Cisco IOS XEおよびGuestShell CA Trust eMudhra信頼ガイド

内容

目的

このドキュメントでは、Mudhraルートおよび中間認証局に対する緊急の不信感を解消するためのCisco IOS® XEのガイダンスを提供します。

このガイドを使用して、影響を受ける階層が構成で許可されているサービスのCA信頼を監査および管理します。製品の適用性と必要な修復方法は、シスコの情報に定義されています。信頼を変更する前に、ご使用のCisco IOS XEリリースのコマンドサポートとサービスへの影響を確認してください。

シスコ情報：cisco-sa-emudhra-info-ScS22A1r

影響を受けるMudhra CA証明書：オリジナルおよびクロス署名

インベントリは2026年10月29日に検証されました。CAキーを発行する8つのeMudhraが影響を受けます。6つはTLS用、2つはコード署名用です。各キーは、元のemSign-issued CA証明書とSectigoのクロス署名CA証明書に含まれています。したがって、インベントリには16の証明書（12のTLS証明書と4つのコード署名証明書）が含まれます。

元のペアとクロス署名ペアは、それぞれ同じサブジェクトと公開キーを持ちますが、発行者、シリアル番号、およびSHA-256証明書フィンガープリントは異なります。この2つのフォームは、代替の認証パスを提供し、両方を監査に含める必要があります。

A.発行者と影響を受ける証明書インベントリ

発行者A：元のTLS証明書：CN=emSign Root TLS CA - G3、O=eMudhra Technologies Limited、C=IN

発行者B：元のコード署名証明書：CN=emSign Root CS CA - G3、O=eMudhra Technologies Limited、C=IN。

発行者C：すべての相互署名証明書：CN=AAA Certificate Services、O=Comodo CA Limited、L=Salford、ST=Greater Manchester、C=GB(Sectigo)。

G3B科目： O=eMudhra BV, C=NL.G3C科目： O=eMudhra Inc, C=US.各バリアントセルには、発行者参照、16進数シリアル、およびSHA-256フィンガープリント（2行）がリストされます。 先頭のゼロのパディングでは、シリアル値は変更されません。

影響を受ける発行側CA(CN)	元の証明書 発行者/シリアル（16進数）/SHA-256	Sectigo相互署名証明書 発行者/シリアル（16進数）/SHA-256
1. EM DV TLS CA - G3B [TLS]	発行者A A4AEE678F23362B4C0E460B772F214 42C4ACCD12C4F59597925E1A08A7A901 02608543809308D50EC427E118BA1493	発行者C A024FDF4BA665C785CE4A020483858B7 45A883F349AEDFD5F3A649B5498FFB12 1712BA8B448F3F2A91DBC5A766FCDDBE
2. EM OV TLS CA:G3B [TLS]	発行者A 6CADA367FB1C100088120EA202D568 98B896625C0C230ADEEC3EB2CF6AAD50 01ECF9156A68644E82374CE5833BC25B	発行者C E94223BDE92F914E5AB9ACABFDAC3710 5F1DED4DD590AF74631290CF073B122A 2AAC6A1FA479778A0BD43579BD832699
3. EM EV TLS CA - G3B [TLS]	発行者A C7C41EFE92D825F05EAE841A4A82C1 E7568FE6E12331F4FD7DE61DA4C7559D 0F8EB192A218D4741A3BA75184333838	発行者C D8F8685E4F2BD9D82402A85A1C1EB731 579480AEA166290BD3C643589E795FC0 D0E54E86505D180F1DFF99EFD3199512
4. EM DV TLS CA - G3C [TLS]	発行者A 588B2E3C2BB8CDA13E1A1209254321 0164D7CEAE5BB909C272AF151FF44B11 77AE9B47D9424B94F1D9558C30934D19	発行者C 91F347F1A5B7D39962B95E0EE4E299D1 62B7F9868263F72F6911FC78CEAF72C1 979158B7368BF648683BB2AA39B02E72
5. EM	発行者A	発行者C

