對CA簽名的伺服器上託管的小工具的Finesse錯誤「SSLPeerUnverifiedException」進行故障排除

目錄

簡介

本文檔介紹對以下場景進行故障排除的步驟：證書頒發機構(CA)簽名的證書鏈上傳到Finesse，用於承載小工具的外部Web伺服器，但小工具在您登入到Finesse時無法載入，並且您看到錯誤「SSLPeerUnverifiedException」。

作者：Gino Schweinsberger，思科TAC工程師。

必要條件

需求

思科建議您瞭解以下主題：

- SSL證書
- Finesse管理
- Windows Server管理
- 使用Wireshark進行資料包捕獲分析

採用元件

本檔案中的資訊是根據以下軟體版本：

- 整合客服中心Express版(UCCX)11.X
- Finesse 11.X

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

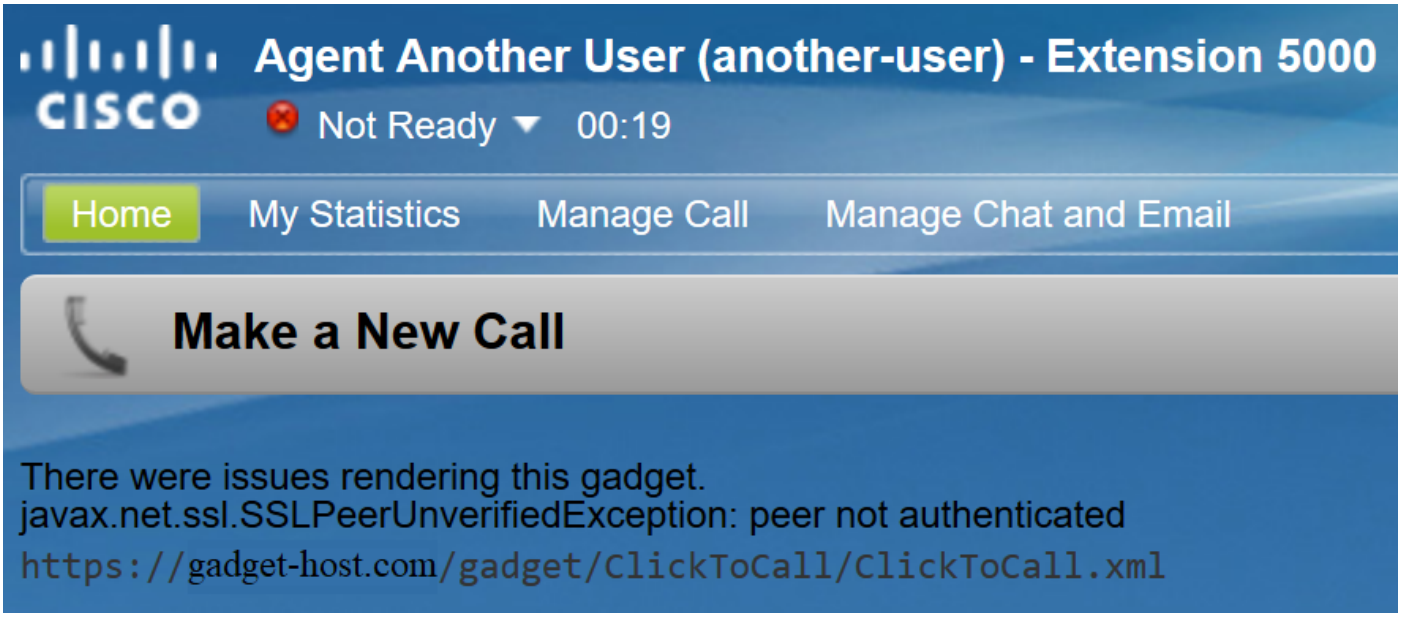
背景資訊

以下是發生錯誤的條件：

- 假設證書信任鏈已上傳到Finesse
- 確保重新啟動了正確的伺服器/服務
- 假定已使用HTTPS URL將小工具新增到Finesse佈局中，並且該URL可訪問

這是代理登入到Finesse時觀察到的錯誤：

「這個小玩意兒在投產上存在問題。javax.net.ssl.SSLPeerUnverifiedException:peer not authenticated"



問題

案例 1:宿主伺服器協商不安全的TLS

當Finesse Server向託管伺服器發出連線請求時，Finesse Tomcat會通告其支援的加密密碼清單。

由於存在安全漏洞，某些密碼不受支援，

如果託管伺服器選擇以下任一密碼，則拒絕連線：

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA

眾所周知，這些密碼在協商連線時使用短暫Diffie-Hellman金鑰，而Logjam漏洞使得這些金鑰對TLS連線來說不是很好的選擇。

按照資料包捕獲中的TLS握手過程檢視協商的密碼。

1. Finesse在**Client Hello** 步驟中顯示其支援的密碼清單：

```
▾ 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.對於此連線，託管伺服器在Server Hello步驟中選擇了TLS_DHE_RSA_WITH_AES_256_CBC_SHA，因為該連線在其首選密碼清單中位於較高位置。

