

**File Name: Exercise 4 Report**

**Student Name: Wuli Zuo**

**Student ID: a1785343**

**Vulnerabilities Assigned:**    **CVE-2018-11087**    **CVE-2018-11797**    **CVE-2017-5647**  
   **CVE-2013-2185,**    **CVE-2018-1295,**    **CVE-2014-0110**

**GitHub account: DoriaSummer**

**Link to the repository of Exercise 4:** [https://github.com/DoriaSummer/sse\\_exercise4](https://github.com/DoriaSummer/sse_exercise4)

In Exercise 4, Python and Git commands are used to analyse commits and identify VCCs. The Python scripts are mainly reused from Exercise 2 and 3, with the improvement of the method of identifying the enclosing scope, optimization of the code structure and modification to make it more suitable for Exercise 4. The file `identify.py` is used to find VCCs; `compare.py` is rewritten from the `analyse.py` of Exercise 2 and used to do the basic analyse of the three new vulnerabilities and compare the VCCs with the fixing commits; and `main.py` calls the functions of `git_identify()` and `git_compare()` in the other two files. Specific comments are included in the source code files. It is most recommended to read the code in Github to get the details.

In this report, answers to each question of Task 3, 4, 5 and relating explanations are included in Section 1, 2, 3, respectively. For the complete output for all the vulnerabilities, please refer to the file `Output.rtf`.

## **1. Task 3: Analysis of the 3 new vulnerabilities**

### **1.1 Case 4**

- CVE-ID: CVE-2013-2185 (duplicated with CVE-2013-4444)
- Link: <https://github.com/apache/tomcat>
- Fixing commit: e246e5fc13307da0a5d3bbf860d64d97be1c40f8

Note for this vulnerability: From the CVE webpage (Picture 1), Apache Tomcat team thinks this vulnerability is duplicated with CVE-2013-4444 and only released the fix of CVE-2013-4444. The following steps are taken for CVE-2013-4444.

CVE-ID	
<b>CVE-2013-2185</b>	<a href="#">Learn more at National Vulnerability Database (NVD)</a> • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
<b>** DISPUTED **</b> The readObject method in the DiskFileItem class in Apache Tomcat and JBoss Web, as used in Red Hat JBoss Enterprise Application Platform 6.1.0 and Red Hat JBoss Portal 6.0.0, allows remote attackers to write to arbitrary files via a NULL byte in a file name in a serialized instance, a similar issue to CVE-2013-2186. NOTE: this issue is reportedly disputed by the Apache Tomcat team, although Red Hat considers it a vulnerability. The dispute appears to regard whether it is the responsibility of applications to avoid providing untrusted data to be deserialized, or whether this class should inherently protect against this issue.	
References	
<b>Note:</b> <a href="#">References</a> are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"> <li>• MLIST:[oss-security] 20130905 Re: CVE-2013-2185 / Tomcat</li> <li>• <a href="http://www.openwall.com/lists/oss-security/2013/09/05/4">URL:http://www.openwall.com/lists/oss-security/2013/09/05/4</a></li> <li>• MLIST:[oss-security] 20141024 Re: Duplicate Request: CVE-2013-4444 as a duplicate of CVE-2013-2185</li> <li>• <a href="http://openwall.com/lists/oss-security/2014/10/24/12">URL:http://openwall.com/lists/oss-security/2014/10/24/12</a></li> <li>• REDHAT:RHSA-2013:1193</li> <li>• <a href="http://rhn.redhat.com/errata/RHSA-2013-1193.html">URL:http://rhn.redhat.com/errata/RHSA-2013-1193.html</a></li> <li>• REDHAT:RHSA-2013:1194</li> <li>• <a href="http://rhn.redhat.com/errata/RHSA-2013-1194.html">URL:http://rhn.redhat.com/errata/RHSA-2013-1194.html</a></li> <li>• REDHAT:RHSA-2013:1265</li> <li>• <a href="http://rhn.redhat.com/errata/RHSA-2013-1265.html">URL:http://rhn.redhat.com/errata/RHSA-2013-1265.html</a></li> </ul>	

Picture 1 Webpage for CVE-2013-2185

- a. CWE Type: CWE-20: Improper Input Validation
- b. Bug report: [http://mail-archives.apache.org/mod\\_mbox/www-announce/201409.mbox/%3C54105978.3060504@apache.org%3E](http://mail-archives.apache.org/mod_mbox/www-announce/201409.mbox/%3C54105978.3060504@apache.org%3E)

No public original bug report has been found in Bugzilla, but a confirmation mail list on 10/Sep/2014 can be found as Picture 2:

Subject	[SECURITY] CVE-2013-4444 Remote Code Execution in Apache Tomcat
Date	Wed, 10 Sep 2014 14:00:24 GMT
-----BEGIN PGP SIGNED MESSAGE----- Hash: SHA1	
CVE-2013-4444 Remote Code Execution	
Severity: Important	
Vendor: The Apache Software Foundation	
Versions Affected: - - Apache Tomcat 7.0.0 to 7.0.39	
Description: In very limited circumstances, it was possible for an attacker to upload a malicious JSP to a Tomcat server and then trigger the execution of that JSP. While Remote Code Execution would normally be viewed as a critical vulnerability, the circumstances under which this is possible are, in the view of the Tomcat security team, sufficiently limited that this vulnerability is viewed as important. For this attack to succeed all of the following requirements must be met: <ol style="list-style-type: none"> <li>a) Using Oracle Java 1.7.0 update 25 or earlier (or any other Java implementation where java.io.File is vulnerable to null byte injection).</li> <li>b) A web application must be deployed to a vulnerable version of Tomcat (see previous section).</li> <li>c) The web application must use the Servlet 3.0 File Upload feature.</li> <li>d) A file location within a deployed web application must be writeable by the user the Tomcat process is running as. The Tomcat security documentation recommends against this.</li> <li>e) A custom listener for JMX connections (e.g. the JmxRemoteListener that is not enabled by default) must be configured and be able to load classes from Tomcat's common class loader (i.e. the custom JMX listener must be placed in Tomcat's lib directory)</li> <li>f) The custom JMX listener must be bound to an address other than localhost for a remote attack (it is bound to localhost by default). If the custom JMX listener is bound to localhost, a local attack will still be possible.</li> </ol>	

Picture 2 Confirmation mail list of CVE-2013-4444

Apache Tomcat log page (Picture 2) also says that this issue was identified by Pierre Ernst of the VMware Security Engineering, Communications and Response group (vSECR) and reported to the Tomcat security team via the Pivotal security team on 5 September 2014. It was made public on 10 September 2014.