影響を受ける発行側CA(CN)	元の証明書 発行者/シリアル（16進数）/SHA-256	Sectigo相互署名証明書 発行者/シリアル（16進数）/SHA-256
OV TLS CA - G3C [TLS]	296926C227A3A1526392B19482A504 CB6E15165AC802E9B3F993F16C25D41F 4E81C21EBDE71AE1B012AC00BF9C7C3F	8D5A4811939EF447D77111CDD9849AE9 7485B51C0ED192878FA1D2F948FB529A E1842CC5086C4B67F7BE5B2039DE9489
6. EM EV TLS CA - G3C [TLS]	発行者A 8009928A64107601AF05FA2F19F109 432EC25F0D28DE9006D97BF752F17BBB 2CD41F669933FCB072D9A1148D0261F3	発行者C 05F078752DECE7A1BB8595D303167171 FD9D5C2DBBD4995AB69B8D50C439D952 7B00529199C2FD01B2174EB58F07021F
7. EM OV CS CA - G3B コード署名	発行者B 7818933F8F85B4619A2B35FBA83D0A 9E39DE2881273ACC225A149C72BA3E0F 51424963669AB633EE1DC7DF5DB067B4	発行者C 41E32B73AC185D6CE0D7480BACF84E69 A496E23AC0CAB8155FC9B117B97EAE36 8EFCBB9041AEC8804D5A308D2D01C2D8
8. EM OV CS CA - G3C コード署名	発行者B 544B7F922C31AE220A4A922916F6E6 4B2460A466F95FD40113BCDCCA2CEB29 B3B093921DE45474DA7DF48874C630B1	発行者C D6A5EF613B390F32EF9CDF349F772841 84112972B8E59D14560AE7E711965447 4AAD119990CF3722BFC1E3B91615E1FD

B.差別化のための認定パス

元のTLSパス

```
emSign ECC Root CA - G3
  +-- emSign Root TLS CA - G3 (issued by emSign ECC Root CA - G3)
       +-- Original EM DV/OV/EV TLS CA - G3B or G3C
            +-- TLS identity certificate
```

元のコード署名パス（ルートCSは直接信頼）

```
emSign Root CS CA - G3
  +-- Original EM OV CS CA - G3B or G3C
       +-- Code-signing identity certificate
```

Sectigo相互署名パス（TLSおよびコード署名）

```
AAA Certificate Services
  +-- One of the eight cross-signed CAs listed above
       +-- TLS or code-signing identity certificate
```

TLSパスは、emSign Root TLS CA(G3)が直接信頼されている場合は、そのCAで終端することもできます。次の図は、関連する発行者関係を示しています。許容パスは、使用サービスのトラストアンカーと検証ポリシーによって異なります。

この例で示されている親/ルートCAは、チェーンのコンテキストを提供し、侵害された8つの発行キーの中にはカウントされません。親/ルートを削除するか信頼しないかの判断は、侵害された中間キーの識別とは別になります。

識別には、証明書の発行者とシリアル番号をSHA-256フィンガープリントと一緒に使用します。件名だけでは不十分です。証明書フィンガープリントは1つの証明書バリアントを識別します。SPKI SHA-256ハッシュは、元のフォームと相互署名フォームで共有される公開キーを識別します。キャッシュされた中間パスだけを削除しても、ピアは残りの信頼できるパスを介して中間パスを提示できます。

ソースと検証日

証明書IDは、2026年10月29日に公開されたインシデント添付ファイルに対して検証されました。元のペアと相互署名されたペアには、それぞれ同じ公開キーが含まれています。製品の適用性と修復ステータスについては、シスコのライブ情報を使用してください。

シスコ情報：cisco-sa-emudhra-info-ScS22A1r

Mozillaインシデントレコード：eMudhra intermediate key compromise(2083449)

元のCA証明書の添付(9664066)

AAA相互署名CA証明書添付(9664067)

Apple：オリジナルおよび相互署名されたTLS証明書のパスとフィンガープリント

2026年10月31日現在、emSignルートCA - G1はこのガイドに記載されているセキュリティ侵害を受けたCAの中で特定されていません。このサービスへの継続的な信頼は、お客様の組織ポリシーとアドバイザリ更新の対象となります。

Cisco IOS XEによる証明書信頼の管理方法

設定されたPKIトラストポイントは、CA/ID証明書および検証ポリシーを保持します。PKIトラストプールには、そのプールを使用するサービス用に組み込みおよびダウンロードされたCA証明書が含まれています。関連する各サービスで使用される信頼の送信元とポリシーを特定します。トラストプールを変更しても、個別に設定されたトラストポイントは置き換えられません。

1. 監査と準備

ソフトウェアリリース、デバイスの時刻、およびインストールされている証明書を確認します。

<#root>

```
show version show clock show crypto pki trustpoints show running-config | section crypto pki trustpoint
```

