Runtime Infrastructure - Bug #5388

handshake failure while testing customer app

05/26/2021 11:40 AM - Eric Faulhaber

Status:	Closed	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	Igor Skornyakov	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Related to Runtime Infrastructure - Bug #5362: Handshake failure while testin...	**Closed**

History

#1 - 05/26/2021 11:40 AM - Eric Faulhaber

- Related to Bug #5362: Handshake failure while testing the simple web cllient added

#2 - 05/26/2021 11:46 AM - Eric Faulhaber

I'm not sure this is actually a different root cause than #5362, but the stack trace is different.

While opening a screen from the main menu of a large customer application, the client stopped responding and I found the following in the server log:

```
[05/24/2021 14:48:15 EDT] (com.goldencode.p2j.net.SSL:WARNING) handshake failure
javax.net.ssl.SSLException: Tag mismatch!
        at sun.security.ssl.Alert.createSSLException(Alert.java:133)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:324)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:267)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:262)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:119)
        at sun.security.ssl.SSLEngineImpl.decode(SSLEngineImpl.java:575)
        at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineImpl.java:531)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:398)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:377)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:484)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:386)
        at com.goldencode.p2j.net.SSL.run(SSL.java:256)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:247)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:172)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: javax.crypto.AEADBadTagException: Tag mismatch!
        at com.sun.crypto.provider.GaloisCounterMode.decryptFinal(GaloisCounterMode.java:620)
        at com.sun.crypto.provider.CipherCore.finalNoPadding(CipherCore.java:1116)
        at com.sun.crypto.provider.CipherCore.fillOutputBuffer(CipherCore.java:1053)
        at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:941)
        at com.sun.crypto.provider.AESCipher.engineDoFinal(AESCipher.java:491)
        at javax.crypto.CipherSpi.bufferCrypt(CipherSpi.java:779)
        at javax.crypto.CipherSpi.engineDoFinal(CipherSpi.java:730)
        at javax.crypto.Cipher.doFinal(Cipher.java:2463)
        at sun.security.ssl.SSLCipher$T12GcmReadCipherGenerator$GcmReadCipher.decrypt(SSLCipher.java:1606)
        at sun.security.ssl.SSLEngineInputRecord.decodeInputRecord(SSLEngineInputRecord.java:240)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:197)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:160)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:109)
        at sun.security.ssl.SSLEngineImpl.decode(SSLEngineImpl.java:575)
```

```
        at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineImpl.java:531)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:398)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:377)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:484)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:386)
        at com.goldencode.p2j.net.SSL.run(SSL.java:256)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:247)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:172)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

I had opened a number of screens from the menu successfully before this occurred, so I do not think it necessarily is specific to the application.

I don't have a reliable recreate; it seems to have been rather random.

I was using 3821c/11446 (plus some local persistence-related changes which should be unrelated).

#3 - 05/26/2021 12:07 PM - Roger Borrello

Sergey, could improper tuning of #5028-39 lead to this issue? Perhaps retry logic is trying to re-utilize a connection that needs to be re-established?

#4 - 05/26/2021 12:18 PM - Greg Shah

> Sergey, could improper tuning of #5028-39 lead to this issue? Perhaps retry logic is trying to re-utilize a connection that needs to be re-established?

#5028-39 is related to the websocket connection between the FWD client and the javascript part of the web client which runs in the browser.

This task is related to the SSL connection between the FWD client and the FWD server. The two are probably not related.

#7 - 06/01/2021 02:16 PM - Greg Shah

This problem is also being seen in #5347.

#8 - 06/01/2021 03:59 PM - Greg Shah

- Assignee set to Igor Skornyakov

#9 - 06/01/2021 04:17 PM - Eric Faulhaber

- Priority changed from Normal to High

#10 - 06/04/2021 10:02 AM - Igor Skornyakov

I've found a similar issue in the internet-
https://stackoverflow.com/questions/53320551/javax-crypto-aeadbadtagexception-tag-mismatch-error-when-encrypting-string/53325590.
It looks like a problem with the decryption implementation. However, in FWD all the cryptography is done by SSLEngine and (default) Cryptographic Service Provider.
Trying to understand how FWD code can affect these components,

#11 - 06/06/2021 03:27 PM - Igor Skornyakov

I've tried to reproduce this issue both with openjdk version "1.8.0_292" and Oracle java version "1.8.0_201", but failed.

Trying to understand what type of encrypted data corruption can cause the exception described in this task. I've written a significant amount of cryptography-related code but have never seen anything like this.

#12 - 06/07/2021 06:35 AM - Greg Shah

Can you put logging code in that will help you figure this out? Eugenie and others see this routinely.

#13 - 06/07/2021 06:43 AM - Igor Skornyakov

Greg Shah wrote:

> Can you put logging code in that will help you figure this out? Eugenie and others see this routinely.

I'm thinking about this, but the problem happens deep in the JRE code and it is unclear what to log. But I will try to provide more information.

What about allow using an open-source security provider (such as BC) at the server-side? This can be enabled by configuration.

#14 - 06/07/2021 06:58 AM - Greg Shah

I'm not opposed to this, as long as it defaults to the JDK version. We already use BouncyCastle for certificate creation.

#15 - 06/07/2021 07:12 AM - Igor Skornyakov

Greg Shah wrote:

> I'm not opposed to this, as long as it defaults to the JDK version. We already use BouncyCastle for certificate creation.

Thank you. Of course, the default will be JDK.

#16 - 06/08/2021 04:05 PM - Igor Skornyakov

We use a very old version of the BouncyCastle library which does not contain the JSSE provider required for TLS/SSL. The minimal version of BC that contains bctls-jdk15on is 1.57. I've tested the large customer app with the most recent version 1.68 and it seems working.
Is it OK to upgrade?
Thank you.

#17 - 06/08/2021 04:49 PM - Greg Shah

Constantin: Any objections?

#18 - 06/08/2021 05:44 PM - Igor Skornyakov

- Added additional information to the log in the unwrap() failure.
- Added optional use of the BouncyCastle JCE and JSSE providers (enabled bu security:bouncycastle:use=true bootstrap param. This requires BC upgrade (see #5388-16).
  This is ready for commit. Waiting for the BC upgrade approval.

#19 - 06/10/2021 08:27 AM - Constantin Asofiei

Igor, does this impact the BouncyCastle usage for SSLCertGenUtil? I mean, did they change APIs?

In any case, please do a SSLCertGenUtil run to re-generate certificates with the new bouncycastle version, and see if it completes OK.

If that works, then I'm OK with the upgrade.

#20 - 06/10/2021 09:04 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, does this impact the BouncyCastle usage for SSLCertGenUtil? I mean, did they change APIs?
>
> In any case, please do a SSLCertGenUtil run to re-generate certificates with the new bouncycastle version, and see if it completes OK.
>
> If that works, then I'm OK with the upgrade.

OK. Thank you.

#21 - 06/10/2021 10:03 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, does this impact the BouncyCastle usage for SSLCertGenUtil? I mean, did they change APIs?
>
> In any case, please do a SSLCertGenUtil run to re-generate certificates with the new bouncycastle version, and see if it completes OK.
>
> If that works, then I'm OK with the upgrade.

I've got the following exception (providing customer directory.xml as a single argument):

```
Exception in thread "main" com.goldencode.p2j.security.SSLCertGenException: com.goldencode.p2j.security.SSLCer
tGenException: The AES key must have 128, 192 or 256 bits!
    at com.goldencode.p2j.security.SSLCertGenUtil.generateRootCA(SSLCertGenUtil.java:1520)
    at com.goldencode.p2j.security.SSLCertGenUtil.generate(SSLCertGenUtil.java:514)
    at com.goldencode.p2j.security.SSLCertGenUtil.main(SSLCertGenUtil.java:2016)
Caused by: com.goldencode.p2j.security.SSLCertGenException: The AES key must have 128, 192 or 256 bits!
    at com.goldencode.p2j.security.BCCertFactory.cipherBytes(BCCertFactory.java:192)
    at com.goldencode.p2j.security.BCCertFactory.decryptPrivateKey(BCCertFactory.java:274)
    at com.goldencode.p2j.security.SSLCertGenUtil.generateRootCA(SSLCertGenUtil.java:1477)
    ... 2 more
```

Analyzing.

#22 - 06/10/2021 10:07 AM - Constantin Asofiei

Use the directory from testcases project, that one is more stable.

#23 - 06/10/2021 11:11 AM - Igor Skornyakov

Constantin Asofiei wrote:

> Use the directory from testcases project, that one is more stable.

Thank you.
Now I have another problem:

```
Exception in thread "main" com.goldencode.p2j.security.SSLCertGenException: com.goldencode.p2j.security.SSLCer
tGenException: org.bouncycastle.crypto.InvalidCipherTextException: pad block corrupted
    at com.goldencode.p2j.security.SSLCertGenUtil.generateRootCA(SSLCertGenUtil.java:1520)
```

```
    at com.goldencode.p2j.security.SSLCertGenUtil.generate(SSLCertGenUtil.java:514)
    at com.goldencode.p2j.security.SSLCertGenUtil.main(SSLCertGenUtil.java:2016)
Caused by: com.goldencode.p2j.security.SSLCertGenException: org.bouncycastle.crypto.InvalidCipherTextException
: pad block corrupted
    at com.goldencode.p2j.security.BCCertFactory.cipherBytes(BCCertFactory.java:223)
    at com.goldencode.p2j.security.BCCertFactory.decryptPrivateKey(BCCertFactory.java:274)
    at com.goldencode.p2j.security.SSLCertGenUtil.generateRootCA(SSLCertGenUtil.java:1477)
    ... 2 more
Caused by: org.bouncycastle.crypto.InvalidCipherTextException: pad block corrupted
    at org.bouncycastle.crypto.paddings.PKCS7Padding.padCount(Unknown Source)
    at org.bouncycastle.crypto.paddings.PaddedBufferedBlockCipher.doFinal(Unknown Source)
    at com.goldencode.p2j.security.BCCertFactory.cipherBytes(BCCertFactory.java:213)
    ... 4 more
```

Analyzing

#24 - 06/10/2021 01:30 PM - Igor Skornyakov

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > Use the directory from testcases project, that one is more stable.
>
> Thank you.
> Now I have another problem:
> [...]
> Analyzing

Actually, the same happens with the previous version of the BC.

#25 - 06/10/2021 02:11 PM - Igor Skornyakov

It seems that I just use the utility with wrong parameters.
What is the root CA private-key encryption password for the testcases directory.xml?

Thank you.

#26 - 06/10/2021 02:25 PM - Sergey Ivanovskiy

It seems that hotel_gui project is more reliable because it was tested with the directory template.

#27 - 06/10/2021 02:29 PM - Igor Skornyakov

Sergey Ivanovskiy wrote:

> It seems that hotel_gui project is more reliable because it was tested with the directory template.

Thank you, I will try. But what is the password for it?
Thanks,

#28 - 06/10/2021 02:36 PM - Sergey Ivanovskiy

- File pk-store-log.txt added

I have this old log for hotel_gui.

#29 - 06/10/2021 02:36 PM - Constantin Asofiei

Igor, do not reuse the root CA - regenerate everything, including this.

#30 - 06/10/2021 02:38 PM - Sergey Ivanovskiy

- File create_certificates added

This is the old log for testcases project.

#31 - 06/10/2021 02:41 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, do not reuse the root CA - regenerate everything, including this.

In this case, everything works fine.

#32 - 06/10/2021 02:46 PM - Constantin Asofiei

Great, now do this:

- revert the directory/store changes so you are on the latest revision. Run again but this time reuse the root CA - for the xfer testcases, the password is in deploy/server/pk-store-log.txt
- save the output from previous run, and run again reusing the newly generated root CA.

I just want to make sure that the change is backwards compatible with previously generated root CA.

#33 - 06/10/2021 03:12 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Great, now do this:
>
> - revert the directory/store changes so you are on the latest revision. Run again but this time reuse the root CA - for the xfer testcases, the password is in deploy/server/pk-store-log.txt
> - save the output from previous run, and run again reusing the newly generated root CA.
>
> I just want to make sure that the change is backwards compatible with previously generated root CA.

When I try to re-use the password from the pk-store-log.txt I get the The AES key must have 128, 192 or 256 bits error. Indeed all generated passwords are 36 characters long (should be 32).

#34 - 06/10/2021 03:18 PM - Constantin Asofiei

You are using the wrong password - use the one from the '... password and the password was NOT SAVED in the directory. Losing this password will prohibit using the root CA to issue new certificates.' line.

#35 - 06/10/2021 03:44 PM - Igor Skornyakov

Here is the log of the full re-generation:

```
Picked up _JAVA_OPTIONS: -Duser.timezone=GMT+3
Initializing service for directory directory.xml...
Reading company configuration...
Using 'US' for [C] Country Code.
Using 'Alpharetta' for [L] Locality (city, etc).
Using 'GoldenCode' for [O] Organization.
Using 'Hotel' for [OU] Organization Unit.
Using 'GA' for [ST] State or Province Name.
Using 10 years for validity.
Adding account /security/accounts/processes/appserver_process
Adding account /security/accounts/processes/appserver_agent
Adding account /security/accounts/processes/standard
Adding account /security/accounts/users/bogus
WARNING: alias shared is set for multiple accounts!
Account /security/accounts/users/lq9503don13zs97x ignored - no alias is defined
Account /security/accounts/users/lnrvfy643s73y926 ignored - no alias is defined
Account /security/accounts/users/0v4ox63dn9m599sa ignored - no alias is defined
Account /security/accounts/users/yli84146oxx1gg26 ignored - no alias is defined
Account /security/accounts/users/97h20it1ki3n83ug ignored - no alias is defined
Account /security/accounts/users/3dx05pr0z9rx63x2 ignored - no alias is defined
Account /security/accounts/users/3r278c73ci37aeil ignored - no alias is defined
Account /security/accounts/users/imu52uq68172d6eg ignored - no alias is defined
Enter the target P2J server identifier (by default "standard"): no
Re-use the encryption passwords currently in the directory (yes/no): no
Re-use the root CA currently in the directory (yes/no): no
Load and use the externally generated root CA (yes/no): no
Generating root CA using [root-ca] alias...
Generating certificate for account with alias [appserver_agent]...
Generating certificate for account with alias [appserver_process]...
Generating certificate for account with alias [shared]...
Generating certificate for server with alias [standard]...
```