**Important: Remote Code Execution [CVE-2013-4444](#)**

In very limited circumstances, it was possible for an attacker to upload a malicious JSP to a Tomcat server and then trigger the execution of that JSP. While Remote Code Execution would normally be viewed as a critical vulnerability, the circumstances under which this is possible are, in the view of the Tomcat security team, sufficiently limited that this vulnerability is viewed as important.

For this attack to succeed all of the following requirements must be met:

1. Using Oracle Java 1.7.0 update 25 or earlier (or any other Java implementation where java.io.File is vulnerable to null byte injection).
2. A web application must be deployed to a vulnerable version of Tomcat.
3. The web application must use the Servlet 3.0 File Upload feature.
4. A file location within a deployed web application must be writeable by the user the Tomcat process is running as. The Tomcat security documentation recommends against this.
5. A custom listener for JMX connections (e.g. the JmxRemoteListener that is not enabled by default) must be configured and be able to load classes from Tomcat's common class loader (i.e. the custom JMX listener must be placed in Tomcat's lib directory).
6. The custom JMX listener must be bound to an address other than localhost for a remote attack (it is bound to localhost by default). If the custom JMX listener is bound to localhost, a local attack will still be possible.

Note that requirements 2 and 3 may be replaced with the following requirement:

7. A web application is deployed that uses Apache Commons File Upload 1.2.1 or earlier.

In this case (requirements 1, 4, 5, 6 and 7 met) a similar vulnerability may exist on any Servlet container, not just Apache Tomcat.

This was fixed in revision [1470437](#).

This issue was identified by Pierre Ernst of the VMware Security Engineering, Communications and Response group (vSECR) and reported to the Tomcat security team via the Pivotal security team on 5 September 2014. It was made public on 10 September 2014.

Affects: 7.0.0 to 7.0.39

Picture 2 Apache Tomcat log

c. Identify the fixing commit: Match

From the log of Apache Tomcat, a fix commit numbered 1479437 (Picture 2&4) can be found. Compare with the provided commit 1470435 (Picture 5&6), they share the same commit messages and are the same repair of the vulnerability for different versions. The provided commit is the fixing commit of the vulnerability CVE-2013-4444.

## Revision 1470437



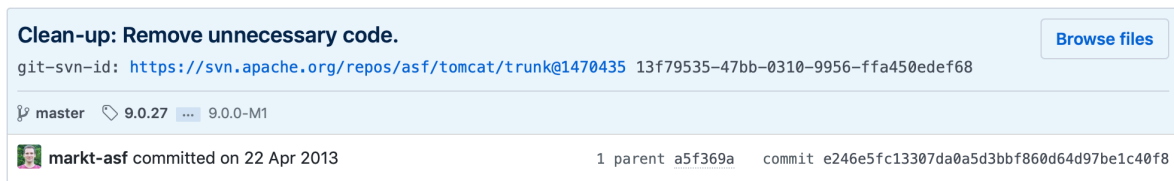
Jump to revision:

**Author:** markt  
**Date:** Mon Apr 22 10:35:10 2013 UTC (6 years, 6 months ago)  
**Changed paths:** 4  
**Log Message:** Clean-up: Remove unnecessary code.

### Changed paths

Path	Details
<a href="#">tomcat/tc7.0.x/trunk/</a>	<a href="#">modified</a> , props changed
<a href="#">tomcat/tc7.0.x/trunk/java/org/apache/tomcat/util/http/fileupload/</a>	<a href="#">modified</a> , props changed
<a href="#">tomcat/tc7.0.x/trunk/java/org/apache/tomcat/util/http/fileupload/FileItem.java</a>	<a href="#">modified</a> , <a href="#">text changed</a>
<a href="#">tomcat/tc7.0.x/trunk/java/org/apache/tomcat/util/http/fileupload/disk/DiskFileItem.java</a>	<a href="#">modified</a> , <a href="#">text changed</a>

Picture 4 Commit found: 1470473



Picture 5 Github page of the fixing commit provided



## Revision 1470435



Jump to revision:   ↩ ↲

**Author:** markt  
**Date:** Mon Apr 22 10:33:43 2013 UTC (6 years, 6 months ago)  
**Changed paths:** 2  
**Log Message:** Clean-up: Remove unnecessary code.

### Changed paths

Path	Details
 <a href="#">tomcat/trunk/java/org/apache/tomcat/util/http/fileupload/FileItem.java</a>	<a href="#">modified</a> , <a href="#">text changed</a>
 <a href="#">tomcat/trunk/java/org/apache/tomcat/util/http/fileupload/disk/DiskFileItem.java</a>	<a href="#">modified</a> , <a href="#">text changed</a>

Picture 6 Commit page of the commit 1470435

d. Affected files:

Output 1: Task 3.d, Case 4:

3.(d) Affected files in fixing commit: 2

java/org/apache/tomcat/util/http/fileupload/FileItem.java

java/org/apache/tomcat/util/http/fileupload/disk/DiskFileItem.java

They are relevant because the vulnerability is caused by using of the outdated java.io.File code, and the fixing is just to remove the outdated code (Picture 7).

3

java/org/apache/tomcat/util/http/fileupload/FileItem.java

@@	-20,7	+20,6	@@
20	20	import java.io.IOException;	
21	21	import java.io.InputStream;	
22	22	import java.io.OutputStream;	
23	-	import java.io.Serializable;	
24	23	import java.io.UnsupportedEncodingException;	
25	24		
26	25	/**	
@@	-46,7	+45,7	@@
46	45	* @version \$Id\$	
47	46	* @since 1.3 additionally implements FileItemHeadersSupport	
48	47	*/	
49	-	public interface FileItem extends Serializable, FileItemHeadersSupport {	
48	+	public interface FileItem extends FileItemHeadersSupport {	
50	49		
51	50	// ----- Methods from javax.activation.DataSource	
52	51		

61

java/org/apache/tomcat/util/http/fileupload/disk/DiskFileItem.java

@@	-24,8	+24,6	@@
24	24	import java.io.FileOutputStream;	
25	25	import java.io.IOException;	
26	26	import java.io.InputStream;	
27	-	import java.io.ObjectInputStream;	
28	-	import java.io.ObjectOutputStream;	
29	27	import java.io.OutputStream;	
30	28	import java.io.UnsupportedEncodingException;	
31	29	import java.util.Map;	

Picture 7 Fixing commit code of CVE-2013-4444

e. VCC:

Output 2: Task 3.e, Case 4:

Analyse repo: <https://github.com/apache/tomcat>

fixing commit: e246e5fc13307da0a5d3bbf860d64d97be1c40f8

3.(e) VCC: bcefe58374c

The parameter ‘-wC’ is used. The option ‘-w’ is to ignore whitespace, and the option ‘-C’ is to detect lines moved or copied from other files that were modified in the same commit, which is useful when the program is recognized and code is moved around across files. By testing all the parameters have with the file `test_blame.py`, the same VCC is given as the result. So there is no need to check the copies by adding more ‘C’, and ‘-wC’ is the most efficient way.

