DCAE R4 Security/Vulnerability - Copy (Problem Code removed)

Note: Copied from - R4 DCAE Security/Vulnerability - Full Content; updates as of 05/16

PLEASE DO NOT UPDATE THIS PAGE - THIS WILL BE REFRESHED FROM MAIN SECOM WIKI PERIODICALLY; FOR ANY UPDATE, PLEASE WORK WITH COMMITTERS.

This template is intended to be used to document the outcome of the impact analysis related to the known vulnerability reported by Nexus-IQ (CLM tab in Jenkins). Nexus-IQ can identify the known vulnerabilities contained in the third party components embedded within ONAP deliverables.

PTLs should report in the template below only the vulnerabilities that Nexus-IQ is reporting as **"Critical"** (Level 7 to 10) and **"Severe"** (Level 4 to 6).

The projects must update this table for each milestone. The final table will be presented to the TSC at Code Freeze milestone (M4).

It is recommended to **first update to the latest version** of the third party components available or to use the Oparent dependency approach (ask Gary Wu for details). In case the latest third party components still reports some vulnerabilities, you must provide an impact analysis as illustrated in the example below.

In the case where you have nested third party components (a third party component embedding another third party component) and there is **NO CVE** number for the upstream third party component (meaning the third party component you are embedding), it is recommended to open a vulnerability issue on the upstream third party component.

Usage

Please make a **Copy** of this template into the "Security Vulnerability" wiki space and then under Casablanca Release page. Be sure to make a Copy (not a Move) by using the ... on the top right corner of this page

Within the M4 checklist create a link toward your copy of this template.

Once this template has been copied into your project wiki space, you can delete this "Tip" section as well as the "Sample of CLM Report" screenshot. This screenshot is just an example.

Between, M4 and RC0, SECCOM (Amy Zwarico, Stephen Terrill and Pawe Pawlak) will review and create a sanitize version for public visibility.

The following table is addressing 2 different scenarios:

- Confirmation of a vulnerability including an action
- False Positive

Repository	Group	Artifact	Version	Impact Analysis	Action
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE The camel-xstream component in Apache Camel before 2.15.5 and 2.16.x before 2.16.1 allow remote attackers to execute arbitrary commands via a crafted serialized Java object in an HTTP request. Explanation Apache Camel is vulnerable to Remote Code Execution (RCE) due to unsafe deserialization of java objects using the `Xstream` component. The `camel-xstream` component uses the `Xstream` component without proper security restrictions allowing arbitrary classes to be serialized /deserialized. This allows a remote attacker to inject malicious payload via a crafted serialized Java object in an HTTP request resulting in arbitrary code execution upon deserialization. **No non-vulnerable version available**	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED

dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE Apache Camel 2.20.0 to 2.20.3 and 2.21.0 Core is vulnerable to XXE in XSD validation processor. Explanation The Apache Camel package is vulnerable to XML eXternal Entity (XXE) Injection. The `doProcess()` function's conditional logic in `ValidatingProcessor.class` can, in some cases, allow external DTDs to be evaluated, even when they are disabled. A remote attacker can exploit this vulnerability to conduct Server Side Request Forgery (SSRF), exfiltrate data, or other XXE related attacks. **No non-vulnerable version available**	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE Netty before 3.9.8.Final, 3.10.x before 3.10.3.Final, 4.0.x before 4.0.28.Final, and 4.1.x before 4.1.0.Beta5 and Play Framework 2.x before 2.3.9 might allow remote attackers to bypass the httpOnly flag on cookies and obtain sensitive information by leveraging improper validation of cookie name and value characters. Explanation Netty is vulnerable to Information Disclosure. Multiple methods in multiple files improperly validate cookie names and values. This allows the presence of single-quote and double-quote characters to break tokenization. A remote attacker can exploit this vulnerability by inducing a victim to send a crafted request containing quote characters in any parameter value that sets a cookie. If that tainted cookie gets reflected in the response, the attacker can then use Cross-Site Scripting (XSS) to potentially retrieve the entire cookie header, despite the presence of an `HttpOnly` flag.	if using netty, update to >= 3.9.8.Final, >= 3.10.3.Final or >= 4.1.0.Beta5 if using Play Framework, update to >= 2.3.9 CLOSED ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE Two four letter word commands "wchp/wchc" are CPU intensive and could cause spike of CPU utilization on Apache ZooKeeper server if abused, which leads to the server unable to serve legitimate client requests. Apache ZooKeeper thru version 3.4.9 and 3.5.2 suffer from this issue, fixed in 3.4.10, 3.5.3, and later.	if using zookeeper, upgrade to >= 3.4.10, >= 3.5.3 CLOSED ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE Apache Camel's Validation Component is vulnerable against SSRF via remote DTDs and XXE. Explanation Apache Camel's Validation Component is vulnerable to Server Side Request Forgery(SSRF). The `createSchemaFactory()` method in the `SchemaReader.class` file doesn't validate for XML External Entities. A remote attacker can use this flaw to inject XML documents with remote DTD URLs or XML External Entities (XXE), which leads to Server Side Request Forgery(SSRF). **No non-vulnerable version available**	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE No authentication/authorization is enforced when a server attempts to join a quorum in Apache ZooKeeper before 3.4.10, and 3.5.0-alpha through 3.5.3-beta. As a result an arbitrary end point could join the cluster and begin propagating counterfeit changes to the leader. Explanation Apache Zookeeper is vulnerable to Insufficient Authorization. The `connectToLeader()` method in the `Learner` class connects to “Leader” servers without ensuring that the server's host name matches the address used by the `Socket` when establishing the connection, and without requiring authentication or authorization. An attacker with access to the Zookeeper quorum can exploit this vulnerability to connect an unauthorized host to the quorum. The attacker could then propagate changes from the malicious host to the “Leader” server, potentially leading to further attacks.	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE WebSocket08FrameDecoder in Netty 3.6.x before 3.6.9, 3.7.x before 3.7.1, 3.8.x before 3.8.2, 3.9.x before 3.9.1, and 4.0.x before 4.0.19 allows remote attackers to cause a denial of service (memory consumption) via a TextWebSocketFrame followed by a long stream of ContinuationWebSocketFrames. Explanation Due to a flaw in the WebSocket08FrameDecoder implementation, Netty is prone to denial of service (DoS) attacks. An out-of-memory error occurs in the WebSocket08FrameDecoder implementation while processing a TextWebSocketFrame that is followed by a long stream of ContinuationWebSocketFrames.	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
dcaegen2-analytics-pnda	ru.yandex.qatools.camelot	camelot-kafka : jar	2.4.4	Description from CVE In Apache Kafka 0.9.0.0 to 0.9.0.1, 0.10.0.0 to 0.10.2.1, 0.11.0.0 to 0.11.0.2, and 1.0.0, authenticated Kafka users may perform action reserved for the Broker via a manually created fetch request interfering with data replication, resulting in data loss. Explanation Apache Kafka is vulnerable to Information Exposure. The `handleFetchRequest` method of the `KafkaApis.scala` file does not check authorizations correctly when handling `Fetch` requests for replication. This allows malicious consumers to impersonate brokers. An authenticated attacker can forge `Fetch` requests that contain a broker ID and therefore confuse the cluster. When the payload executes, it can cause multiple security issues.	Closed ☑ ~~DCAEGEN2-1207~~ - dcaegen2/analytics/pnda security vulnerabilities CLOSED
onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.core	jackson-databind	2.9.6	Description from CVE FasterXML jackson-databind 2.x before 2.9.7 might allow remote attackers to conduct server-side request forgery (SSRF) attacks by leveraging failure to block the axis2-jaxws class from polymorphic deserialization. Explanation `jackson-databind` is vulnerable to Server-Side Request Forgery (SSRF) via Deserialization of Untrusted Data. The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in the execution of SSRF attacks if the application attempts to deserialize it.	Request Exception

onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.core	jackson-databind	2.9.6	Description from CVE FasterXML jackson-databind 2.x before 2.9.7 might allow remote attackers to execute arbitrary code by leveraging failure to block the slf4j-ext class from polymorphic deserialization. Explanation `jackson-databind` is vulnerable to Remote Code Execution (RCE). The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it.	Request Exception
onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.core	jackson-databind	2.9.6	Description from CVE FasterXML jackson-databind 2.x before 2.9.7 might allow remote attackers to execute arbitrary code by leveraging failure to block the blaze-ds-opt and blaze-ds-core classes from polymorphic deserialization. Explanation `jackson-databind` is vulnerable to Remote Code Execution (RCE). The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it.	Request Exception
onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.core	jackson-databind	2.9.6	Description from CVE FasterXML jackson-databind 2.x before 2.9.7 might allow attackers to conduct external XML entity (XXE) attacks by leveraging failure to block unspecified JDK classes from polymorphic deserialization. Explanation `jackson-databind` is vulnerable to XML eXternal Entity (XXE) attacks via Deserialization of Untrusted Data. The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in the execution of XXE attacks if the application attempts to deserialize it.	Request Exception
onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.core	jackson-databind	2.9.6	Explanation `jackson-databind` is vulnerable to Remote Code Execution (RCE). The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it.	Request Exception
onap-dcaegen2-analytics-tca-gen2	com.fasterxml.jackson.datatype	jackson-datatype-jsr310	2.9.6	Description from CVE Fasterxml Jackson version Before 2.9.8 contains a CWE-20: Improper Input Validation vulnerability in Jackson-Databind that can result in Causes a denial-of-service (DoS). This attack appear to be exploitable via The victim deserializes malicious input, specifically very large values in the nanoseconds field of a time value. This vulnerability appears to have been fixed in 2.9.8. Explanation The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the InstantDeserializer class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation.	Request Exception
onap-dcaegen2-analytics-tca-gen2	org.springframework : spring-web : 5.0.9.RELEASE			Description from CVE Spring Framework, version 5.1, versions 5.0.x prior to 5.0.10, versions 4.3.x prior to 4.3.20, and older unsupported versions on the 4.2.x branch provide support for range requests when serving static resources through the ResourceHttpRequestHandler, or starting in 5.0 when an annotated controller returns an org.springframework.core.io.Resource. A malicious user (or attacker) can add a range header with a high number of ranges, or with wide ranges that overlap, or both, for a denial of service attack. This vulnerability affects applications that depend on either spring-webmvc or spring-webflux. Such applications must also have a registration for serving static resources (e.g. JS, CSS, images, and others), or have an annotated controller that returns an org.springframework.core.io.Resource. Spring Boot applications that depend on spring-boot-starter-web or spring-boot-starter-webflux are ready to serve static resources out of the box and are therefore vulnerable. Explanation The Spring Framework is vulnerable to Denial of Service (DoS). The `toResourceRegions()` and `parseRanges()` methods in the `HttpRange` class process range requests with a large number of extensive ranges which can overlap causing additional resource consumption. An attacker can exploit this vulnerability by submitting a request with a malicious range header that includes overlapping and/or extensive ranges which exhaust the server's resources leading to a DoS.	Closed (04/08) Upgrade to >=5.0.10.RELEASE and <=5.0.12.RELEASE ☑ ~~DCAEGEN2-1208~~ - dcaegen2/analytics/tca-gen2 security vulnerabilities CLOSED
onap-dcaegen2-analytics-tca-gen2	io.undertow : undertow-core : 1.4.25.Final			Description from CVE In Undertow 2.x before 2.0.0.Alpha2, 1.4.x before 1.4.17.Final, and 1.3.x before 1.3.31.Final, it was found that the fix for CVE-2017-2666 was incomplete and invalid characters are still allowed in the query string and path parameters. This could be exploited, in conjunction with a proxy that also permitted the invalid characters but with a different interpretation, to inject data into the HTTP response. By manipulating the HTTP response the attacker could poison a web-cache, perform an XSS attack, or obtain sensitive information from requests other than their own.	05/06 - CLOSED Switch to 2.0.17.Final ☑ ~~DCAEGEN2-1457~~ - onap-dcaegen2-analytics-tca-gen2 - 2019-04-19 CLOSED
onap-dcaegen2-analytics-tca-gen2	io.undertow : undertow-core : 1.4.25.Final			Description from CVE Get requests in JBoss Enterprise Application Platform (EAP) 7 disclose internal IP addresses to remote attackers. Explanation The `undertow-core` package is vulnerable to Information Exposure. The `getHostAndPort()` method in the `HttpServerExchange` class exposes an internal IP address via the `Location` header during a `302` redirect if the host header field is not set. A remote attacker can exploit this issue by submitting a GET request that results in a `302` redirect response. The attacker can leverage this vulnerability to exfiltrate an internal IP address that can potentially be used for further attacks.	05/06 - CLOSED Switch to 2.0.17.Final ☑ ~~DCAEGEN2-1457~~ - onap-dcaegen2-analytics-tca-gen2 - 2019-04-19 CLOSED
onap-dcaegen2-analytics-tca-gen2	io.undertow : undertow-core : 1.4.25.Final			Description from CVE An information leak vulnerability was found in Undertow. If all headers are not written out in the first write() call then the code that handles flushing the buffer will always write out the full contents of the writevBuffer buffer, which may contain data from previous requests. Explanation The `undertow` package is vulnerable to Denial-of-Service (DoS). The `processWrite()` method in the `HttpResponseConduit` Java class file does not restrict the buffer allocation size. An attacker can exploit this vulnerability by crafting a request that consists of a large header size and sending it to the server. The request, once processed, would exceed the allocated buffer size resulting in an application crash or unintended behavior.	05/06 - CLOSED Switch to 2.0.17.Final ☑ ~~DCAEGEN2-1457~~ - onap-dcaegen2-analytics-tca-gen2 - 2019-04-19 CLOSED

