CA 서명 서버에서 호스트되는 가젯에 대한 Finesse 오류 "SSLPeerUnverifiedException" 트러블슈팅

목차

소개

이 문서에서는 CA(인증 기관)가 서명한 인증서 체인이 가젯을 호스팅하는 외부 웹 서버에 대해 Finesse에 업로드되지만 Finesse에 로그인할 때 가젯이 로드되지 않고 "SSLPeerUnverifiedException" 오류가 표시되는 시나리오를 해결하는 단계를 설명합니다.

기고자: Gino Schweinsberger, Cisco TAC 엔지니어

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- SSL 인증서
- Finesse 관리
- Windows Server 관리
- Wireshark를 이용한 패킷 캡처 분석

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 버전을 기반으로 합니다.

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

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든

명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

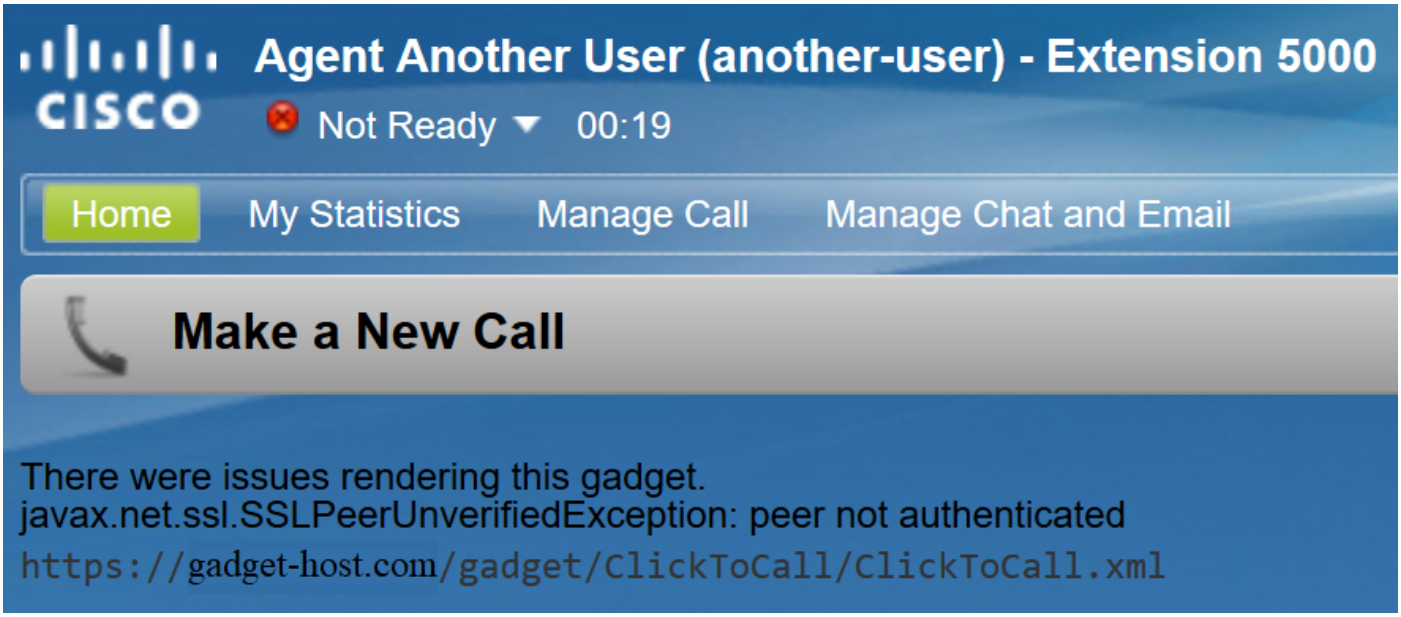
배경 정보

다음은 오류가 발생하는 조건입니다.

- 인증서 신뢰 체인이 Finesse에 업로드되었다고 가정합니다.
- 올바른 서버/서비스가 다시 시작되었는지 확인합니다.
- 가젯이 HTTPS URL을 사용하여 Finesse 레이아웃에 추가되었고 URL에 연결할 수 있다고 가정합니다

이 오류는 에이전트가 Finesse에 로그인할 때 발생합니다.

"이 가젯을 렌더링하는 동안 문제가 발생했습니다. javax.net.ssl.SSLPeerUnverifiedException: 피어가 인증되지 않음"



문제

시나리오 1: 호스팅 서버가 안전하지 않은 TLS를 협상합니다

Finesse Server에서 호스팅 서버에 대한 연결 요청을 하면 Finesse Tomcat은 지원하는 암호화 암호 목록을 광고합니다.

일부 암호는 보안 취약성으로 인해 지원되지 않습니다.

호스팅 서버가 이러한 암호 중 하나를 선택하면 연결이 거부됩니다.

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA

이러한 암호는 연결을 협상할 때 약한 단명 Diffie-Hellman 키를 사용하는 것으로 알려져 있으며, Logjam 취약성으로 인해 이러한 암호는 TLS 연결에 적합하지 않은 선택이 됩니다.

패킷 캡처의 TLS 핸드셰이크 프로세스에 따라 협상된 암호를 확인합니다.

1. Finesse는 **Client Hello** 단계에서 지원되는 암호 **목록을** 표시합니다.

```
▾ 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. **서버 Hello** 단계 동안 호스팅 서버**가 기본** 암호 목록에서 더 높기 때문에 이 연결에서 TLS**_DHE_RSA_WITH_AES_256_CBC_SHA**를 선택했습니다.