## 1.2 Case 5

- CVE ID: CVE-2018-1295
  - Link: <https://github.com/apache/ignite>
  - Fixing commit: 340569b8f4e14a4cb61a9407ed2d9aa4a20bdf49
- a. CWE Type: CWE-502: Deserialization of Untrusted Data
- b. Bug report: <https://lists.apache.org/thread.html/45e7d5e2c6face85aab693f5ae0616563132ff757e5a558da80d0209@%3Cdev.ignite.apache.org%3E>

CVE-2018-1295: Possible Execution of Arbitrary Code Within Deserialization Endpoints of Apache Ignite

Severity: Important

Vendor: The Apache Software Foundation

Versions Affected: Apache Ignite 2.3 or earlier

Impact:

An attacker can execute arbitrary code on Ignite nodes in the case when Ignite classpath contains arbitrary vulnerable classes.

Description:

Apache Ignite serialization mechanism does not have a list of classes allowed for serialization/deserialization, which makes it possible to run arbitrary code when 3-rd party vulnerable classes are present in Ignite classpath. The vulnerability can be exploited if the one sends a specially prepared form of a serialized object to one of the deserialization endpoints of some Ignite components - discovery SPI, Ignite persistence, Memcached endpoint, socket steamer.

Mitigation:

- All Ignite versions: make sure there are no vulnerable classes among your custom code used in Apache Ignite.
- Ignite 2.3 or earlier users: upgrade to Ignite 2.4 and use `IGNITE_MARSHALLER_WHITELIST` and/or `IGNITE_MARSHALLER_BLACKLIST` system properties to define classes allowed for deserialization

Credit:

The vulnerability was discovered by Man Yue Mo of lgtm.com.

References:

- \* <http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-1295>

Picture 8 Mail including mitigation of the vulnerability

The above link and Picture 8 is a confirmation mail list from Ignite. The Original bug report may have been deleted or not reported publicly and is not found in JIRA.

From the CVE webpage (Picture 8), it can be told that this vulnerability was first reported around 07/Dec/2017 and fixed around 02/Arp/2018. Different conditions are used to search for the bug report in JIRA (Picture 9&10) as well as in Google, but no bug report relating to this vulnerability is found around and between the above two dates. I even tried to open the report link with the number 6643 found in the provided fixing commit, but the page showed ‘deleted or no permission’ (Picture 11), which indicates that the original bug report was deleted or not reported publicly.

CVE-ID	
<b>CVE-2018-1295</b>	<a href="#">Learn more at National Vulnerability Database (NVD)</a>
<a href="#">CVSS Severity Rating</a> • <a href="#">Fix Information</a> • <a href="#">Vulnerable Software Versions</a> • <a href="#">SCAP Mappings</a> • <a href="#">CPE Information</a>	
Description	
In Apache Ignite 2.3 or earlier, the serialization mechanism does not have a list of classes allowed for serialization/deserialization, which makes it possible to run arbitrary code when 3-rd party vulnerable classes are present in Ignite classpath. The vulnerability can be exploited if the one sends a specially prepared form of a serialized object to one of the deserialization endpoints of some Ignite components - discovery SPI, Ignite persistence, Memcached endpoint, socket steamer.	
References	
<b>Note:</b> References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"> <li>BID:103692</li> <li>URL:<a href="http://www.securityfocus.com/bid/103692">http://www.securityfocus.com/bid/103692</a></li> <li>MLIST:[dev]<a href="#">[20180402]</a>[CVE-2018-1295]: Possible Execution of Arbitrary Code Within Deserialization Endpoints of Apache Ignite</li> <li>URL:<a href="https://lists.apache.org/thread.html/45e7d5e2c6face85aab693f5ae0616563132ff757e5a558da80d0209@%3Cdev.ignite.apache.org%3E">https://lists.apache.org/thread.html/45e7d5e2c6face85aab693f5ae0616563132ff757e5a558da80d0209@%3Cdev.ignite.apache.org%3E</a></li> <li>REDHAT:RHSA-2018:2405</li> <li>URL:<a href="https://access.redhat.com/errata/RHSA-2018:2405">https://access.redhat.com/errata/RHSA-2018:2405</a></li> </ul>	
Assigning CNA	
Apache Software Foundation	
Date Entry Created	
<b>20171207</b>	Disclaimer: The <a href="#">entry creation date</a> may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Picture 8 Webpage for CVE-2018-1295

Search

Save as

✓ project = Ignite AND text ~ serialization ORDER BY createdDate DESC

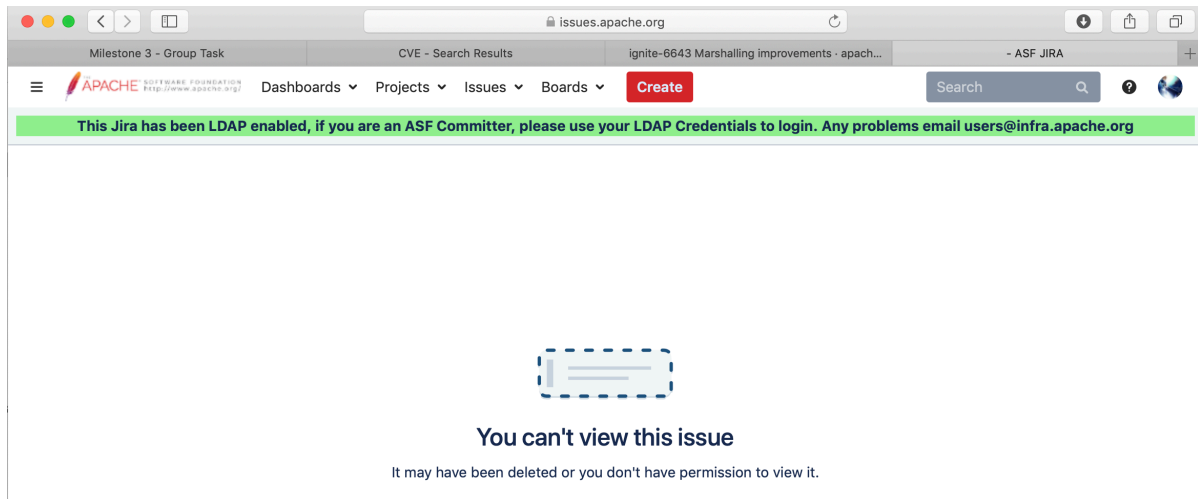
Picture 9 Search condition in JIRA – example condition 1

Search

Save as

✓ project = Ignite AND affectedVersion = 2.3 AND fixVersion =2.4 ORDER BY createdDate DESC

Picture 10 Search condition in JIRA – example condition 2



Picture 11 JIRA Page <https://issues.apache.org/jira/browse/IGNITE-6643>

c. Identify the fixing commit: Match

By reading the mail from the vendor (Picture 8) and the provided fixing commit (Picture 13), it is clear that the mitigation of this vulnerability is to use `IGNITE_MARSHALLER_WHITELIST` and/or `IGNITE_MARSHALLER_BLACKLIST` system properties to define classes allowed for deserialization, and the provided fixing commit just does this fix. It is believed that the provided fixing commit is the corresponding fixing commit to the vulnerability.