dcaegen2-analytics-tca	c3p0 : c3p0 : 0.9.1.1			c3p0 0.9.5.2 allows XXE in extractXmlConfigFromInputStream in com/mchange/v2/c3p0/cfg/C3P0ConfigXmlUtils.java during initialization. **Impacted method/class not used**.	False Positive
dcaegen2-analytics-tca	com.fasterxml.jackson.core : jackson-databind : 2.4.4			Workaround: Do not use the default typing. Instead you will need to implement your own. *It is also possible to customize global defaulting, using ObjectMapper.setDefaultTyping(...) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all aspects of type information. Builder itself is just a short-cut for building actual handlers.*	Request Exception
dcaegen2-analytics-tca	com.fasterxml.jackson.core : jackson-core : 2.4.4			The application is vulnerable by using this component when `WRITE_BIGDECIMAL_AS_PLAIN` is explicitly enabled. By default, `WRITE_BIGDECIMAL_AS_PLAIN` is disabled `jackson-core` is vulnerable to Denial of Service (DoS). The `_reportInvalidToken()` function in the `UTF8StreamJsonParser` and `ReaderBasedJsonParser` classes allows large amounts of extraneous data to be printed to the server log	Request Exception
dcaegen2-analytics-tca	commons-codec : commons-codec : 1.6			Apache Commons Codec - Base32 would decode some invalid Base32 encoded string into arbitrary value	Upgrade to 1.10 CLOSED ☑ ~~DCAEGEN2-1209~~ - dcaegen2/analytics/tca security vulnerabilties CLOSED
dcaegen2-analytics-tca	com.google.guava : guava : 13.0.1			The application is vulnerable by using this component if it uses Java deserialization or GWT-RPC to deserialize untrusted data. **Data for TCA is coming from known DCAE sources (through Dmaap) hence this vulnerability does not apply**	False Positive
dcaegen2-collectors-datafile	com.fasterxml.jackson.core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Request Exception
dcaegen2-collectors-datafile	com.fasterxml.jackson.core: jackson-databind: 2.97			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2-collectors-datafile	com.fasterxml.jackson.core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2-collectors-datafile	com.fasterxml.jackson.datatype: jackson-datatype-jsr310: 2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the InstantDeserializer class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or InstantDeserializer classes of this component to deserialize untrusted data.	Request Exception
dcaegen2-collectors-datafile	org.springframework : spring-web : 5.1.0.RELEASE			The Spring Framework is vulnerable to Denial of Service (DoS). The `toResourceRegions()` and `parseRanges()` methods in the `HttpRange` class process range requests with a large number of extensive ranges which can overlap causing additional resource consumption	CLOSED Upgrade to 5.1.2.RELEASE if impacted ☑ ~~DCAEGEN2-1210~~ - dcaegen2/collectors/datafile security vulnerabilties CLOSED

