

# EEMを利用してユーザへの安全な電子メールを自動化

## 内容

---

[はじめに](#)

[使用例](#)

[背景](#)

[Gmailアカウントの設定](#)

[ベースEEMの設定](#)

[デフォルトの証明書のみがインストールされている場合の問題](#)

[SMTPを保護するための証明書](#)

[証明書をより簡単に見つける方法](#)

[セキュアSMTPを使用したEEMの再テスト](#)

[その他の注意事項と考慮事項](#)

[@記号を含むユーザー名](#)

[結論](#)

---

## はじめに

このドキュメントでは、Cisco IOS® XE内のEmbedded Event Manager(EEM)の「メールサーバ」アクションを利用して、ポート587でTransport Layer Security(TLS)を使用するSimple Mail Transfer Protocol(SMTP)サーバに**セキュア**な電子メールを送信するために必要なプロセスについて説明します。

このプロセス中に発生する可能性のある注意事項が多数あるため、この記事は、このプロセスを実行するために必要な手順を文書化するように作成されています。

## 使用例

多くのお客様は、特定のイベントが発生した後に電子メール通知を自動的に受信することに価値を見出しています。EEMサブシステムは、ネットワークイベントの検出とオンボードの自動化を実現する強力なツールであり、Cisco IOS XEデバイスで電子メール通知を自動化する効率的な方法を提供します。たとえば、IPSLAトラックを監視し、状態の変化を示すsyslogに応答して、何らかのアクションを実行し、ネットワーク管理者に電子メールでイベントのアラートを送信することができます。この「電子メール通知」の考え方は、他の多くのシナリオにも適用でき、ハイライトしたい特定のイベントに注意を喚起する手段となります。

## 背景

PEMは「Privacy Enhanced Mail (プライバシー強化メール)」の略で、証明書と鍵を表すためによく使用される形式です。これは、Cisco IOS XEデバイスが使用する証明書形式です。セキュ

アなアプリケーション ( HTTPSやセキュアSMTPなど ) では、次のような複数の証明書が含まれる「スタックPEM」が使用されることがよくあります。

- ルート証明書
- 署名 ( 中間 ) 証明書
- エンドユーザ ( またはサーバ ) 証明書

## Gmailアカウントの設定

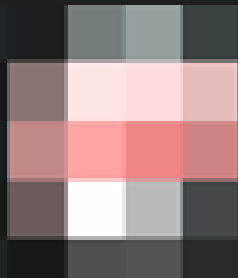
この記事では、GoogleのSMTPサービスを例として使用します。前提条件は、Gmailアカウントが以前に設定されていることです。

Googleを使用すると、リモートクライアントからGmailに電子メールを送信できます。以前はGmailに「保護されていないアプリ」の設定があり、この設定がGoogle側で許可されていない場合、アプリケーションはエラーに直面していました。この設定は削除され、代わりに「アプリケーションの保護」オプションが用意されています。このオプションには、次の方法でアクセスできます。

mail.google.com > プロフィールをクリック(#1) > Googleアカウントの管理(#2) > セキュリティ(#3) > Googleへのサインイン方法 > 2段階認証(#4)



1



Manage your Google Account

2



Add another account



Sign out

[Privacy Policy](#) • [Terms of Service](#)

- Home
- Personal info
- Data & privacy
- Security**
- People & sharing
- Payments & subscriptions
- About

## Security

Settings and recommendations to help you keep your account secure

### You have security tips

Security tips found in the Security Checkup



[Review security tips](#)

### Recent security activity

New sign-in on Mac

3:55 PM



[Review security activity](#)

### How you sign in to Google

Make sure you can always access your Google Account by keeping this information up to date

2-Step Verification



On since Jul 20, [blurred]



このページで、2段階認証がオンになっていることを確認します。

## ← 2-Step Verification

2-Step Verification is ON since Jul 20, [blurred]

「アプリパスワード」まで下にスクロールして、2段階認証をサポートしていないアプリケーションからGoogleアカウントにサインインするために使用できるパスワードをGmailに生成させることができます。

## App passwords

App Passwords aren't recommended and are unnecessary in most cases. To help keep your account secure, use "Sign in with Google" to connect apps to your Google Account.

### App passwords

None



## ← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.

Mail



Select device

iPhone

iPad

BlackBerry

Mac

Windows Phone

Windows Computer

Other (*Custom name*)

GENERATE

## ← App passwords

---

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

You don't have any app passwords.

Select the app and device you want to generate the app password for.


MyRouter ×

GENERATE

## ← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

### Your app passwords

Name	Created	Last used	
MyRouter	4:03 PM	-	

Select the app and device you want to generate the app password for.

Select app

Select device

GENERATE

### Generated app password

#### Your app password for your device



#### How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above. Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

DONE

このスクリーンショットの16文字のアプリケーションパスワードは、個人のGmailアカウントに結び付けられているため、ぼやけています。

Gmailのアプリケーションパスワードを取得したので、これをGmailアカウント名と共に、電子メールの転送に使用する電子メールサーバーとして使用できます。 サーバを指定する形式は「username:password@host」です。

## ベースEEMの設定

ニーズに合わせてEEMスクリプトをカスタマイズする方法は多数ありますが、この例は、安全な電子メール機能を実行するための基本的なEEMスクリプトです。

```
(config)# event manager environment _email_from <username@gmail.com>
(config)# event manager environment _email_to <EMAIL@domain.com>
(config)# event manager environment _email_server <username>:<password>@smtp.gmail.com

(config)# event manager applet SendSecureEmailEEM
(config-applet)# event none
(config-applet)# action 0010 mail server "$_email_server" to "$_email_to" from "$_email_from" cc "$_
```

設定では、最初に3つのEEM環境変数（\_email\_from、\_email\_to、および\_email\_server）を作成します。 各変数は、設定の変更を容易にするために変数で定義されます。 次に、SendSecureEmailEEMスクリプトを作成します。 ここでトリガーされるイベントは「none」です。これにより、特定のイベントがトリガーされるのを待つのではなく、「# event manager run SendSecureEmailEEM」を使用して、EEMスクリプトを任意に手動で実行できます。 次に、電子メールの生成を処理する1つの「メールサーバ」アクションがあります。「secure tls」および「port 587」オプションは、Gmailサーバがリッスンするポート587でTLSをネゴシエートするようにデバイスに指示します。

また、「from」フィールドが有効であることも確認する必要があります。「Alice」として認証を行っているにもかかわらず、「Bob」から電子メールを送信しようとする、Aliceが他のユーザの電子メールアドレスをスプーフィングしているため、エラーが発生します。「from」フィールドは、サーバで電子メールを送信するために使用するアカウントに合わせる必要があります。

## デフォルトの証明書のみがインストールされている場合の問題

EEMはopensslを使用してSMTPサーバと接続します。 セキュアな通信のために、サーバはCisco IOSdで実行されているopensslに証明書を返信します。 IOSdは、その証明書に関連付けられたトラストポイントを探します。

Cisco IOS XEデバイスでは、Gmail SMTPサーバの証明書はデフォルトではインストールされません。 信頼を確立するには、手動でインポートする必要があります。 証明書がインストールされていないと、「不正な証明書」が原因でTLSハンドシェイクが失敗します。

次のデバッグは、証明書の問題のデバッグに非常に役立ちます。

```
debug event manager action mail
debug crypto pki API
```



```
debug crypto pki callbacks
debug crypto pki messages
debug crypto pki scep
debug crypto pki server
debug crypto pki transactions
debug crypto pki validation
debug ssl openssl errors
debug ssl openssl ext
debug ssl openssl msg
debug ssl openssl states
```

ルータでEmbedded Packet Capture(EPC)を開始すると、EEMがトリガーされたときに電子メールサーバとの間で送受信されるすべてのトラフィックをキャプチャできます。

! Trigger the EEM:

```
# event manager run SendSecureEmailEEM
```

<SNIP>

```
*Mar 15 21:51:32.798: CRYPTO_PKI: (A0693) Check for identical certs
```

```
*Mar 15 21:51:32.798: CRYPTO_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-
```

```
*Mar 15 21:51:32.798: CRYPTO_PKI: looking for cert in handle=7F41EE523CE0, digest=
94 40 D1 90 A0 A3 5D 47 E5 B5 31 F6 63 AD 1B 0A
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: Cert record not found for issuer serial.
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI : (A0693) Validating non-trusted cert
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: (A0693) Create a list of suitable trustpoints
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: crypto_pki_get_cert_record_by_issuer()
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: Unable to locate cert record by issuername
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: No trust point for cert issuer, looking up cert chain
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: (A0693) No suitable trustpoints found
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: (A0693) Removing verify context
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: destroying ca_req_context type PKI_VERIFY_CHAIN_CONTEXT,ident 32, ref
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: ca_req_context released
```

```
*Mar 15 21:51:32.799: CRYPTO_OPSSL: Certificate verification has failed
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: Rcvd request to end PKI session A0693.
```

```
*Mar 15 21:51:32.799: CRYPTO_PKI: PKI session A0693 has ended. Freeing all resources.
```

```
*Mar 15 21:51:32.800: >>> ??? [length 0005]
```

```
*Mar 15 21:51:32.800: 15 03 03 00 02
```

```
*Mar 15 21:51:32.800:
```

```
*Mar 15 21:51:32.800: >>> TLS 1.2 Alert [length 0002], fatal bad_certificate
```

```
*Mar 15 21:51:32.800: 02 2A
```

```
*Mar 15 21:51:32.800:
```

```
*Mar 15 21:51:32.800: SSL3 alert write:fatal:bad certificate
```

```
*Mar 15 21:51:32.801: P11:C_OpenSession slot 1 flags 6
```

```
*Mar 15 21:51:32.801: SSL_connect:error in error
```

```
*Mar 15 21:51:32.801: 0:error:1416F086:SSL routines:tls_process_server_certificate:certificate verify f
```

最終的に、opensslはSMTPサーバとのセキュアなTLSセッションを確立できないため、「不正な証明書」エラーをスローします。その結果、EEMの実行が停止します。

```
*Mar 15 21:51:32.801: %HA_EM-3-FMPD_SMTP: Error occurred when sending mail to SMTP server: username:pas
*Mar 15 21:51:32.802: %HA_EM-3-FMPD_ERROR: Error executing applet SendSecureEmailEEM statement 0010
```

