted Store Interface: True

**Local Device: P:C9300-48UN,S:**

**, state[2], Trust Data INSTALLED TrustId:612  
Overall Trust: INSTALLED (2) <--- Trust code installed**

要求が送信されたが、CSSMからの応答がない場合、show license history message コマンドでは特定の応答のJSONデータが表示されず、障害に関するより詳細な情報を取得できません。

```
<#root>
```

```
Switch#
```

```
show license history message
```

```
! <--- Output omitted for brevity --->
```

```
Trust Establishment:
```

```
REQUEST: Feb 21 16:54:49 2025 UTC
```

```
{"request":{"header":{"request_type":"ID_TOKEN_TRUST"},"sudi":{"udi_pid":"C9300X-12Y",
```

```
RESPONSE: Feb 21 16:54:49 2025 UTC
```

```
<--- The line is empty, which means there was no response from CSSM
```

```
REQUEST: Feb 21 16:55:19 2025 UTC
```

```
{"request":{"header":{"request_type":"ID_TOKEN_TRUST"},"sudi":{"udi_pid":"C9300X-12Y",
```

```
! <--- Output omitted for brevity --->
```

```
Switch#
```

```
show license tech support | sec Trust
```

```
Trust Establishment:
```

```
Attempts: Total=2, Success=0,
```

```
Fail=2
```

```
Ongoing Failure: Overall=2 Communication=2
```

```
Last Response: NO REPLY on Feb 21 16:55:39 2025 UTC
```

```
<--- Failure reason was NO REPLY
```

```
Failure Reason: <none>
```

```
Last Success Time: <none>
```

```
Last Failure Time: Feb 21 16:55:39 2025 UTC
```

```
Trust Acknowledgement:
```

```
Attempts: Total=0, Success=0, Fail=0 Ongoing Failure: Overall=0 Communication=0
Last Response: <none>
Failure Reason: <none>
Last Success Time: <none>
Last Failure Time: <none>
Trust Sync:
Attempts: Total=0, Success=0, Fail=0 Ongoing Failure: Overall=0 Communication=0
Last Response: <none>
Failure Reason: <none>
Last Success Time: <none>
Last Failure Time: <none>
Trusted Store Interface: True
Local Device: P:C9300-48UN,S:<SN>, state[1],
```

**NOT INSTALLED**

TrustId:605

**<--- Trust point exists but it is not installed yet**

Overall Trust: No ID

#### 4. CSSMと同期するためにRUMレポートを送信するようにスイッチをトリガーします。

**<#root>**

Switch#

**show clock**

\*23:38:54.683 UTC Fri Jan 24 2025

Switch#

**show license tech support | i Utility**

Utility:

Start Utility Measurements: Jan 24 23:35:55 2025 UTC (4 minutes, 38 seconds remaining)

**Send Utility RUM reports: Feb 23 23:20:56 2025 UTC (29 days, 23 hours, 49 minutes, 39 seconds remaining)**

Process Utility RUM reports: Jan 25 23:30:58 2025 UTC (23 hours, 59 minutes, 41 seconds remaining)

Switch#

**show license history message | i REQUEST**

REQUEST: Jan 24 23:18:59 2025 UTC

REQUEST: Jan 24 23:20:50 2025 UTC

REQUEST: Jan 24 23:25:55 2025 UTC

REQUEST: Jan 24 23:19:41 2025 UTC

Switch#

**license smart sync all <--- Trigger synchronization**

Switch#

**show license history message | i REQUEST**

REQUEST: Jan 24 23:18:59 2025 UTC  
REQUEST: Jan 24 23:20:50 2025 UTC  
REQUEST: Jan 24 23:25:55 2025 UTC  
REQUEST: Jan 24 23:19:41 2025 UTC

**REQUEST: Jan 24 23:39:05 2025 UTC**

**<--- New RUM report was sent**

Switch#

**show license tech support | sec Trust**

Trust Establishment:

Attempts: Total=1, Success=1, Fail=0 Ongoing Failure: Overall=0 Communication=0  
Last Response: OK on Jan 24 23:19:05 2025 UTC

Failure Reason:

Last Success Time: Jan 24 23:19:05 2025 UTC

Last Failure Time:

Trust Acknowledgement:

Attempts: Total=0, Success=0, Fail=0 Ongoing Failure: Overall=0 Communication=0

Last Response:

Failure Reason:

Last Success Time:

Last Failure Time:

Trust Sync:

Attempts:

**Total=2, Success=2**

, Fail=0 Ongoing Failure: Overall=0 Communication=0

**Last Response: OK on Jan 24 23:39:14 2025 UTC <--- Successful response from CSSM**

Failure Reason:

Last Success Time: Jan 24 23:39:14 2025 UTC

Last Failure Time:

Trusted Store Interface: True

Local Device: P:C9300-48UN,S:<SN>, state[2], Trust Data INSTALLED TrustId:612

Overall Trust: INSTALLED (2)

5. スイッチがCSSMとのHTTPS接続を確立していることを検証するために、パケットキャプチャを取得できます。次に、プロキシサーバを使用した接続の成功を示すパケットキャプチャの例を示します。

<#root>

Switch#

**sh ip cef 10.31.104.78**

0.0.0.0/0

nexthop 10.31.121.65 Vlan10

Switch#

sh ip arp 10.31.121.65

| Protocol | Address      | Age (min) | Hardware Addr | Type | Interface |
|----------|--------------|-----------|---------------|------|-----------|
| Internet | 10.31.121.65 | 0         |               |      |           |

2c31.24b1.6bc6

ARPA Vlan10

Switch#

show mac address-table address 2c31.24b1.6bc6

| Mac Address Table |                |         |       |
|-------------------|----------------|---------|-------|
| Vlan              | Mac Address    | Type    | Ports |
| 10                | 2c31.24b1.6bc6 | DYNAMIC |       |

Fi1/0/48

Total Mac Addresses for this criterion: 1

Switch#

monitor capture CSSM interface Fi1/0/48 both match any

Switch#

monitor capture CSSM start

Started capture point : CSSM

Switch#

show clock

\*

15:41:10.058 UTC

Fri Jan 24 2025

Switch#

license smart sync all

Switch#sh license hist mess | i REQUEST

REQUEST: Jan 24 15:35:17 2025 UTC

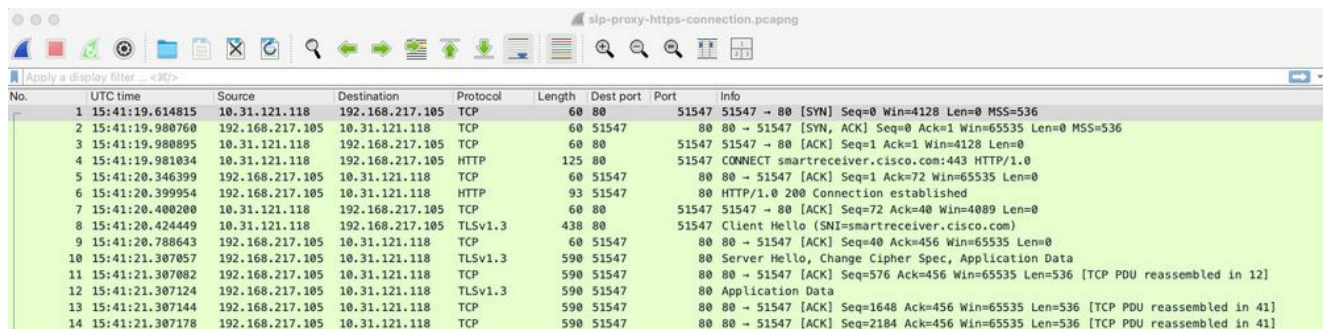
REQUEST: Jan 24 15:41:58 2025 UTC

Switch#

monitor capture CSSM stop

Switch#

monitor capture CSSM export location flash:slp-proxy-https-connection.pcap  
Export Started Successfully



