CA署名付きサーバでホストされるガジェットのFinesseエラー「SSLPeerUnverifiedException」のトラブルシューティング

内容

[概要](#)
[前提条件](#)
[要件](#)
[使用するコンポーネント](#)
[背景説明](#)
[課題](#)
[シナリオ 1：ホスティングサーバが非セキュアTLSをネゴシエートする](#)
[解決方法](#)
[シナリオ 2：証明書にサポートされていない署名アルゴリズムがあります](#)
[解決方法](#)

概要

このドキュメントでは、ガジェットをホストする外部Webサーバの認証局(CA)署名付き証明書チェーンがFinesseにアップロードされるが、Finesseにログインしたときにガジェットがロードされず、エラー「SSLPeerUnverifiedException」が表示されるシナリオのトラブルシューティング手順について説明します。

著者：Cisco TACエンジニア、Gino Schweinsberger

前提条件

要件

次の項目に関する知識があることが推奨されます。

- SSL 証明書
- Finesse管理
- Windows Serverの管理
- Wiresharkによるパケットキャプチャ分析

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアのバージョンに基づいています。

- Unified Contact Center Express(UCCX)11.X
- Finesse 11.X

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、初期（デフォルト）設定の状態から起動しています。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。

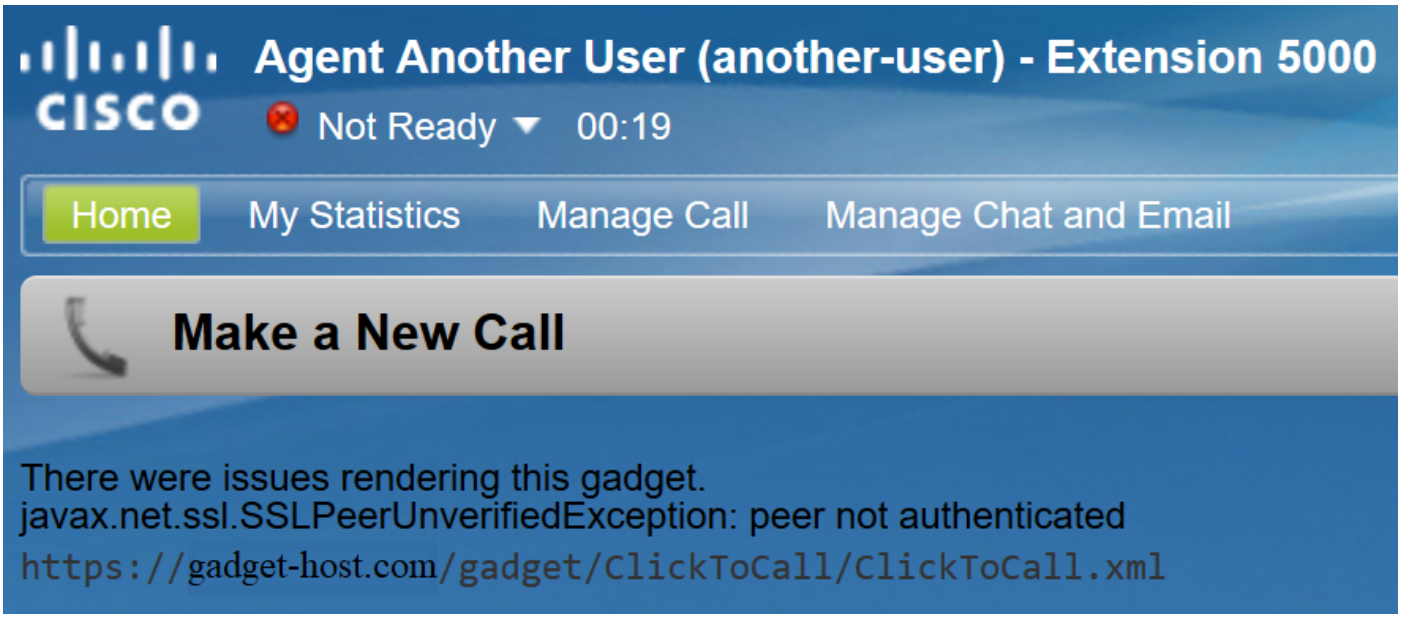
背景説明

エラーが発生する条件は次のとおりです。

- 証明書信頼チェーンがFinesseにアップロードされていると仮定する
- 正しいサーバ/サービスが再起動されたことを確認します。
- ガジェットがHTTPS URLを使用してFinesseレイアウトに追加され、URLが到達可能であると仮定します

