Smart Software Manager-satelliet (Ssms) 5.1.0 Installatie-falen in KVM-gebaseerde Kernel

Inhoud

Inleiding

In dit document wordt de oplossing beschreven voor het probleem dat zich voordoet wanneer de installatie van Smart Software Manager satelliet (Ssms) 5.1.0 op het toetsenbord/video/muis (KVM)-gebaseerde kern, inclusief het Cisco Cloud Service Platform, faalt.

Probleem

De installatie wordt voltooid via console en de gebruikersinterface (UI) is toegankelijk.

Tijdens de installatie van de CSSM-registratie is opgemerkt dat de registratie niet verloopt tijdens de registratie van het netwerk, maar ook nadat de handmatige registratie is uitgevoerd. De tomcat-versie is gevalideerd, kernel en Java virtuele machine (JVM) in een op KVM gebaseerd systeem. Let erop dat JVM 1.8.0_102-b14 en kernel 3.10.0-514.el7 draait. Vergelijk met ESXI-gebaseerde setup, waarbij kernel 3.10.0-862.14.4.el7 en JVM 1.8.0_191-b12 draait.

```
[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
```

Componenten

Platform: KVM-gebaseerde kernel

Software: Classic 5.1 ISO-afbeelding

Oplossing

Stap 1. Navigeer naar **cd/opt/tomcat/logs/**.

Stap 2. Open **catalina.out-**stammen en vind de uitzondering die op het moment van het registratieproces plaatsvindt met CSSM.

IAIK provider IAIK-JCE is een Java Cryptografie Extension die een reeks API's heeft en cryptografische functies kan implementeren. Het wordt gebruikt ter ondersteuning van extra beveiligingsfuncties aan de JDK. De LCS-module genereert geen sleutelpaar voor het CSR-bestand vanwege de onbeschikbaarheid van het IAIK-bestand.

```
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
```

Stap 3. Plaats de gewenste beveiligingsaanbieder in de gevarenzone; **cp /opt/tomcat/webapps/Lindos/WEB-INF/lib/iaik_jce-5.1.jar/usr/lib/jvm/java/jre/lib/ext/**.

Stap 4. Zorg ervoor dat de pot door andere modules kan worden gelezen; **chmod o+r/usr/lib/jvm/java/jre/lib/ext/iaik_jce-5.1.jar**.

Stap 5. Bewaar **java.security** bestandspad naar een temp-variabele; **java_security=/usr/lib/jvm/java/jre/lib/security/java.security**.

Stap 6. Verhoging van bestaande aanbieders prioriteit door; **perl -pi -e's/^security.provider. (\d+)/"security.provider." . ($1+1)/e' $java_security**.

Stap 7. Plaats IAIK als eerste provider in de lijst (let op de backslash die aan newline ontsnapt); **sed-i '/security.provider.2/i **

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

Stap 8. Herstart de taak om wijzigingen door te voeren met de opdracht; **start de opdracht opnieuw**.

Stap 9. Registreer de satelliet met CSSM en wanneer de registratie in satelliet wordt voltooid, zal de UI niet opnieuw beginnen.

Stap 10. Vouw beide x509-certificaten die gebruikt worden voor TLS-verbindingen (Transport

Layer Security) op poorten 443 en 8443 om te voldoen aan het PEM-formaat (Privacy Enhanced Email Security); **vouw 64 /drbd/certs/rails_ssl.crt > /drbd/certs/rails_ssl_folded.crt & mv /drbd/certs/rails_ssl_folded.crt /drbd/certs/rails_ssl.crt**

voudig - w 64 /drbd/certs/pi_ssl.crt > /drbd/certs/pi_ssl_folded.crt & mv /drbd/certs/pi_ssl_folded.crt /drbd/certs/pi_ssl.crt.

> Opmerking: Voer deze opdrachten niet uit, vouwt uit en verplaats verschillende regels omdat ze de 64-gecodeerde PEM-cert beschadigen.

Stap 11.Start ng; **systeembegin nginx**.

> Opmerking: Als de UI na een synchronisatie niet naar voren komt, is het te wijten aan het feit dat deze certs worden bijgewerkt/vervangen. Daarom moeten stap 8-10 worden herhaald.

Nadat u deze stappen volgt, hebt u toegang tot de UI en u kunt postsynchronisatie met CSSM zien en de definitieve registratie is succesvol.

U kunt de inventaris- en licentiesectie van de licentie zien die aan de VA-indeling is gekoppeld. U kunt slimme productvoorbeelden registreren via satelliet.