File	Line	Code
	@@ -524,6 +508,12 @@	
524	508	<code>public static final String IGNITE_BINARY_MARSHALLER_USE_STRING_SERIALIZATION_VER_2 =</code>
525	509	<code>"IGNITE_BINARY_MARSHALLER_USE_STRING_SERIALIZATION_VER_2";</code>
526	510	
511	+	<code>/** Defines path to the file that contains list of classes allowed to safe deserialization.*/</code>
512	+	<code>public static final String IGNITE_MARSHALLER_WHITELIST = "IGNITE_MARSHALLER_WHITELIST";</code>
513	+	
514	+	<code>/** Defines path to the file that contains list of classes disallowed to safe deserialization.*/</code>
515	+	<code>public static final String IGNITE_MARSHALLER_BLACKLIST = "IGNITE_MARSHALLER_BLACKLIST";</code>
516	+	
527	517	<code>/**</code>
528	518	<code>* If set to <code>{@code true}</code>, then default selected keys set is used inside</code>
529	519	<code>* <code>{@code GridNioServer}</code> which lead to some extra garbage generation when</code>

Picture 12 Github page of the fixing commit for CVE-2018-1295

d. Affected files:

Output 3: Task 3.d, Case 5:

3.(d) Affected files in fixing commit: 35

`modules/core/src/main/java/org/apache/ignite/IgniteSystemProperties.java`



---

modules/core/src/main/java/org/apache/ignite/internal/ClassSet.java

modules/core/src/main/java/org/apache/ignite/internal/GridKernalContextImpl.java

modules/core/src/main/java/org/apache/ignite/internal/IgniteKernal.java

modules/core/src/main/java/org/apache/ignite/internal/MarshallerContextImpl.java

modules/core/src/main/java/org/apache/ignite/internal/client/marshaller/optimized/GridClientOptimizedMarshaller.java

modules/core/src/main/java/org/apache/ignite/internal/marshaller/optimized/OptimizedMarshallerUtils.java

modules/core/src/main/java/org/apache/ignite/internal/marshaller/optimized/OptimizedObjectInputStream.java

modules/core/src/main/java/org/apache/ignite/internal/marshaller/optimized/OptimizedObjectOutputStream.java

modules/core/src/main/java/org/apache/ignite/internal/processors/cache/persistence/wal/reader/StandaloneGridKernalContext.java

modules/core/src/main/java/org/apache/ignite/internal/processors/platform/utils/PlatformUtils.java

modules/core/src/main/java/org/apache/ignite/internal/processors/rest/protocols/tcp/GridTcpRestParser.java

modules/core/src/main/java/org/apache/ignite/internal/processors/rest/protocols/tcp/GridTcpRestProtocol.java

modules/core/src/main/java/org/apache/ignite/internal/util/IgniteUtils.java

modules/core/src/main/java/org/apache/ignite/marshaller/MarshallerContext.java

modules/core/src/main/java/org/apache/ignite/marshaller/MarshallerUtils.java

modules/core/src/main/java/org/apache/ignite/marshaller/jdk/JdkMarshaller.java

modules/core/src/main/java/org/apache/ignite/marshaller/jdk/JdkMarshallerObjectInputStream.java

modules/core/src/main/java/org/apache/ignite/spi/discovery/tcp/TcpDiscoverySpi.java

modules/core/src/main/java/org/apache/ignite/stream/StreamAdapter.java

modules/core/src/main/java/org/apache/ignite/stream/socket/SocketStreamer.java

modules/core/src/test/config/class\_list\_exploit\_excluded.txt

modules/core/src/test/config/class\_list\_exploit\_included.txt

modules/core/src/test/java/org/apache/ignite/internal/ClassSetTest.java

---

---

```
modules/core/src/test/java/org/apache/ignite/internal/MarshallerContextLockingSelfTest.java
modules/core/src/test/java/org/apache/ignite/internal/binary/GridBinaryMarshallerCtxDisabledSelfTest.java
modules/core/src/test/java/org/apache/ignite/internal/processors/cache/GridCacheEntryMemorySizeSelfTest.java
modules/core/src/test/java/org/apache/ignite/marshaller/MarshallerContextSelfTest.java
modules/core/src/test/java/org/apache/ignite/marshaller/MarshallerContextTestImpl.java
modules/core/src/test/java/org/apache/ignite/spi/discovery/tcp/DiscoveryUnmarshalVulnerabilityTest.java
modules/core/src/test/java/org/apache/ignite/stream/socket/SocketStreamerUnmarshalVulnerabilityTest.java
modules/core/src/test/java/org/apache/ignite/testframework/junits/GridTestKernalContext.java
modules/core/src/test/java/org/apache/ignite/testsuites/IgniteBasicTestSuite.java
modules/core/src/test/java/org/apache/ignite/testsuites/IgniteSpiDiscoverySelfTestSuite.java
modules/core/src/test/java/org/apache/ignite/testsuites/IgniteStreamSelfTestSuite.java
```

---

The vulnerability is caused by that the serialization mechanism does not have a list of classes allowed for serialization/deserialization. As stated in Question c, the mitigation is stated as to use `IGNITE_MARSHALLER_WHITELIST` and/or `IGNITE_MARSHALLER_BLACKLIST` system properties to define classes allowed for deserialization, which is just what the fixing commit does (Picture 13). That is why they are relevant.

e. VCC:

Output 4: Task 3.e, Case 5:

---

Analyse repo: <https://github.com/apache/ignite>

fixing commit: 340569b8f4e14a4cb61a9407ed2d9aa4a20bdf49

3.(e) VCC: bdbba0ee9a5

---

The parameter '-wC' is used. The option '-w' is to ignore whitespace, and the option '-C' is to detect lines moved or copied from other files that were modified in the same commit, which is useful when the program is recognized and code is moved around across files. By testing all the parameters have with the file `test_blame.py`, the same VCC is given as the result. So there is no need to check the copies by adding more 'C', and '-wC' is the most efficient way.

