对CA签名服务器上托管的小工具的Finesse错误“SSLPeerUnverifiedException”进行故障排除

目录

简介

本文档介绍对以下场景进行故障排除的步骤：证书颁发机构(CA)签名的证书链上传到Finesse用于托管小工具的外部Web服务器，但小工具在您登录Finesse时加载失败，并且您看到错误“SSLPeerUnverifiedException”。

作者：Gino Schweinsberger，Cisco TAC工程师。

先决条件

要求

Cisco 建议您了解以下主题：

- SSL证书
- Finesse管理
- Windows Server管理
- 使用Wireshark进行数据包捕获分析

使用的组件

本文档中的信息基于以下软件版本：

- 统一联络中心快捷版(UCCX)11.X
- Finesse 11.X

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

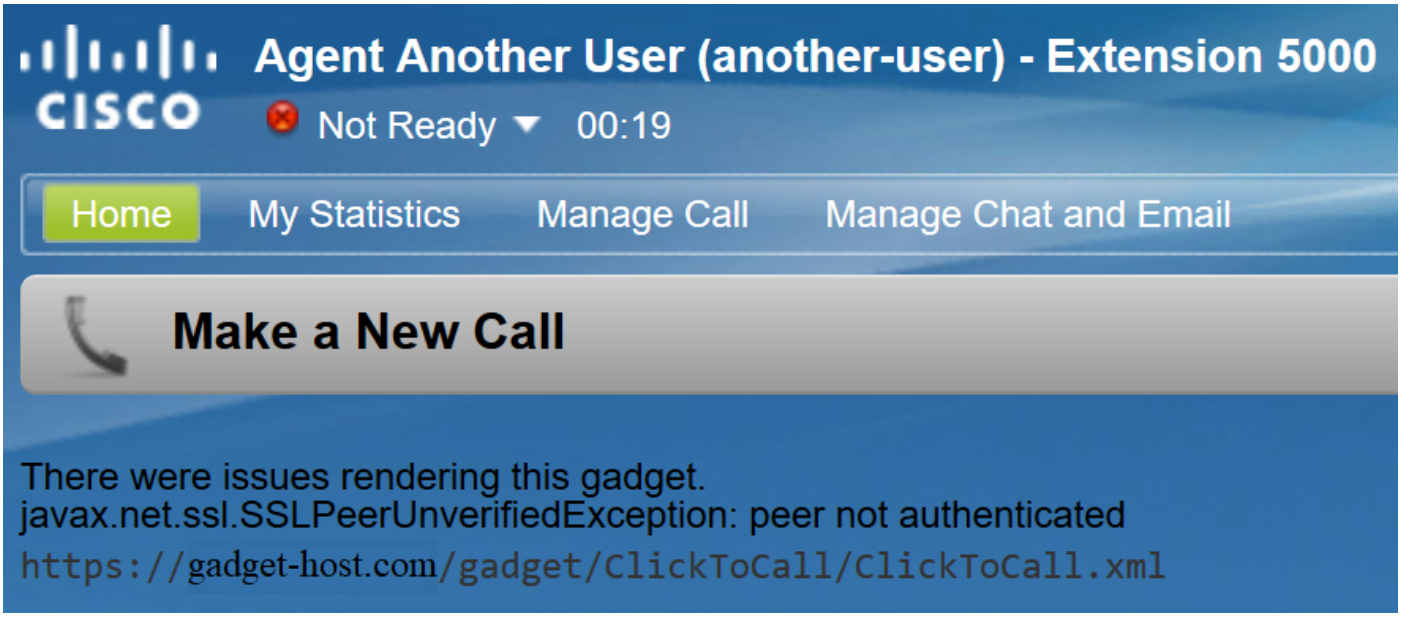
背景信息

以下是导致出现错误的条件：

- 假设证书信任链已上传到Finesse
- 确保重新启动了正确的服务器/服务
- 假设已使用HTTPS URL将小工具添加到Finesse布局中，并且该URL可访问

这是当代理登录到Finesse时观察到的错误：

“这个小玩意儿在制作时出了问题。javax.net.ssl.SSLPeerUnverifiedException:对等体未验证”



问题

情形 1：托管服务器协商不安全的TLS

当Finesse服务器向托管服务器发出连接请求时，Finesse Tomcat会通告其支持的加密密码列表。

由于存在安全漏洞，某些密码不受支持，

如果托管服务器选择以下任一密码，则拒绝连接：

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA

已知这些密码在协商连接时会使用较弱的短暂Diffie-Hellman密钥，而Logjam漏洞使得这些密码不适合TLS连接。

在数据包捕获过程中执行TLS握手过程，以查看协商的密码。

1. Finesse在**Client Hello**步骤中显示其支持的密码列表：

```
▾ 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.对于此连接，TLS_DHE_RSA_WITH_AES_256_CBC_SHA在Server Hello步骤期间由托管服务器选定，因为此值在其首选密码列表中较高。

```
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发送致命警报并结束连接：

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

解决方案

为了防止使用这些密码，必须将托管服务器配置为赋予这些密码较低的优先级，或者必须将其从可用密码列表中完全删除。这可以在Windows服务器上使用Windows组策略编辑器(gpedit.msc)完成。

