La instalación elegante 5.1.0 del satélite del administrador de software (SS) falla en el corazón basado KVM

Contenido

Introducción

Este documento describe la solución al problema que ocurre cuando la instalación elegante 5.1.0 del satélite del administrador de software (SS) falla en el corazón basado del teclado/del vídeo/del ratón (KVM) que incluye la Plataforma de servicio de la nube de Cisco.

Problema

La instalación completa vía la consola y la interfaz de usuario (UI) es accesible.

A la hora del proceso de configuración del registro CSSM, se nota que el registro falla mientras que se realiza el registro de la red, así como la inscripción manual. Se valida la versión del tomcat, el corazón y la máquina virtual Java (JVM) en el sistema basado KVM. tome la nota que el JVM funciona con 1.8.0_102-b14 y el corazón 3.10.0-514.el7. Compare con la configuración basada ESXI, donde el corazón ejecuta 3.10.0-862.14.4.el7 y JVM 1.8.0_191-b12.

```
[root@satellite bin]# ./version.sh
Using CATALINA_BASE: /opt/tc
Using CATALINA_HOME: /opt/tc
Using CATALINA_TMPDIR: /opt/tc/temp
Using JRE_HOME: /
Using CLASSPATH: /opt/tc/bin/bootstrap.jar:/opt/tc/bin/tomcat-juli.jar
Using CATALINA_PID: /opt/tomcat/temp/tomcat.pid
Server version: Apache Tomcat/9.0.1
Server built: Sep 27 2017 17:31:52 UTC
Server number: 9.0.1.0
OS Name: Linux
OS Version: 3.10.0-514.el7.x86_64
Architecture: amd64
JVM Version: 1.8.0_102-b14
JVM Vendor: Oracle Corporation
```

Componentes

Plataforma: El KVM basó el corazón

Software: Imagen ISO de la obra clásica 5.1

Solución

Paso 1. Navegue a **cd/opt/tomcat/logs/**.

Paso 2. Abra los **registros catalina.out** y encuentre la excepción que ocurre a la hora del proceso de inscripción con CSSM.

El proveedor IAIK-JCE IAIK es una extensión de la criptografía de las Javas que tiene un conjunto de los API y puede implementar las funciones criptográficas. Se utiliza para soportar las funciones de la seguridad complementaria al JDK. El módulo LC no puede generar el par clave para el archivo de la petición CSR debido a la indisponibilidad del archivo JAR IAIK.

```
2019-05-15 20:35:01,604 [http-nio-8080-exec-9] INFO controller.LindosController - Invoked GET
/lcsSetupStatus
2019-05-15 20:35:01,606 [http-nio-8080-exec-9] INFO controller.LindosController - LCS Setup
Status = 0
2019-05-15 23:53:12,226 [http-nio-8080-exec-10] INFO controller.LindosController - Invoked GET
/lcsSetupStatus
2019-05-15 23:53:12,230 [http-nio-8080-exec-10] INFO controller.LindosController - LCS Setup
Status = 0
2019-05-15 23:53:12,241 [http-nio-8080-exec-1] INFO controller.LindosController - Invoked
/lcsSetup
2019-05-15 23:53:12,243 [http-nio-8080-exec-1] DEBUG controller.LindosController - Setup Status
= 0 (0=empty, 1=key/CSR generated, 2=Signer certs installed)
2019-05-15 23:53:12,243 [http-nio-8080-exec-1] DEBUG controller.LindosController - First time
setup invoked (ID element not present in JSON). CN=5fc62a80-59a0-0137-54ab-023a01ab3207
2019-05-15 23:53:12,243 [http-nio-8080-exec-1] DEBUG domain.LcsSignerSetup - In LcsSignerSetup
2019-05-15 23:53:12,244 [http-nio-8080-exec-1] DEBUG domain.LcsSignerSetup - Generating Key
Pair...
2019-05-15 23:53:12,244 [http-nio-8080-exec-1] ERROR error.RestResponseEntityExceptionHandler -
java.security.NoSuchProviderException: no such provider: IAIK
com.cisco.ias.lindos.data.domain.LcsSetupException: java.security.NoSuchProviderException: no
such provider: IAIK
at com.cisco.ias.lindos.data.domain.LcsSignerSetup.<init>(LcsSignerSetup.java:50)
at com.cisco.ias.lindos.web.controller.LindosController.setupLcs(LindosController.java:126)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at
org.springframework.web.method.support.InvocableHandlerMethod.invoke(InvocableHandlerMethod.java
:215)
at
org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerM
ethod.java:132)
at
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHan
dle(ServletInvocableHandlerMethod.java:104)
at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandleM
ethod(RequestMappingHandlerAdapter.java:749)
at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInterna
l(RequestMappingHandlerAdapter.java:690)
at
org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMe
thodAdapter.java:83)
at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:945)
at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:876)
```

```
at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:961)
at org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:863)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:660)
at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:837)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231
)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193
)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:140)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81)
at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:651)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:342)
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:500)
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:754)
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1376)
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.lang.Thread.run(Thread.java:745)
2019-05-15 23:53:12,254 [http-nio-8080-exec-2] INFO controller.LindosController - Invoked GET
/lcsSetupStatus
2019-05-15 23:53:12,256 [http-nio-8080-exec-2] INFO controller.LindosController - LCS Setup
Status = 0
```

