排除精良错误"故障; SSLPeerUnverifiedException"在CA签名的服务器主机的小配件

Contents

Introduction

本文描述步骤排除Certificate Authority (CA) -的方案故障签字的证书链被加载到主机一个小配件，但是小配件不能装载外部Web服务器的精良，当您登陆对精良时，并且您看到错误“SSLPeerUnverifiedException”。

贡献用吉诺Schweinsberger， Cisco TAC工程师。

Prerequisites

Requirements

Cisco 建议您了解以下主题：

- SSL证书
- 精良管理
- Windows服务器管理
- 与Wireshark的信息包获取分析

Components Used

本文档中的信息基于以下软件版本：

- 统一的Contact Center Express (UCCX) 11.X
- 精良11.X

The information in this document was created from the devices in a specific lab environment.All of the devices used in this document started with a cleared (default) configuration.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。

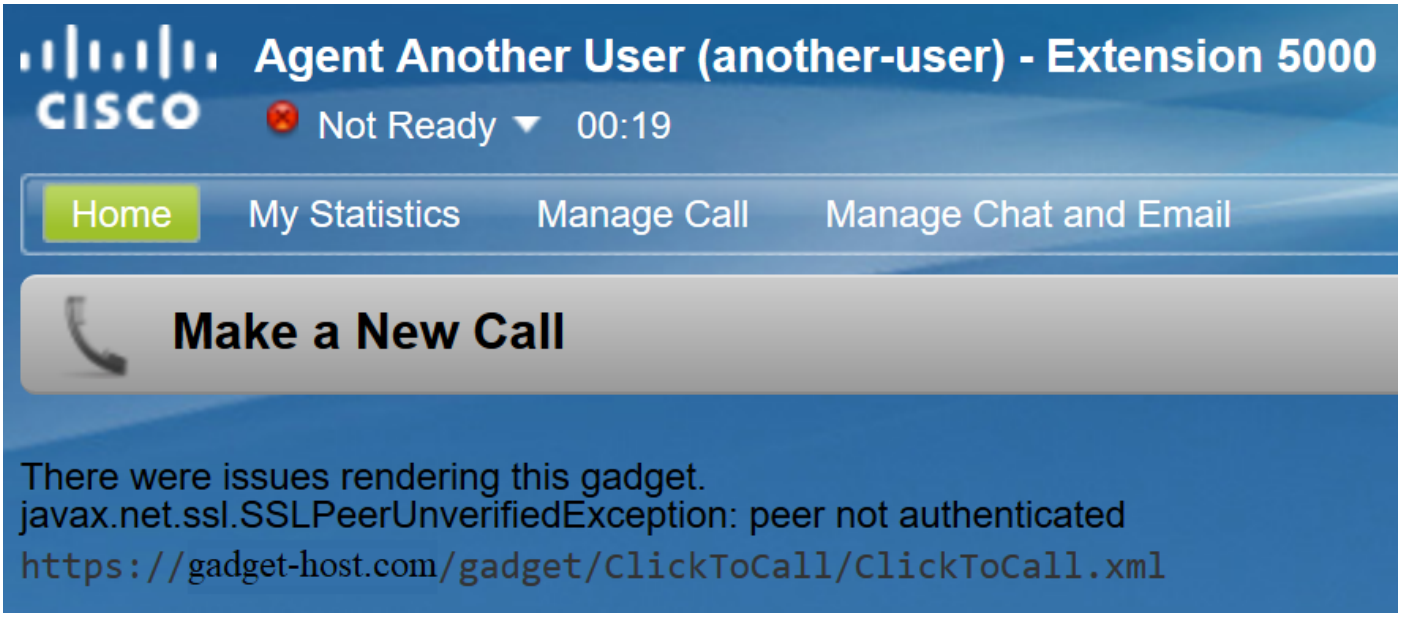
背景信息

这些是错误的条件能发生：

- 假设认证信任一系列被加载到精良
- 保证正确的服务器/服务被重新启动了
- 假设小配件被添加了到与HTTPS URL的精良布局，并且URL可及的

当代理程序登录对精良时，这是被观察的错误：

“有回报此小配件的问题。javax.net.ssl.SSLPeerUnverifiedException ：没验证的对等体”