dcaegen2-collectors-datafile	com.jcraft : jsch : 0.1.53			Directory traversal vulnerability in JCraft JSch before 0.1.54 on Windows, when the mode is ChannelSftp.OVERWRITE, allows remote SFTP servers to write to arbitrary files via a ..\ (dot dot backslash) in a response to a recursive GET command.	CLOSED Upgrade to 0.1.54 if impacted ☑ ~~DCAEGEN2-1210~~ - dcaegen2/collectors/datafile security vulnerabilties CLOSED
dcaegen2/collector/hv-ves	com.fasterxml.jackson.core : jackson-databind : 2.9.6			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x. Workaround: Do not use the default typing. Instead you will need to implement your own. *It is also possible to customize global defaulting, using ObjectMapper.setDefaultTyping(…) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all aspects of type information. Builder itself is just a short-cut for building actual handlers.*	Request Exception
dcaegen2/collector/hv-ves	com.fasterxml.jackson.datatype:jackson-datatype-jsr310: 2.9.4			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the InstantDeserializer class allow arbitrarily large `BigDecimal` initialization values.	Request Exception
dcaegen2/collector/hv-ves	com.google.guava : guava : 19.0			Unbounded memory allocation in Google Guava 11.0 through 24.x before 24.1.1 allows remote attackers to conduct denial of service attacks against servers that depend on this library and deserialize attacker-provided data, because the AtomicDoubleArray class (when serialized with Java serialization) and the CompoundOrdering class The application is vulnerable by using this component if it uses Java deserialization or GWT-RPC to deserialize untrusted data.	CLOSED Upgrade to 23.6.1-jre ☑ ~~DCAEGEN2-1211~~ - dcaegen2/collectors/hv-ves security vulnerabilities CLOSED
dcaegen2-collectors-restconf	org.apache.tomcat.embed : tomcat-embed-core : 8.0.36			The `ResourceLinkFactory` class of Apache Tomcat is vulnerable to Authorization Bypass. The `addResourceLink() andremoveResourceLink()methods of theNamingContextListener`` class allows the ability to modify unauthorized resource links on global JNDI resources not linked to the web application.	Closed ~~Upgrade to 8.5.35~~ ☑ ~~DCAEGEN2-1212~~ - dcaegen2/collectors/restconf security vulnerabilties CLOSED
dcaegen2-collectors-restconf	com.fasterxml.jackson.core : jackson-databind : 2.8.11			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Do not use the default typing. Instead you will need to implement your own. *It is also possible to customize global defaulting, using ObjectMapper.setDefaultTyping(…) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all*	Closed ☑ ~~DCAEGEN2-1212~~ - dcaegen2/collectors/restconf security vulnerabilties CLOSED
dcaegen2-collectors-restconf	org.apache.tomcat : tomcat-catalina : 8.0.36			`Apache Tomcat` is vulnerable to Insufficient Authorization. The `forwardToLoginPage()`,`forwardToErrorPage()` methods of `FormAuthenticator` class, the `invoke()` method of `StandardHostValve` class, and the `asyncDispatch()` method of `CoyoteAdapter` class allows malicious requests to be processed as it does not use the appropriate facade object when running an untrusted application under a SecurityManage	Closed ~~Upgrade to 8.5.35~~ ☑ ~~DCAEGEN2-1212~~ - dcaegen2/collectors/restconf security vulnerabilties CLOSED