サポートコンテキストとしてサブジェクトを使用して、完全な証明書の発行者と16進数のシリアル番号をインベントリと比較します。Cisco IOS XEがMD5およびSHA1フィンガープリントのみを表示する場合、その出力ではインベントリSHA-256フィンガープリントを直接比較できません。これは、デバイス外部の証明書ファイルを検証するために提供されます。シリアル番号の先頭の00パディングは異なる場合があります。ベンダー名検索では、候補のみが識別されます。各トラストポイントに依存するサービスを記録し、元のパスと相互署名パスの両方（直接信頼される中間証明書を含む）を確認します。

例：ルートCAおよび影響を受ける下位CA

詳細証明書の短縮された例が表示されます。他のフィールドは省略されます。件名、発行者、シリアル番号を比較します。

ルートCAの例：emSign ECC Root CA - G3

```
CA Certificate
  Certificate Serial Number (hex): 3CF607A968700EDA8B84
  Issuer:
    cn=emSign ECC Root CA - G3
  Subject:
    cn=emSign ECC Root CA - G3
  Associated Trustpoints:
```

影響を受ける下位CAの例：元のemSign発行証明書

```
CA Certificate
  Certificate Serial Number (hex): 00A4AEE678F23362B4C0E460B772F214
  Issuer:
    cn=emSign Root TLS CA - G3
  Subject:
    cn=EM DV TLS CA - G3B
  Associated Trustpoints:
```

影響を受ける下位CAの例：Sectigo相互署名証明書

```
CA Certificate
  Certificate Serial Number (hex): 00A024FDF4BA665C785CE4A020483858B7
  Issuer:
    cn=AAA Certificate Services
  Subject:
    cn=EM DV TLS CA - G3B
  Associated Trustpoints:
```

両方のSubCAフォームに同じSubjectがありますが、発行者とシリアルが異なります。発行者全体とシリアル番号をインベントリに一致させます。先頭の00パディングは異なる場合があります。所有しているトラストポイントまたはプールを見つけるには、関連付けられたトラストポイントを記録します。示されているルートはチェーンのコンテキストであり、侵害された8つの発行キーの1つではありません。

リカバリアクセス方式と承認済みの交換バンドルを常に利用可能な状態にしておきます。必要に応じて、承認済みの証明書/キーバックアップ手順を使用します。コンフィギュレーションをバックアップします。このコピーだけでは、秘密キーのバックアップにはなりません。

<#root>

copy running-config

かぎカッコのプレースホルダを、承認されたデバイス固有の値に置き換えます。show file systemsで表示されるファイルシステム（flash：やbootflash：など）を使用します。

2. 信頼パスの確認

直接信頼された該当SubCAを削除する

トラストポイントおよびダウンロードしたトラストプールバンドルから、直接信頼できる該当SubCAを削除します。侵害されたCAの直接信頼を無効にするために、失効設定を使用しないでください。

ルートCAの削除（オプション）

ルートCAの削除は、追加の保護のためのオプションの推奨事項です。実際のサービスCRL/OCSPチェックが現在のステータスを取得し、該当するSubCAを信頼できる親の下で評価し、該当する元のチェーン、相互署名されたチェーン、該当する長いチェーンを拒否し、設定されたどちらの方法でも使用可能なステータスが提供されない場合は検証に失敗すると、必要なルートは信頼されたままになります。検証された条件の下で、失効チェックは、ルートを削除することなく、特定された侵害された発行者から保護します。

不要なルート/トラストアンカーを削除して、失効サービスへの依存を減らすことを検討してください。最初にサービスの依存関係を評価します。AAA証明書サービスを削除すると、無関係な正当なチェーンが中断される可能性があります。eMudhraとAAAの両方のパスをカバーします。キャッシュされたSubCAのみを削除しても、残りの信頼された親を介したピア指定のコピーはブロックされません。

選択したトラストポイントの削除

削除対象として選択された管理者設定のトラストポイントの場合：

<#root>

```
configure terminal no crypto pki trustpoint

        end
```

同意する前に確認を確認します。トラストポイントの削除により、関連付けられている証明書が破棄されます。デバイスIDの交換が必要な場合は、デバイスキーと一致する交換証明書を取得し、サービス登録手順を使用してサービスバインディングを更新します。新しいCA証明書だけをインストールしても、デバイスIDは置き換えられません。

ダウンロードされたトラストプールエントリの場合は、バンドル交換手順を使用します。ビルトインまたは機能管理アンカーには、サポートされている製品アップデートまたはCisco TACガイダンスが必要です。