この交換から文書化されたパケットキャプチャは、「NoCertificateInstalled.pcap」として添付されます。 ルータ(10.122.x.x)からGmail SMTPサーバ(142.251.163.xx)への最後のTLSパケットでは、デバッグで前述したのと同じ「無効な証明書」メッセージが原因で、TLSネゴシエーションが終了したことが示されています。

```
Frame 33: 61 bytes on wire (488 bits), 61 bytes captured (488 bits)
Ethernet II, Src: Cisco_a3:c5:f0 (74:86:0b:a3:c5:f0), Dst: Cisco_f0:44:45 (00:08:30:f0:44:45)
Internet Protocol Version 4, Src: 10.122.xx.xx, Dst: 142.251.163.xx
Transmission Control Protocol, Src Port: 13306, Dst Port: 587, Seq: 189, Ack: 4516, Len: 7
Transport Layer Security
TLsv1.2 Record Layer: Alert (Level: Fatal, Description: Bad Certificate)
Content Type: Alert (21)
Version: TLS 1.2 (0x0303)
Length: 2
Alert Message
Level: Fatal (2)
Description: Bad Certificate (42)
```

## SMTPを保護するための証明書

Cisco IOS XEデバイスがGmailのサーバを信頼できる証明書が不足しているため、デバイスのトラストポイントにこれらの証明書の1つまたはすべてをインストールすることで修正できます。

たとえば、前のテストの完全なデバッグでは、次の証明書検索が行われたことが示されています。

```
CRYPTO_PKI(Cert Lookup) issuer="cn=GTS CA 1C3,o=Google Trust Services LLC,c=US" serial number= 52 87 E0
CRYPTO_PKI(Cert Lookup) issuer="cn=GTS Root R1,o=Google Trust Services LLC,c=US" serial number= 02 03 B
CRYPTO_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-sa,c=BE" serial number=
```

これらの各発行者の証明書は、デバイスがGmail SMTPサーバとのセキュアなセッションを確立できるように、トラストポイントの下にインストールする必要があります。 次の設定を使用して、各発行者のトラストポイントを作成できます。

```
crypto pki trustpoint CA-GTS-1C3
  enrollment terminal
  revocation-check none
  chain-validation stop
```

```
crypto pki trustpoint CA-GTS-Root-R1
```

```
enrollment terminal
revocation-check none
chain-validation stop
```

```
crypto pki trustpoint CA-GlobalSign-Root
enrollment terminal
revocation-check none
chain-validation stop
```

```
crypto pki trustpoint CA-gmail-SMTP
enrollment terminal
revocation-check none
chain-validation stop
```

これで、各発行者のトラストポイントが設定されました。ただし、実際の証明書はまだ関連付けられていません。これらは基本的に空白のトラストポイントです。

```
# show run | sec crypto pki certificate chain CA-
crypto pki certificate chain CA-GTS-1C3
crypto pki certificate chain CA-GTS-Root-R1
crypto pki certificate chain CA-GlobalSign-Root
crypto pki certificate chain CA-gmail-SMTP
```

これらの証明書の場所を追跡し、デバイスにインストールする必要があります。

「Google Trust Services 1C3」をオンラインで検索すると、証明書のGoogle Trust Servicesリポジトリがすぐに表示されます。

<https://pki.goog/repository/>

そのページ上のすべての証明書を展開した後、「1C3」を検索し、「Action」ドロップダウンをクリックして、PEM証明書をダウンロードできます。

GTS CA <b>1C3</b>	RSA	23:ec:b0:3e:ec:17:33:8c:4e:33:a6:b4:8a:41:dc:3c:da:12:28:1b:bc:3f:f8:13:c0:58:9d:6c:c2:38:75:22	2027-09-30	Action ^
GTS CA 1D4	RSA	64:e2:86:b7:60:63:60:2a:37:2e:fd:60:cd:e8:db:26:56:a4:9e:e1:5e:825:4b:3d:6e:b5:fe:38:f4:28:8b		Preview Certificate View Certificate Details
GTS CA 1D8	RSA	c0:e8:b1:c1:95:cd:ff:7b:51:37:b9:ad:35:13:a6:12:0b:1d:bf:f4:9e:5e:8c:ea:32:73:bc:8d:76:18:77		Downloads Certificate (PEM) Certificate (DER) Partitioned CRLs (JSON)
GTS CA 1P5	RSA	97:d4:20:03:e1:32:55:29:46:09:7f:20:ef:95:5f:5b:1c:d5:70:aa:43:727:80:03:3a:65:ef:be:69:75:8d		
		11:c6:97:87:87:32:05:6d:e1:7c:1d:a1:34:e9:d2:b6:d2:3c:f1:de:95:b		

テキストエディタでダウンロードしたPEMファイルを開くと、これは先ほど作成したトラストポイントでCisco IOS XEデバイスにインポートできる証明書であることがわかります。

```
-----BEGIN CERTIFICATE-----
MIIFljCCA36gAwIBAgINAg08U1lrNMcY9QFQZjANBgkqhkiG9w0BAQsFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFN1cnZpY2VzIExMQzEU
<snip>
AJ2xDx8hcFH1mt0G/FX0Kw4zd8NLQsLxdxP8c4CU6x+7Nz/OAipmsHMDmQyYbDKw
juDEI/9bfU1lcKwrmz302+BtjjKAvpafkm0817tdufThcV4q508DIrGKZTqPwJN1
1IXNDw9bg1kWRxYtnCQ6yICmJhSFm/Y3m6xv+cXDB1Hz4n/FsRC6UfTd
-----END CERTIFICATE-----
```

設定コマンドを使用して、「CA-GTS-1C3」トラストポイントにインポートできます。

```
(config)# crypto pki authenticate CA-GTS-1C3

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

MIIFljCCA36gAwIBAgINAg08U1lrNMcY9QFQZjANBgkqhkiG9w0BAQsFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFN1cnZpY2VzIExMQzEU
<snip>
juDEI/9bfU1lcKwrmz302+BtjjKAvpafkm0817tdufThcV4q508DIrGKZTqPwJN1
1IXNDw9bg1kWRxYtnCQ6yICmJhSFm/Y3m6xv+cXDB1Hz4n/FsRC6UfTd

Certificate has the following attributes:
Fingerprint MD5: 178EF183 43CCC9E0 ECB0E38D 9DEA03D8
Fingerprint SHA1: 1E7EF647 CBA15028 1C608972 57102878 C4BD8CDC
Certificate validated - Signed by existing trustpoint CA certificate.

Trustpoint CA certificate accepted.
% Certificate successfully imported

(config)#
```

次に、証明書がインストールされたことを確認します。

```
# show run | sec crypto pki certificate chain CA-GTS-1C3
crypto pki certificate chain CA-GTS-1C3
certificate ca 0203BC53596B34C718F5015066
 30820596 3082037E A0030201 02020D02 03BC5359 6B34C718 F5015066 300D0609
 2A864886 F70D0101 0B050030 47310B30 09060355 04061302 55533122 30200603
 55040A13 19476F6F 676C6520 54727573 74205365 72766963 6573204C 4C433114
<snip>
E1715E2A E4EF0322 B18A653A 8FC09365 D485CD0F 0F5B8359 1647162D 9C243AC8
80A62614 859BF637 9BAC6FF9 C5C30651 F3E27FC5 B110BA51 F4DD
quit
```

```
#show crypto pki certificates verbose CA-GTS-1C3
CA Certificate
  Status: Available
  Version: 3
  Certificate Serial Number (hex): 0203BC53596B34C718F5015066
```

Certificate Usage: Signature  
Issuer:  
cn=GTS Root R1  
o=Google Trust Services LLC  
c=US  
Subject:  
cn=GTS CA 1C3  
o=Google Trust Services LLC  
c=US  
CRL Distribution Points:  
<http://crl.pki.goog/gtsr1/gtsr1.crl>  
Validity Date:  
start date: 00:00:42 UTC Aug 13 2020  
end date: 00:00:42 UTC Sep 30 2027  
Subject Key Info:  
Public Key Algorithm: rsaEncryption  
RSA Public Key: (2048 bit)  
Signature Algorithm: SHA256 with RSA Encryption  
Fingerprint MD5: 178EF183 43CCC9E0 ECBOE38D 9DEA03D8  
Fingerprint SHA1: 1E7EF647 CBA15028 1C608972 57102878 C4BD8CDC  
X509v3 extensions:  
X509v3 Key Usage: 86000000  
Digital Signature  
Key Cert Sign  
CRL Signature  
X509v3 Subject Key ID: 8A747FAF 85CDEE95 CD3D9CD0 E24614F3 71351D27  
X509v3 Basic Constraints:  
CA: TRUE  
X509v3 Authority Key ID: E4AF2B26 711A2B48 27852F52 662CEFF0 8913713E  
Authority Info Access:  
OCSP URL: <http://ocsp.pki.goog/gtsr1>  
CA ISSUERS: <http://pki.goog/repo/certs/gtsr1.der>  
X509v3 CertificatePolicies:  
Policy: 2.23.140.1.2.2  
Policy: 2.23.140.1.2.1  
Policy: 1.3.6.1.4.1.11129.2.5.3  
Qualifier ID: 1.3.6.1.5.5.7.2.1  
Qualifier Info: <https://pki.goog/repository/>  
Extended Key Usage:  
Client Auth  
Server Auth  
Cert install time: 02:31:20 UTC Mar 16 2023  
Cert install time in nsec: 1678933880873946880  
Associated Trustpoints: CA-GTS-1C3

次に、他の2つの発行者の証明書をインストールできます。

CA-GTS-Root-R1:

設定 :

[スポイラー](#) ( 参照用に強調表示 )

```
(config)# crypto pki authenticate CA-GTS-Root-R1
```

Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself

```
MIIFVzCAAz+gAwIBAgINAgP1k28xsBNJiGuiFzANBgkqhkiG9w0BAQwFADBHMQsw
CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZpY2VzIExMQzEU
<snip>
2tIMPNuzj-smhDYAPexZ3FL//2wmUsp08IFgV6dtxQ/PeEMMA3Kgq1bbC1j+Qa3bb
bP6MvPJwNQzcmRk13NFIRmPVNnGuV/u3gm3c
```

```
Certificate has the following attributes:
Fingerprint MD5: 05FED0BF 71A8A376 63DA01E0 D852DC40
Fingerprint SHA1: E58C1CC4 913B3863 4BE9106E E3AD8E6B 9DD9814A
```

```
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

```
(config)# end
```

(config)# crypto pki authenticate CA-GTS-Root-R1Base 64でエンコードされたCA証明書を入力し  
ます。空白行または「quit」という単語だけを入力して終了します。

```
qgewJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFNlcnZpY2VzIExMQzEU<snip>2tIMPNuzj-smhDYAP
peEMMA3Kgq1bbC1j+Qa3bbP6MvPJwNQzcmRk13NFIRmPVNnGuV/u3gm3c証明書の属性
: Fingerprint MD5: 05FED0BF 71A8A3763DA01E0 D852DC40 Fingerprint SHA e1: E58C1CC4
913B3863 4BE9106E E3AD8E6B 9DD9814A%この証明書を受け入れますか。[yes/no]:yesトラス
トポイントCA証明書が承認されました。%証明書が正常にインポートされました(config)# end
```

実行コンフィギュレーションの検証：

[スポイラー](#) (参照用に強調表示)

