Smart Software Manager Satellite (SSMS) 5.1.0 Installation Fails in KVM Based Kernel

Contents

Introduction

This document describes the solution to the problem that occurs when Smart Software Manager Satellite (SSMS) 5.1.0 installation fails in Keyboard/Video/Mouse (KVM) based kernel which includes the Cisco Cloud Service Platform.

Problem

The installation completes via console and the User Interface (UI) is accessible.

At the time of the CSSM Registration setup process, it is noticed that the registration fails while Network registration, as well as Manual registration, is performed. The tomcat version is validated, kernel and Java Virtual Machine (JVM) in KVM based system. take note that JVM runs 1.8.0_102-b14 and kernel 3.10.0-514.el7. Compare with ESXI based setup, where kernel runs 3.10.0-862.14.4.el7 and 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
```

Components

Platform: KVM based kernel

Software: Classic 5.1 ISO image

Solution

Step 1. Navigate to **cd/opt/tomcat/logs/**.

Step 2. Open **catalina.out** logs and find the exception that takes place at the time of the registration process with CSSM.

IAIK provider IAIK-JCE is a Java Cryptography Extension that has a set of APIs and can implement cryptographic functionality. It is used in order to support additional security functionalities to the JDK. The LCS module fails to generate key pair for CSR request file due to unavailability of IAIK jar file.

```
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
```

Step 3. Place the required security provider in classpath; **cp /opt/tomcat/webapps/Lindos/WEB-INF/lib/iaik_jce-5.1.jar /usr/lib/jvm/java/jre/lib/ext/**.

Step 4. Ensure that the jar is readable by other modules; **chmod o+r /usr/lib/jvm/java/jre/lib/ext/iaik_jce-5.1.jar**.

Step 5. Store **java.security** file path to a temp variable; **java_security=/usr/lib/jvm/java/jre/lib/security/java.security**.

Step 6. Increment existing providers priority by; **perl -pi -e 's/^security.provider.(\d+)/"security.provider." . ($1+1)/e' $java_security**.

Step 7. Insert IAIK as the first provider in the list (note the backslash that escapes newline); **sed -i '/security.provider.2/i **

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

Step 8. Restart tomcat for changes in order to take effect with the command; **systemctl restart tomcat**.

Step 9. Register the satellite with CSSM and when the registration in satellite is completed, the UI will fail to restart.

Step 10. Fold both x509 certificates used for Transport Layer Security (TLS) connections on ports

443 and 8443 in order to meet Privacy Enhanced Email (PEM) format; **fold -w 64 /drbd/certs/rails_ssl.crt > /drbd/certs/rails_ssl_folded.crt && mv /drbd/certs/rails_ssl_folded.crt /drbd/certs/rails_ssl.crt**

fold -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.

> **Note**: Do not execute these commands fold as well as move-in different line as they corrupt the 64-encoded PEM cert.

Step 11.Start nginx; **systemctl start nginx**.

> **Note**: If the UI fails to come up after a synchronization, then it is due to these certs being updated/replaced. Therefore, steps 8-10 will have to be repeated.

After you follow these steps, access the UI and you can see post synchronization with CSSM and final registration is sucess.

You can see inventory and license section the license mapped from VA. You can Register smart product instances to Satellite.