3. 必要に応じて、ダウンロードした信頼プールを更新します

ダウンロードしたバンドルの交換が必要な場合は、次の手順を実行します。必要な親/ルートは、ステップ2で検証された失効条件の下で信頼された状態を維持できます。

3.1ダウンロードしたバンドルがインストールされ、使用されているかどうかを確認する

<#root>

```
show running-config | section crypto pki certificate pool show crypto pki trustpool verbose show crypto
```

Crypto pki certificate poolの下で、Trustpool: DownloadedとマーキングされたCAレコードのケーブルエントリを探します。バンドルファイル名は、ios.p7bまたはios_core.p7bとは異なることがあります。保存されたファイルまたは構成された更新URLだけでは、証明書がサービスによってインストールまたは使用されていることを証明しません。サービスが信頼プールを使用していることを確認し、そのCAレコードを影響を受けるインベントリと比較します。ダウンロードされたCAがインストールされていない場合は、クリーンアップと置換をスキップし、他の信頼ソースを引き続き評価します。

3.2ダウンロードしたトラストプールの消去

クリーンアップの前に、回復アクセスと承認済み回復バンドルを使用可能な状態に保ちます。信頼が削除されることに依存する転送の場合、後でダウンロードすると失敗する可能性があります。依存サービスの中断を計画します。

<#root>

```
configure terminal crypto pki trustpool clean end
```

```
show crypto pki trustpool verbose
```

ダウンロードしたCAレコードが削除されたことを確認します。クリーンアップは、ダウンロードされたすべてのトラストプール証明書とその管理対象バンドルファイルを削除します。組み込みCAと個別のトラストポイントはそのまま残ります。交換とサービスチェックが成功するまで保存しないでください。

3.3承認済みの交換バンドルの入手とインストール

承認済みのシスコ署名付きバンドルを入手します。必要なサービスとしてios.p7bまたはios_core.p7bを選択し、前述の該当する証明書および不要なトラストアンカーを除外することを確認します。2026年10月31日にチェックされたレガシーバンドルには、emSign Root CA - G1 in ios.p7bが含まれていますが、eMudhraもAAA Certificate Services CA in ios_core.p7bも含まれていません。コアの信頼セットは小さくなります。切り替えの前にサービスの依存関係を確認し、インストールするバンドルの内容を確認してください。承認されたHTTPSまたはSCP転送を使用して、管理されたトラストプールストレージの外部にステージングします。

CiscoフルCAバンドル：ios.p7b

シスコのコアCAバンドル：ios_core.p7b

<#root>

```
show file systems copy
```

```
configure terminal crypto pki trustpool import url
```

```
end show crypto pki trustpool verbose
```

たとえば、ステージングされたURLはflash:approved-ios.p7bにすることができます。インポートが失敗した場合は、保存する前に承認済みのバンドルを復元します。目的のCAと必要な接続を確認します。クリーンアップは証明書の拒否リストではありません：残りのトラストポイント、組み込みCA、および代替パスを確認すると、強制された失効またはトラストパスの削除によって影響を受けるチェーンが拒否されます。

4. 検証および更新ポリシーの構成

変更を行う前に、既存のトラストプールポリシー、影響を受けるトラストポイントポリシー、証明書マップのバインディング、および更新URLを記録し、必要に応じて復元できるようにします。

関連する設定されたトラストポイントについて：

<#root>

```
configure terminal crypto pki trustpoint
```

```
revocation-check crl ocsp exit end
```

トラストプールポリシーを使用するサービスの場合：

<#root>

```
configure terminal crypto pki trustpool policy chain-validation revocation-check crl ocsp exit end show
```

表示されたトラストプールポリシーがルートへの検証を続行し、crl ocspをリストすることを確認します。デバイスは最初にCRLを試行し、CRLの取得が利用できないときにOCSPを試行します。無効になった結果は、影響を受ける証明書が評価されたときに検証に失敗します。正常な失効確認が必要な場合は、noneを追加しないでください。

正確な時刻、到達可能なCA公開ステータスエンドポイント、受け入れられる応答形式、および適用可能な失効スキップ例外がないことを確認します。影響を受けるSubCAのステータスは、信頼できる親によって評価される必要があります。SubCAの直接信頼は、親がチェックされる前に検証を停止できます。実際のサービスが使用可能な現在のステータスを取得し、影響を受けるチェーンを拒否することを確認します。ステータスサーバへのアクセスと拒否の動作を確認できない場合は、設定だけを確認済みの保護として扱わないでください。軽減計画で必要とされる場合は、関連するトラストパスを削除してください。利用不可ステータスをバイパスするには、noneを追加しないでください。