```
# show run | sec crypto pki certificate chain CA-GTS-Root-R1
crypto pki certificate chain CA-GTS-Root-R1
certificate ca 0203E5936F31B01349886BA217
 30820557 3082033F A0030201 02020D02 03E5936F 31B01349 886BA217 300D0609
 2A864886 F70D0101 0C050030 47310B30 09060355 04061302 55533122 30200603
<snip>
6775C119 3A2B474E D3428EFD 31C81666 DAD20C3C DBB38EC9 A10D800F 7B167714
BFFFDB09 94B293BC 205815E9 DB7143F3 DE10C300 DCA82A95 B6C2D63F 906B76DB
6CFE8CBC F270350C DC991935 DCD7C846 63D53671 AE57FBB7 826DDC
quit
```

```
#show run | sec crypto pki certificate chain CA-GTS-Root-R1crypto pki certificate chain CA-GTS-
Root-R1 certificate ca 0203E5936F31B01349886BA217 30820557 3082033F A0030201
02020D02 03E5936F 31B01349 886BA217300D0602 a864886 F70D0101 0C050030 47310B30
09060355 04061302 55533122 30200603 <snip> 6775C119 3A2B474E D3428EFD 31C81666
DAD20C3C DBB38EC9 A10D800F 7B167714 BFFFDB094B 293BC 205815E9 DB7143F3
DE10C300 DCA82A95 B6C2D63F 906B76DB 6CFE8CBC F270350C DC991935
DCD7C8463D53671 AE57FBB7 826DDC quit
```

暗号検証の表示：

[スポイラー](#) (参照用に強調表示)

```
# show crypto pki certificates verbose CA-GTS-Root-R1
```

## CA Certificate

Status: Available  
Version: 3  
Certificate Serial Number (hex): 0203E5936F31B01349886BA217  
Certificate Usage: Signature  
Issuer:  
  cn=GTS Root R1  
  o=Google Trust Services LLC  
  c=US  
Subject:  
  cn=GTS Root R1  
  o=Google Trust Services LLC  
  c=US  
Validity Date:  
  start date: 00:00:00 UTC Jun 22 2016  
  end date: 00:00:00 UTC Jun 22 2036  
Subject Key Info:  
  Public Key Algorithm: rsaEncryption  
  RSA Public Key: (4096 bit)  
Signature Algorithm: SHA384 with RSA Encryption  
Fingerprint MD5: 05FED0BF 71A8A376 63DA01E0 D852DC40  
Fingerprint SHA1: E58C1CC4 913B3863 4BE9106E E3AD8E6B 9DD9814A  
X509v3 extensions:  
  X509v3 Key Usage: 86000000  
    Digital Signature  
    Key Cert Sign  
    CRL Signature  
  X509v3 Subject Key ID: E4AF2B26 711A2B48 27852F52 662CEFF0 8913713E  
  X509v3 Basic Constraints:  
    CA: TRUE  
  Authority Info Access:  
    Cert install time: 14:39:38 UTC Mar 13 2023  
    Cert install time in nsec: 1678718378546968064  
    Associated Trustpoints: CA-GTS-Root-R1 Trustpool

# show crypto pki certificates verbose CA-GTS-Root-R1CA証明書の状態：使用可能なバージョン  
：3証明書シリアル番号 ( 16進数 ) : 0203E5936F31B01349886BA217証明書の使用方法：署名発  
行者：cn=GTS Root R1 o=Google Trust Services LLC c=US Subject: cn=GTS Root R1=Google  
Trust Services LLC=US有効期間：開始日：00:00:00 UTC Jun 22 2016 end date: 00:00:00 UTC  
Jun 22 2036 Subject Key Info: Public Key Algorithm: rsaEncryption RSA Public Key: ( 4096ビット )  
署名アルゴリズム：SHA384 with RSA Encryption Fingerprint MD5: 05FED0BF 71A8A36  
63DA01E0 D852DC40フィンガープリントSHA1: E58C1CC4 913B3863 4BE9106E E3AD8E6B  
9DD9814A X509v3拡張：X509v3キー使用法：86000000デジタル署名キー証明書署名CRL署名  
X509v3サブジェクトキーID: E4AF2B26 711A2B48 27852F52 662CEFF0 8913713E X509v3基  
本制約：CA:TRUE Authority Info Access：証明書のインストール時間：14:39:38 UTC Mar 13  
2023証明書のインストール時間 ( 単位：nsec ) :1678718378546968064関連トラストポイント  
：CA-GTS-Root-R1 Trustpool

CA-GlobalSign-Root:

この証明書は次の場所で見つかりました：

<https://support.globalsign.com/ca-certificates/root-certificates/globalsign-root-certificates>

設定：



## スポイラー ( 参照用に強調表示 )

```
(config)# crypto pki authenticate CA-GlobalSign-Root
```

Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself

```
MIIDdTCCA12gAwIBAgILBAAAAAABFUtaW5QwDQYJKoZIhvcNAQEFBQAwVzELMAkG
A1UEBHMtMCQkUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2ExEDAOBgNVBAsTB1Jv
<snip>
DKqC5J1R3XC321Y9YeRq4VzW9v493kHMB65jUr9TU/Qr6cf9tveCX4XSQRjbgbME
HMUfpIBvFSDJ3gyICh3WZlXi/EjJKSZp4A==
```

Certificate has the following attributes:

Fingerprint MD5: 3E455215 095192E1 B75D379F B187298A

Fingerprint SHA1: B1BC968B D4F49D62 2AA89A81 F2150152 A41D829C

% Do you accept this certificate? [yes/no]: yes

Trustpoint CA certificate accepted.

% Certificate successfully imported

```
(config)# end
```

(config)# crypto pki authenticate CA-GlobalSign-RootBase 64でエンコードされたCA証明書を入力  
します。空白行または「quit」という単語だけを入力して終了します。

```
kUxGTAXBgNVBAoTEEdsb2JhbFNpZ24gbnYtc2ExEDAOBgNVBAsTB1Jv<snip>DKqC5J1R3XC321Y9Ye
cf9tveCX4XSQRjbgbMEHMUfpIBvFSDJ3gyICh3WZlXi/EjJKSZp4A==証明書の属性：フィンガー
プリントMD5: 3E455215 095192E1 B75D379F B187298AフィンガープリントSHA1:
B1BC968B4F4 9D62 2AA89A81 F2150152 A41D829C%この証明書に同意しますか
? [yes/no]:yesトラストポイントCA証明書が承認されました。%証明書が正常にインポートされ
ました(config)# end
```

実行コンフィギュレーションの検証：

## スポイラー ( 参照用に強調表示 )

```
# show run | sec crypto pki certificate chain CA-GlobalSign-Root
```

```
crypto pki certificate chain CA-GlobalSign-Root
```

```
certificate ca 040000000001154B5AC394
```

```
30820375 3082025D A0030201 02020B04 00000000 01154B5A C394300D 06092A86
```

```
<snip>
```

```
2AC45631 95D06789 852BF96C A65D469D 0CAA82E4 9951DD70 B7DB563D 61E46AE1
```

```
5CD6F6FE 3DDE41CC 07AE6352 BF5353F4 2BE9C7FD B6F7825F 85D24118 DB81B304
```

```
1CC51FA4 806F1520 C9DE0C88 0A1DD666 55E2FC48 C9292669 E0
```

```
quit
```

```
#show run | sec crypto pki certificate chain CA-GlobalSign-Rootcrypto pki certificate chain CA-
GlobalSign-Root certificate ca 040000000001154B5AC394 30820375 3082025D A0030201
02020B04 00000000 01154B5A C394300D 06092A86 <snip> 2AC45631 95D06789 852BF96C
A65D469D 0CAA82E4 9 951DD70 B7DB563D 61E46AE1 5CD6F6FE 3DDE41CC 07AE6352
BF5353F4 2BE9C7FD B6F7825F 85D24118 DB81B304 1CC51FA4 06F1520 C9DE0C88
0A1DD666 55E2FC48 C9292669 E0終了
```



暗号検証の表示：

[スプイラー](#) ( 参照用に強調表示 )

```
#show crypto pki certificates verbose CA-GlobalSign-Root
CA Certificate
Status: Available
Version: 3
Certificate Serial Number (hex): 04000000001154B5AC394
Certificate Usage: Signature
Issuer:
cn=GlobalSign Root CA
ou=Root CA
o=GlobalSign nv-sa
c=BE
Subject:
cn=GlobalSign Root CA
ou=Root CA
o=GlobalSign nv-sa
c=BE
Validity Date:
start date: 12:00:00 UTC Sep 1 1998
end date: 12:00:00 UTC Jan 28 2028
Subject Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Signature Algorithm: SHA1 with RSA Encryption
Fingerprint MD5: 3E455215 095192E1 B75D379F B187298A
Fingerprint SHA1: B1BC968B D4F49D62 2AA89A81 F2150152 A41D829C
X509v3 extensions:
X509v3 Key Usage: 6000000
Key Cert Sign
CRL Signature
X509v3 Subject Key ID: 607B661A 450D97CA 89502F7D 04CD34A8 FFFCFD4B
X509v3 Basic Constraints:
CA: TRUE
Authority Info Access:
Cert install time: 03:03:01 UTC Mar 16 2023
Cert install time in nsec: 1678935781942944000
Associated Trustpoints: CA-GlobalSign-Root
```

```
#show crypto pki certificates verbose CA-GlobalSign-RootCA CertificateStatus: AvailableVersion:
3Certificate Serial Number (hex): 04000000001154B5AC394Certificate Usage: SignatureIssuer:
cn=GlobalSign Root CAou=Root CAo=GlobalSign nv-sac=BESubject: cn=GlobalSign Root
CAou=Root CAo=GlobalSign-nv sac=BEVAIIDVAIID ITY日付：開始日：12:00:00 UTC 9月1日
1998終了日：12:00:00 UTC Jan 28 2028件名キー情報：公開キーアルゴリズム：
rsaEncryptionRSA公開キー：(2048ビット)署名アルゴリズム：SHA1 with RSA
EncryptionFingerprint MD5: 3E455215E1 B75D379F095192 187298AフィンガープリントSHA1:
B1BC968B D4F49D62 2AA89A81 F2150152 A41D829C X509v3拡張：X509v3キー使用法：
6000000キー証明書SignCRL署名X509v3サブジェクトキーID: 607B661A 50D97CA 89502F7D
04CD34A8 FFFCFD4B X509v3基本制約：CA: TRUEAuthority情報アクセス：証明書インストール
時間：03:03:01 UTC Mar 16 2023証明書インストール時間(nsec): 1678935781942944000関連ト
ラストポイント：CA-GlobalSign-Root
```

CA-gmail-SMTP:

Gmailのサーバ(CA-gmail-SMTP)のTLS証明書が、次の手順を使用して見つかりました。「[セキュアな転送にTLS証明書を使用する](#)」

設定 :

[スポイラー](#) ( 参照用に強調表示 )