Paso 3. Coloque el proveedor de seguridad requerido en el classpath; **cp /opt/tomcat/webapps/Lindos/WEB-INF/lib/iaik_jce-5.1.jar /usr/lib/jvm/java/jre/lib/ext/**.

Paso 4. Asegúrese de que el tarro sea legible por otros módulos; **chmod o+r /usr/lib/jvm/java/jre/lib/ext/iaik_jce-5.1.jar**.

Paso 5. Salve el **trayecto del archivo java.security a una** variable TEMP; **java_security=/usr/lib/jvm/java/jre/lib/security/java.security.**

Paso 6. Prioridad existente de los proveedores del incremento por; **Perl - pi - e 's/^security.provider.)/"(\ d+ security.provider." . ($1+1)/e $java_security.**

Paso 7. Inserte IAIK como el primer proveedor en la lista (observe la barra que escapa el newline); **sed - i '/security.provider.2/i **

security.provider.1=iaik.security.provider.IAIK $java_security.

Paso 8. Recomience el tomcat para los cambios para tomar el efecto con el comando; **tomcat del reinicio del systemctl**.

Paso 9. Registre el satélite con CSSM y cuando el registro en el satélite se completa, el UI no podrá recomenzar.

Paso 10. Plegable ambos Certificados x509 usados para las conexiones de Transport Layer

Security (TLS) en los puertos 443 y 8443 para resolver el formato aumentado aislamiento del correo electrónico (PEM); **doblez - w 64 /drbd/certs/rails_ssl.crt > && milivoltio /drbd/certs/rails_ssl_folded.crt /drbd/certs/rails_ssl.crt de /drbd/certs/rails_ssl_folded.crt**

doblez - w 64 /drbd/certs/pi_ssl.crt > && milivoltio /drbd/certs/pi_ssl_folded.crt /drbd/certs/pi_ssl.crt de /drbd/certs/pi_ssl_folded.crt.

> Nota: No ejecute estos comandos plegable así como diversa línea de la instalación como corrompen el CERT 64-encoded PEM.

Nginx del paso 11.Start; **nginx del comienzo del systemctl**.

> Nota: Si el UI no puede subir después de una sincronización, después es debido a estos certs que son actualizados/substituidos. Por lo tanto, los pasos 8-10 tendrán que ser relanzados.

Después de que usted siga los siguientes pasos, acceda el UI y usted puede ver que sincronización del poste con CSSM y el registro final es éxito.

Usted puede ver el inventario y la licencia seccionar la licencia asociada del VA. Usted puede registrar los casos elegantes del producto al satélite.