保護を確認する前にキャッシュされたCRLを更新する

ローカルキャッシュまたはデータベースに既に存在する使用可能なCRLは、CAが新しいCRLを発行した後でも再利用できます。status-server reachabilityを確認し、保護を検証する前に古いCRLをクリアします。

<#root>

```
show crypto pki crls clear crypto pki crl show crypto pki crls
```

clearコマンドは、すべてのトラストポイントのローカルCRLを削除します。それ自体で置換をダウンロードするわけではありません。古いエントリが削除されたことを確認してから、影響を受けるサービスで新しい接続または証明書の検証を開始します。show crypto pki crlsを再度実行し、関連する各発行者、LastUpdate、NextUpdate、および配布ポイントをCAの現在のパブリケーションと照合します。ID証明書の発行者のCRLだけでなく、影響を受けるSubCAをカバーする親の発行者のCRLを含めてください。該当するチェーンが拒否されたことを確認します。ステータス・データのフェッチ中にクリアすると、新しい接続に一時的に影響が及ぶ場合があります。フェールクローズ検証は保持されます。

バンドルの更新ソースの確認

接続されたデバイスに対して、オプションで承認済みの有効期限駆動バンドル更新ソースを設定します。

<#root>

configure terminal crypto pki trustpool policy cabundle url

exit end show crypto pki trustpool policy

設定されているすべてのバンドルソースを確認します。有効期限による更新は、即時のセキュリティ更新に置き換わるものではなく、不要な信頼パスを復元することはできません。

オプションの証明書マップフィルタ：カバレッジと制限

Cisco IOS XE証明書マップ(cert-map)は、発行者名などの証明書属性と一致します。追加の手段として、このフィルタを使用すると、リストされている8つのCAから直接発行されたID (リーフ) 証明書を拒否できます。トラストポイントの既存の照合証明書ポリシーを確認してから適用します。

2026年10月29日に、実験用ラボデバイスでEM DV TLS CA(G3B)の下でラボIDを使用して動作がテストされました。PKIピア証明書の検証では、フィルタは直接発行されたIDを拒否しましたが、追加の子SubCAによって発行されたIDをブロックしませんでした。リストにない正当な発行者が渡されました。動作カバレッジは、フィルタを分離するために一時的なラボトラストポイントでのみ無効化された失効で、影響を受ける1つの発行者を使用しました。アプリケーションのハンドシェイクやその他のリリースは、このフィルタでは検証されませんでした。実際のサービスに依存する前にテストしてください。

<#root>

configure terminal crypto pki certificate map EMSIGN-ISSUER-FILTER 10 issuer-name nc cn=EM DV TLS CA - G

match certificate EMSIGN-ISSUER-FILTER exit end

ここで、ncは「含まない」を意味します。 8つの負のissuer-name条件すべてを同じエントリ（シーケンス10）内に保持します。このフィルタに証明書が合格するには、すべてが一致している必要があります。リストされた名前に一致する直接発行者はフィルタに失敗します。

Cisco IOS XEは、issuer-name nc cn = em dv tls ca - g3bなどの正規化されたエントリを表示できます。動作テストでは、この小文字とスペースの形式が使用されました。マップと、サービスによって実際に使用される検証トラストポイントへのバインドを確認します。

<#root>

show running-config | section crypto pki certificate map show running-config | section crypto pki trustp

フィルタは、チェーン全体のCA拒否リストを提供しません。以下に、いくつかの例を示します。

```
Trusted root CA
  +-- Affected SubCA
       +-- New child SubCA
            +-- ID (leaf) certificate
```

ID証明書issuerは新しい子SubCAであるため、そのissuer-nameフィルタは、チェーン内で上位にある影響を受けるCAを識別しません。CAの制約や失効など、その他のチェックは引き続き適用されます。このフィルタは、このガイドの失効検証とオプションのルート削除手順とともに、追加の保護としてのみ使用してください。直接発行された証明書、子SubCA証明書、および正当な証明書を、実際のサービスでテストします。