```
(ca-trustpoint)# crypto pki authenticate CA-gmail-SMTP
```

```
Enter the base 64 encoded CA certificate.  
End with a blank line or the word "quit" on a line by itself
```

```
MIIEhjCCA26gAwIBAgIQUofgQKT+9wcSaLBP3d3w9DANBgkqhkiG9w0BAQsFADBG  
MQswCQYDVQQGEwJVUzEiMCAgA1UEChMZR29vZ2x1IFRydXN0IFN1cnZpY2VzIEEM  
<snip>  
b1J2gZAyjd4nfFRG1jeL5KrsfUR9hIXufqySv1PUoPuKSi3fvsIS21BYEXEe8uZ  
gBxJaeTUjncvow==
```

```
Trustpoint 'CA-gmail-SMTP' is a subordinate CA.  
but certificate is not a CA certificate.  
Manual verification required  
Certificate has the following attributes:  
Fingerprint MD5: 19651FBE 906A414D 6D57B783 946F30A2  
Fingerprint SHA1: 4EF392CB EEB46D5E 47433953 AAEF313F 4C6D2825
```

```
% Do you accept this certificate? [yes/no]: yes  
Trustpoint CA certificate accepted.  
% Certificate successfully imported
```

```
(config)#
```

```
(ca-trustpoint)# crypto pki authenticate CA-gmail-SMTPEnter the base 64エンコードCA証明書  
.End with a blank line or or the line by
```

```
itselfMIIEhjCCA26gAwIBAgIQUofgQKT+9wcSaLBP3d3w9DANBgkqhkiG9w0BAQsFADGBMQswCQYDVQQG  
EwJVUzEiMCAgA1UEChMZR29vZ2x1IFRydXN0IFN1cnZpY2VzIEEMb1J2gZAyjd4nfFRG1jeL5KrsfUR9hIXufqySv1PUoPuKSi3fvsIS21BYEXEe8uZgBxJaeTUjncvow==  
i3fvsIS21BYEXEe8uZgBxJaeTUjncvow==トラストポイント'CA-gmail-SMTP'は下位CAですが、  
証明書がCA証明書ではありません。手動検証が必要な証明書の属性 : Fingerprint MD5:  
19651FBE 906A414D 6D57B783946F30A2 Fingerprint ef1: 4EF392CB EEB46D5E 47433953  
AAEF313F 4C6D2825%この証明書に同意しますか ? [yes/no]:yesトラストポイントCA証明書が承認  
されました。%証明書が正常にインポートされました(config)#
```

実行コンフィギュレーションの検証 :

[スポイラー](#) ( 参照用に強調表示 )

```
# show run | sec crypto pki certificate chain CA-gmail-SMTP  
crypto pki certificate chain CA-gmail-SMTP  
certificate ca 5287E040A4FEF7071268B04FDDDDF0F4  
30820486 3082036E A0030201 02021052 87E040A4 FEF70712 68B04FDD DDF0F430  
0D06092A 864886F7 0D01010B 05003046 310B3009 06035504 06130255 53312230  
<snip>  
92ABB1F5 11F61217 B9FAB24A F94F5283 EE2928B7 7EFB084B 6D416045 C47BCB99  
801C4969 E4D48E77 2FA3
```

quit

```
#show run | sec crypto pki証明書チェーンCA-gmail-SMTP crypto pki証明書チェーンCA-gmail-SMTP証明書チェーンCA 5287E040A4FEF7071268B04FDDDDF0F4 30820486 3082036E A0030201 02021052 87E040A4 FEF70712 68B04FDD DDF0F430 0D06092A 864886F7 0D01010B 05003046 311310B 3 009 06035504 06130255 53312230 <snip> 92ABB1F5 11F61217 B9FAB24A F94F5283 EE2928B7 7EFB084B 6D416045 C47BCB99 801C4969 E4D48E77 2FA3終了
```

暗号検証の表示：

スポイラー ( 参照用に強調表示 )

```
# show crypto pki certificates verbose CA-gmail-SMTP
CA Certificate
Status: Available
Version: 3
Certificate Serial Number (hex): 5287E040A4FEF7071268B04FDDDDF0F4
Certificate Usage: Signature
Issuer:
cn=GTS CA 1C3
o=Google Trust Services LLC
c=US
Subject:
cn=smtp.gmail.com
CRL Distribution Points:
http://crls.pki.goog/gts1c3/moVdfISia2k.crl
Validity Date:
start date: 09:15:03 UTC Feb 20 2023
end date: 09:15:02 UTC May 15 2023
Subject Key Info:
Public Key Algorithm: ecEncryption
EC Public Key: (256 bit)
Signature Algorithm: SHA256 with RSA Encryption
Fingerprint MD5: 19651FBE 906A414D 6D57B783 946F30A2
Fingerprint SHA1: 4EF392CB EEB46D5E 47433953 AAEF313F 4C6D2825
X509v3 extensions:
X509v3 Key Usage: 80000000
Digital Signature
X509v3 Subject Key ID: 5CC36972 D07FE997 510E1A67 8A8ECC23 E40CFB68
X509v3 Basic Constraints:
CA: FALSE
X509v3 Subject Alternative Name:
smtp.gmail.com
IP Address :
OtherNames :
X509v3 Authority Key ID: 8A747FAF 85CDEE95 CD3D9CD0 E24614F3 71351D27
Authority Info Access:
OCSP URL: http://ocsp.pki.goog/gts1c3
CA ISSUERS: http://pki.goog/repo/certs/gts1c3.der
X509v3 CertificatePolicies:
Policy: 2.23.140.1.2.1
Extended Key Usage:
Server Auth
Cert install time: 03:10:41 UTC Mar 16 2023
Cert install time in nsec: 1678936241822955008
Associated Trustpoints: CA-gmail-SMTP
```

```
# show crypto pki certificates verbose CA-gmail-SMTPCA CertificateStatus: AvailableVersion:
3Certificate Serial Number (hex): 5287E040A4FEF7071268B04FDDDF0F4Certificate Usage:
SignatureIssuer: cn=GTS CA 1C3o=Google Trust Services LLCc=USsubject:
cn=smtp.gmail.comCRLDistribution points: http://crls.pki.goog/gts1c3/moVDfISia2k.crlValidity
Date:: start date: 0 2023年2月20日UTC:15:03 UTC終了日 : 09:15:02 UTC:2023年5月15日件名キ
ー情報 : 公開キーアルゴリズム : ecEncryptionEC公開キー : ( 256ビット ) 署名アルゴリズム
: SHA256 with RSA EncryptionFingerprint MD5: 19651FBE 906A414D6D57B778 3 946F30A2フ
ィンガープリントSHA1: 4EF392CB EEB46D5E 47433953 AAEF313F 4C6D2825 X509v3拡張
: X509v3キー使用法 : 80000000デジタル署名X509v3サブジェクトキーID: 5CC36972
D07FE99751 0E1A67 8A8ECC23 E40CFB68 X509v3基本制約 : CA: FALSEX509v3サブジェクト
代替名 : smtp.gmail.com IPアドレス : その他の名前 : X509v3認証局キーID: 8A747FAF
85CDEE95 CD3D9CD0 E24614F3 71351 D27 Authority Info Access:OCSP
URL:http://ocsp.pki.goog/gts1c3CA ISSUERS:http://pki.goog/repo/certs/gts1c3.derX509v3
CertificatePolicies:Policy:2.23.140.1.2.1Extended Key Usage:Server AuthCert install time:03:10:41
UTC Mar 16 2023 Cert install time in nsec:1678936241822955008関連トラストポイント : CA-
gmail-SMTP
```

## 証明書をより簡単に見つける方法

または、デバッグを使用したり、Googleを検索して追跡したりすることなく、SMTPサーバから証明書を簡単に取得する方法として、サーバ/ラップトップからのopensslコールを使用することもできます。

```
openssl s_client -showcerts -verify 5 -connect gmail-smtp-in.l.google.com:25 -starttls smtp
```

use smtp.gmail.comも使用できます。

```
openssl s_client -showcerts -verify 5 -connect smtp.gmail.com:25 -starttls smtp
```

このコールからの出力には、「crypto pki authenticate <trustpoint>」設定に使用できる実際の証明書自体が含まれます。

## セキュアSMTPを使用したEEMの再テスト

証明書がCisco IOS XEデバイスに適用されると、EEMスクリプトはセキュリティで保護されたSMTPメッセージを期待どおりに送信します。

```
# event manager run SendSecureEmailEEM
```

スプーラで完全な暗号化およびsslデバッグ出力を確認します。

[スポイラー](#) ( 参照用に強調表示 )

```
# event manager run SendSecureEmailEEM
```

```
*Mar 16 03:28:50.673: CRYPTO_OPSSL: Allocated the memory for OPSSLContext
*Mar 16 03:28:50.673: CRYPTO_OPSSL: Set cipher specs to mask 0x02FC0000 for version 128
*Mar 16 03:28:50.674: Set the Default EC Curve list: 0x70Set the EC curve list: secp521r1:secp384r1:prime256v1
*Mar 16 03:28:50.674: opssl_SetPKIInfo entry
*Mar 16 03:28:50.674: CRYPTO_PKI: (A069B) Session started - identity selected (TP-self-signed-486541296)
*Mar 16 03:28:50.674: CRYPTO_PKI: Begin local cert chain retrieval.
*Mar 16 03:28:50.674: CRYPTO_PKI(Cert Lookup) issuer="cn=IOS-Self-Signed-Certificate-486541296" serial=1234567890

*Mar 16 03:28:50.674: CRYPTO_PKI: looking for cert in handle=7F41EE523CE0, digest=
1C 7F 3D 52 67 66 D5 59 E2 66 58 E7 8B E7 9B 8E

*Mar 16 03:28:50.675: CRYPTO_PKI: Done with local cert chain fetch 0.
*Mar 16 03:28:50.675: CRYPTO_PKI: Rcvd request to end PKI session A069B.
*Mar 16 03:28:50.675: CRYPTO_PKI: PKI session A069B has ended. Freeing all resources.TP-self-signed-486541296
*Mar 16 03:28:50.675: opssl_SetPKIInfo done.
*Mar 16 03:28:50.675: CRYPTO_OPSSL: Common Criteria is disabled on this session.Disabling Common Criteria