```
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. Finesse에서 Fatal(치명적) 알림을 전송하고 연결을 종료합니다.

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

솔루션

이러한 암호를 사용하지 않으려면 이러한 암호에 낮은 우선 순위를 부여하도록 호스팅 서버를 구성해야 합니다. 그렇지 않으면 사용 가능한 암호 목록에서 완전히 제거해야 합니다. 이 작업은 Windows 그룹 정책 편집기(gpedit.msc)가 있는 Windows 서버에서 수행할 수 있습니다.

참고: Finesse의 Logjam과 gpedit 사용에 대한 자세한 내용은 다음을 확인하십시오.

시나리오 2: 인증서에 지원되지 않는 서명 알고리즘이 있습니다.

Windows Server 인증 기관에서는 최신 서명 표준을 사용하여 인증서를 서명할 수 있습니다. SHA보다 뛰어난 보안을 제공하는 경우에도 Microsoft 제품 이외의 표준은 채택률이 낮으므로 관리

자가 상호운용성 문제에 부딪힐 가능성이 높습니다.

Finesse Tomcat은 Java의 SunMSCAPI 보안 제공자를 사용하여 Microsoft에서 사용하는 다양한 서명 알고리즘 및 암호화 기능을 지원합니다. 모든 현재 버전의 Java(1.7, 1.8 및 1.9)는 다음 서명 알고리즘만 지원합니다.

- MD5withRSA
- MD2withRSA
- 없음RSA
- SHA1withRSA
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA

Finesse 서버에서 실행되는 Java 버전을 확인하여 해당 버전에서 지원되는 알고리즘을 확인하는 것이 좋습니다. 다음 명령을 사용하여 루트 액세스에서 버전을 확인할 수 있습니다. **java -버전**

```
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 ~]#
```

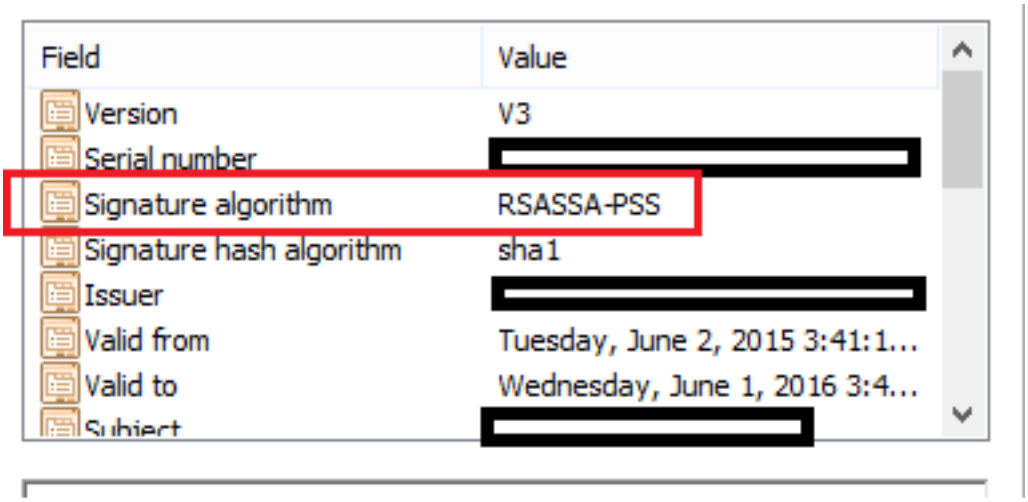
> **참고**: Java SunMSCAPI 공급자에 대한 자세한 내용은 https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunMSCAPI을 참조하십시오.

인증서에 위에 나열된 서명 이외의 서명이 있는 경우 Finesse는 인증서를 사용하여 호스팅 서버에 대한 TLS 연결을 생성할 수 없습니다. 여기에는 지원되는 서명 유형으로 서명되었지만 자체 중간 인증서와 루트 인증서가 다른 서명된 인증 기관에서 발급한 인증서가 포함됩니다.

패킷 캡처를 살펴보면 Finesse는 "치명적 경고: 이미지에 표시된 것처럼 Certificate Unknown(인증서 알 수 없음)" 오류입니다.

```
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)
```

이 시점에서 호스팅 서버에서 제공하는 인증서를 확인하고 지원되지 않는 서명 알고리즘을 찾아야 합니다. 일반적으로 RSASSA-PSS는 **다음과 같은 문제**가 있는 서명 알고리즘으로 간주됩니다.



체인의 인증서가 RSASSA-PSS로 서명된 경우 연결이 실패합니다. 이 경우 패킷 캡처는 루트 CA가 자체 인증서에 RSASSA-PSS를 사용함을 보여줍니다.

```
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...
```

솔루션

이 문제를 해결하려면 앞에서 설명한 대로 전체 인증서 체인에 나열된 지원되는 SunMSCAPI 서명 유형 중 하나만 사용하는 CA 공급자로부터 새 인증서를 발급해야 합니다.

> **참고**: RSASSA-PSS 서명 알고리즘에 대한 자세한 내용은 https://pkisolutions.com/pkcs1v2-1rsassa-pss/을 참조하십시오.

> **참고:** 이 문제는 결함 CSCve에서 추적됩니다79330

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.