```
Encrypt the in-directory private keys using the same password (yes/no): yes
Private key for root CA [root-ca] was encrypted using the [SiCxzom3cUT47hrp!Y)29T8Y!1&J3wDF] password and the
password was NOT SAVED in the directory.  Losing this password will prohibit using the root CA to issue new ce
rtificates.
Adding certificate for alias [root-ca] to the /security/certificates/cas/root-ca node...
Certificate for alias [root-ca] has checksum [be2de69e16e6b3c032785f6566cf9826]
Private key for alias [appserver_agent] was encrypted using the [voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je] password an
d saved in the /security/certificates/private-keys/appserver_agent node.
Use the access:password:keyentry-appserver_agent key to set this password in the server's bootstrap config fil
e and delete  the /security/certificates/private-keys/appserver_agent node from the directory.
Adding certificate for alias [appserver_agent] to the /security/certificates/peers/appserver_agent node...
Certificate for alias [appserver_agent] has checksum [b840d1b97143abcfedcc91a3510b94e1]
Private key for alias [appserver_process] was encrypted using the [voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je] password
and saved in the /security/certificates/private-keys/appserver_process node.
Use the access:password:keyentry-appserver_process key to set this password in the server's bootstrap config f
ile and delete  the /security/certificates/private-keys/appserver_process node from the directory.
Adding certificate for alias [appserver_process] to the /security/certificates/peers/appserver_process node...
Certificate for alias [appserver_process] has checksum [d4bcccb11edd2313521f175d657f05e8]
Private key for alias [shared] was encrypted using the [voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je] password and saved i
n the /security/certificates/private-keys/shared node.
Use the access:password:keyentry-shared key to set this password in the server's bootstrap config file and del
ete  the /security/certificates/private-keys/shared node from the directory.
Adding certificate for alias [shared] to the /security/certificates/peers/shared node...
Certificate for alias [shared] has checksum [96b3f35f35c319da4164f681f589cd7d]
Private key for alias [standard] was encrypted using the [voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je] password and saved
 in the /security/certificates/private-keys/standard node.
Use the access:password:keyentry-standard key to set this password in the server's bootstrap config file and d
elete  the /security/certificates/private-keys/standard node from the directory.
Adding certificate for alias [standard] to the /security/certificates/peers/standard node...
Certificate for alias [standard] has checksum [3ed4ee4ea80ca9a9adc50076a40e0e51]
The private key for certificate [appserver_agent] was saved in the [appserver_agent-private-key.store] key sto
re, using store password [#Lgu6x4#jOj=DMzzj(h11G2EHhv8yngr5qz0] and key-entry password [#uAFH8?9m7rS4Ssw04XXOt
1riUFfk<Ll=jD9].
The private key for certificate [appserver_process] was saved in the [appserver_process-private-key.store] key
 store, using store password [TLTgY4Nj%AL)HJ920hQeEb68rS3JD3nDi~)Y] and key-entry password [+bW76_6)iKx101Xvav
KLaVjxJJelmK6w+6pK].
The private key for certificate [shared] was saved in the [shared-private-key.store] key store, using store pa
ssword [XJC*09T95wUGKiNQv(bo(IyN4ItmTX689lI#] and key-entry password [VwbrWEozOn=JcS85I-PI309fTJ171ja_<gPE].
The private key for certificate [standard] was saved in the [standard-private-key.store] key store, using stor
e password [k~>K2Mm9h_2SgnSjRFa83`onop5u01CpKfNE] and key-entry password [1zCE1Kufa0@oH1eF4B#H89_neKcLO#RsRI2Q
].
The servers' certificates [standard] were saved in the [srv-certs.store] key store, using password [xnLtbtjW16
dEiGXk5bYO4TZ(n2-n!6Xr88q)].
The root CA's private key and certificate were saved in the [root-ca-pk.store] store.
The key store was encrypted using the [9qz6Sc@nF4kOW3ANQ@yw7umYH-U8>Jv7B7Xr] password, while the root CA's pri
vate key was encrypted using the [_p0U=OdocQDNmgld1uWp42Eo!GP2py10>YM6] password.
Done.
WARNING! Any configuration set at the client.xml or server.xml files or via bootstrap config arguments will ha
ve priority over the in-directory keys or certificates.
WARNING! The private key encryption passwords for all the account certificates are saved unencrypted in the di
rectory.  The root CA's private key can be safely deleted, if it is not required to issue other certificates u
sing this CA.
WARNING! All private keys are encrypted using the same [voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je] password. If needed,
 delete the 'key-password' nodes from the directory manually, and set this password using the access:password:
masterkeyentry bootstrap config at server startup.
```

As you can see the password for the root CA private key is 36 chars long.

The password voN4CVDeY4dp=UK4Gt*xl0h=~a1446Je has a correct length but it cannot be used for the root CA PK (decryption fails)

The situation with the previous version of BC is the same.

What I'm doing wrong?
Thank you.

#36 - 06/10/2021 04:11 PM - Constantin Asofiei

I get this too, with the xfer project. I might have merged the directory.xml wrong or something else, when I re-generated the certificates.

Anyway, if I run cleanly and re-use the root CA, it works with 3821c.

So, my advice: re-generate everything (including root CA) with 3821c, and after that do the tests in #5388-32

#37 - 06/10/2021 04:16 PM - Igor Skornyakov

Constantin Asofiei wrote:

> I get this too, with the xfer project. I might have merged the directory.xml wrong or something else, when I re-generated the certificates.
>
> Anyway, if I run cleanly and re-use the root CA, it works with 3821c.
>
> So, my advice: re-generate everything (including root CA) with 3821c, and after that do the tests in #5388-32

But this is exactly what I do. Which password should I use on a second run?
As I mentioned, I see only one with a correct length but it doesn't work for the root CA private key.
Thank you.

#38 - 06/10/2021 04:21 PM - Constantin Asofiei

Igor Skornyakov wrote:

> But this is exactly what I do. Which password should I use on a second run?

The one on this line:

```
Private key for root CA [root-ca] was encrypted using the [SiCxzom3cUT47hrp!Y)29T8Y!1&J3wDF] password and the password was NOT SAVED in the directory.  Losing this password will prohibit using the root CA to issue new certificates.
```

This is the password for the root CA's private-key saved (and encrypted) in the directory.

The weird part is that I've reverted all changes in deploy/server and tried again with the password from pk-store-log.txt, and it worked.

#39 - 06/10/2021 04:29 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor Skornyakov wrote:
>
> > But this is exactly what I do. Which password should I use on a second run?
>
> The one on this line:
> [...]
> This is the password for the root CA's private-key saved (and encrypted) in the directory.

Well, it works now. Thanks a lot!

#40 - 06/10/2021 04:30 PM - Igor Skornyakov

Igor Skornyakov wrote:

> Well, it works now. Thanks a lot!

Can I commit my changes with BC upgrade?
Thank you.

#41 - 06/10/2021 04:34 PM - Constantin Asofiei

Igor Skornyakov wrote:

> Igor Skornyakov wrote:
>
> > Well, it works now. Thanks a lot!
>
> Can I commit my changes with BC upgrade?
> Thank you.

Yes.

#42 - 06/10/2021 04:51 PM - Igor Skornyakov

Committed to 3821c/12527.
I cannot reproduce the issue described in this and related tasks.

Those, which experience this, please switch to the BouncyCastle JCE/JSSE using security:bouncycastle:use=true bootstrap param. Id the problems will be seen again, please provide a stack trace of the exception and the log of the corresponding peer client.
There is no need to switch to BC on both sides.

Thank you.

#43 - 06/10/2021 08:42 PM - Roger Borrello

Igor Skornyakov wrote:

> Committed to 3821c/12527.
> I cannot reproduce the issue described in this and related tasks.
>
> Those, which experience this, please switch to the BouncyCastle JCE/JSSE using security:bouncycastle:use=true bootstrap param. Id the problems will be seen again, please provide a stack trace of the exception and the log of the corresponding peer client.
> There is no need to switch to BC on both sides.
>
> Thank you.

Which file should get updated? Is that directory.xml or client.xml?

#44 - 06/11/2021 03:28 AM - Igor Skornyakov

Roger Borrello wrote:

> Which file should get updated? Is that directory.xml or client.xml?

The issue is seen on the server side. So you should update server.xml like this:

```
<security>
   <server id="standard"/>
   <bouncycastle use="true"/>
</security>
```

or add a security:bouncycastle:use=true command-line option to the server.sh.

Please do not forget to remove old BC jars (bc*-jdk15on-1.56.jar) from the deploy/lib.

#45 - 06/11/2021 09:09 AM - Roger Borrello

Igor Skornyakov wrote:

> We use a very old version of the BouncyCastle library which does not contain the JSSE provider required for TLS/SSL. The minimal version of BC that contains bctls-jdk15on is 1.57. I've tested the large customer app with the most recent version 1.68 and it seems working.
> Is it OK to upgrade?
> Thank you.

What are the steps for upgrading BC?

#46 - 06/11/2021 09:31 AM - Igor Skornyakov

Roger Borrello wrote:

> Igor Skornyakov wrote:
>
> > We use a very old version of the BouncyCastle library which does not contain the JSSE provider required for TLS/SSL. The minimal version of BC that contains bctls-jdk15on is 1.57. I've tested the large customer app with the most recent version 1.68 and it seems working.
> > Is it OK to upgrade?
> > Thank you.
>
> What are the steps for upgrading BC?

Just rebuild the project. You may also need to manually update the project settings in your IDE

#47 - 06/13/2021 11:04 AM - Igor Skornyakov

Please note that I've experienced strange problems with BoucyCastle JSSE and Oracle JDK.
Do not use this combination until I will resolve the issue.
Sorry for the inconvenience.

#48 - 06/15/2021 09:48 AM - Roger Borrello

Just to confirm, running SSLCertGenUtil to create new certificates is **not** required because they are backwardly compatible? I need to know in order to properly update a customer server.

#49 - 06/15/2021 09:49 AM - Igor Skornyakov

Roger Borrello wrote:

> Just to confirm, running SSLCertGenUtil to create new certificates is **not** required because they are backwardly compatible? I need to know in order to properly update a customer server.

Yes, there is no need to re-generate the certificates.

#50 - 06/15/2021 10:43 AM - Roger Borrello

I received this error in server.log:

```
java.lang.NoSuchMethodError: org.bouncycastle.util.encoders.Hex.decodeStrict(Ljava/lang/String;)[B
    at org.bouncycastle.tls.TlsUtils.<clinit>(Unknown Source)
    at org.bouncycastle.jsse.provider.CipherSuiteInfo.forCipherSuite(Unknown Source)
    at org.bouncycastle.jsse.provider.ProvSSLContextSpi.addCipherSuite(Unknown Source)
    at org.bouncycastle.jsse.provider.ProvSSLContextSpi.addCipherSuite13(Unknown Source)
    at org.bouncycastle.jsse.provider.ProvSSLContextSpi.createSupportedCipherSuiteMap(Unknown Source)
    at org.bouncycastle.jsse.provider.ProvSSLContextSpi.<clinit>(Unknown Source)
    at org.bouncycastle.jsse.provider.BouncyCastleJsseProvider$3.createInstance(Unknown Source)
    at org.bouncycastle.jsse.provider.BouncyCastleJsseProvider$BcJsseService.newInstance(Unknown Source)
    at sun.security.jca.GetInstance.getInstance(GetInstance.java:236)
    at sun.security.jca.GetInstance.getInstance(GetInstance.java:218)
    at javax.net.ssl.SSLContext.getInstance(SSLContext.java:236)
    at com.goldencode.p2j.security.SecurityManager.getSecureSocketContext(SecurityManager.java:1913)
    at com.goldencode.p2j.security.SecurityManager.getSecureSocketContext(SecurityManager.java:1889)
    at com.goldencode.p2j.security.SecurityManager.<init>(SecurityManager.java:739)
    at com.goldencode.p2j.security.SecurityManager.createInstance(SecurityManager.java:791)
    at com.goldencode.p2j.main.StandardServer.bootstrap(StandardServer.java:887)
    at com.goldencode.p2j.main.ServerDriver.start(ServerDriver.java:485)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:444)
    at com.goldencode.p2j.main.ServerDriver.process(ServerDriver.java:209)
    at com.goldencode.p2j.main.ServerDriver.main(ServerDriver.java:863)
```

The 1.68 jar files were in place. It went away when I set <bouncycastle use="false"/> in the server.xml.

Any suggestions what I did wrong? Maybe there's an old version somewhere earlier in the classpath?

#51 - 06/15/2021 12:18 PM - Igor Skornyakov

Roger Borrello wrote:

> I received this error in server.log:
> [...]
>
> The 1.68 jar files were in place. It went away when I set <bouncycastle use="false"/> in the server.xml.
>
> Any suggestions what I did wrong? Maybe there's an old version somewhere earlier in the classpath?

Please double check that you've removed **all** old BC jars (bc*-jdk15on-1.56.jar) from the deploy/lib

#52 - 06/15/2021 04:49 PM - Igor Skornyakov

The problem is that with the combination of Oracle JDK and BouncyCastle JCR/JSST providers the server fails to select a cipher suite during the handshake. I've found a description of several similar use cases on the Internet but the provided workarounds do not work in our case.
I'm trying to compare the execution flow with Oracle JDK and OpenJDK using a debugger but so far do not understand what is the root difference.

#53 - 06/16/2021 04:54 AM - Igor Skornyakov

It looks that the problem can be a race condition. When I step through the cipher suite selection with a debugger and JVM suspend on breakpoints, the correct suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 is selected. Investigating.

#54 - 06/16/2021 09:44 AM - Roger Borrello

Igor Skornyakov wrote:

> Please double check that you've removed **all** old BC jars (bc*-jdk15on-1.56.jar) from the deploy/lib

Thanks... on the deployment server, I deploy to /opt/<path>/client and /opt/<path>/server and the older ones were in the server/lib folder. It will be active the next server restart.

#55 - 06/16/2021 10:46 AM - Greg Shah

What can we do to disallow BouncyCastle in OracleJDK? I'm fine with that for now. I don't want to spend any more time on Oracle's bugs.

#56 - 06/16/2021 11:10 AM - Igor Skornyakov

Greg Shah wrote:

> What can we do to disallow BouncyCastle in OracleJDK? I'm fine with that for now. I don't want to spend any more time on Oracle's bugs.

BouncyCastle is disallowed by default. However, I'm not 100% sure that it is Oracle's bug. My guess is that the way we initialize SSLContext is not 100% correct and it can cause problems in the future.

#57 - 06/16/2021 11:21 AM - Greg Shah

Using BouncyCastle is really just a hopeful attempt at a workaround for some low level network issue which you can't recreate. Is that correct? We don't even know that it will give us any relief. I really don't want to spend time on this unless you are SURE that it is actually an important thing to support.

#58 - 06/16/2021 11:27 AM - Igor Skornyakov

Greg Shah wrote:

> Using BouncyCastle is really just a hopeful attempt at a workaround for some low level network issue which you can't recreate. Is that correct? We don't even know that it will give us any relief. I really don't want to spend time on this unless you are SURE that it is actually an important thing to support.

I see. So, I suspend working on this and switch to #4551 as per the latest instructions from Eric.

#59 - 06/16/2021 11:46 AM - Greg Shah

Yes, work on #4551 while we wait for Roger to get some results on this.

Roger: Were you previously able to recreate the issue locally? If so, then you can use the OpenJDK to test this out. What JDK is used on the customer's test server?

#60 - 06/16/2021 11:54 AM - Greg Shah

> It looks that the problem can be a race condition. When I step through the cipher suite selection with a debugger and JVM suspend on breakpoints, the correct suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 is selected. Investigating.

One thing I don't understand here is how there can be a race condition. Our server startup is single threaded. The only thing that comes to mind here is that at some point we do start appserver agents/batch processes. Is that involved? If not, what threads are involved in the race condition?

#61 - 06/16/2021 11:58 AM - Roger Borrello

Greg Shah wrote:

> Roger: Were you previously able to recreate the issue locally? If so, then you can use the OpenJDK to test this out. What JDK is used on the customer's test server?

It wasn't actually a recreate. It was very sporadic, and did happen on my local laptop. The customer is using OpenJDK.

#62 - 06/16/2021 12:04 PM - Greg Shah

OK. Please enable the BC stuff in both places. Let us know if you see the problem.

#63 - 06/16/2021 12:10 PM - Igor Skornyakov

Greg Shah wrote:

> One thing I don't understand here is how there can be a race condition. Our server startup is single threaded. The only thing that comes to mind here is that at some point we do start appserver agents/batch processes. Is that involved? If not, what threads are involved in the race condition?

I also do not understand the details. I'm talking about race condition because based on my experience the issue that disappears when debugging is most likely caused by something like this. Please note that SSL handshaking uses two auxiliary threads.

#64 - 06/16/2021 04:04 PM - Roger Borrello

Greg Shah wrote:

> OK. Please enable the BC stuff in both places. Let us know if you see the problem.

It has been enabled.

#65 - 06/22/2021 08:31 AM - Greg Shah

Roger: Have you seen the issue recur? By now would you have expected to see the issue recur?

#66 - 06/22/2021 09:25 AM - Roger Borrello

- File server.log added

I just looked at a log from yesterday, where I was working on the Sikuli testcases. I see some NPE and other exceptions which are not quite the same. There is mention of the com function for PSTimer, so I am not sure it is related. The who log is attached:

```
[06/21/2021 16:10:28 EDT] (Protocol.Writer.run():WARNING) {Writer [00000006:bogus]} failure in writing loop
java.lang.NullPointerException
        at org.bouncycastle.tls.crypto.TlsCryptoUtils.hkdfExpandLabel(Unknown Source)
        at org.bouncycastle.tls.TlsUtils.update13TrafficSecret(Unknown Source)
        at org.bouncycastle.tls.TlsUtils.update13TrafficSecret(Unknown Source)
        at org.bouncycastle.tls.TlsUtils.update13TrafficSecretLocal(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.send13KeyUpdate(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.writeApplicationData(Unknown Source)
        at org.bouncycastle.jsse.provider.ProvSSLEngine.wrap(Unknown Source)
        at javax.net.ssl.SSLEngine.wrap(SSLEngine.java:471)
        at com.goldencode.p2j.net.SSL.wrap(SSL.java:435)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:385)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.send(SSL.java:230)
        at com.goldencode.p2j.net.BlockingSSL.send(BlockingSSL.java:165)
        at com.goldencode.p2j.net.NIOSslSocket.write(NIOSslSocket.java:247)
        at com.goldencode.p2j.net.NIONetSocketBase.write(NIONetSocketBase.java:196)
        at com.goldencode.p2j.net.Queue.write(Queue.java:955)
        at com.goldencode.p2j.net.Protocol$Writer.run(Protocol.java:642)
        at java.lang.Thread.run(Thread.java:748)
```

#67 - 06/24/2021 05:34 AM - Constantin Asofiei

I still get this with a customer application:

```
javax.net.ssl.SSLException: org.bouncycastle.tls.TlsFatalAlert: bad_record_mac(20)
        at org.bouncycastle.jsse.provider.ProvSSLEngine.unwrap(Unknown Source)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: org.bouncycastle.tls.TlsFatalAlert: bad_record_mac(20)
        at org.bouncycastle.tls.crypto.impl.TlsAEADCipher.decodeCiphertext(Unknown Source)
        at org.bouncycastle.tls.RecordStream.decodeAndVerify(Unknown Source)
        at org.bouncycastle.tls.RecordStream.readFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.safeReadFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.jsse.provider.ProvSSLEngine.unwrap(Unknown Source)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
```