ステップ4のロールバック

記録された失効方法、チェーン検証設定、および証明書マップのバインドを復元します。この手順で追加された更新URLを削除し、以前に変更されたURLをすべて復元します。次の例では、追加された更新URLとフィルタバインディングのみを削除し、その他のポリシー設定は復元しません。

<#root>

configure terminal crypto pki trustpool policy no cabundle url

```
       exit
```

```
crypto pki trustpoint
```

```
       no match certificate EMSIGN-ISSUER-FILTER exit end
```

この手順でURLとバインディングが追加された場所でのみこれらの削除を実行します。以前の証明書マップのバインドが置き換えられた場合は、復元します。新しく作成したマップは、他の設定がそのマップを参照していないことを確認した後にのみ削除します。

デフォルトへのリセットが意図された承認済みの結果である場合にのみ、no crypto pki trustpool policyを使用します。この大幅なリセットにより、設定済みの更新URLがクリアされ、失効確認が「なし」にリセットされます。以前のカスタムポリシーは復元されません。ポリシーロールバックでは、削除された証明書の復元、インストールされたバンドルの復元、クリアされたCRLの復元は行われません。保存する前に、必要なサービスと影響を受けるチェーンの拒否を再確認します。

5. 保護の確認と保存

- 影響を受ける元のチェーンと相互署名されたチェーンが、ピア指定の中間品を含めて拒否されることを確認します。より長いチェーンが関係する場合は、影響を受けるCAがチェーン内でより高いときに拒否を確認します。
- 必要な正当な接続が引き続き機能し、交換IDが目的のサービスにバインドされていることを確認します。
- 失効データが利用できない場合の動作を確認します。既存のTLS/IPsecセッションは、ポリシーの変更によって自動的に再検証されることはありません。必要に応じて、サービスサポートされている再接続手順を使用してください。

チェックが成功した後：

<#root>

```
show crypto pki certificates verbose show crypto pki trustpool policy write memory
```

次回の計画的なリロード後、および必要に応じてスタンバイデバイスで持続性を確認します。結果が予想される信頼ポリシーと異なる場合は、ログを保持し、検証された信頼の変更を検討する前にCisco TACに連絡してください。

参考資料

Cisco IOS XE PKIの信頼プール管理

シスコ：Cisco IOS XE PKIを使用したCA署名付き証明書の設定

Cisco:CRLキャッシングおよび失効チェック

Cisco IOS XE GuestShell CA信頼ガイド

ステップ1：デバイスGuestShellの展開状態の監査

修復が必要なのは、GuestShellがインストールされ、RUNNING状態またはDEPLOYED状態にあるCisco IOS XEシステムだけです。GuestShellがDEPLOYED状態の場合は、修復コマンドを実行する前に、コマンド「guestshell enable」を使用してまず有効にする必要があります。show app-hosting listで「No App found」と表示される（GuestShellがデプロイされていないか有効になっている）場合、GuestShellでアクションを実行する必要はありません。

A. DEPLOYED状態のGuestShell（アクションが必要：最初に有効にする必要があります）

show app-hosting listでDEPLOYED状態のGuestShellが報告される場合、GuestShellコンテナストレージは存在しますが、コンテナは停止します。bashを入力する前に'guestshell enable'を実行する必要があります（修復完了後に停止状態に戻った場合は、オプションで'guestshell disable'を実行します）。

<#root>

Router#

show app-hosting list

```
App id                                    State
------------------------------------------------------------
guestshell                                DEPLOYED
```

B. RUNNING状態のGuestShell（操作が必要：直接続行）

show app-hosting listでRUNNING状態のGuestShellが報告される場合、GuestShellはアクティブに実行されており、bashに直接進むことができます。

<#root>

```
Router#

show app-hosting list

App id                                    State
------------------------------------------------------------
guestshell                                RUNNING
```

B. GuestShellが有効でないか、または展開されていない（アクションは不要）

show app-hosting listで「No App found」と表示された場合、GuestShellはシステムにインストールされていないか有効になっていません。この場合は特に対処の必要はありません。

<#root>

```
Router#

show app-hosting list

No App found
```

ステップ2:GuestShellへのアクセスとCA不信感の適用

Cisco IOS XE特権EXEC(「enable」)プロンプトから、次のコマンドを実行します。GuestShellがDEPLOYED状態の場合は、最初に有効にします。次に、bashシェルに入り、ルートにエスカレートし、動的なフォルダ選択を使用してCAブロックリストを設定し、すべてのCisco IOS XEリリース間で互換性を確保します。