*Mar 16 03:28:50.675: CRYPTO_OPSSL: ciphersuites ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384
*Mar 16 03:28:50.676: Handshake start: before SSL initialization
*Mar 16 03:28:50.676: SSL_connect:before SSL initialization
*Mar 16 03:28:50.676: >>> ??? [length 0005]
*Mar 16 03:28:50.676: 16 03 01 00 95
*Mar 16 03:28:50.676:
*Mar 16 03:28:50.676: >>> TLS 1.2 Handshake [length 0095], ClientHello
*Mar 16 03:28:50.676: 01 00 00 91 03 03 26 4B 9F B3 44 94 FD 5F FD A1
<snip>
*Mar 16 03:28:50.679: 03 03 01 02 01
*Mar 16 03:28:50.679:
*Mar 16 03:28:50.679: SSL_connect:SSLv3/TLS write client hello
*Mar 16 03:28:50.692: <<< ??? [length 0005]
*Mar 16 03:28:50.692: 16 03 03 00 3F
*Mar 16 03:28:50.692:
*Mar 16 03:28:50.692: SSL_connect:SSLv3/TLS write client hello
*Mar 16 03:28:50.692: <<< TLS 1.2 Handshake [length 003F], ServerHello
*Mar 16 03:28:50.692: 02 00 00 3B 03 03 64 12 7E 05 25 F6 7A BD A0 2E
*Mar 16 03:28:50.692: 58 83 12 7F 90 CD F4 AB E2 69 53 A8 C7 FC 44 4F
*Mar 16 03:28:50.692: 57 4E 47 52 44 01 00 C0 2B 00 00 13 00 17 00 00
*Mar 16 03:28:50.693: FF 01 00 01 00 00 0B 00 02 01 00 00 23 00 00
*Mar 16 03:28:50.693: TLS server extension "unknown" (id=23), len=0
TLS server extension "renegotiate" (id=65281), len=1

*Mar 16 03:28:50.693: 00
*Mar 16 03:28:50.693: TLS server extension "EC point formats" (id=11), len=2

*Mar 16 03:28:50.693: 01 00
*Mar 16 03:28:50.693: TLS server extension "session ticket" (id=35), len=0