```
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傳送致命警報並結束連線：

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

解決方案

為了防止使用這些密碼，必須將託管伺服器配置為賦予它們低優先順序，或者必須將其從可用密碼清單中完全刪除。這可以在Windows伺服器上使用Windows組策略編輯器(gpedit.msc)完成。

注意：有關Finesse中Logjam的影響以及gpedit使用的詳細資訊，請檢視：

案例 2:證書具有不受支援的簽名演算法

Windows Server證書頒發機構可以使用較新的簽名標準來簽署證書。即使它提供比SHA更高的安全性，在Microsoft產品之外採用這些標準的程度也很低，管理員可能遇到互操作性問題。

Finesse Tomcat依靠Java的SunMSCAPI安全提供程式來支援Microsoft使用的各種簽名演算法和加密函式。所有當前版本的Java（1.7、1.8和1.9）僅支援以下簽名演算法：

- MD5與RSA
- MD2與RSA
- NONEwithRSA
- SHA1 withRSA
- 帶RSA的SHA256
- 帶RSA的SHA384
- 帶RSA的SHA512

檢查在Finesse伺服器上運行的Java版本以確認該版本支援哪些演算法，這是個好主意。可以通過以下命令從根訪問檢查版本：**java -version**

```
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 ~]#
```

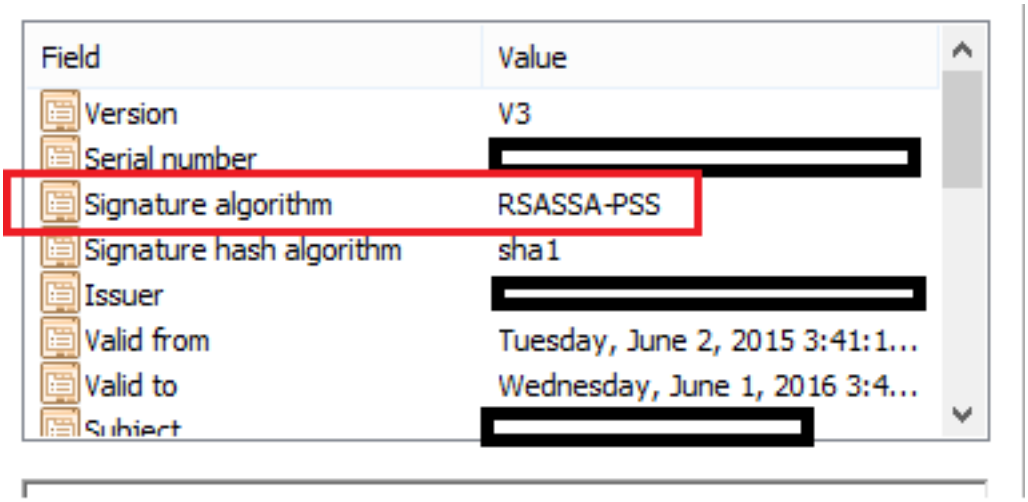
> **註**：有關Java SunMSCAPI提供程式的詳細資訊，請參閱 https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunMSCAPI

如果證書提供的簽名不是上面列出的簽名，則Finesse無法使用該證書建立與宿主伺服器的TLS連線。這包括使用支援的簽名型別進行簽名，但由證書頒發機構頒發的證書，這些證書擁有自己的中間和根證書，這些證書由其他證書簽名。

如果您檢視資料包捕獲，Finesse會使用「Fatal alert:憑證未知」錯誤，如圖所示。

```
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)
```

此時必須檢查託管伺服器提供的證書並查詢不受支援的簽名演算法。**RSASSA-PSS**通常被視為有問題的簽名演算法：



如果鏈中的任何證書使用RSASSA-PSS簽名，則連線失敗。在這種情況下，資料包捕獲顯示根CA將RSASSA-PSS用於自己的證書：

```
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...
```

解決方案

為了解決此問題，必須從CA提供程式頒發新證書，該提供程式僅使用整個證書鏈中列出的受支援的SunMSCAPI簽名型別之一，如前所述。

> **註**：有關RSASA-PSS簽名演算法的詳細資訊，請參閱https://pkisolutions.com/pkcs1v2-1rsassa-pss/

> **附註**：此問題會在缺陷CSCve79330中追蹤

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。