| No. | UTC time        | Source          | Destination     | Protocol | Length | Dest port | Port  | Info                                                                            |
|-----|-----------------|-----------------|-----------------|----------|--------|-----------|-------|---------------------------------------------------------------------------------|
| 1   | 15:41:19.614815 | 10.31.121.118   | 192.168.217.105 | TCP      | 60     | 80        | 51547 | 51547 → 80 [SYN] Seq=0 Win=4128 Len=0 MSS=536                                   |
| 2   | 15:41:19.980760 | 192.168.217.105 | 10.31.121.118   | TCP      | 60     | 51547     | 80    | 80 → 51547 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=536                       |
| 3   | 15:41:19.980895 | 10.31.121.118   | 192.168.217.105 | TCP      | 60     | 80        | 51547 | 51547 → 80 [ACK] Seq=1 Ack=1 Win=4128 Len=0                                     |
| 4   | 15:41:19.981034 | 10.31.121.118   | 192.168.217.105 | HTTP     | 125    | 80        | 51547 | CONNECT smartreceiver.cisco.com:443 HTTP/1.0                                    |
| 5   | 15:41:20.346399 | 192.168.217.105 | 10.31.121.118   | TCP      | 60     | 51547     | 80    | 80 → 51547 [ACK] Seq=1 Ack=72 Win=65535 Len=0                                   |
| 6   | 15:41:20.399954 | 192.168.217.105 | 10.31.121.118   | HTTP     | 93     | 51547     | 80    | HTTP/1.0 200 Connection established                                             |
| 7   | 15:41:20.400200 | 10.31.121.118   | 192.168.217.105 | TCP      | 60     | 80        | 51547 | 51547 → 80 [ACK] Seq=72 Ack=40 Win=4089 Len=0                                   |
| 8   | 15:41:20.424449 | 10.31.121.118   | 192.168.217.105 | TLSv1.3  | 438    | 80        | 51547 | Client Hello (SNI=smartreceiver.cisco.com)                                      |
| 9   | 15:41:20.788643 | 192.168.217.105 | 10.31.121.118   | TCP      | 60     | 51547     | 80    | 80 → 51547 [ACK] Seq=40 Ack=456 Win=65535 Len=0                                 |
| 10  | 15:41:21.307057 | 192.168.217.105 | 10.31.121.118   | TLSv1.3  | 590    | 51547     | 80    | Server Hello, Change Cipher Spec, Application Data                              |
| 11  | 15:41:21.307082 | 192.168.217.105 | 10.31.121.118   | TCP      | 590    | 51547     | 80    | 80 → 51547 [ACK] Seq=576 Ack=456 Win=65535 Len=536 [TCP PDU reassembled in 12]  |
| 12  | 15:41:21.307124 | 192.168.217.105 | 10.31.121.118   | TLSv1.3  | 590    | 51547     | 80    | Application Data                                                                |
| 13  | 15:41:21.307144 | 192.168.217.105 | 10.31.121.118   | TCP      | 590    | 51547     | 80    | 80 → 51547 [ACK] Seq=1648 Ack=456 Win=65535 Len=536 [TCP PDU reassembled in 41] |
| 14  | 15:41:21.307178 | 192.168.217.105 | 10.31.121.118   | TCP      | 590    | 51547     | 80    | 80 → 51547 [ACK] Seq=2184 Ack=456 Win=65535 Len=536 [TCP PDU reassembled in 41] |

プロキシサーバを使用したCSSMとの正常なHTTPS接続のパケットキャプチャ

6. HTTPS接続の確立が試行されているが失敗したことをパケットキャプチャが示す場合は、TLSまたはSSLハンドシェイクの失敗が原因である可能性があります。これらのデバッグは、さらに調査するために使用できます。

```
debug ip http client all
debug ssl openssl states
debug ssl openssl errors
debug crypto pki messages
debug crypto pki transactions
```

7. インストールされている信頼コードをスイッチから削除する必要がある場合は、スマートライセンスの工場出荷時設定へのリセットを実行できます。このプロセスではリロードが必要です。工場出荷時設定にリセットした後、新しい信頼コードをインストールできます。このコマンドを使用すると、ポリシーを含むすべてのライセンス情報が削除されます。

```
<#root>
```

```
Switch#
```

```
license smart factory reset
```

```
%Warning: reload required after "license smart factory reset" command
```

```
Switch#
```

```
show license status | include Trust
```

```
Trust Code Installed:
```

```
<--- Installed trust code now shows none
```



Switch#

reload

## 関連情報

- [Cisco Catalyst 9000シリーズスイッチのポリシーを使用したスマートライセンス](#)
- [CatalystスイッチのSmart Licensingについて](#)
- [Cisco Live：ポリシーを使用したスマートライセンスの概要](#)
- [シスコのテクニカルサポートとダウンロード](#)

## 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。