*Mar 16 03:28:50.693: <<< ??? [length 0005]
*Mar 16 03:28:50.693: 16 03 03 0F 9A
*Mar 16 03:28:50.694:
*Mar 16 03:28:50.702: SSL_connect:SSLv3/TLS read server hello
*Mar 16 03:28:50.702: <<< TLS 1.2 Handshake [length 0F9A], Certificate
*Mar 16 03:28:50.702: 0B 00 0F 96 00 0F 93 00 04 8A 30 82 04 86 30 82
*Mar 16 03:28:50.702: 03 6E A0 03 02 01 02 02 10 52 87 E0 40 A4 FE F7
```

<snip>

\*Mar 16 03:28:50.763: 82 35 CF 62 8B C9 24 8B A5 B7 39 0C BB 7E 2A 41

\*Mar 16 03:28:50.763: BF 52 CF FC A2 96 B6 C2 82 3F

\*Mar 16 03:28:50.763:

\*Mar 16 03:28:50.765: CC\_DEBUG: Entering shim layer app callback function

\*Mar 16 03:28:50.765: CRYPTO\_PKI: (A069C) Session started - identity not specified

\*Mar 16 03:28:50.765: CRYPTO\_PKI: (A069C) Adding peer certificate

\*Mar 16 03:28:50.767: CRYPTO\_PKI: Added x509 peer certificate - (1162) bytes

\*Mar 16 03:28:50.767: CRYPTO\_PKI: (A069C) Adding peer certificate

\*Mar 16 03:28:50.768: CRYPTO\_PKI: Added x509 peer certificate - (1434) bytes

\*Mar 16 03:28:50.768: CRYPTO\_PKI: (A069C) Adding peer certificate

\*Mar 16 03:28:50.770: CRYPTO\_PKI: Added x509 peer certificate - (1382) bytes

\*Mar 16 03:28:50.770: CRYPTO\_OPSSL: Validate Certificate Chain Callback

\*Mar 16 03:28:50.770: CRYPTO\_PKI(Cert Lookup) issuer="cn=GTS CA 1C3,o=Google Trust Services LLC,c=US" s

\*Mar 16 03:28:50.770: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=  
A7 CC 4B 0F 36 C3 AC D1 2F 77 DD 1D 9A 37 DC FC

\*Mar 16 03:28:50.770: CRYPTO\_PKI(Cert Lookup) issuer="cn=GTS Root R1,o=Google Trust Services LLC,c=US" s

\*Mar 16 03:28:50.771: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=  
03 9F CF 59 82 EE 09 CC 4F 53 AE D8 02 7E 4B AF

\*Mar 16 03:28:50.771: CRYPTO\_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-

\*Mar 16 03:28:50.771: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=  
94 40 D1 90 A0 A3 5D 47 E5 B5 31 F6 63 AD 1B 0A

\*Mar 16 03:28:50.771: CRYPTO\_PKI: Cert record not found for issuer serial.

\*Mar 16 03:28:50.772: CRYPTO\_PKI: crypto\_pki\_get\_cert\_record\_by\_subject()

\*Mar 16 03:28:50.772: CRYPTO\_PKI: Found a subject match

\*Mar 16 03:

#28:50.772: CRYPTO\_PKI: ip-ext-val: IP extension validation not required:Incrementing refcount for cont

\*Mar 16 03:28:50.773: CRYPTO\_PKI: create new ca\_req\_context type PKI\_VERIFY\_CHAIN\_CONTEXT,ident 35

\*Mar 16 03:28:50.773: CRYPTO\_PKI: (A069C)validation path has 1 certs

\*Mar 16 03:28:50.773: CRYPTO\_PKI: (A069C) Check for identical certs

\*Mar 16 03:28:50.773: CRYPTO\_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-

\*Mar 16 03:28:50.774: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=  
94 40 D1 90 A0 A3 5D 47 E5 B5 31 F6 63 AD 1B 0A

\*Mar 16 03:28:50.774: CRYPTO\_PKI: Cert record not found for issuer serial.

\*Mar 16 03:28:50.774: CRYPTO\_PKI : (A069C) Validating non-trusted cert

\*Mar 16 03:28:50.774: CRYPTO\_PKI: (A069C) Create a list of suitable trustpoints

\*Mar 16 03:28:50.774: CRYPTO\_PKI: crypto\_pki\_get\_cert\_record\_by\_issuer()

\*Mar 16 03:28:50.774: CRYPTO\_PKI: Found a issuer match

\*Mar 16 03:28:50.774: CRYPTO\_PKI: (A069C) Suitable trustpoints are: CA-GlobalSign-Root,

\*Mar 16 03:28:50.775: CRYPTO\_PKI: (A069C) Attempting to validate certificate using CA-GlobalSign-Root p

\*Mar 16 03:28:50.775: CRYPTO\_PKI: (A069C) Using CA-GlobalSign-Root to validate certificate

\*Mar 16 03:28:50.775: CRYPTO\_PKI(make trusted certs chain)

\*Mar 16 03:28:50.775: CRYPTO\_PKI: Added 1 certs to trusted chain.

\*Mar 16 03:28:50.775: CRYPTO\_PKI: Prepare session revocation service providers

\*Mar 16 03:28:50.776: P11:C\_CreateObject:

\*Mar 16 03:28:50.776: CKA\_CLASS: PUBLIC KEY

\*Mar 16 03:28:50.776: CKA\_KEY\_TYPE: RSA

\*Mar 16 03:28:50.776: CKA\_MODULUS:

DA 0E E6 99 8D CE A3 E3 4F 8A 7E FB F1 8B 83 25

6B EA 48 1F F1 2A B0 B9 95 11 04 BD F0 63 D1 E2

<snip>

\*Mar 16 03:28:50.780: CKA\_PUBLIC\_EXPONENT: 01 00 01

\*Mar 16 03:28:50.780: CKA\_VERIFY\_RECOVER: 01  
\*Mar 16 03:28:50.780: CRYPTO\_PKI: Deleting cached key having key id 45  
\*Mar 16 03:28:50.781: CRYPTO\_PKI: Attempting to insert the peer's public key into cache  
\*Mar 16 03:28:50.781: CRYPTO\_PKI:Peer's public inserted successfully with key id 46  
\*Mar 16 03:28:50.781: P11:C\_CreateObject: 131118  
\*Mar 16 03:28:50.781: P11:C\_GetMechanismInfo slot 1 type 3 (invalid mechanism)  
\*Mar 16 03:28:50.781: P11:C\_GetMechanismInfo slot 1 type 1  
\*Mar 16 03:28:50.781: P11:C\_VerifyRecoverInit - 131118  
\*Mar 16 03:28:50.781: P11:C\_VerifyRecover - 131118  
\*Mar 16 03:28:50.781: P11:found pubkey in cache using index = 46  
\*Mar 16 03:28:50.781: P11:public key found is :  
30 82 01 22 30 0D 06 09 2A 86 48 86 F7 0D 01 01  
01 05 00 03 82 01 0F 00 30 82 01 0A 02 82 01 01  
<snip>  
CF 02 03 01 00 01

\*Mar 16 03:28:50.788: P11:CEAL:CRYPTO\_NO\_ERR  
\*Mar 16 03:28:50.788: P11:C\_DestroyObject 2:2002E  
\*Mar 16 03:28:50.788: CRYPTO\_PKI: Expiring peer's cached key with key id 46  
\*Mar 16 03:28:50.788: CRYPTO\_PKI: (A069C) Certificate is verified  
\*Mar 16 03:28:50.788: CRYPTO\_PKI: Remove session revocation service providers  
\*Mar 16 03:28:50.788: CRYPTO\_PKI: Remove session revocation service providersCA-GlobalSign-Root:validat  
\*Mar 16 03:28:50.788: CRYPTO\_PKI: (A069C) Certificate validated without revocation check:cert refcount  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: Populate AAA auth data  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: Unable to get configured attribute for primary AAA list authorization  
\*Mar 16 03:28:50.790: PKI: Cert key-usage: Digital-Signature , Certificate-Signing , CRL-Signing  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C)chain cert was anchored to trustpoint CA-GlobalSign-Root, and  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C) Removing verify context

\*Mar 16 03:28:50.790: CRYPTO\_PKI: destroying ca\_req\_context type PKI\_VERIFY\_CHAIN\_CONTEXT,ident 35, ref  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: ca\_req\_context released  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C) Validation TP is CA-GlobalSign-Root  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C) Certificate validation succeeded  
\*Mar 16 03:28:50.790: CRYPTO\_OPSSL: Certificate verification is successful  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: Rcvd request to end PKI session A069C.  
\*Mar 16 03:28:50.790: CRYPTO\_PKI: PKI session A069C has ended. Freeing all resources.:cert refcount aft  
\*Mar 16 03:28:50.791: <<< ??? [length 0005]  
\*Mar 16 03:28:50.791: 16 03 03 00 93  
\*Mar 16 03:28:50.791:  
\*Mar 16 03:28:50.791: SSL\_connect:SSLv3/TLS read server certificate  
\*Mar 16 03:28:50.791: <<< TLS 1.2 Handshake [length 0093], ServerKeyExchange  
\*Mar 16 03:28:50.791: 0C 00 00 8F 03 00 17 41 04 3D 49 34 A3 52 D4 EB  
\*Mar 16 03:28:50.791: DE A2 9E CC B0 91 AA CB 1B 39 D0 26 1B 7D FF 31  
\*Mar 16 03:28:50.792: E0 D7 D5 9C 75 C0 7D 5B D6 B2 0A B5 CC EA E1 4B  
\*Mar 16 03:28:50.792: 4E E5 72 7B 54 5D 9B B2 95 91 E0 CC D6 A5 8E CE  
\*Mar 16 03:28:50.792: 8D 36 C9 83 42 B0 4D AC 0C 04 03 00 46 30 44 02  
\*Mar 16 03:28:50.792: 20 67 B3 F1 DA D1 BF 13 72 DD B6 B2 11 3B 6E 6F  
\*Mar 16 03:28:50.793: 87 52 D9 00 F7 44 31 C3 C2 5E BE 2D FF 93 4E F0  
\*Mar 16 03:28:50.793: A8 02 20 24 42 91 BE B7 10 1C D1 C0 12 28 FB 1F  
\*Mar 16 03:28:50.793: E4 DE 81 0B AA 66 19 CD 28 5A A0 30 7D 3C 4A 56  
\*Mar 16 03:28:50.793: 0D 94 E2  
\*Mar 16 03:28:50.793:  
\*Mar 16 03:28:50.794: P11:C\_FindObjectsInit:  
\*Mar 16 03:28:50.794: CKA\_CLASS: PUBLIC KEY  
\*Mar 16 03:28:50.794: CKA\_KEY\_TYPE: : 00 00 00 03

\*Mar 16 03:28:50.794: CKA\_ECDSA\_PARAMS:  
30 59 30 13 06 07 2A 86 48 CE 3D 02 01 06 08 2A  
86 48 CE 3D 03 01 07 03 42 00 04 63 B6 D3 1A 28  
<snip>

\*Mar 16 03:28:50.796: P11:C\_FindObjectsFinal

\*Mar 16 03:28:50.796: P11:C\_VerifyInit - Session found  
\*Mar 16 03:28:50.796: P11:C\_VerifyInit - key id = 131073  
\*Mar 16 03:28:50.796: P11:C\_Verify  
\*Mar 16 03:28:50.800: P11:CEAL:CRYPTO\_NO\_ERR  
\*Mar 16 03:28:50.800: <<< ??? [length 0005]  