### 1.3 Case 6

- CVE-ID: CVE-2014-0110
  - Link: <https://github.com/apache/cxf>
  - Fixing commit: 8f4799b5bc5ed0fe62d6e018c45d960e3652373e
- a. CWE Type: CWE-399: Resource Management Errors
- b. Bug report: <http://cxf.apache.org/security-advisories.data/CVE-2014-0110.txt.asc?version=1&modificationDate=1398873378628&api=v2>

CVE-2014-0110: Large invalid content could cause temporary space to fill

Severity: Major

Vendor: The Apache Software Foundation

Versions Affected:

This vulnerability affects all versions of Apache CXF prior to 2.6.14 and 2.7.11.

Description:

If a SOAP message generates a fault on parsing or processing, but is not fully consumed, it is possible to cause the server to read all of the remaining data and to save it to a temp file. By dynamically creating data, you can cause the entire /tmp directory to fill.

This has been fixed in revisions:

<https://git-wip-us.apache.org/repos/asf?p=cxf.git;a=commit;h=8f4799b5bc5ed0fe62d6e018c45d960e3652373e>

Migration:


CXF 2.6.x users should upgrade to 2.6.14 or later as soon as possible.  
CXF 2.7.x users should upgrade to 2.7.11 or later as soon as possible.

Picture 13 Confirmation mail list from CXF


The above link and Picture 13 is a confirmation mail list from CXF. The Original bug report may not exist or not be reported publicly in JIRA.

Similar situation as Case 5, no relating bug report can be found from JIRA. From the log page of CXF, it can be told that their commits usually contain an bug report number (Picture 14), but the fixing commit for this vulnerability does not contain such [CXF-xxxx] number (Picture 15), so it can be inferred that the bug report may not exist or the bug may not be reported publicly.

**5 years ago**      [\[CXF-5639\] Minor updates to StreamingResponseProvider](#)


[commit](#) | [commitdiff](#) | [tree](#)  
Sergey Beryozkin [Tue, 25 Mar 2014 21:45:53 +1030 (11:15 +0000)]   
[CXF-5639] Minor updates to StreamingResponseProvider

**5 years ago**      [\[CXF-5639\] Introducing StreamingResponse, can be used with/without WebSocket](#)

[commit](#) | [commitdiff](#) | [tree](#)  
Sergey Beryozkin [Tue, 25 Mar 2014 21:29:23 +1030 (10:59 +0000)]   
[CXF-5639] Introducing StreamingResponse, can be used with/without WebSocket

Picture 14 CXF log page: commits with a bug report usually contain the [CXF-xxxx] number

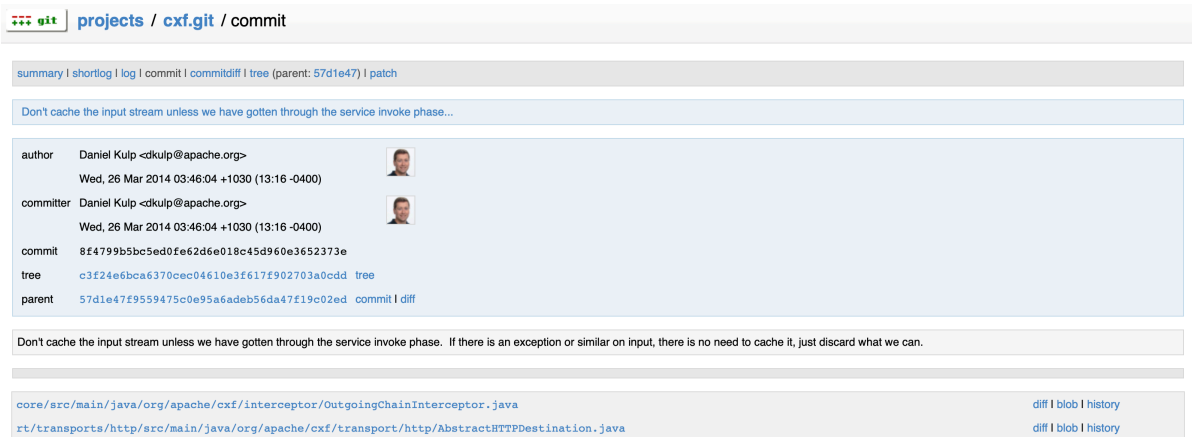
**5 years ago**      [Don't cache the input stream unless we have gotten through the service invoke phase...](#)

[commit](#) | [commitdiff](#) | [tree](#)  
Daniel Kulp [Wed, 26 Mar 2014 03:46:04 +1030 (13:16 -0400)]   
Don't cache the input stream unless we have gotten through the service invoke phase. If there is an exception or

Picture 15 CXF log page: the fixing commit for CVE-2014-0110 does not contain a [CXF-xxxx] number

- c. Identify the fixing commit: Match

From the confirmation mail list from CXF (Picture 13), a fixing link (Picture 17) can be found, and it is the same as the provided one.



d. Affected files:

3.(d) Affected files in fixing commit: 2

The vulnerability is described as that if large invalid content is not fully consumed, the remaining data could cause temporary space to fill. The commit fix this vulnerability by limit the caching of the input stream only when having gotten through the service invoke phase. They are relevant.

Output 6: Task 3.e, Case 6:

fixing commit: 8f4799b5bc5ed0fe62d6e018c45d960e3652373e

The parameter ‘-wC’ is used. The option ‘-w’ is to ignore whitespace, and the option ‘-C’ is to detect lines moved or copied from other files that were modified in the same commit, which is useful when the program is recognized and code is moved

around across files. By testing all the parameters have with the file `test_blame.py`, the same VCC is given as the result. So there is no need to check the copies by adding more 'C', and '-wC' is the most efficient way.

## **2. Task 4: Analysis of each pair of the VCC and the fixing commit**

Since the output of the six pairs of the VCC and the fixing commit share some similarities, this report first explains the phenomenon in general and then shows the output of each case in Section 2.1-2.6.

- a. Is the current VCC an initial commit?

By trying to show the log of the previous commit of the VCC, found that none of the VCCs is an initial commit, because all the previous commits exist.

- b. Is the developer of the current fixing commit and its corresponding VCC the same? If not, give your reflections on the difference in their experience.