dcaegen2-collectors-restconf	org. apache. httpcomponents : httpclient : 4.5			The Apache httpcomponents component is vulnerable to Directory Traversal. The `normalizePath()` function in the `URIBuilder` class allows directory traversal characters such as `../`.	Closed ~~Upgrade to 4.5.3~~ ☑ ~~DCAEGEN2-1212~~ - dcaegen2/collectors/restconf security vulnerabilties CLOSED
dcaegen2-collectors-restconf	com. fasterxml. jackson. core: jackson-databind: 2.9.7			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Added 3/20/19 - No non vulnerable version available Request Exception
dcaegen2-collectors-restconf	com. fasterxml. jackson. core: jackson-databind: 2.9.7			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Added 3/20/19 - No non vulnerable version available Request Exception
dcaegen2-collectors-restconf	com. fasterxml. jackson. core: jackson-databind: 2.9.7			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Added 3/20/19 - No non vulnerable version available Request Exception
dcaegen2-collectors-restconf	com. fasterxml. jackson. datatype: jackson-datatype-jsr310: 2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Added 3/20/19 - No non vulnerable version available Request Exception
dcaegen2-collectors-restconf	org. springframework. security : spring-security-web : 5.1.1. RELEASE			The `spring-security-web` package is vulnerable to Cross-Site Request Forgery (CSRF). The `doFilter()` method in the `SwitchUserFilter`, which is reachable via a GET request, does not require any form of confirmation that the user sending the request intended to do so **Not applicable; as the specified method is not invoked**	Added 3/20/19 - No non vulnerable version available **False positive**
dcaegen2-collectors-restconf	com. googlecode. libphonenumber : libphonenumber : 6.2			sonatype-2015-0090 - libphonenumber - A Cross Site Scripting vulnerability was found which is exploitable by manipulating the inputs Not applicable	Added 3/20/19 - No non vulnerable version available Request Exception
dcaegen2-collectors-restconf	org. springframework. security : spring-security-core : 5.1.1. RELEASE			Spring Security versions 4.2.x prior to 4.2.12, 5.0.x prior to 5.0.12, and 5.1.x prior to 5.1.5 contain an insecure randomness vulnerability when using SecureRandomFactoryBean#setSeed to configure a SecureRandom instance. Upgrade to 5.1.5.RELEASE **Not applicable; as the specified method is not invoked**	False Positive
dcaegen2-collectors-restconf	javax. mail : mailapi : 1.4.3			JavaMail is vulnerable to Information Exposure. The `getUniqueMessageIDValue()` method in the `UniqueValue` class file appends the username and the hostname of the Java process when generating the `Message-Id` for an email. This can lead to unintended information leakage in the email headers and potentially lead to security issues. **Not applicable; as the specified method is not invoked**	Added 3/20/19 - No non vulnerable version available False Positive
dcaegen2-collectors-ves	com. fasterxml. jackson. core: jackson-databind: 2.9.7			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Request Exception
dcaegen2-collectors-ves	com. fasterxml. jackson. core: jackson-databind: 2.9.7			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception

dcaegen2-collectors-ves	com.fasterxml.jackson.core:jackson-databind:2.9.7			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2-collectors-ves	com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Request Exception
dcaegen2-collectors-ves	org.springframework.security : spring-security-web : 5.1.1.RELEASE			The `spring-security-web` package is vulnerable to Cross-Site Request Forgery (CSRF). The `doFilter()` method in the `SwitchUserFilter`, which is reachable via a GET request, does not require any form of confirmation that the user sending the request intended to do so **Not applicable; as the specified method is not invoked**	False Positive
dcaegen2-collectors-ves	org.springframework.security : spring-security-core : 5.1.1.RELEASE			Spring Security versions 4.2.x prior to 4.2.12, 5.0.x prior to 5.0.12, and 5.1.x prior to 5.1.5 contain an insecure randomness vulnerability when using SecureRandomFactoryBean#setSeed to configure a SecureRandom instance. Upgrade to 5.1.5.RELEASE **Not applicable; as the specified method is not invoked**	False positive
dcaegen2-collectors-ves	com.googlecode.libphonenumber : libphonenumber : 6.2			sonatype-2015-0090 - libphonenumber - A Cross Site Scripting vulnerability was found which is exploitable by manipulating the inputs Not applicable	False Positive
dcaegen2-collectors-ves	javax.mail : mailapi : 1.4.3			JavaMail is vulnerable to Information Exposure. The `getUniqueMessageIDValue()` method in the `UniqueValue` class file appends the username and the hostname of the Java process when generating the `Message-Id` for an email. This can lead to unintended information leakage in the email headers and potentially lead to security issues. **Not applicable; as the specified method is not invoked**	False Positive
dcaegen2/platform/inventory-api	com.fasterxml.jackson.core : jackson-databind : 2.8.7			A deserialization flaw was discovered in the jackson-databind, versions before 2.6.7.1, 2.7.9.1 and 2.8.9, Workaround: Do not use the default typing. Instead you will need to implement your own. *It is also possible to customize global defaulting, using ObjectMapper.setDefaultTyping(…) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all aspects of type information. Builder itself is just a short-cut for building actual handlers.*	Request Exception
dcaegen2/platform/inventory-api	com.fasterxml.jackson.datatype : jackson-datatype-jsr310 : 2.8.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. Detection The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Request Exception
onap-dcaegen2-services-pm-mapper	io.undertow : undertow-core : 2.0.16.Final			Description from CVE Get requests in JBoss Enterprise Application Platform (EAP) 7 disclose internal IP addresses to remote attackers. Explanation The undertow-core package is vulnerable to Information Exposure. The getHostAndPort() method in the HttpServerExchange class exposes an internal IP address via the Location header during a 302 redirect if the host header field is not set. A remote attacker can exploit this issue by submitting a GET request that results in a 302 redirect response. The attacker can leverage this vulnerability to exfiltrate an internal IP address that can potentially be used for further attacks.	~~To be assessed if risk noted is valid or if dependency can be removed.~~ False Positive ~~DCAEGEN2-1224~~ - dcaegen2/services/pm-mapper security vulnerabilities CLOSED