```
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.IllegalStateException
        at org.bouncycastle.tls.crypto.impl.jcajce.Exceptions.illegalStateException(Unknown Source)
        at org.bouncycastle.tls.crypto.impl.jcajce.JceAEADCipherImpl.doFinal(Unknown Source)
        at org.bouncycastle.tls.crypto.impl.TlsAEADCipher.decodeCiphertext(Unknown Source)
        at org.bouncycastle.tls.RecordStream.decodeAndVerify(Unknown Source)
        at org.bouncycastle.tls.RecordStream.readFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.safeReadFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.jsse.provider.ProvSSLEngine.unwrap(Unknown Source)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: javax.crypto.AEADBadTagException: Tag mismatch!
        at com.sun.crypto.provider.GaloisCounterMode.decryptFinal(GaloisCounterMode.java:620)
        at com.sun.crypto.provider.CipherCore.finalNoPadding(CipherCore.java:1116)
        at com.sun.crypto.provider.CipherCore.fillOutputBuffer(CipherCore.java:1053)
        at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:941)
        at com.sun.crypto.provider.AESCipher.engineDoFinal(AESCipher.java:491)
        at javax.crypto.Cipher.doFinal(Cipher.java:2115)
        at org.bouncycastle.tls.crypto.impl.jcajce.JceAEADCipherImpl.doFinal(Unknown Source)
        at org.bouncycastle.tls.crypto.impl.TlsAEADCipher.decodeCiphertext(Unknown Source)
        at org.bouncycastle.tls.RecordStream.decodeAndVerify(Unknown Source)
        at org.bouncycastle.tls.RecordStream.readFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.safeReadFullRecord(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.tls.TlsProtocol.offerInput(Unknown Source)
        at org.bouncycastle.jsse.provider.ProvSSLEngine.unwrap(Unknown Source)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

I'm using:

```
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

#68 - 06/24/2021 06:35 AM - Igor Skornyakov

Constantin Asofiei wrote:

> I still get this with a customer application:
> [...]
>
> I'm using:
> [...]

Constantin.
Do you see any errors/unusual messages in the client log?
Thank you.

#69 - 06/24/2021 06:36 AM - Constantin Asofiei

Igor Skornyakov wrote:

> Constantin Asofiei wrote:
>
> > I still get this with a customer application:
> > [...]
> >
> > I'm using:
> > [...]
>
> Constantin.
> Do you see any errors/unusual messages in the client log?
> Thank you.

Just this:

```
WARNING: handshake failure
javax.net.ssl.SSLException: Tag mismatch!
        at sun.security.ssl.Alert.createSSLException(Alert.java:133)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:324)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:267)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:262)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:119)
        at sun.security.ssl.SSLEngineImpl.decode(SSLEngineImpl.java:575)
        at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineImpl.java:531)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:398)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:377)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
```

```
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: javax.crypto.AEADBadTagException: Tag mismatch!
        at com.sun.crypto.provider.GaloisCounterMode.decryptFinal(GaloisCounterMode.java:620)
        at com.sun.crypto.provider.CipherCore.finalNoPadding(CipherCore.java:1116)
        at com.sun.crypto.provider.CipherCore.fillOutputBuffer(CipherCore.java:1053)
        at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:941)
        at com.sun.crypto.provider.AESCipher.engineDoFinal(AESCipher.java:491)
        at javax.crypto.CipherSpi.bufferCrypt(CipherSpi.java:779)
        at javax.crypto.CipherSpi.engineDoFinal(CipherSpi.java:730)
        at javax.crypto.Cipher.doFinal(Cipher.java:2463)
        at sun.security.ssl.SSLCipher$T12GcmReadCipherGenerator$GcmReadCipher.decrypt(SSLCipher.java:1606)
        at sun.security.ssl.SSLEngineInputRecord.decodeInputRecord(SSLEngineInputRecord.java:240)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:197)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:160)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:109)
```

#70 - 06/24/2021 07:16 AM - Constantin Asofiei

I got this 3 times already while in the customer app web client, trying to type something in a FILL-IN or EDITOR (and I was typing fast, not letter-by-letter, but once happened when I hit the first key).

I've tried recreating this with a standalone program, trying to type in an EDITOR, added a trigger... but no luck. I wonder if is not something else at play, some background threads which try to use the socket, a certain package (length?) which triggers this, or some weird concurrency issue.

#71 - 06/24/2021 07:46 AM - Igor Skornyakov

Constantin Asofiei wrote:

> I got this 3 times already while in the customer app web client, trying to type something in a FILL-IN or EDITOR (and I was typing fast, not letter-by-letter, but once happened when I hit the first key).
>
> I've tried recreating this with a standalone program, trying to type in an EDITOR, added a trigger... but no luck. I wonder if is not something else at play, some background threads which try to use the socket, a certain package (length?) which triggers this, or some weird concurrency issue.

I cannot understand. I've also tried to reproduce this, but with no luck as well. What is strange is that Sun JCE was used while the code configures BC as the primary JCE provider. This may be related to the problem with Oracle JDK + BC JCE/JSSE. The investigation was suspended but I suspect that it is also a result of some weird concurrency issue.

#72 - 06/24/2021 07:57 AM - Constantin Asofiei

How can I enable logging for the SSL code in FWD?

#73 - 06/24/2021 08:16 AM - Igor Skornyakov

Constantin Asofiei wrote:

> How can I enable logging for the SSL code in FWD?

Constantin.
You should enable the FINE level for the root logger and com.goldencode.p2j.net logger.

Unfortunately, the detailed logging is **extremely** verbose. In particular, because it will be polluted by the output from the SecurityManager. It can be useful for investigating the reproducible issues but in this case, I think it will just add a lot of overhead. Moreover, if the problem root is a concurrency issue this overhead can seriously distort the picture.

I have suggested starting using a more common approach to logging but was not supported.

#74 - 06/24/2021 08:23 AM - Greg Shah

Why can't you reduce the SecurityManager logging level?

#75 - 06/24/2021 08:25 AM - Igor Skornyakov

Greg Shah wrote:

> Why can't you reduce the SecurityManager logging level?

Because it uses some tricky logic for loggers' creation which I do not understand.

#76 - 06/24/2021 08:49 AM - Greg Shah

```
      DirectoryService ds = DirectoryService.getInstance();

      // override debug level from directory
      Level lvl = readDebugLevel(ds);

      if (lvl != null)
         debugLevel = lvl;

      // initialize logging (get an anonymous logger)
      LOG = LogHelper.getSecureLogger(bc.isServer());
      LOG.setLevel(debugLevel);
```

The core of readDebugLevel() is String lvl = ds.getNodeString("/security/config/debug-level", "value");. You can see the logging level names in that same method.

If you set that value to something less verbose, then it will apply. Is there something I'm misunderstanding here?

#77 - 06/24/2021 08:56 AM - Igor Skornyakov

Greg Shah wrote:

> [...]
>
> The core of readDebugLevel() is String lvl = ds.getNodeString("/security/config/debug-level", "value");. You can see the logging level names in that same method.
>
> If you set that value to something less verbose, then it will apply. Is there something I'm misunderstanding here?

I see. Thank you. I will test it. However, the SSL detailed logging is very verbose per see. Moreover, in this particular issue, the problem looks like encrypted data corruption which makes the analysis even more complicated.
I'm busy now with UDFs re-implementation and can resume the investigation after it will be done if you do not think that this is of higher priority.

#78 - 06/24/2021 09:04 AM - Greg Shah

I want to enable Constantin to gather useful information that can be posted here. Since you cannot recreate the issue, we need to help him gather what you need. For now, I just want you to figure out what logging is needed. If there is more logging to add to FWD to support your needs, then that can be done too.

#79 - 06/24/2021 09:24 AM - Igor Skornyakov

Greg Shah wrote:

> I want to enable Constantin to gather useful information that can be posted here. Since you cannot recreate the issue, we need to help him gather what you need. For now, I just want you to figure out what logging is needed. If there is more logging to add to FWD to support your needs, then that can be done too.

I see. Well, in this case, Constantin has to enable the detailed SSL logging as described in #5388-73 and reduce the log level for the SecurityManager to INFO.

I've just added more details to the logging on the unwrap failure. Committed to 3821c/12567.

#80 - 06/24/2021 09:57 AM - Greg Shah

> reduce the log level for the SecurityManager to INFO

The default logging level for SecurityManager is WARNING (same as Level.WARNING) which is less verbose than STAT (which is the same as Level.INFO).

Many of our project directories have this:

```
        <node class="string" name="debug-level">
          <node-attribute name="value" value="STAT"/>
        </node>
```

This means that most projects already set INFO as the logging level. This seems to be too verbose. Are you sure that is what you need?

#81 - 06/24/2021 10:02 AM - Igor Skornyakov

Greg Shah wrote:

> > reduce the log level for the SecurityManager to INFO
>
> The default logging level for SecurityManager is WARNING (same as Level.WARNING) which is less verbose than STAT (which is the same as Level.INFO).
>
> Many of our project directories have this:
>
> [...]
>
> This means that most projects already set INFO as the logging level. This seems to be too verbose. Are you sure that is what you need?

In my environment, there is no special configuration for the SecurityManager in the "logging" section, and "/security/config/debug-level" is set to DATA. However, when I set console handler level to FINEST I see a log of SecurityManager messages in the log.

#82 - 06/24/2021 10:19 AM - Greg Shah

Yes, the SecurityManager logger is still governed by the console handler, so whatever you set there is important. But in most environments, DATA is not used. You must have added that at some point. You can still control the SecurityManager logging and greatly limit it by using something like WARNING level.

#83 - 06/24/2021 10:21 AM - Igor Skornyakov

Greg Shah wrote:

> Yes, the SecurityManager logger is still governed by the console handler, so whatever you set there is important. But in most environments, DATA is not used. You must have added that at some point. You can still control the SecurityManager logging and greatly limit it by using something like WARNING level.

Got it. Thank you.

#84 - 06/24/2021 11:41 AM - Constantin Asofiei

Igor, I see an inconsistency - FWD server uses org.bouncycastle.jsse.provider.ProvSSLEngine as SSL.engine, and the FWD client has sun.security.ssl.SSLEngineImpl. Is this expected?

#85 - 06/24/2021 12:27 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, I see an inconsistency - FWD server uses org.bouncycastle.jsse.provider.ProvSSLEngine as SSL.engine, and the FWD client has sun.security.ssl.SSLEngineImpl. Is this expected?

Constantin,
This should not matter since both peers correctly implement SSL.

#86 - 06/24/2021 01:19 PM - Constantin Asofiei

Igor, I have a reliable recreate for this. It consists using a 'custom modified' converted program.

Basically, have an empty .p program and change its content to this once was converted:

```
   /**
    * External procedure (converted to Java from the 4GL source code
    * in c1.p).
    */
   @LegacySignature(type = Type.MAIN, name = "c1.p")
   public void execute()
   {
      externalProcedure(C1.this, new Block((Body) () ->
```

```
      {
         new V1().start();
         new V1().start();

         LogicalTerminal.pause();
      }));
   }

   public static class V1
   extends AssociatedThread
   implements Runnable
   {
      public V1()
      {
         super(() ->
         {
            while (true)
            {
               LogicalTerminal.getClient().getScreenLines();
            }
         });
      }

      @Override
      public synchronized void start()
      {
         SessionManager.get().registerAsyncThread(this);
         super.start();
      }
   }
```

This starts two threads which make async calls to the FWD client. It will abend in the first 2 seconds with the 'tag mismatch'. It will **not abend** if you run it in insecure (non-ssl) mode.

I could duplicate this without the web client, but you need to configure the FWD server and ChUI client to run in secure mode.

I've used this server.xml (with the server.xml net:server:insecure_port=3436 net:server:secure_port=3336 net:connection:secure=true command-line arguments):

```
<?xml version="1.0"?>
<!-- Server side bootstrap configuration
-->
<node type="server">
   <net>
      <router threads="2"/>
   </net>

   <security>
      <server id="standard"/>
      <keystore filename="standard-private-key.store"/>
      <keystore alias="standard"/>
      <bouncycastle use="true"/>
   </security>

   <directory>
      <backend type="xml"/>
      <xml filename="directory.xml"/>
   </directory>

   <access>
      <password keystore="pRNWF5>KjU37bQEwSqE^8FUBT-$hf2snc367"/>
      <password keyentry="CH9Q@Q6gp8qRd9zpic&amp;amp;$ylBwov`72J1V1cMF"/>
   </access>
</node>
```

and this client.xml (with the client.xml client:cmd-line-option:startup-procedure=c1.p command line arguments):

```
<?xml version="1.0"?>
```

```
<!-- Client side bootstrap configuration for the standard client application
     with the two way certificate validation
-->

<node type="client">
   <net>
      <connection secure="true" />
      <server host="localhost" />
      <server secure_port="3336" />
      <server insecure_port="3436" />
   </net>

   <security>
      <certificate validate="false" />
      <truststore filename="../server/srv-certs.store" />
      <truststore alias="standard" />
      <keystore filename="../server/some-new-process-alias-private-key.store" />
      <keystore processalias="some-new-process-alias" />
   </security>

   <access>
      <password truststore="NGf=8BrEeR8h1TSn^-0kBSxPyc&lt;78LzT6J8b" />
      <password keystore="xFt(XK4Z)&lt;8xEK1zy+yQK0XaC6j1Rj75vIYK" />
      <password keyentry="K3`g5uA465O8BWuHIGtfiQj#ex3W+ddzY%2l"/>
   </access>

   <client>
      <driver type="chui_swing"/>
      <chui rows="24"/>
      <chui columns="80"/>
      <chui background="0x000000"/>
      <chui foreground="0xFFA500"/>
      <chui selection="0x0000FF"/>
      <chui fontname="monospaced"/>
      <chui fontsize="12"/>
   </client>
</node>
```

You will need to adjust the settings with your directory.xml configuration (process alias, store filenames, passwords, etc); if you don't know the store passwords, re-generate everything using SSLCertGenUtil.

If you find this difficult, you can use the Web client, as this will be shown there, too.

The stacktrace I get is this:

```
avax.net.ssl.SSLException: Tag mismatch!
        at sun.security.ssl.Alert.createSSLException(Alert.java:133)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:324)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:267)
        at sun.security.ssl.TransportContext.fatal(TransportContext.java:262)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:119)
        at sun.security.ssl.SSLEngineImpl.decode(SSLEngineImpl.java:575)
        at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineImpl.java:531)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:398)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:377)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: javax.crypto.AEADBadTagException: Tag mismatch!
        at com.sun.crypto.provider.GaloisCounterMode.decryptFinal(GaloisCounterMode.java:620)
        at com.sun.crypto.provider.CipherCore.finalNoPadding(CipherCore.java:1116)
        at com.sun.crypto.provider.CipherCore.fillOutputBuffer(CipherCore.java:1053)
        at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:941)
```

```
        at com.sun.crypto.provider.AESCipher.engineDoFinal(AESCipher.java:491)
        at javax.crypto.CipherSpi.bufferCrypt(CipherSpi.java:779)
        at javax.crypto.CipherSpi.engineDoFinal(CipherSpi.java:730)
        at javax.crypto.Cipher.doFinal(Cipher.java:2463)
        at sun.security.ssl.SSLCipher$T12GcmReadCipherGenerator$GcmReadCipher.decrypt(SSLCipher.java:1606)
        at sun.security.ssl.SSLEngineInputRecord.decodeInputRecord(SSLEngineInputRecord.java:240)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:197)
        at sun.security.ssl.SSLEngineInputRecord.decode(SSLEngineInputRecord.java:160)
        at sun.security.ssl.SSLTransport.decode(SSLTransport.java:109)
        at sun.security.ssl.SSLEngineImpl.decode(SSLEngineImpl.java:575)
        at sun.security.ssl.SSLEngineImpl.readRecord(SSLEngineImpl.java:531)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:398)
        at sun.security.ssl.SSLEngineImpl.unwrap(SSLEngineImpl.java:377)
        at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:626)
        at com.goldencode.p2j.net.SSL.unwrap(SSL.java:491)
        at com.goldencode.p2j.net.SSL.handshake(SSL.java:387)
        at com.goldencode.p2j.net.SSL.run(SSL.java:257)
        at com.goldencode.p2j.net.SSL.notify(SSL.java:248)
        at com.goldencode.p2j.net.BlockingSSL.processInput(BlockingSSL.java:207)
        at com.goldencode.p2j.net.BlockingSSL.checkInput(BlockingSSL.java:113)
        at com.goldencode.p2j.net.NIOSslSocket.lambda$new$1(NIOSslSocket.java:173)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

#87 - 06/24/2021 01:40 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Igor, I have a reliable recreate for this. It consists using a 'custom modified' converted program.
>
> Basically, have an empty .p program and change its content to this once was converted:
> [...]
>
> This starts two threads which make async calls to the FWD client. It will abend in the first 2 seconds with the 'tag mismatch'. It will **not abend** if you run it in insecure (non-ssl) mode.
>
> I could duplicate this without the web client, but you need to configure the FWD server and ChUI client to run in secure mode.
>
> I've used this server.xml (with the server.xml net:server:insecure_port=3436 net:server:secure_port=3336 net:connection:secure=true command-line arguments):
> [...]
> and this client.xml (with the client.xml client:cmd-line-option:startup-procedure=c1.p command line arguments):
> [...]
>
> You will need to adjust the settings with your directory.xml configuration (process alias, store filenames, passwords, etc); if you don't know the store passwords, re-generate everything using SSLCertGenUtil.
>
> If you find this difficult, you can use the Web client, as this will be shown there, too.