In Case 3 and Case 4, the VCC and the fixing commit sharing the same developer, all the others have different developers of the VCC and the fixing commit. Usually, the developer of the fixing commit is more experienced (three times to tens or hundreds of times more) than the developer of the VCC, which can be told from the numbers of their commits in the repositories. The only exception is Case 5 with the developer of the VCC is more experienced than the one who fixed it. We can say that usually the vulnerability is fixed by the same developer who is responsible for it or someone who is more experienced, but it is not necessary because the time of the commit of VCC and fixing may be long enough to allow personnel changes of the development team.

- c. How many days were between the current fixing commit and its corresponding VCC? Is the current VCC fixed immediately (within the same day)?

The time between the fixing commits and their corresponding VCCs vary between two hundred days and ten years. None of the VCCs is fixed immediately.

- d. If the current VCC is not fixed immediately (within the same day), was there any other vulnerability, bug or special request happening between the current fixing commit and VCC that might require higher priority? Include your detailed reflections on why it had remained unfixed.

By checking 10 commits follow the VCC to see if there was any special events. The vulnerabilities, bugs or special requests that may have caused the delay of the fixing is highlighted.

There is another speculation for the project with no commits identified as special events (Case 4) that it may not be a single event delayed the fixing for such long time. Consider that these vulnerabilities are issues relating to security, the developers usually would not allow it to be left there once they are aware of them. The reasonable explanation for such long gaps in between may be that, although the vulnerabilities were caused by some quite early versions of codes, but they had not been triggered or found until some new functionalities were added.

## 2.1 Case 1

- CVE-ID: CVE-2018-11087
- Link: <https://github.com/spring-projects/spring-amqp>
- Fixing Commit: 444b74e95bb299af5e23ebf006fbb45d574fb95 (the given commit)
- VCC: 127d6aabc

Output 7: Task 4.a-4.d, Case 1:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Zachary DeLuca, Number of commits has made: 1

The developer of the fixing commit: Gary Russell, Number of commits has made: 623

--> Different developers.

---

---

4.(c) The number of days between the VCC and the fixing commit: 284.94

The VCC is not fixed immediately.

4.(d) Commits after the VCC:

1ddc74f233fdb2d4ba0322fea7e17aa32891fe24 AMQP-784: Fix multi method @RabbitListener

1d4b99c4bcad5a8ebcf569270d545b112af01cc1 AMQP-776: More Consumer Events

9fbf48a989afd6628712235651932409b54da415 Fix Event Test

1abc07e767f92e10624432aea7a5423e2fcdd110 AMQP-785: SMLC Lifecycle fixes

cc138bbde8fb0db4d3901045b08dc07a954a6d61 AMQP-789: MPP Javadoc polishing

dec63b11a79bc16daeb75409c5e9711ccf97d551 [artifactory-release] Release version

## 2.0.1.RELEASE

895c7b04683c64ca7a308ccd52d4172a27d4d44f [artifactory-release] Next development version

86eb43f683f92c1f2fcfaabb5db47b908ee47c0e Fix Sonar false positives

dc0bbc4104ee8400d2b5ddb8042e536e843ffade More Sonar Polishing

c576b274935c40d9c34266137cacd67287f1860c AMQP-790: Fix after receive MPPs with

send/receive

---

## 2.2 Case 2

- CVE-ID: CVE-2018-11797
- Link: <https://github.com/apache/pdfbox>
- Fixing Commit: 4fa98533358c106522cd1bfe4cd9be2532af852
- VCC: 0043363995

Output 8: Task 4.a-4.d, Case 2:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Jukka Zitting, Number of commits has made: 236

The developer of the fixing commit: Tilman Hausherr, Number of commits has made:

3853

--> Different developers.

4.(c) The number of days between the VCC and the fixing commit: 3652.11

The VCC is not fixed immediately.

---



---

4.(d) Commits after the VCC:

dc8ef25fa1a6f9de1c73e3c3479037332a5d6950 PDFBOX-2: Migrate PDFBox sources to Apache

d18984f70b1197cbf53e2e3f6472018b26ced7b5 PDFBOX-2: Migrate PDFBox sources to Apache

d8915013d5466c6b6a5e2584bcc64e8ce469be7d PDFBOX-2: Migrate PDFBox sources to Apache

623a0af377233776f57213cf74b1a909c23272 Set svn:eol-style, remove trailing spaces

5392a1d4ea19beb4aecdd0d8d385102d1b408cd44 Set svn:eol-style, remove trailing spaces

831055345926ef7eb987ed169de1a36eceb09755 Set svn:eol-style, remove trailing spaces

e7b411f94a48ee1c24bc8ce5d9382b85f7a519ed PDFBOX-364: Resolved use of auto-boxing to

restore Java 1.4 compatibility.

5657e8819eb9c0ec8cd66e15ae8d220db037cc2e First batch of changes to prepare the website.

38e6ec5bf11daefc26e73180c57d31133f230ab2 Make site deployable via ForrestBot. See README.txt for instructions.

c4c86bb2078cc309454075ddc0f85a0547602e27 Absolute path names won't work since the PDFBox website isn't deployed at the root level of a subdomain.

---

## 2.3 Case 3

- CVE-ID: CVE-2017-5647
- Link: <https://github.com/apache/tomcat80>
- Fixing Commit: ec10b8c785d1db91fe58946436f854dde04410fd
- VCC: a89540b76d

Output 9: Task 4.a-4.d, Case 3:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Mark Thomas, Number of commits has made: 10186

The developer of the fixing commit: Mark Thomas, Number of commits has made: 10186

--> The same developer.

4.(c) The number of days between the VCC and the fixing commit: 746.36

The VCC is not fixed immediately.

4.(d) Commits after the VCC:

---

---

ccc6f78aec6b7ea08f6bd5379cd7dbbf2e8ac85 Make processing of trailer headers for chunked input optional. Trailer headers to process must be added to the allowedTrailerHeader list or they will be ignored

035b6670d984dc6d4c811d3798d6d741a65d414c Treat \*.pem files (used in Tomcat test cases) as text. I set svn:eol-style and update "text.files" patterns in build.xml.

bb8cf97101027c2f766b8f28f9be24d2864d0ec1 Update link to SPNEGO documentation in Apache Geronimo wiki The old link went 404, as reported by André Warnier on the users@ list.

ac1b8b072ed66700fbad1aeb5f9cd442c04fadb7 Defer the initialization of StartTLS settings to the startInternal stage. That way logging is possible and errors in tls/ssl will lead to stopping the context.

7a2eb950cfdccb4bc42433abe76bd6da299810e3 Use the right variable for null check; the method variable, that is.

2cba2ad4cacbb5197e57360dd4fd2b7f5aa9aade Fix trailing space in comment lines in \*.pem files, to make checkstyle happy. Followup to r1666503.