<#root>

```
! 1. If GuestShell is in DEPLOYED state, enable it first:
Router#
```

```
guestshell enable


! 2. Enter GuestShell bash shell:
Router#

guestshell run bash


! 3. Escalate to root user inside GuestShell:
[guestshell@guestshell ~]$

sudo -s


! 4. Prepare the ca-trust blocklist directory (compatible with 17.15.x and 17.18.x):
d=/etc/pki/ca-trust/source/blocklist
[ -d "$d" ] || d=/etc/pki/ca-trust/source/blacklist
mkdir -p "$d"
```

侵害されたMudhra証明書の信頼を失う

trust extractコマンドを実行して、侵害されたeMudhra証明書をca-trust blocklistディレクトリに直接抽出してから、システム信頼ストアを再構築します。

```
trust extract --overwrite --format=pem-bundle \
  --filter='pkcs11:object=emSign%20ECC%20Root%20CA%20-%20G3;type=cert' \
  "$d/emSign_ECC_Root_CA_-_G3.pem"

trust extract --overwrite --format=pem-bundle \
  --filter='pkcs11:object=Comodo%20AAA%20Services%20root;type=cert' \
  "$d/Comodo_AAA_Services_root.pem"

trust extract --overwrite --format=pem-bundle \
  --filter='pkcs11:object=emSign%20ECC%20Root%20CA%20-%20C3;type=cert' \
  "$d/emSign_ECC_Root_CA_-_C3.pem"

trust extract --overwrite --format=pem-bundle \
  --filter='pkcs11:object=emSign%20Root%20CA%20-%20C1;type=cert' \
  "$d/emSign_Root_CA_-_C1.pem"

update-ca-trust extract
```

ステップ3:GuestShellで信頼できない証明書を確認する

次の検証コマンドを使用して、抽出された4つの証明書がシステムの信頼/ブロックリストポリシーに含まれ、アクティブなトラストアンカーから除外されていることを確認します。コマンドは、証明書がブロックリストポリシーに登録されていることを確認します。

```
###Verify with below command
[guestshell@guestshell ~]$ trust list --filter=trust-policy --purpose=server-auth | grep -E -A3 'emSign
    label: AAA Certificate Services
    trust: distrusted
    category: authority

--
    label: emSign ECC Root CA - C3
    trust: distrusted
    category: authority

--
    label: emSign ECC Root CA - G3
    trust: distrusted
    category: authority

--
    label: emSign Root CA - C1
    trust: distrusted
    category: authority

--
    label: emSign Root CA - G1
    trust: anchor
    category: authority

Output will vary based on CentOS Stream version
###Verify CentOS Stream Version
[guestshell@guestshell ~]$ cat /etc/redhat-release
CentOS Stream release 8

OS                          CentOS Stream 8       CentOS Stream 9
Directory                   blacklist only        blocklist
Result shown                 trust: blacklisted    trust: distrusted
```

ステップ4：状態の持続性とコンテナのライフサイクル

コンテナの持続性に関する注意：これらの証明書の不信変更は、/etc/pki/ca-trustが永続的なコンテナストレージに存在するため、「guestshell enable」および「guestshell disable」操作とデバイスの再起動を行っても保持されます。

DEPLOYED状態に戻る（該当する場合）:GuestShellがもともとDEPLOYED状態にあり、この不信手順の実行のみを有効にしている場合は、bashを終了し、IOS特権EXECプロンプトから「guestshell disable」を実行して、GuestShellを以前の停止状態に戻します。

<#root>

```
[root@guestshell ~]#
```

exit

```
[guestshell@guestshell ~]$
```

```
exit

Router#

guestshell disable
```

ライフサイクルに関する重要な考慮事項：GuestShellが「guestshell destroy」で完全に破棄されると、コンテナファイルシステムは削除されます。その後、GuestShellを再度有効にすると、コンテナはベースシステムイメージから再作成されるため、これらの不信回復手順を再適用する必要があります。

ステップ5：ロールバック手順（CA信頼の復元）

任意のCisco IOS XEリリースで不信設定を元に戻し、デフォルトのCA信頼を復元するには、アクティブなブロックリストディレクトリ($d)を動的に選択し、この手順で追加された4つの証明書ファイル（Comodo AAA Servicesルートを含む）を明示的に削除し、信頼ストアを再構築します。