> The stacktrace I get is this:
> [...]

This is great! Thanks a lot!

Greg.
Should I switch to this issue immediately or can finish with UDFs first?
Thank you.

#88 - 06/24/2021 01:50 PM - Greg Shah

Keep going with the UDFs. I think there will be a natural pause there while Eric reviews. At that point you can fix this.

#90 - 08/20/2021 08:17 AM - Greg Shah

Roger: Can this problem be recreated consistently using the test setup/scenario in #5611?

#91 - 08/20/2021 08:35 AM - Roger Borrello

Not consistently. But it seems likely to happen if you are impatient and trying to click on things before they are fully positioned.

#92 - 08/20/2021 08:56 AM - Greg Shah

Igor: Please try to recreate the problem on your local system using #5611/#5571. This is pretty urgent since it freezes the application and makes it unusable. I think Roger's idea of the recreate makes some sense. Perhaps the recursive event processing is causing some network flows that result in the problem. If you can't recreate it there, you will need to work on the customer system where the problem can be seen more often. In that scenario you may need to add instrumentation or other logic since traditional debugging may not be possible.

#93 - 08/20/2021 02:57 PM - Igor Skornyakov

Greg Shah wrote:

> Igor: Please try to recreate the problem on your local system using #5611/#5571. This is pretty urgent since it freezes the application and makes it unusable. I think Roger's idea of the recreate makes some sense. Perhaps the recursive event processing is causing some network flows that result in the problem. If you can't recreate it there, you will need to work on the customer system where the problem can be seen more often. In that scenario you may need to add instrumentation or other logic since traditional debugging may not be possible.

Greg,
I've tried to reproduce it but w/o any success. However, Constantin suggested a way to reproduce it reliably (#5388-86).
Should I suspend my work on #4551 and switch to this issue?
Thank you.

#94 - 08/20/2021 06:34 PM - Greg Shah

Yes, please focus on this.

#95 - 08/21/2021 03:24 AM - Igor Skornyakov

Greg Shah wrote:

> Yes, please focus on this.

OK, thank you.

#96 - 08/23/2021 06:13 PM - Roger Borrello

Igor, are you making progress on this?

I received this today, only the Tag mismatch was caused by javax.crypto.AEADBadTagException instead of javax.net.ssl.SSLException. Does that make any difference?

#97 - 08/24/2021 04:00 AM - Igor Skornyakov

Roger Borrello wrote:

> Igor, are you making progress on this?
>
> I received this today, only the Tag mismatch was caused by javax.crypto.AEADBadTagException instead of javax.net.ssl.SSLException. Does that make any difference?

Roger,
I'm still working on this. Fortunately, I can reliably reproduce the issue as Constantin suggested so I hope to fix it soon. T think that what you saw is due to the same reason than other similar issues,

#98 - 08/24/2021 05:24 PM - Igor Skornyakov

I can reliably reproduce the issue as suggested in #5388-86. At this moment I think that it is a concurrency issue, but it is unclear if it happens at the client side or at the server one. I plan to add the dump of the network traffic to the log to figure that. Hopefully, the overhead will not change the picture.

#99 - 08/25/2021 04:06 PM - Igor Skornyakov

The situation looks weird.
I've added encrypted packets' dump on both sides and I see that the packet that caused unwrap failure comes to the server-side exactly as created at the client-side. It is very unlikely that it was corrupted during encryption since I see the same even when I use a local wrap() method variable as the

encryption target.
The packet is also not corrupted during unwrapping (I dump it again after failure).
The error is essentially the same with different Oracle JDK/OpenJDK and Sun/BouncyCastle JCE/JSSE combinations (record MAC validation fails).
Continue investigation.

#100 - 08/26/2021 11:59 AM - Igor Skornyakov

The current state of the investigation is:

- The issue can be seen at the client-side as well.
- When adding logging to the send method the test typically runs much longer before encountering the problem.
- The problem is encountered with block ciphers operating both in Galois/counter mode (GCM) and Cipher block chaining (CBC) modes.

Since both GCM and CBC use the previous ciphertext block as an initialization vector for encrypting the next one, my guess is that somehow ciphertext blocks are sent not in the same order as encrypted (or such an inversion happens at the peer side on decryption).
Please note that this can happen only if multiple threads use the same SSL connection. In real apps, this can happen due to user activity (as was noticed by Roger). In Constantin's test, it is simulated at the server-side.

Continue investigation.

#101 - 08/26/2021 04:21 PM - Igor Skornyakov

Igor Skornyakov wrote:

> The current state of the investigation is:
>
> - The issue can be seen at the client-side as well.
> - When adding logging to the send method the test typically runs much longer before encountering the problem.
> - The problem is encountered with block ciphers operating both in Galois/counter mode (GCM) and Cipher block chaining (CBC) modes.
>
> Since both GCM and CBC use the previous ciphertext block as an initialization vector for encrypting the next one, my guess is that somehow ciphertext blocks are sent not in the same order as encrypted (or such an inversion happens at the peer side on decryption).
> Please note that this can happen only if multiple threads use the same SSL connection. In real apps, this can happen due to user activity (as was noticed by Roger). In Constantin's test, it is simulated at the server-side.
>
> Continue investigation.

After adding logic for tracking ciphertext blocks' order I indeed see that it is possible that blocks are processed in the incorrect order.
Working on the fix.

#102 - 08/30/2021 09:05 AM - Igor Skornyakov

The issue is (hopefully) fixed.
Committed to 3821c/12867.

I've retained additional logging and optional instrumentation that was added for the investigation.

- To enable hex dump of the network traffic the net:ssl:dump bootstrap config parameter should be set to "true". The default value is of course "false".
- To add tracking of the order in which the SSL packets are sent/received the net:ssl:trackSeqNo bootstrap config parameter should be set to "true".
- I've added an option to disable the Galois/counter mode for TLS ciphers via setting the net:ssl:useGCM bootstrap config parameter to "false". In theory, this can also reduce the encryption/decryption CPU consumption.

The fix was tested with the large customer application and with Constantin's stress test for all combinations of the Oracle JDK/OpenJDK and Sun/BouncyCastle JCE/JSSE. When testing Oracle/BouncyCastle I've experienced the known NPE issue (see e.g. https://githubmemory.com/repo/bcgit/bc-java/issues/921) and have upgraded BC version to 1.69 to resolve it. After that, I had to upgrade Oracle JDK to version 301 to resolve the problem with the cipher selection I've got with version 201.

#103 - 08/30/2021 10:23 AM - Roger Borrello

Igor, thank you for your efforts on this one. It certainly didn't look like an easy one to chase down! I'll get the customer's environment updated later today, so your fix can get exercised!

#104 - 08/30/2021 12:09 PM - Greg Shah

Roger: Is the customer using OpenJDK?

#105 - 08/30/2021 12:10 PM - Greg Shah

> When testing Oracle/BouncyCastle I've experienced the known NPE issue (see e.g. https://githubmemory.com/repo/bcgit/bc-java/issues/921) and have upgraded BC version to 1.69 to resolve it. After that, I had to upgrade Oracle JDK to version 301 to resolve the problem with the cipher selection I've got with version 201.

Did you find any similar BC issue with OpenJDK? I'm inclined to upgrade BC to the later level unless there are reasons to avoid it.

#106 - 08/30/2021 12:23 PM - Roger Borrello

The customer is using OpenJDK.

#107 - 08/30/2021 12:48 PM - Igor Skornyakov

Greg Shah wrote:

> When testing Oracle/BouncyCastle I've experienced the known NPE issue (see e.g. https://githubmemory.com/repo/bcgit/bc-java/issues/921) and have upgraded BC version to 1.69 to resolve it. After that, I had to upgrade Oracle JDK to version 301 to resolve the problem with the cipher selection I've got with version 201.

> Did you find any similar BC issue with OpenJDK? I'm inclined to upgrade BC to the later level unless there are reasons to avoid it.

If you're asking about NPE, I'm not sure, but It seems that it is the case - the problem happens deep inside BC code after a (relatively) long TLS conversation. The cipher selection issue with BC 1.69 was only with Oracle 1.8-201. No problems with OpenJDK (n my environment it is **build 1.8.0_292-8u292-b10**).