问题

情形 1：主服务器协商不安全的TLS

当精良服务器做连接请求到主服务器时，精良Tomcat通告支持的加密密码列表。

一些密码不支持的归结于安全漏洞，

如果主服务器选择这些密码之一，连接被拒绝：

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA

这些密码知道使用弱的短暂Diffie-Hellman键，当它协商连接时，并且木材堵塞弱点使这些TLS连接的一个坏选择做出。

按照在信息包获取的TLS握手进程发现哪个密码协商。

1. 精良提交在**客户端Hello**步骤的支持的密码其列表：

```
▾ 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，因为那列在首选密码其列表前边， TLS_DHE_RSA_WITH_AES_256_CBC_SHA由主服务器选择在**服务器问候**步骤期间。

```
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. 精良发送一次致命戒备并且结束连接：

```
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，必须配置主服务器产生这些低优先级，或者必须从可用的密码列表完全地去除他们。这在一Windows服务器可以执行用Windows组策略编辑器(gpedit.msc)。

Note: 欲了解更详细的信息在木材堵塞的作用在精良的和使用gpedit，检查：

方案 2：认证有一种不支持的签署的算法

Windows服务器认证权限能使用更新的签名标准签署证书。它比SHA提供更加巨大的安全，这些标准的采用在Microsoft产品外面是低的，并且管理员可能遇到互操作性问题。

精良Tomcat依靠SunMSCAPI安全供应商从Java到Microsoft和密码功能的enable (event)技术支持使用的多种签名算法。所有当前JAVA版本(1.7， 1.8和1.9)技术支持仅这些签名算法：

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

它是一个好想法检查在精良服务器运行确认的JAVA版本该版本支持哪些算法。版本可以从根访问权限被检查用此命令：**Java -版本**

```
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 ~]#
```

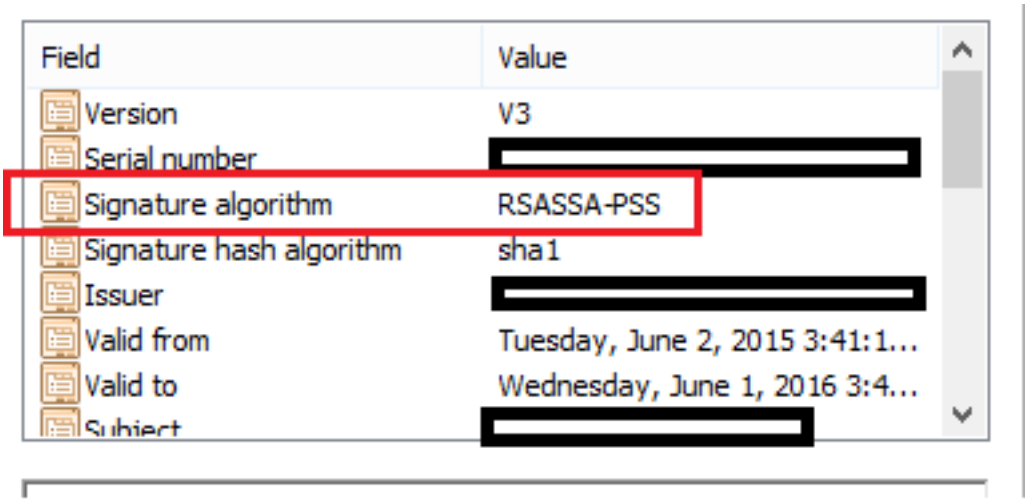
> **Note**: 欲了解更详细的信息在Java SunMSCAPI供应商请参见 https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunMSCAPI

如果认证带有签名除以上所列的那些之外，精良不能使用认证创建与主服务器的TLS连接。包括签字与一种支持的签名类型的证书，但是有他们自己的中间和根证明签字与其他的认证权限发出这。

如果查看信息包获取，精良断开与“致命戒备的连接：认证未知”错误，如镜像所显示。

```
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)
```

这时必要的is is检查主服务器提交的证书和寻找不支持的签名算法。它是普通发现**RSASSA-PSS**作为有问题的签名算法：



如果在一系列的任何认证签字与RSASSA-PSS，连接发生故障。 在这种情况下信息包获取表示，根CA使用RSASSA-PSS其自己的认证：

```
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

为了解决此问题，必须从只使用其中一种支持的SunMSCAPI签名类型列出在整个证书链中按照说明以前的CA供应商发出新证书。

Note: 欲了解更详细的信息在RSASSA-PSS签名算法，请参阅 https://pkisolutions.com/pkcs1v2-1rsassa-pss/

Note:此问题在缺陷CSCve79330被跟踪