

A instalação 5.1.0 esperta do satélite do gerenciador de software (SS) falha no núcleo baseado KVM

Índice

[Introdução](#)

[Problema](#)

[Componentes](#)

[Solução](#)

Introdução

Este documento descreve a solução ao problema que ocorre quando a instalação 5.1.0 esperta do satélite do gerenciador de software (SS) falha no núcleo baseado do teclado/vídeo/rato (KVM) que inclui a plataforma do serviço da nuvem de Cisco.

Problema

A instalação termina através do console e a interface do utilizador (UI) é acessível.

Na altura do processo de instalação do registo CSSM, observa-se que o registo falha quando o registo da rede, assim como o registo manual, forem executados. A versão de TomCat é validada, o núcleo e a máquina virtual de java (JVM) no sistema baseado KVM. tome a nota que o JVM executa 1.8.0_102-b14 e núcleo 3.10.0-514.el7. Compare com a instalação baseada ESXI, onde o núcleo executa 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
```

Componentes

Plataforma: O KVM baseou o núcleo

Software: Imagem ISO do clássico 5.1

Solução

Etapa 1. Navegue a `cd/opt/tomcat/logs/`.

Etapa 2. Abra `logs catalina.out` e encontre a exceção que ocorre na altura do processo de registro com CSSM.

O fornecedor IAIK-JCE IAIK é uma extensão da criptografia das Javas que tenha um grupo de API e possa executar a funcionalidade criptograficamente. É usado a fim apoiar funcionalidades da segurança adicional ao JDK. O módulo LC não gerencie o par de chaves para o arquivo do pedido CSR devido à indisponibilidade do arquivo 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(InvocableHandlerMethod.java:132)
at
org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:104)
at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandleMethod(RequestMappingHandlerAdapter.java:749)
at
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:690)
at
org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.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

```

Etapa 3. Coloque o provedor de segurança exigido no caminho de classe; **cp /opt/tomcat/webapps/Lindos/WEB-INF/lib/iaik_jce-5.1.jar /usr/lib/jvm/java/jre/lib/ext/.**

Etapa 4. Assegure-se de que o frasco esteja legível por outros módulos; **chmod o+r /usr/lib/jvm/java/jre/lib/ext/iaik_jce-5.1.jar.**

Etapa 5. Armazene o caminho de arquivo **java.security** a uma variável do temp; **java_security=/usr/lib/jvm/java/jre/lib/security/java.security.**

Etapa 6. Prioridade existente dos fornecedores do incremento por; **Perl - pi - e 's/^security.provider.) (\ d+/"security.provider." . (\$1+1)/e \$java_security.**

Etapa 7. Introduza IAİK como o primeiro fornecedor na lista (note o corte traseiro que escapa o newline); **sed - i '/security.provider.2/i **

security.provider.1=iaik.security.provider.IAİK \$java_security.

Etapa 8. Reinicie TomCat para mudanças a fim tomar o efeito com o comando; **reinício TomCat do systemctl.**

Etapa 9. Registrar o satélite com CSSM e quando o registro no satélite é terminado, o UI não reiniciará.

Etapa 10. Dobre ambos os Certificados x509 usados para conexões do Transport Layer Security

(TLS) nas portas 443 e 8443 a fim encontrar o formato aumentado privacidade do email (PEM); dobra - w 64 /drbd/certs/rails_ssl.crt > && milivolt /drbd/certs/rails_ssl_folded.crt /drbd/certs/rails_ssl.crt de /drbd/certs/rails_ssl_folded.crt

dobra - w 64 /drbd/certs/pi_ssl.crt > && milivolt /drbd/certs/pi_ssl_folded.crt /drbd/certs/pi_ssl.crt de /drbd/certs/pi_ssl_folded.crt.

Nota: Não execute estes comandos dobram-se assim como linha diferente da instalação como corrompem o CERT 64-encoded PEM.

Nginx da etapa 11.Start; **nginx do começo do systemctl.**

Nota: Se o UI não vem acima após uma sincronização, a seguir é devido a estes certs que estão sendo atualizados/substituídos. Conseqüentemente, as etapas 8-10 terão que ser repetidas.

Depois que você segue estas etapas, alcance o UI e você pode ver que sincronização do cargo com CSSM e registro final é sucesso.

Você pode ver o inventário e a licença seccionar a licença traçada do VA. Você pode registrar exemplos espertos do produto ao satélite.