#108 - 08/31/2021 08:00 AM - Constantin Asofiei

Review for 3821c/12867

- ServerDriver.start:
  - the catch exceptions need to be on their own line
  - what happens if the algorithm can't be found? This most likely is a configuration issue, and IMO the server should not be allowed to start.
- RouterSessionManager$Incoming.run - you added a synchronized (this), but I don't understand why is it needed. There is always one single Incoming instance, handled on its own dedicated thread. Did you mean to synchronize on the RouterSessionManager instance?
- SSL.inpGuard and outGuard need javadoc.
- only SSL.unwrap and handshake are still synchronized. I can't tell if there is a problem or not:
  - unwrap uses the inpGuard r/w lock, and is in conflict with the synchronized mode. I assume this is because inpGuard is used in notify, which can be called from another thread (and notify is not synchronized).
  - both wrap and unwrap are called from handshake, which is synchronized. I see that inpGuard (for unwrap) is used in notify, too, which I assume can be called from a different thread. But outGuard, which is used in wrap, I don't think has any effect, as the call is already under the handshake synchronized lock.

#109 - 08/31/2021 01:07 PM - Igor Skornyakov

Constantin Asofiei wrote:

> Review for 3821c/12867
>
> - ServerDriver.start:
>   - the catch exceptions need to be on their own line

Fixed.

- what happens if the algorithm can't be found? This most likely is a configuration issue, and IMO the server should not be allowed to start.

The problem with the cipher selection always happens during the SSL handshake and it depends on the configuration of **both sides**. With this is mind I do not think that shutting down the server is a good solution since it can be a problem of the (miss)configuration of a particular client. Othe sessions can be already active.

- RouterSessionManager$Incoming.run - you added a synchronized (this), but I don't understand why is it needed. There is always one single Incoming instance, handled on its own dedicated thread. Did you mean to synchronize on the RouterSessionManager instance?

The synchronized (this) is actually not required. Removed.

- SSL.inpGuard and outGuard need javadoc.

Fixed.

- only SSL.unwrap and handshake are still synchronized. I can't tell if there is a problem or not:
  - unwrap uses the inpGuard r/w lock, and is in conflict with the synchronized mode. I assume this is because inpGuard is used in notify, which can be called from another thread (and notify is not synchronized).
  - both wrap and unwrap are called from handshake, which is synchronized. I see that inpGuard (for unwrap) is used in notify, too, which I assume can be called from a different thread. But outGuard, which is used in wrap, I don't think has any effect, as the call is already under the handshake synchronized lock.

The synchronized modifiers of the SSL.handshake and SSL.unwrap and BlockingSSL.send are removed.

The changes were tested with the stress test and a large customer app.

Committed to 3821c/12875.

#110 - 08/31/2021 01:13 PM - Constantin Asofiei

Igor Skornyakov wrote:

- what happens if the algorithm can't be found? This most likely is a configuration issue, and IMO the server should not be allowed to start.

> The problem with the cipher selection always happens during the SSL handshake and it depends on the configuration of **both sides**. With this is mind I do not think that shutting down the server is a good solution since it can be a problem of the (miss)configuration of a particular client. Othe sessions can be already active.

I mean this code in ServerDriver.start:

```
      if (bc.getBoolean("security", "bouncycastle", "use", false))
      {
         Security.insertProviderAt(BCHolder.JCE, 1);
         Security.insertProviderAt(BCHolder.JSSE, 2);
         try
         {
            KeyManagerFactory.getInstance("PKIX", BouncyCastleJsseProvider.PROVIDER_NAME);
            TrustManagerFactory.getInstance("PKIX", BouncyCastleJsseProvider.PROVIDER_NAME);
         }
         catch (NoSuchAlgorithmException | NoSuchProviderException e)
         {
            e.printStackTrace();
         }
      }
```

This code is executed **before** the FWD server is listening for connections, just when it tries to start. If the provider/algorithm can't be found, what happens? The server will continue to start, but will it behave OK?

So my point was to not allow the FWD server to start if catch (NoSuchAlgorithmException | NoSuchProviderException e) is reached.

#111 - 08/31/2021 01:23 PM - Greg Shah

> This code is executed **before** the FWD server is listening for connections, just when it tries to start. If the provider/algorithm can't be found, what happens? The server will continue to start, but will it behave OK?
>
> So my point was to not allow the FWD server to start if catch (NoSuchAlgorithmException | NoSuchProviderException e) is reached.

I'd rather have a safe default. Can't we fallback to security:bouncycastle:use=false if the BC ciphers aren't there?

#112 - 08/31/2021 01:29 PM - Igor Skornyakov

Greg Shah wrote:

> > This code is executed **before** the FWD server is listening for connections, just when it tries to start. If the provider/algorithm can't be found, what happens? The server will continue to start, but will it behave OK?
> >
> > So my point was to not allow the FWD server to start if catch (NoSuchAlgorithmException | NoSuchProviderException e) is reached.If
>
> I'd rather have a safe default. Can't we fallback to security:bouncycastle:use=false if the BC ciphers aren't there?

Sorry, what do you mean? If BC jars are in the classpath, some ciphers are defined and no exception will be thrown in the try block. If they are not then BCHolder initialization will fail before the try block and server start will fail. Please note that one has to explicitly enable using of BC. By default, we use JDK JCE/JSSE.

#113 - 08/31/2021 02:06 PM - Greg Shah

> If they are not then BCHolder initialization will fail before the try block and server start will fail. Please note that one has to explicitly enable using of BC. By default, we use JDK JCE/JSSE.

Why would we ever have the server fail if we can always safely use JDK JCE/JSSE? It doesn't matter if the user explicitly enabled it. If it can't work then we should have a safe fallback.

#114 - 08/31/2021 02:09 PM - Igor Skornyakov

Greg Shah wrote:

> If they are not then BCHolder initialization will fail before the try block and server start will fail. Please note that one has to explicitly enable using of BC. By default, we use JDK JCE/JSSE.

Why would we ever have the server fail if we can always safely use JDK JCE/JSSE? It doesn't matter if the user explicitly enabled it. If it can't work then we should have a safe fallback.

OK. I will think about how to implement this. This should be easy.

#115 - 08/31/2021 02:20 PM - Igor Skornyakov

Igor Skornyakov wrote:

> Greg Shah wrote:
>
> > If they are not then BCHolder initialization will fail before the try block and server start will fail. Please note that one has to explicitly enable using of BC. By default, we use JDK JCE/JSSE.
>
> Why would we ever have the server fail if we can always safely use JDK JCE/JSSE? It doesn't matter if the user explicitly enabled it. If it can't work then we should have a safe fallback.
>
> OK. I will think about how to implement this. This should be easy.

Greg,
Sorry. I think I've misunderstood the idea. With the properly built and deployed FWD BC jars always present.
Do you mean that we have to fall back to JDK JCR/JSSE if the handshake failed due to the missed cipher (this can be more tricky)?
Thank you.

#116 - 08/31/2021 02:46 PM - Greg Shah

> Do you mean that we have to fall back to JDK JCR/JSSE if the handshake failed due to the missed cipher (this can be more tricky)?

No.

I'm saying that we should back out the provider changes and safely allow the built-in JSSE to work if there is any failure during BC init.

> With the properly built and deployed FWD BC jars always present.

That is something that cannot be known from the Java code itself. We are dependent upon the runtime environment being setup properly and that is not guaranteed. If the BC jars are not there at runtime, then we would normally be able to work with the JSSE built-in support.

#117 - 08/31/2021 02:53 PM - Igor Skornyakov

Greg Shah wrote:

> Do you mean that we have to fall back to JDK JCR/JSSE if the handshake failed due to the missed cipher (this can be more tricky)?
>
> No.
>
> I'm saying that we should back out the provider changes and safely allow the built-in JSSE to work if there is any failure during BC init.
>
> With the properly built and deployed FWD BC jars always present.
>
> That is something that cannot be known from the Java code itself. We are dependent upon the runtime environment being setup properly and that is not guaranteed. If the BC jars are not there at runtime, then we would normally be able to work with the JSSE built-in support.

Got it. Will be done.

#118 - 09/01/2021 05:03 AM - Igor Skornyakov

Added check for the presence of the BouncyCastle providers.
Committed to 3821c/12879

I've realized that forgot to commit my changes to the build.gradle regarding the upgrade of the BouncyCastle to the version 1.69. Committed now.
Please do not forget to delete bc*-1.68.jar from deploy/lib.

#119 - 11/22/2021 11:55 AM - Greg Shah

Igor: What is the status of this task?

#120 - 11/22/2021 12:11 PM - Igor Skornyakov

Greg Shah wrote:

> Igor: What is the status of this task?

I understand that it is fixed.

#121 - 11/22/2021 02:24 PM - Greg Shah

- Status changed from New to Closed

- % Done changed from 0 to 100

Files

pk-store-log.txt	6.02 KB	06/10/2021	Sergey Ivanovskiy
create_certificates	4.94 KB	06/10/2021	Sergey Ivanovskiy
server.log	2.13 MB	06/22/2021	Roger Borrello