Java Update Enforces CRL Checks by Default Which Prevents NSP and Guest Flows

TAC **Document ID: 116444**

Contributed by Sam Hertica, Jesse Dubois, and John Newman, Cisco TAC Engineers.
Aug 07, 2013

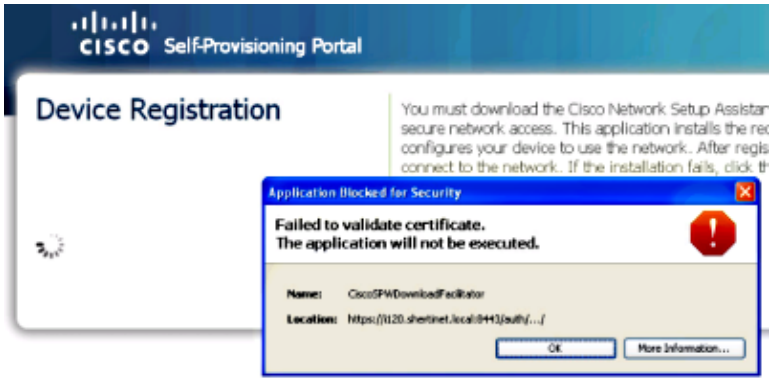
Contents

Introduction

This document describes a problem encountered where the latest Java update breaks supplicant provisioning and some guest flows that use Access Control Lists (ACLs) and redirection.

Background Information

The error is in the CiscoSPWDownloadFacilitator and reads "Failed to validate certificate. The application will not be executed."



If you click ***More Information***, you receive output that complains about the Certificate Revocation List (CRL).

```
java.security.cert.CertificateException: java.security.cert.
CertPathValidatorException: java.io.IOException: DerInputStream.getLength():
lengthTag=127, too big.
        at com.sun.deploy.security.RevocationChecker.checkOCSP(Unknown Source)
        at com.sun.deploy.security.RevocationChecker.check(Unknown Source)
        at com.sun.deploy.security.TrustDecider.checkRevocationStatus(Unknown Source)
        at com.sun.deploy.security.TrustDecider.getValidationState(Unknown Source)
        at com.sun.deploy.security.TrustDecider.validateChain(Unknown Source)
        at com.sun.deploy.security.TrustDecider.isAllPermissionGranted(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.isTrustedByTrustDecider
```

```
        (Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.getTrustedCodeSources(Unknown Source)
        at com.sun.deploy.security.CPCallbackHandler$ParentCallback.strategy
        (Unknown Source)
        at com.sun.deploy.security.CPCallbackHandler$ParentCallback.openClassPathElement
        (Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath$JarLoader.getJarFile
        (Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath$JarLoader.access$1000
        (Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath$JarLoader$1.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at com.sun.deploy.security.DeployURLClassPath$JarLoader.ensureOpen
        (Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath$JarLoader.<init>(Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath$3.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at com.sun.deploy.security.DeployURLClassPath.getLoader(Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath.getLoader(Unknown Source)
        at com.sun.deploy.security.DeployURLClassPath.getResource(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader$2.run(Unknown Source)
        at java.security.AccessController.doPrivileged(Native Method)
        at sun.plugin2.applet.Plugin2ClassLoader.findClassHelper(Unknown Source)
        at sun.plugin2.applet.Applet2ClassLoader.findClass(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadClass0(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadClass(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadClass0(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadClass(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadClass(Unknown Source)
        at java.lang.ClassLoader.loadClass(Unknown Source)
        at sun.plugin2.applet.Plugin2ClassLoader.loadCode(Unknown Source)
        at sun.plugin2.applet.Plugin2Manager.initAppletAdapter(Unknown Source)
        at sun.plugin2.applet.Plugin2Manager$AppletExecutionRunnable.run(Unknown Source)
        at java.lang.Thread.run(Unknown Source)
        Suppressed: com.sun.deploy.security.RevocationChecker$StatusUnknownException
                at com.sun.deploy.security.RevocationChecker.checkCRLs(Unknown Source)
                ... 34 more
Caused by: java.security.cert.CertPathValidatorException:
java.io.IOException: DerInputStream.getLength(): lengthTag=127, too big.
        at sun.security.provider.certpath.OCSP.check(Unknown Source)
        at sun.security.provider.certpath.OCSP.check(Unknown Source)
        at sun.security.provider.certpath.OCSP.check(Unknown Source)
        ... 35 more
Caused by: java.io.IOException: DerInputStream.getLength(): lengthTag=127, too big.
        at sun.security.util.DerInputStream.getLength(Unknown Source)
        at sun.security.util.DerValue.init(Unknown Source)
        at sun.security.util.DerValue.<init>(Unknown Source)
        at sun.security.provider.certpath.OCSPResponse.<init>(Unknown Source)
        ... 38 more
```

