Installazione di Smart Software Manager Satellite (SSMS) 5.1.0 non riuscita nel kernel basato su KVM

Sommario

Introduzione

Questo documento descrive la soluzione al problema che si verifica quando l'installazione di Smart Software Manager Satellite (SSMS) 5.1.0 non riesce nel kernel basato su tastiera/video/mouse (KVM) che include Cisco Cloud Service Platform.

Problema

L'installazione viene completata tramite la console e l'interfaccia utente è accessibile.

Durante il processo di configurazione della registrazione CSM, si noti che la registrazione non riesce durante l'esecuzione della registrazione di rete e della registrazione manuale. La versione tomcat viene convalidata, kernel e Java Virtual Machine (JVM) in un sistema basato su KVM. notare che JVM esegue 1.8.0_102-b14 e kernel 3.10.0-514.el7. Confrontare con l'installazione basata su ESXI, in cui il kernel esegue 3.10.0-862.14.4.el7 e 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
```

Componenti

Piattaforma: kernel basato su KVM

Software: Immagine ISO classica 5.1

Soluzione

Passaggio 1. Passare a **cd/opt/tomcat/logs/**.

Passaggio 2. Aprire i log di **catalina.out** e individuare l'eccezione che si verifica al momento del processo di registrazione con CSM.

Il provider IAIK IAIK-JCE è un'estensione di crittografia Java che dispone di un set di API e può implementare funzionalità di crittografia. È utilizzato per supportare funzionalità di sicurezza aggiuntive per JDK. Il modulo LCS non è in grado di generare la coppia di chiavi per il file di richiesta CSR a causa della mancata disponibilità del file 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
```

Passaggio 3. Inserire il provider della sicurezza richiesto in classpath; **cp /opt/tomcat/webapps/Lindos/WEB-INF/lib/iaik_jce-5.1.jar /usr/lib/jvm/java/jre/lib/ext/**.

Passaggio 4. Verificare che il file jar sia leggibile da altri moduli: **chmod o+r /usr/lib/jvm/java/jre/lib/ext/iaik_jce-5.1.jar**.

Passaggio 5. Archiviare il percorso del file **java.security** in una variabile temporanea; **java_security=/usr/lib/jvm/java/jre/lib/security/java.security**.

Passaggio 6. Aumentare la priorità dei provider esistenti di; **perl -pi -e 's/^security.provider.(\d+)/"security.provider." . ($1+1)/e' $java_security**.

Passaggio 7. Inserire IAIK come primo provider dell'elenco (si noti la barra rovesciata che fuoriesce dalla nuova riga); **sed -i '/security.provider.2/i **

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

Passaggio 8. Per rendere effettive le modifiche, riavviare tomcat con il comando; **systemctl restart tomcat**.

Passaggio 9. Registrare il satellite con CSSM e al termine della registrazione nel satellite non sarà possibile riavviare l'interfaccia utente.

Passaggio 10. Per soddisfare i requisiti del formato PEM (Privacy Enhanced Email), compilare

entrambi i certificati x509 utilizzati per le connessioni Transport Layer Security (TLS) sulle porte 443 e 8443. **Piegare -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.

> **Nota:** Non eseguire questi comandi piegare e spostare in una riga diversa in quanto danneggiano il certificato PEM con codifica 64.

Passaggio 11. Inginx; **systemctl start inginx**.

> **Nota:** Se l'interfaccia utente non viene visualizzata dopo una sincronizzazione, ciò è dovuto all'aggiornamento o alla sostituzione dei certificati. È quindi necessario ripetere i passaggi da 8 a 10.

Dopo aver eseguito questi passaggi, accedere all'interfaccia utente e verificare che la post-sincronizzazione con CSM abbia esito positivo.

Èpossibile visualizzare l'inventario e la sezione delle licenze della licenza mappata da VA. È possibile registrare istanze di smart product in Satellite.