3d1a2bb161686246caca796d4fdcfcea9fce23ba2 Merged revision 1666637 from tomcat/trunk: Unused type parameter T

4210f7699291523d8de3e7ddaf583122053915b7 Fix [https://bz.apache.org/bugzilla/show\\_bug.cgi?id=57704](https://bz.apache.org/bugzilla/show_bug.cgi?id=57704) Merged revision 1666649 from tomcat/trunk: Access instanceManager via get/set methods. Fix potential NPEs. In web app start if a problem occur prior to instanceManager initialization then: - SCI.onStart will fail if it tries to use instanceManager - During web app stop, StandardContext.listenerStop will fail if it tries to use instanceManager

88355985104acfc2ed726f805dbb111645006906 Fix [https://bz.apache.org/bugzilla/show\\_bug.cgi?id=57377](https://bz.apache.org/bugzilla/show_bug.cgi?id=57377) Remove the restriction that prevented the use of SSL when specifying a bind address. Enable SSL to be configured for the registry as well as the server.

92baa3317df474506833e5e601fdb8256f2e48bf Use the correct capitalisation.

---

## 2.4 Case 4

- CVE-ID: CVE-2013-2185
- Link: <https://github.com/apache/tomcat>
- Fixing commit: e246e5fc13307da0a5d3bbf860d64d97be1c40f8
- VCC: bcefe58374c

Output 10: Task 4.a-4.d, Case 4:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Mark Emlyn David Thomas, Number of commits has made: 6483

The developer of the fixing commit: Mark Emlyn David Thomas, Number of commits has made: 6483

--> The same developer.

4.(c) The number of days between the VCC and the fixing commit: 1251.59

The VCC is not fixed immediately.

4.(d) Commits after the VCC:

5e7ae383f189e2409febfeb9c1f8196a47891956 Test some edge cases with = in cookie values (current code fails these tests - patch to follow shortly)

01d825bf74491435c1a55589bcd4d8bd3961af1c Better handle edge cases when allowing = in cookie value

4ecee94347289d7feff25329a541f62dc82ff01b Fix an Eclipse warning

771ffdbe3de52db374ba63c04ebb2e7f19f102b0 Add support for multipart config in web.xml (partially complete) Review and fix issues in WebXml merge code

576bf2bb753784985f100fb7a25f25f63c344ae8 Remove deprecated code

30131cb0480d63e911c1f7912a4e45687d2636ab Remove some more deprecated code

16db530b7af9969670938b6ab6aba319b7ce6f88 Remove unused imports

d750f01655152f35f1d738e6a5b7f52282ccde06 Add @Override annotations

8b22089526cad157662152cd5a34eb3dfd0df860 Consolidation of protocol attributes into a base class

b5c8c1ce5a3e1f000c400257a8de65d6d9f95bc5 Generics

---

## 2.5 Case 5

- CVE ID: CVE-2018-1295
- Link: <https://github.com/apache/ignite>
- Fixing commit: 340569b8f4e14a4cb61a9407ed2d9aa4a20bdf49
- VCC: bdbba0ee9a5

Output 11: Task 4.a-4.d, Case 5:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Dmitriy Govorukhin, Number of commits has made: 368

The developer of the fixing commit: Andrey Gura, Number of commits has made: 43

--> Different developers.

4.(c) The number of days between the VCC and the fixing commit: 227.61

The VCC is not fixed immediately.

4.(d) Commits after the VCC:

6bf5ce46c1e6c4e3dcf042ac91a5b61a726c5821 IGNITE-5267 - Moved ignite-ps module to ignite-core

4dc81ca86347107848328d1e2e206b796976fb23 IGNITE-5267 - Added missing class

ca6cd142d43a47399c51a586d43c8eab6f82163d ignite-5272 Do not always disable one-phase commit if there is near cache, instead switch to non one-phase mode if there are readers.

0879ca88553ac1299fe21bd1e60cf64e3edf4096 IGNITE-5414 Fixed cluster topology serialization.

b843f078fa97c6a34dd4bb375c534bfa798bcda2 IGNITE-5460 Added cache group name.

195147d573d7cb3fc637f74937ee561b03a3c574 IGNITE-5267 - Activate nodes after start

d4f19e7f3977332c1e64465310627167b39a0bdf Fixed failure in IgniteCacheNoClassQuerySelfTest.

4df693704eddfeea2899ec7bf6db21304116f648 .NET: Fix TestAsciiChars

d38dfcd5c4a49b6ec4be1cfdb4f28f7d152cd14a ignite-5267-merge remove pds dependency, restore data structures only in snapshot operation

cf886fd5816534b46a1247d3bdc41061087608e4 ignite-2.1.1 fix java doc

---

## 2.6 Case 6

- CVE-ID: CVE-2014-0110
- Link: <https://github.com/apache/cxf>
- Fixing commit: 8f4799b5bc5ed0fe62d6e018c45d960e3652373e
- VCC: be286c70c1e

Output 12: Task 4.a-4.d, Case 6:

---

4.(a) Previous commit of the VCC found, the current VCC is not an initial commit.

4.(b) Developer(s):

The developer of the VCC: Sergey Beryozkin, Number of commits has made: 1899

The developer of the fixing commit: Daniel Kulp, Number of commits has made: 3388

--> Different developers.

4.(c) The number of days between the VCC and the fixing commit: 753.18

The VCC is not fixed immediately.

4.(d) Commits after the VCC:

a4209e39c7299839e6517afb51842688070d7b8a [CXF-4153] Fixing Beanspector wrongly

getting the property names in case of 'is'

35a39a7b1d1b6109b02b7dbe44f60c3151394e16 [CXF-4153] Adding the missing test file

efbe607e0f1d5cfb477f1b38296d90149f38f77a Do not set passwords to the empty String if they are null. - This is causing the keystore.private.password property of WSS4J to not work

f133afbfe582ac0451d601f8b12da64cb5742f32 [CXF-4143] Make the ending interceptor public so the name can easily be accessed.

37fe50cd5d7c5a9f3d5c0ab2724926fed1466c56 [CXF-4136] Handle wsdl:s not as a dependency  
Patch from Owen Farrell applied

eb44c422d0b003d8a7e0d197c77558123cf1c9cf [CXF-4131] Make sure the conduit is only added as a listener once

dc90e3449f2ea219796eaedbffe9fba925bed7b2 [CXF-4130] Fix problems with Providers  
writing bodies into headers Patch from Seumas Soltysik applied

da3ea3cb722a8c32a89b5a103ad2346fbf9e61cf Upgrading to WSS4J 1.6.5

5511da825cf956f803f8acda6f54bee5b80d6df5 [CXF-4160] - Support signing a SAML token  
using the requested signature and canonicalization algorithm

faadb5887fa92e72f3ce8c103620918421341548 [CXF-3589] Prototyping the skeleton code  
for supporting saml web sso on the SP side based on Colm's test

---

### 3. Task 5: Assessment of each vulnerability

Each vulnerability is assessed with CVSS 2.0 and 3.0. The metrics and base scores by this exercise and NVD are included in Section 3.1-3.6, with the relating explanation and comparison.