Problem

In the latest version of Java (Version 7, Update 25 − released August 5, 2013), Oracle introduced a new default setting that forces the client to validate the certificate associated with any applet against any CRL or Online Certificate Status Protocol (OCSP).

The signing certificate Cisco associates with these applets has a listed CRL and OCSP with Thawte. Because of this new change, when the Java client attempts to reach out to Thawte, it is blocked by either a port ACL and/or a redirect ACL.

The problem is tracked under Cisco bug ID CSCui46739.

Solution

Option 1 − Switch or Wireless Controller Side Fix

1. Rewrite any redirect or port−based ACLs in order to allow traffic to Thawte and Verisign. Unforunately, one limitation with this option is that ACLs cannot be created from domain names.
2. Resolve the CRL list manually, and put it in the redirect ACL.

Note: Firewall rules might need to be updated if the client needs to communicate through a firewall.

```
[user@user−linux logs]$ nslookup
> crl.thawte.com
Server:         64.102.6.247
Address:        64.102.6.247#53

Non−authoritative answer:
crl.thawte.com  canonical name = crl.ws.symantec.com.edgekey.net.
crl.ws.symantec.com.edgekey.net canonical name = e6845.ce.akamaiedge.net.
Name:   e6845.ce.akamaiedge.net
Address: 23.5.245.163

> ocsp.thawte.com
Server:         64.102.6.247
Address:        64.102.6.247#53

Non−authoritative answer:
ocsp.thawte.com canonical name = ocsp.verisign.net.
Name:   ocsp.verisign.net
Address: 199.7.48.72
```

If these DNS names change and clients resolve something else, rewrite the redirect URL with the updated addresses.

Example redirect ACL:

```
     5 remark ISE IP address
    10 deny ip any host X.X.X.X (467 matches)
    15 remark crl.thawte.com
    20 deny ip any host 23.5.245.163 (22 matches)
    25 remark ocsp.thawte.com
    30 deny ip any host 199.7.52.72
    40 deny udp any any eq domain (10 matches)
    50 permit tcp any any eq www (92 matches)
    60 permit tcp any any eq 443 (58 matches)
```

Testing has shown the OSCP and CRL URLs resolve to these IP addresses:

OCSP
199.7.48.72
199.7.51.72
199.7.52.72
199.7.55.72
199.7.54.72
199.7.57.72
199.7.59.72

CRL

23.4.53.163
23.5.245.163
23.13.165.163
23.60.133.163
23.61.69.163
23.61.181.163

This might not be a complete list and might change based on geography, so testing is required to discover what IP address(es) the hosts resolve to in each instance.

Option 2 − Client Side Fix

Inside the ***Advanced*** section of the Java Control Panel, set ***Perform certificate revocation checks on*** to ***Do not check (not recommended)***.

OSX: ***System Preferences > Java***
Advanced
Perform certificate revocation using: Change to 'Do not check (not recommended)'

Windows: ***Control Panel > Java***
Advanced
Perform certificate revocation using: Change to 'Do not check (not recommended)'

Updated: Aug 07, 2013 Document ID: 116444