\*Mar 16 03:28:50.800: 16 03 03 00 04  
\*Mar 16 03:28:50.800:  
\*Mar 16 03:28:50.800: SSL\_connect:SSLv3/TLS read server key exchange  
\*Mar 16 03:28:50.800: <<< TLS 1.2 Handshake [length 0004], ServerHelloDone  
\*Mar 16 03:28:50.801: 0E 00 00 00  
\*Mar 16 03:28:50.801:  
\*Mar 16 03:28:50.801: SSL\_connect:SSLv3/TLS read server done  
\*Mar 16 03:28:50.810: >>> ??? [length 0005]  
\*Mar 16 03:28:50.810: 16 03 03 00 46  
\*Mar 16 03:28:50.811:  
\*Mar 16 03:28:50.811: >>> TLS 1.2 Handshake [length 0046], ClientKeyExchange  
\*Mar 16 03:28:50.811: 10 00 00 42 41 04 26 C3 EF 02 05 6C 82 D1 90 B3  
\*Mar 16 03:28:50.811: 17 31 9A CD DD 8C 81 91 BA E8 C0 86 40 7B 2C E4  
\*Mar 16 03:28:50.811: 9A 2C 18 9D D1 6A C0 56 A0 98 2E B7 3B AB B3 EB  
\*Mar 16 03:28:50.811: BB CD 5E 42 C5 76 C0 C4 BF 15 F4 87 F2 7C AD 74  
\*Mar 16 03:28:50.812: 97 0A 97 2B 06 B5  
\*Mar 16 03:28:50.812:  
\*Mar 16 03:28:50.812: SSL\_connect:SSLv3/TLS write client key exchange  
\*Mar 16 03:28:50.812: >>> ??? [length 0005]  
\*Mar 16 03:28:50.812: 14 03 03 00 01  
\*Mar 16 03:28:50.812:  
\*Mar 16 03:28:50.812: >>> TLS 1.2 ChangeCipherSpec [length 0001]  
\*Mar 16 03:28:51.116: >>> ??? [length 0005]  
\*Mar 16 03:28:51.116: 17 03 03 00 35  
\*Mar 16 03:28:51.116:  
\*Mar 16 03:28:51.116: >>> ??? [length 0005]  
\*Mar 16 03:28:51.116: 17 03 03 00 1A  
\*Mar 16 03:28:51.116:  
\*Mar 16 03:28:51.116: >>> ??? [length 0005]  
\*Mar 16 03:28:51.116: 17 03 03 00 30  
\*Mar 16 03:28:51.116:  
\*Mar 16 03:28:51.116: >>> ??? [length 0005]  
\*Mar 16 03:28:51.116: 17 03 03 00 1B  
\*Mar 16 03:28:51.117:  
\*Mar 16 03:28:51.713: <<< ??? [length 0005]  
\*Mar 16 03:28:51.713: 17 03 03 00 6D  
\*Mar 16 03:28:51.713:  
\*Mar 16 03:28:51.714: >>> ??? [length 0005]  
\*Mar 16 03:28:51.714: 17 03 03 00 1E  
\*Mar 16 03:28:51.714:  
\*Mar 16 03:28:51.732: <<< ??? [length 0005]  
\*Mar 16 03:28:51.732: 17 03 03 00 71  
\*Mar 16 03:28:51.732:

# event manager run SendSecureEmailEEM\*Mar 16 03:28:50.673: CRYPTO\_OPSSL: Allocated the memory for OPSSLContext\*Mar 16 03:28:50.673: CRYPTO\_OPSSL: Set cipher specs to mask 0x02FC000 for version 128\*Mar 16 03:28:50.674: デフォルトのEC曲線リストを設定します。 0x70EC曲線リストを設定します。 secp521r1:secp384r1:prime256v1\*Mar 16 03:28:50.674: opssl\_SetPKIInfo entry\*Mar 16 03:28:50.674: CRYPTO\_PKI: (A069B) Session started identity - selected自己署名486541296)xTP-self-signed-486541296:refcount after increment = 1\*Mar 16 03:28:50.674: CRYPTO\_PKI : ローカル証明書チェーンの取得を開始します。\*Mar 16 03:28:50.674: CRYPTO\_PKI (証明書検索) issuer="cn=IOS-Self-Signed-Certificate-486541296" serial number= 01\*Mar 16 03:28:50.674: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0,

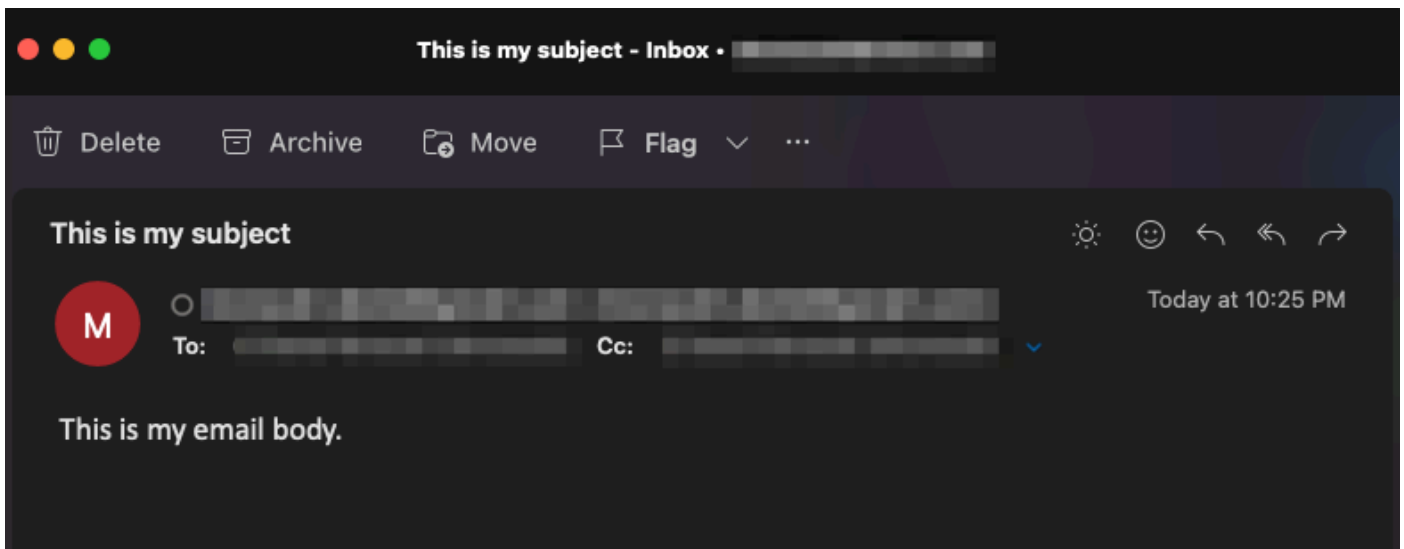


digest=1C 7F 3D 52 67 66 D5 59 E2 66 58 E7 8B E7 9B 8E\*Mar 16 03:28:50.675: CRYPTO\_PKI: Done with local cert chain fetch 0.\*Mar 16 03:28:50.675: CRYPTO\_PKI: Rcvd request to end PKI session A069B.\*Mar 16 03:28:50.675: CRYPTO\_PKI: PKI session A069B has ended.すべてのリソースを解放します。TP-self-signed-486541296:unlockedトラストポイントTP-self-signed-486541296, refcount is 0\*Mar 16 03:28:50.675: opssl\_SetPKInfo done.\*Mar 16 03:28:50.675: CRYPTO\_OPSSL: Common Criteria is disabled on this session.Disabled Common Criteria mode functionality in CiscoSSL CTX7F441F 28EAF8\*3月16日03:28:50.675: CRYPTO\_OPSSL: ciphersuites ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:AES256-GCM-SHA384:AES256-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:AES128-GCM-SHA256:AES128-SHA256\*3月16日03:28:50.676 : ハンドシェイクの開始 : SSL初期化前\*3月16日03:28:50.676: SSL\_connect:SSL初期化前\*3月16日03:28:50.676: >> ???[length 0005]\*Mar 16 03:28:50.676: 16 03 01 00 95\*Mar 16 03:28:50.676: \*Mar 16 03:28:50.676: >> TLS 1.2ハンドシェイク[length 0095], ClientHello\*Mar 16 03:28:50.676: 1 00 00 91 03 03 26 4B 9F B3 44 94 FD 5F FD A1<snip>\*3月16日03:28:50.679: 03 03 01 02 01\*3月16日03:28:50.679: \*3月16日03:28:50.679: SSL\_connect:SSL TLS書き込みクライアントhello\*3月16日03:28:50.692: << ???[length 0005]\*Mar 16 03:28:50.692: 16 03 03 00 3F\*Mar 16 03:28:50.692: \*Mar 16 03:28:50.692: SSL\_connect:SSLv3/TLS write client hello\*Mar 16 03:28:50.692:< TLS 1.2 handshake [length 003F], ServerHello\*Mar 16 03:28:50.692: 02 00 00 3B 03 03 64 12 7E 05 25 F6 7A BD A0 2E\*Mar 16 03:28:50.692: 583 12 7F 90 CD F4 AB E2 6953 A c7 FC 44 4F\*3月16日03:28:50.692: 57 4E 47 52 44 01 00 C0 2B 00 00 13 00 17 00 00\*3月16日03:28:50.693: FF 01 0 00 0 0100 0B 00200 0\*3月16日03:28:50.693: TLSサーバーの拡張「不明」(id=23)、len=0TLSサーバーの拡張「再交渉」(id=65281)、len=1\*3月16日03:28:50.693: 00\*3月16日03:28:50.693: TLSサーバーの拡張「ECポイントフォーマット」(id=1111)、len=2\*3月63:2:50.693: 01 00\*3月16日03:28:50.693: TLSサーバ拡張「セッションチケット」(id=35), len=0\*3月16日03:28:50.693: << ???[長さ0005]\*3月16日03:28:50.693: 16 03 03 0F 9A\*3月16日03:28:50.694: \*3月16日03:28:50.702: SSL\_connect:SSLv3/TLS読み取りサーバ\*3月16日03:28:50.70 handshake [length 0F9A], Certificate\*Mar 16 03:28:50.702: 0B 00 0F 96 00 0F 93 00 04 8A 30 82 04 86 30 82\*Mar 16 03:28:50.702: 03 6E A0 03 02 01 02 10 5287 E0 40 4 FE F7<snip>\*3月16日03:28:50.763: 82 35 CF 62 8B C9 24 8B A5 B7 39 0C BB 7E 2A 41\*3月16日03:28:50.763: BF 52 CF FC A2 96 B6 C2 82F\*1603:2 :50.763: \*Mar 16 03:28:50.765: CC\_DEBUG: Entering shim layer app callback function\*Mar 16 03:28:50.765: CRYPTO\_PKI: (A069C) Session started - identity not specified\*Mar 16 03:28:50.765: CRYPTO\_PKI: (A069C) peer certificate\*Mar 16 03:28:50.767: CRYPTO\_PKI: Added x509 peer certificate - (1162) bytes\*Mar 16 03:28:50.767: CRYPTO\_PKI: (A069C) Adding peer certificate\*Mar 16 03:28:50.768: CRYPTO\_PKI: Added x509 - (1434) bytes\*Mar 16 03:28:50.768: CRYPTO\_PKI: (A069C)ピア証明書を追加\*Mar 16 03:28:50.770: CRYPTO\_PKI: Added x509 peer certificate - (1382) bytes\*Mar 16 03:28:50.770: CRYPTO\_OPSSL:証明書チェーンコールバック\*Mar 16 03:28:50.770: CRYPTO\_PKI(Cert Lookup) issuer="cn=GTS CA 1C3,o=Google Trust Services LLC,c=US"シリアル番号= 52 87 E0 40 A4 FE F7 07 12 68 B0 4F DD DD F0 F4\*1603:28:50.770: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=A7 CC 4B 0F 36 C3 AC D1 2F 77 DD 1D 9A 37 DC FC\*Mar 16 03:28:50.770: CRYPTO\_PKI(Cert Lookup) issuer="cn=GTS Root R1,o=Google Trust services LLC,c=US" serial number= 02 03 BC 53 59 6B 34 C7 18 F5 01 50 66\*Mar 16 03:28:50.771: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=03 9F CF 5982 EE 09 CC 4F 53 Ae 8 02 7E 4B AF\*Mar 16 03:28:50.771: CRYPTO\_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-sa,c=BE" serial number= 77 BD 0D

6C DB 36 F9 1A EA 21 0F C4 F0 58 D3 0D\*Mar 16 3:28:50.771: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=94 40 D1 90 A0 A3 5D 47 E5 B5 31 F6 63 AD 1B 0A\*Mar 16 03:28:50.771: CRYPTO\_PKI: Cert record not found for issuer serial.\*1 6 03:28:50.772: CRYPTO\_PKI: crypto\_pki\_get\_cert\_record\_by\_subject()\*Mar 16 03:28:50.772: CRYPTO\_PKI: Found a subject match\*Mar 16 03:28:50.772: CRYPTO\_PKI: ip-ext-val: IP extension validation not required:Incrementing refcount for context-35 to\*1\*Mar 16 3:28:50.773: CRYPTO\_PKI : 新しいca\_req\_contextタイプを作成しますPKI\_VERIFY\_CHAIN\_CONTEXT,ident 35\*3月16日 03:28:50.773: CRYPTO\_PKI: (A069C)検証パスには1つの証明書があります\*3月16日 03:28:50.773: CRYPTO\_PKI )同一証明書の確認\*Mar 16 03:28:50.773: CRYPTO\_PKI(Cert Lookup) issuer="cn=GlobalSign Root CA,ou=Root CA,o=GlobalSign nv-sa,c=BE" serial number= 77 BD 0D 6C DB 36 F9 1A EA 21 0F C4 F0 58 D3 0D\*Mar 16 03:28:50.774: CRYPTO\_PKI: looking for cert in handle=7F41EE523CE0, digest=94 40 D1 90 A0 A3 5D 47 E5 B5 31 F6 63 AD 1B 0A\*Mar 16 03:28:50.774: CRYPTO\_PKI: Cert record not found for issuer.\*Mar 16 03:28:50.774: CRYPTO\_PKI : (A069C)信頼できない証明書の検証\*3月16日 03:28:50.774: CRYPTO\_PKI: (A069C)適切なトラストポイントのリストを作成\*3月16日 03:28:50.774: CRYPTO\_PKI: crypto\_pki\_get\_cert\_record\_by issuer()\*16 3:28:50.774: CRYPTO\_PKI: Found a issuer match\*Mar 16 03:28:50.774: CRYPTO\_PKI: (A069C)適切なトラストポイント : CA-GlobalSign-Root,\*Mar 16 03:28:50.775: CRYPTO\_PKI: (A069C) Attempting to validate certificate using CA -GlobalSign-Root policy\*Mar 16 03:28:50.775: CRYPTO\_PKI: (A069C) Using CA-GlobalSign-Root to validate certificate\*Mar 16 03:28:50.775: CRYPTO\_PKI(make trusted certs chain)\*Mar 16 03:28:50.775: CRYPTO\_PKI: Added to certs trusted chain.