注意：有关Finesse中Logjam的影响以及gpedit使用的详细信息，请查看：

方案 2：证书具有不受支持的签名算法

Windows Server证书颁发机构可以使用较新的签名标准来签署证书。即使它提供比SHA更高的安全性，在Microsoft产品之外对这些标准的采用率也很低，管理员可能会遇到互操作性问题。

Finesse Tomcat依赖于Java的SunMSCAPI安全提供程序，以支持微软使用的各种签名算法和加密功能。所有当前版本的Java（1.7、1.8和1.9）仅支持以下签名算法：

- MD5与RSA
- MD2与RSA
- NONEwithRSA
- SHA1与RSA
- 带RSA的SHA256
- 带RSA的SHA384
- 带RSA的SHA512

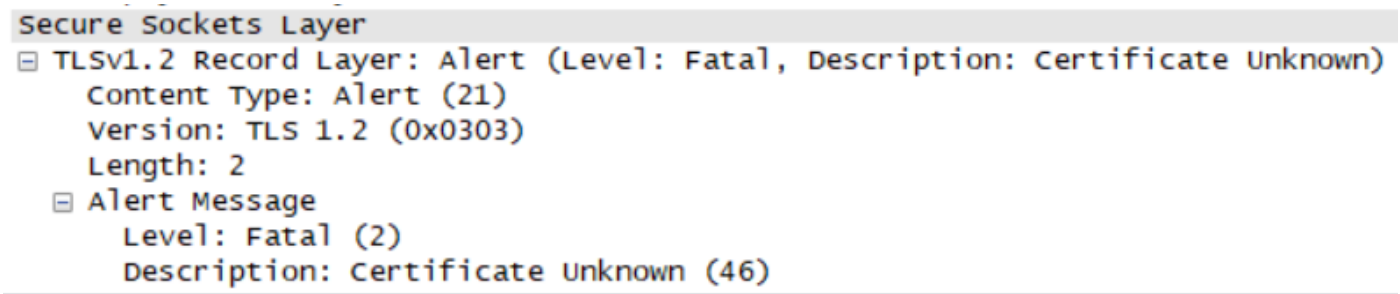
检查在Finesse服务器上运行的Java版本以确认该版本支持哪些算法是一个好主意。可通过以下命令从根访问检查版本：**java -version**

```
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 ~]#
```

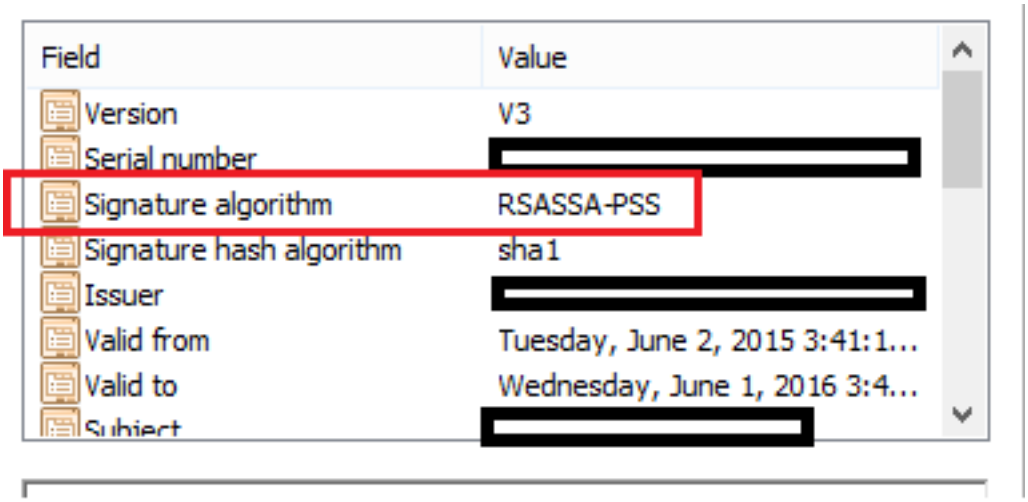
> **注：**有关Java SunMSCAPI提供程序的详细信息，请参阅 https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunMSCAPI

如果证书提供的签名不是上面列出的签名，Finesse将无法使用该证书创建与托管服务器的TLS连接。这包括使用受支持的签名类型签名的证书，这些证书由证书颁发机构颁发，这些颁发机构拥有自己的中间和根证书，这些证书颁发机构使用其他证书签名。

如果您查看数据包捕获，Finesse会使用“致命警报：证书未知"错误，如图所示。



此时必须检查托管服务器提供的证书并查找不受支持的签名算法。通常，**RSASSA-PSS**被视为有问题的签名算法：



如果链中的任何证书使用RSASSA-PSS签名，则连接失败。在这种情况下，数据包捕获显示根CA将RSASSA-PSS用于自己的证书：

```
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...
```

解决方案

要解决此问题，必须从CA提供程序颁发新证书，该提供程序仅使用整个证书链中列出的受支持的SunMSCAPI签名类型之一，如前所述。

> **注**：有关RSASSA-PSS签名算法的详细信息，请参阅https://pkisolutions.com/pkcs1v2-1rsassa-pss/

> **注意**：此问题在缺陷CSCve中跟踪79330

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。