<#root>

```
! Escalate to root user inside GuestShell:
[guestshell@guestshell ~]$

sudo -s


d=/etc/pki/ca-trust/source/blocklist
[ -d "$d" ] || d=/etc/pki/ca-trust/source/blacklist

! Remove only the four certificate files added during remediation:
rm -f "$d/emSign_ECC_Root_CA_-_G3.pem" "$d/Comodo_AAA_Services_root.pem" \
      "$d/emSign_ECC_Root_CA_-_C3.pem" "$d/emSign_Root_CA_-_C1.pem"

! Rebuild the system CA trust store:
update-ca-trust extract
```

削除したファイルがブロックリストディレクトリに含まれていないことを確認します。

<#root>

```
[root@guestshell ~]#

ls -l "$d"/

total 0
```

CA Trust Restorationの確認：手順3の確認コマンド（すべてのエントリがtrust: anchorと報告される）を再実行して、5つの証明書がすべてアクティブなトラストアンカーに戻ったことを確認します。

<#root>

```
[root@guestshell ~]#

trust list --filter=trust-policy --purpose=server-auth | grep -E -A3 'emSign|AAA'

    trust: anchor
    trust: anchor
    trust: anchor
    trust: anchor
    trust: anchor
```

エンドツーエンドの実行トランスクリプトの例

このトランスクリプトは、Cisco IOS XEデバイス上で成功した、完全なGuestShell監査、修復、および検証セッションを示しています。

<#root>

```
Router#

guestshell run bash

[guestshell@guestshell ~]$ sudo -s
bash-4.4#

d=/etc/pki/ca-trust/source/blocklist

bash-4.4#

[ -d "$d" ] || d=/etc/pki/ca-trust/source/blacklist

bash-4.4#

mkdir -p "$d"

bash-4.4#

trust extract --overwrite --format=pem-bundle \ >   --filter='pkcs11:object=emSign%20ECC%20Root%20CA%20-

bash-4.4#

trust extract --overwrite --format=pem-bundle \ >   --filter='pkcs11:object=Comodo%20AAA%20Services%20ro
```

```
bash-4.4#

trust extract --overwrite --format=pem-bundle \ >   --filter='pkcs11:object=emSign%20ECC%20Root%20CA%20-

bash-4.4#

trust extract --overwrite --format=pem-bundle \ >   --filter='pkcs11:object=emSign%20Root%20CA%20-%20C1;

bash-4.4#

update-ca-trust extract

bash-4.4#

trust list --filter=trust-policy --purpose=server-auth | grep -E -A3 'emSign|AAA'

    label: emSign ECC Root CA - G3
    trust: blacklisted
    category: authority

--
    label: AAA Certificate Services
    trust: blacklisted
    category: authority

--
    label: emSign ECC Root CA - C3
    trust: blacklisted
    category: authority

--
    label: emSign Root CA - C1
    trust: blacklisted
    category: authority

--
    label: emSign Root CA - G1
    trust: anchor
    category: authority

bash-4.4#

cat /etc/redhat-release

CentOS Stream release 8
bash-4.4#

d=/etc/pki/ca-trust/source/blocklist

bash-4.4#

[ -d "$d" ] || d=/etc/pki/ca-trust/source/blacklist

bash-4.4#
```

```
rm -f "$d/emSign_ECC_Root_CA_-_G3.pem" "$d/Comodo_AAA_Services_root.pem" \ >       "$d/emSign_ECC_Root_C

bash-4.4#
bash-4.4#

update-ca-trust extract

bash-4.4#

ls -l "$d"/

total 0
```

コマンドの要約マトリックス

運用タスク	CLI/
GuestShell状態の監査	show app-hosting list
GuestShellを有効にする（展開されている場合）	guestshell enable
GuestShell Bashと入力します。	guestshell run bash
ルートユーザへの切り替え	sudo -s
ブロックリストディレクトリの準備	d=/etc/pki/ca-trust/source/blocklist; [-d "$d"] || d=/etc/pki/ca-trust/source/blacklist;
ブロックリストへの	trust extract --overwrite --format=pem-bundle ... (x4 CAs)

CAの抽出	
CA信頼ストアの再構築	`update-ca-trust extract`
信頼できないポリシーの確認	`trust list --filter=trust-policy --purpose=server-auth`
展開済み状態の復元（オプション）	`guestshell disable`
信頼のロールバック/リストア	`rm -f "$d/emSign_ECC_Root_CA_-_G3.pem" "$d/Comodo_AAA_Services_root.pem" "$d/emSign_ECC_Ro`

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。