\*Mar 16 03:28:50.775: CRYPTO\_PKI: Prepare session revocation service providers\*Mar 16 03:28:50.776: P11:C\_CreateObject:\*Mar 16 03:28:50.776: CKA\_CLASS: PUBLIC KEY\*Mar 16 03:28:50.776: CKA\_KEY\_TYPE: RSA\*Mar 16 03:28:50.776: CKA\_MODULUS: DA 0E E6 99 8D CE A3 E3 4F 8A 7E FB F1 8B 83 25 6B EA 48 1F F1 2A B0 B9 95 1104 BD F0 63 D1 E2 <snip>\*Mar 16 03:28:50.780: CKA\_PUBLIC\_EXPONENT: 01 00 01\*3月16日 03:28:50.780: CKA\_VERIFY\_RECOVER: 01\*3月16日 03:28:50.780: CRYPTO\_PKI : キーID 45\*3月16日 03:28:50.781: crypto\_PKI : ピアの公開キーをキャッシュに挿入しようとしています\*Mar 16 03:28:50.781: CRYPTO\_PKI:Peer's public inserted successfully with key id 46\*Mar 16 03:28:50.781: P11:C\_CreateObject: 131118\*Mar 16 03:28:50.781: P11:C\_GetMechanism Info Slot 1タイプ3 ( 無効なメカニズム ) \*Mar 16 03:28:50.781: P11:C\_GetMechanismInfo slot 1タイプ 1\*Mar 16 03:28:50.781: P11:C\_VerifyRecoverInit - 131118\*Mar 16 03:28:50.781: P11:C\_VerifyRecover - 131118\*Mar 16 03:28:50.781: P11:found pubkey in cache using index = 46\*Mar 16 03:28:50.781: P11:public key found is : 30 82 01 22 30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 00 32 01 0F 00 30 82 0A 02 82 01 01 <snip>CF 02 03 01 00 01\*3月16日 03:28:50.788: P11:CEAL:CRYPTO\_NO\_ERR\*3月16日 03:28:50.788: P11:C\_DestroyObject 2:2002E\*3月 16 03:28:50.788: CRYPTO\_PKI : キーIDが46\*Mar 16のピアのキャッシュされたキーの期限切れ03:28:50.788: CRYPTO\_PKI: (A069C)証明書が確認されました\*Mar 16 03:28:50.788: CRYPTO\_PKI: Remove session revocation service providers\*Mar 16 03:28:50.788: CRYPTO\_PKI: Remove session revocation service providersCA-GlobalSign-Root:validation status - CRYPTO\_VALID\_CERT\_WITH\_WARNING\*Mar 16 03:28:50.788: CRYPTO\_PKI: (A069C)証明書は失効チェックなしで検証されました : cert refcount after increment = 1\*Mar 16 03:28:50.790: CRYPTO\_PKI: Populate AAA AUTH data\*Mar 16 03:28:50.790: CRYPTO\_PKI : プライマリ AAAリスト認証に対して設定された属性を取得できません。\*Mar 16 03:28:50.790: PKI : 証明書キーの使用 : デジタル署名、証明書署名、CRL署名\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C)chain cert was anchored to trustpoint CA-Global Sign-Root, and chain validation result was: CRYPTO\_VALID\_CERT\_WITH\_WARNING\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C) Removing verify context\*Mar 16 03:28:50.790: CRYPTO\_PKI: destroying ca\_req\_context type

PKI\_VERIFY\_CHAIN\_CONTEXT,ident35, ref:1 コンテキストid-35のrefcountを0\*Mar 16に減らします03:28:50.790: CRYPTO\_PKI: ca\_req\_context released\*Mar 16 03:28:50.790: CRYPTO\_PKI: (A069C) Validation TP is CA-GlobalSign-Root\*Mar 16 03:28:50.790: CRYPTO\_PKI: A069C)証明書の検証に成功\*3月16日03:28:50.790:CRYPTO\_OPSSL : 証明書の検証に成功\*3月16日03:28:50.790:CRYPTO KI\_PKI:PKIセッションA069Cを終了するためのRCVD要求。\*Mar 16 03:28:50.790: CRYPTO\_PKI: PKIセッションA069Cが終了しました。すべてのリソースを解放します。:cert refcount after decrement = 0\*Mar 16 03:28:50.791: << ???[長さ0005]\*3月16日03:28:50.791: 16 03 03 00 93\*3月16日03:28:50.791: \*3月16日03:28:50.791: SSL\_connect:SSLv3/TLS読み取りサーバ証明書\*3月16日03:2:5:50.5 handshake [length 0093], ServerKeyExchange\*Mar 16 03:28:50.791: 0C 00 00 8F 03 00 17 41 04 3D 49 34 A3 52 D4 EB\*Mar 16 03:28:50.791: DE A2 9E CC B0 91 AA CB 1B 39 D0 6 1B 7D FF 31\*3月16日03:28:50.792: E0 D7 D5 9C 75 C0 7D 5B D6 B2 0A B5 CC EA E1 4B\*3月16 03:28:50.792: 4E E5 72 7B 54 5D 9B2 9591 E0 CC D6 a5 8E CE\*Mar 16 03:28:50.792: 8D 36 C9 83 42 B0 4D AC 0C 04 03 00 46 30 44 02\*Mar 16 03:28:50.792: 20 67 B3 F1 DA D1 BF 13 72 DD B6 B2 11 6B 6E F\*Mar 16 03:28:50.793: 87 52 D9 00 F7 44 31 C3 C2 5E BE 2D FF 93 4E F0\*Mar 16 03:28:50.793: A8 02 20 24 42 91 BE B7 10 1C D1 C0 128 FB 1F\*1 Mar 6 03:28:50.793: E4 DE 81 0B AA 66 19 CD 28 5A0 30 7D 3C 4A 56\*Mar 16 03:28:50.793: 0D 94 E2\*Mar 16 03:28:50.793: \*Mar 16 03:28:50.794: p11:C\_FindObjectsInit:\*3月16日03:28:50.794: CKA\_CLASS : 公開キー\*3月16日03:28:50.794: CKA\_KEY\_TYPE: : 00 00 00 03\*3月16日03:28:50.794: CKA\_ECDSA\_PARAMS: 35 9 30 13 06 07 2A 86 48 CE 3D 02 01 06 08 2A 86 48 CE 3D 03 01 07 03 42 00 04 63 B6 D3 1A 28 <snip>\*3月16日03:28:50.796: P11:C\_FindObjects最終\*3月16:28:50.796: P11:C\_VerifyInit – セッションが見つかりました\*3月16日03:28:50.796: P11:C\_VerifyInit – キーID = 131073\*3月16日03:28:50.796: P11:C\_Verify\*3月16日03:28:50.800: P1:CEAL NO\_ERR\*Mar 16 03:28:50.800: << ???[長さ0005]\*3月16日03:28:50.800: 16 03 03 00 04\*3月16日03:28:50.800: \*3月16日03:28:50.800: SSL\_connect:SSLv3/TLS read server key exchange\*3月16日03:28:50.800 Handshake [length 0004], ServerHelloDone\*Mar 16 03:28:50.801: 0E 00 00 00\*Mar 16 03:28:50.8 01: \*Mar 16 03:28:50.801: SSL\_connect:SSLv3/TLS read server done\*Mar 16 03:28:50.810: >> ???[length 0005]\*Mar 16 03:28:50.810: 16 03 03 00 46\*Mar 16 03:28:50.811: \*Mar 16 03:28:50.811: >>> TLS 1.2ハンドシェイク[length 0046], ClientKeyExchange\*Mar 16 03:28:50.81 : 10 00 00 42 41 04 26 C3 EF 02 05 6C 82 D1 90 B3\*Mar 16 03:28:50.811: 17 31 9A CD DD 8C 81 91 BA E8 86 40 7B 2C E4\*Mar 1603:28:50.811: 9A c 18 9D D1 6A C0 56 A0 98 2E B7 3B AB B3 EB\*Mar 16 03:28:50.811: BB CD 5E 42 C5 76 C0 C4 BF 15 F4 87 F2 7C AD 74\*Mar 16 03:28:50.812: 97 A 97 b 06 B5\*3月16日03:28:50.812: \*3月16日03:28:50.812: SSL\_connect:SSLv3/TLS write client key exchange\*3月16日03:28:50.812: >> ???[長さ0005]\*3月16日03:28:50.812: 14 03 03 00 01\*3月16日03:28:50.812: \*3月16 03:28:50.812: >> TLS 1.2 ChangeCipherSpec [length 0001]\*Mar 16 03:28:51.116: >> ???[長さ0005]\*3月16日03:28:51.116: 17 03 03 00 35\*3月16日03:28:51.116: \*3月16日03:28:51.116: >> ???[長さ0005]\*3月16日03:28:51.116: 17 03 03 00 1A\*3月16日03:28:51.116: \*3月16日03:28:51.116: >> ???[長さ0005]\*3月16日03:28:51.116: 17 03 03 00 30\*3月16日03:28:51.116: \*3月16日03:28:51.116: >> ???[長さ0005]\*3月16日03:28:51.116: 17 03 03 00 1B\*3月16日03:28:51.117: \*3月16日03:28:51.713: << ???[長さ0005]\*3月16日03:28:51.713: 17 03 03 00 6D\*3月16日03:28:51.713: \*3月16日03:28:51.714: >> ???[長さ0005]\*3月16日03:28:51.714: 17 03 03 00 1E\*3月16日03:28:51.714: \*3月16日03:28:51.732: << ???[長さ0005]\*Mar 16 03:28:51.732: 17 03 03 00 71\*Mar 16 03:28:51.732:

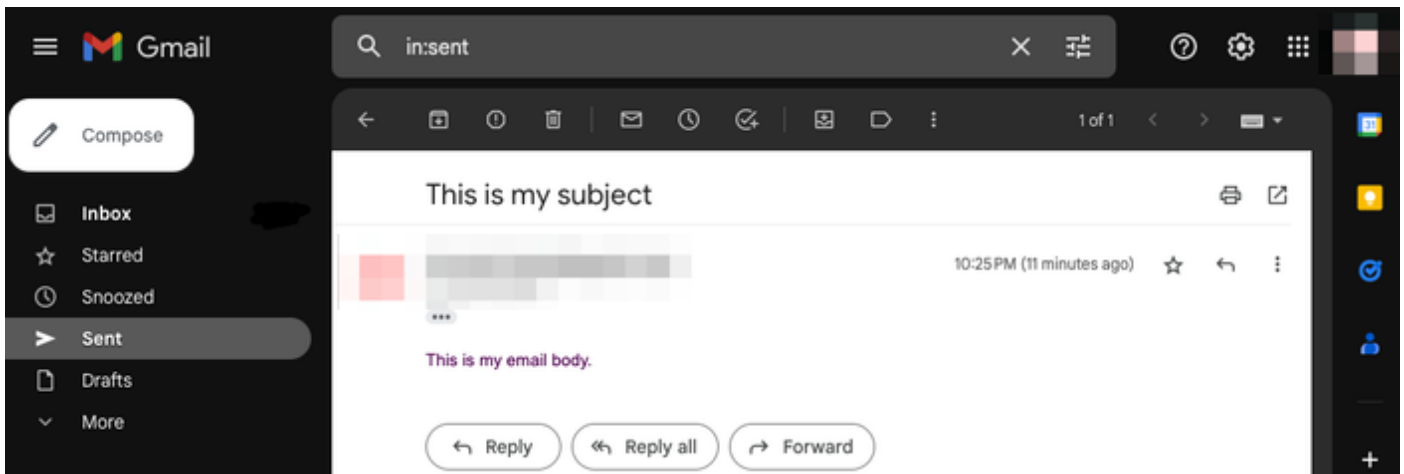
電子メールが受信され、すべてのフィールド(to、from、cc、subject、body)が正しく入力されたことを確認できます。



また、Cisco IOS XEデバイス ( 「WorkingSMTPwithTLS.pcap」として添付 ) のパケットキャプチャからTLSハンドシェイクおよびセッションが行われたことを確認できます。

No.	Time	Source	Destination	Protocol	Length	ID	Info
11	2023-03-16 03:28:50...	10.122.144.150	142.251.163.109	TLSv1.2	208	0x8790 (34704)	Client Hello
12	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	590	0x7641 (30273)	Server Hello
32	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	439	0x7649 (30281)	Certificate, Server Key Exchange, Server Hello Done
33	2023-03-16 03:28:50...	10.122.144.150	142.251.163.109	TLSv1.2	180	0x879d (34717)	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
34	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	349	0x764a (30282)	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
36	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	107	0x879f (34719)	Application Data
38	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	306	0x764c (30284)	Application Data
39	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	116	0x87a0 (34720)	Application Data
41	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	101	0x764e (30286)	Application Data
42	2023-03-16 03:28:50...	142.251.163.109	10.122.144.150	TLSv1.2	109	0x87a1 (34721)	Application Data

使用している電子メールアカウントの「送信済み」フォルダに電子メールが反映されているかどうかを確認することもできます。



## その他の注意事項と考慮事項

### @記号を含むユーザ名

SMTPリレーを使用しようとする、問題が発生する可能性があります。SMTPリレーにより、サーバ文字列は次の形式になります ( ユーザ名の「@」 )。

event manager environment \_email\_server email.relay@My.Domain.Name:MyPasswordString@smtp-relay.gmail.com

ユーザ名とパスワードを解析するコードは、最初に出現する「@」記号の文字列を分割します。その結果、システムはサーバのホスト名が最初の「@」記号の直後から残りの文字列の直後に始まると見なし、その前をすべて「ユーザ名:パスワード」として解釈します。

SMTPのTCL実装では、このユーザ名/パスワード/サーバ情報を異なる方法で処理する正規表現(regex)を使用します。この違いにより、TCLでは「@」記号を使用したユーザ名を使用できません。ただし、Cisco IOS XE TCLは暗号化をサポートしていないため、TLS経由で安全な電子メールを送信するオプションはありません。

## まとめ

- 電子メールを安全にする必要がある場合は、TCLで送信することはできません。
- ユーザ名に「@」が含まれている場合は、EEMで送信できません。

Cisco Bug ID [CSCwe75439](#)は、EEMの電子メール機能を改善するためにこの機会に対処するために提出されましたが、現在、この拡張要求に対するロードマップはありません。

## 結論

ここに示すように、Embedded Event Manager(EEM)アプレットを使用して、TLSを使用したSMTP経由でセキュアな電子メールを送信できます。信頼を可能にするために必要な証明書を設定するだけでなく、サーバ側で何らかの設定を行う必要がありますが、自動化された安全な電子メール通知を生成したい場合は可能です。

## 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。