これは、エージェントがFinesseにログインするときに観察されるエラーです。

「このガジェットのレンダリングで問題が発生しました。
javax.net.ssl.SSLPeerUnverifiedException:peer not authenticated



課題

シナリオ 1：ホスティングサーバが非セキュアTLSをネゴシエートする

Finesse Serverがホスティングサーバに接続要求を行うと、Finesse Tomcatはサポートする暗号化方式のリストをアドバタイズします。

セキュリティ上の脆弱性のため、一部の暗号はサポートされていません。

ホスティングサーバがこれらの暗号のいずれかを選択すると、接続は拒否されます。

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA

これらの暗号は、接続をネゴシエートするときに弱い一時的なDiffie-Hellman(DH)キーを使用することが知られており、Logjamの脆弱性により、これらはTLS接続に適していません。

パケットキャプチャのTLSハンドシェイクプロセスに従って、ネゴシエートされた暗号を確認します。

1. Finesseは、サポートされている暗号のリストをClient Helloステップで表示します。

```
▼ TLSv1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 67
  ▼ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 63
      Version: TLS 1.0 (0x0301)
    ▶ Random: 5cacb293b5efdb4cf1bb34464d7de9f5060b00a9beb81d29…
      Session ID Length: 0
      Cipher Suites Length: 24
    ▼ Cipher Suites (12 suites)
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
        Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
        Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
        Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
        Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
        Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
        Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
        Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Cipher Suite: TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff)
      Compression Methods Length: 1
    ▶ Compression Methods (1 method)
```

2.この接続の場合、TLS_DHE_RSA_WITH_AES_256_CBC_SHAがServer Helloステップ中にホスティングサーバによって選択されました。これは、優先暗号のリストの方が上位にあるためです。

```
TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 2557
  Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.0 (0x0301)
    Random: 5cacb292c4d7183627f620a066f9b6ce6460dcb849b59cae…
    Session ID Length: 32
    Session ID: 4c290000ce66098cc994a33e193b0da1244cb9f083f69c26…
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
    Compression Method: null (0)
    Extensions Length: 5
    Extension: renegotiation_info (len=1)
  Handshake Protocol: Certificate
  Handshake Protocol: Server Key Exchange
    Handshake Type: Server Key Exchange (12)
    Length: 1032
    Diffie-Hellman Server Params
  Handshake Protocol: Server Hello Done
    Handshake Type: Server Hello Done (14)
    Length: 0
```

3. FinesseがFatalアラートを送信し、接続を終了します。

```
TLSv1 Record Layer: Alert (Level: Fatal, Description: Internal Error)
  Content Type: Alert (21)
  Version: TLS 1.0 (0x0301)
  Length: 2
  Alert Message
```

解決方法

これらの暗号の使用を防止するには、ホスティングサーバがこれらを低い優先順位に設定するか、使用可能な暗号のリストから完全に削除する必要があります。これは、Windowsグループポリシーエディタ(gpedit.msc)を使用してWindows Serverで実行できます。

注:FinesseでのLogjamの影響とgpeditの使用の詳細については、次を確認してください。

シナリオ2：証明書にサポートされていない署名アルゴリズムがあります

Windows Serverの証明機関は、新しい署名標準を使用して証明書に署名できます。SHAよりも優れたセキュリティを提供しますが、Microsoft製品以外でこれらの標準を採用することは少なく、

管理者は相互運用性の問題に遭遇する可能性があります。

Finesse Tomcatは、JavaのSunMSCAPIセキュリティプロバイダを利用して、Microsoftが使用するさまざまな署名アルゴリズムと暗号化機能のサポートを有効にします。Javaの現在のすべてのバージョン（1.7、1.8、および1.9）は、次の署名アルゴリズムのみをサポートしています。

- MD5withRSA
- MD2withRSA
- NONEwithRSA
- SHA1withRSA
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA

Finesseサーバで実行されているJavaのバージョンを確認し、そのバージョンでサポートされているアルゴリズムを確認することをお勧めします。バージョンは、次のコマンドでルートアクセスから確認できます。**javaバージョン**

```
Using username "root".
Last login: Tue Apr 16 13:11:00 2019 from
[root@uccx12pub ~]# java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (rhel-2.6.14.8.el6_9-i386 u181-b00)
OpenJDK Server VM (build 24.181-b00, mixed mode)
[root@uccx12pub ~]#
```