#### 3.1 Case 1: CVE-2018-11087

- CVSS 2.0

Table 1 CVE-2018-11087 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Remote attack can be exploited through network
		AC	H	H	Man in the middle attack related
		AU	N	N	No authentication is required
	Impact metrics	C	P	P	Data in transit can be viewed
		I	N	N	No Integrity is violated
		A	N	N	No Availability is violated
Base Score			4.3	4.3	Same

- CVSS 3.0

Table 2 CVE-2018-11087 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Remote attack can be exploited through network
		AC	H	H	Man in the middle attack related
		PR	N	N	No authorization is required
		UI	N	N	No interaction is required
	Scope		U	U	The scope is not changed
	Impact metrics	C	H	H	Data in transit can be viewed
		I	N	N	No Integrity is violated
		A	N	N	No Availability is violated
Base Score			5.9	5.9	Same

### 3.2 Case 2: CVE-2018-11797

- CVSS 2.0

Table 3 CVE-2018-11797 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Remote attack can be exploited through network
		AC	M	M	A carefully crafted PDF file is required
		AU	N	N	No authentication needed
	Impact metrics	C	N	N	No Confidentiality is violated
		I	N	N	No Integrity is violated
		A	M	M	A long running computation can be triggered
Base Score			4.3	4.3	Same

- CVSS 3.0

Table 4 CVE-2018-11797 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	L	Remote attack can be exploited through network
		AC	M	M	A carefully crafted PDF file is required
		PR	N	N	No authorization is required
		UI	R	R	A carefully crafted PDF file is required to be uploaded
	Scope		U	U	The scope is not changed
	Impact metrics	C	N	N	No Confidentiality is violated
		I	N	N	No Integrity is violated
		A	H	H	A long running computation is triggered
Base Score			6.5	5.5	The worse attack vector is chosen as 'N', because the vulnerability allows remote attack through network. I have no idea why NVD chose 'L' here, while 'N' is chosen in CVSS 2.0

### 3.3 Case 3: CVE-2017-5647

- CVSS 2.0

Table 5 CVE-2017-5647 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Wrong request sent when uploading files, network related
		AC	L	L	User request could trigger this bug, low complexity for attackers
		AU	N	N	No authentication is required
	Impact metrics	C	P	P	Responds could be sent to a wrong user
		I	N	N	No Integrity is violated
		A	N	N	No Availability is violated
	Base Score		5.0	5.0	Same

- CVSS 3.0

Table 6 CVE-2017-5647 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Wrong request sent when uploading files, network related
		AC	L	L	User request could trigger this bug, low complexity for attackers
		PR	N	N	No authorization is required
		UI	N	N	No interaction is required
	Scope		U	U	The scope is not changed
	Impact metrics	C	H	H	Responds could be sent to a wrong user
		I	N	N	No Integrity is violated
		A	N	N	No Availability is violated
	Base Score		7.5	7.5	Same



### 3.4 Case 4: CVE-2013-4444 (Assess CVE-2013-4444 instead of CVE-2013-2185)

- CVSS 2.0

Table 7 CVE-2013-4444 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	Remote attack can be exploited by uploading files
		AC	M	M	Arbitrary code can be executed by uploading and accessing a JSP file, medium complexity
		AU	N	N	No authentication is required
	Impact metrics	C	P	P	Arbitrary code can be executed, so
		I	P	P	Confidentiality, Integrity and Availability are all
		A	P	P	violated
Base Score			6.8	6.8	Same

- CVSS 3.0

Table 8 CVE-2013-4444 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N/A	Remote attack can be exploited by uploading files
		AC	L	N/A	Arbitrary code can be executed by uploading and accessing a JSP file, medium complexity
		PR	N	N/A	No authorization is required
		UI	R	N/A	A JSP file need to be accessed
	Scope		U	U	The scope is not changed
	Impact metrics	C	H	N/A	Arbitrary code can be executed, so
		I	H	N/A	Confidentiality, Integrity and Availability are all
		A	H	N/A	violated with high level
Base Score			8.8	N/A	

### 3.5 Case 5: CVE-2018-1295

- CVSS 2.0

Table 9 CVE-2018-1295 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	The vulnerability can be triggered by sending a specially prepared form of a serialized object, network related
		AC	L	L	Sending an object is of low complexity
		AU	N	N	No authentication is required
	Impact metrics	C	P	P	The vulnerability makes it possible to run arbitrary code, thus Confidentiality, Integrity and Availability are all violated with high level
		I	P	P	
		A	P	P	
Base Score			7.5	7.5	Same

- CVSS 3.0

Table 10 CVE-2018-1295 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	The vulnerability can be triggered by sending a specially prepared form of a serialized object, network related
		AC	L	L	Sending an object is of low complexity
		PR	N	N	No authorization is required
		UI	N	N	No interaction is required
	Scope		U	U	The scope is not changed
	Impact metrics	C	H	H	The vulnerability makes it possible to run arbitrary code, thus Confidentiality, Integrity and Availability are all violated with high level
		I	H	H	
		A	H	H	
Base Score			9.8	9.8	Same

### 3.6 Case 6: CVE-2014-0110

- CVSS 2.0

Table 11 CVE-2014-0110 CVSS 2.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N	The vulnerability allows for remote attacks
		AC	M	M	The vulnerability is triggered by a large invalid SOAP message, medium complexity
		AU	N	N	No authentication is required
	Impact metrics	C	N	N	No Confidentiality is violated
		I	N	N	No Integrity is violated
		A	P	P	A denial of service can be caused
Base Score			4.3	4.3	Same

- CVSS 3.0

Table 12 CVE-2014-0110 CVSS 3.0

Group		Metrics	Exercise	NVD	Explanation
Base	Exploitability metrics	AV	N	N/A	The vulnerability allows for remote attacks
		AC	L	N/A	The vulnerability is triggered by a large invalid SOAP message, low complexity
		PR	N	N/A	No authorization is required
		UI	R	N/A	A large invalid SOAP message is required to be sent
	Scope		U	U	The scope is not changed
	Impact metrics	C	N	N/A	No Confidentiality is violated
		I	N	N/A	No Integrity is violated
		A	H	N/A	A denial of service can be caused
Base Score			6.5	N/A	