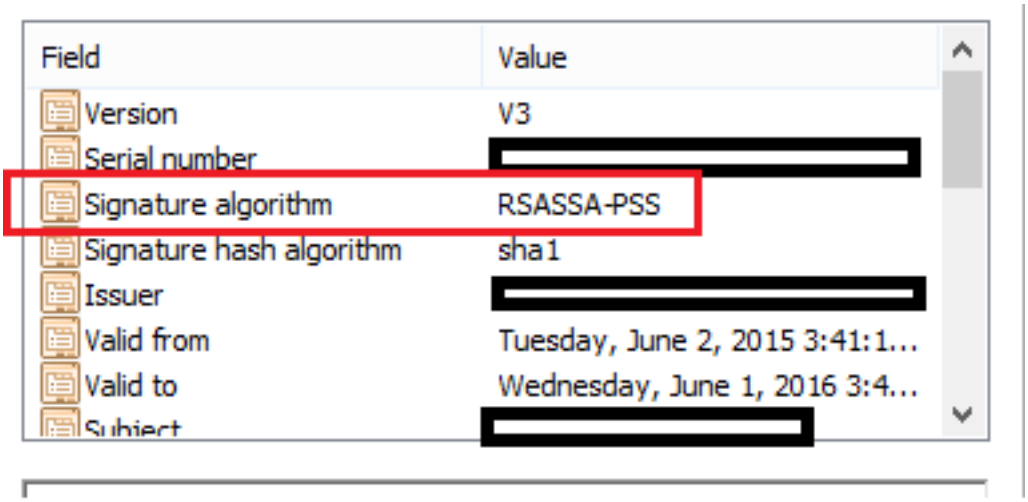
> **注**:Java SunMSCAPIプロバイダーの詳細については、https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunMSCAPIを参照してください。

証明書に上記の署名以外の署名が指定されている場合、Finesseはその証明書を使用してホスティングサーバへのTLS接続を作成できません。これには、サポートされている署名タイプで署名された証明書が含まれますが、それ以外で署名された独自の中間証明書とルート証明書を持つ証明機関によって発行されました。

パケットキャプチャを見ると、Finesseは「Fatal alert:図に示すように、Certificate Unknown」エラーが発生します。

```
Secure Sockets Layer
⊟ TLSv1.2 Record Layer: Alert (Level: Fatal, Description: Certificate Unknown)
    Content Type: Alert (21)
    Version: TLS 1.2 (0x0303)
    Length: 2
  ⊟ Alert Message
      Level: Fatal (2)
      Description: Certificate Unknown (46)
```

この時点で、ホスティングサーバによって提示された証明書を確認し、サポートされていない署名アルゴリズムを探す必要があります。問題のあるシグニチャアルゴリズムとして**RSASSA-PSS**を参照するのが一般的です。



チェーン内のいずれかの証明書がRSASSA-PSSで署名されている場合、接続は失敗します。この場合、パケットキャプチャは、ルートCAが自身の証明書にRSASSA-PSSを使用していることを示します。

```
Certificates (3906 bytes)
  Certificate Length: 1728
  Certificate: 308206bc308205a4a003020102021374000000243b805da9... (id-at-commonName=[redacted]
    signedCertificate
    algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: e6230df257be9d34c0f57bc2f88c081c4186aaad092c8155...
  Certificate Length: 1114
  Certificate: 308204563082033ea003020102021316000000a93cd17d6... (id-at-commonName=[redacted]Issuing Authority[redacted]
    signedCertificate
    algorithmIdentifier (sha256WithRSAEncryption)
      Padding: 0
      encrypted: 889be6a1125c758cd0009b392d3b90a69b64546dcee09c84...
  Certificate Length: 1055
  Certificate: 3082041b308202cfa00302010202107b70dbb7c2760da74f... (id-at-commonName=[redacted]Root CA [redacted]
    signedCertificate
    algorithmIdentifier (id-RSASSA-PSS)
      Algorithm Id: 1.2.840.113549.1.1.10 (id-RSASSA-PSS)
      RSASSA-PSS-params
    Padding: 0
    encrypted: d8e9151adc76b4e55f9277fce916613ce26199e3b50dcb54...
```

解決方法

この問題を解決するには、前に説明したように、証明書チェーン全体でリストされているサポートされているSunMSCAPI署名タイプの1つだけを使用するCAプロバイダから新しい証明書を発行する必要があります。

> **注**：RSASSA-PSS署名アルゴリズムの詳細については、https://pkisolutions.com/pkcs1v2-1rsassa-pss/を参照してください。

> **注**：この問題は、不具合CSCve79330

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。