onap-dcaegen2-services-pm-mapper	io.undertow : undertow-core : 2.0.16.Final			Description from CVE An information leak vulnerability was found in Undertow. If all headers are not written out in the first write() call then the code that handles flushing the buffer will always write out the full contents of the writevBuffer buffer, which may contain data from previous requests. Explanation The undertow package is vulnerable to Denial-of-Service (DoS). The processWrite() method in the HttpResponseConduit Java class file does not restrict the buffer allocation size. An attacker can exploit this vulnerability by crafting a request that consists of a large header size and sending it to the server. The request, once processed, would exceed the allocated buffer size resulting in an application crash or unintended behavior.	~~To be assessed if risk noted is valid or if dependency can be removed.~~ False Positive ~~DCAEGEN2-1224~~ - dcaegen2/services/pm-mapper security vulnerabilities CLOSED
onap-dcaegen2-services-pm-mapper	org.jboss.gwt.elemento : elemento-testsuite-standalone : 0.9			Description from CVE Get requests in JBoss Enterprise Application Platform (EAP) 7 disclose internal IP addresses to remote attackers. Explanation The undertow-core package is vulnerable to Information Exposure. The getHostAndPort() method in the HttpServerExchange class exposes an internal IP address via the Location header during a 302 redirect if the host header field is not set. A remote attacker can exploit this issue by submitting a GET request that results in a 302 redirect response. The attacker can leverage this vulnerability to exfiltrate an internal IP address that can potentially be used for further attacks.	~~No non-vulnerable version available; to be assessed if risk noted is valid or if dependency can be removed.~~ False Positive ~~DCAEGEN2-1224~~ - dcaegen2/services/pm-mapper security vulnerabilities CLOSED
onap-dcaegen2-services-pm-mapper	org.jboss.gwt.elemento : elemento-testsuite-standalone : 0.9			Description from CVE An information leak vulnerability was found in Undertow. If all headers are not written out in the first write() call then the code that handles flushing the buffer will always write out the full contents of the writevBuffer buffer, which may contain data from previous requests. Explanation The undertow package is vulnerable to Denial-of-Service (DoS). The processWrite() method in the HttpResponseConduit Java class file does not restrict the buffer allocation size. An attacker can exploit this vulnerability by crafting a request that consists of a large header size and sending it to the server. The request, once processed, would exceed the allocated buffer size resulting in an application crash or unintended behavior.	~~No non-vulnerable version available; to be assessed if risk noted is valid or if dependency can be removed.~~ CLOSED ~~DCAEGEN2-1224~~ - dcaegen2/services/pm-mapper security vulnerabilities CLOSED
onap-dcaegen2-services-bbs-event-processor	org.hibernate : hibernate-validator : 5.2.4.Final			Hibernate Validator 5.2.x before 5.2.5 final, 5.3.x, and 5.4.x, it was found that when the security manager's reflective permissions, which allows it to access the private members of the class, are granted to Hibernate Validator, a potential privilege escalation can occur. By allowing the calling code to access those private members without the permission an attacker may be able to validate an invalid instance and access the private member value via ConstraintViolation#getInvalidValue().	CLOSED Upgrade to 5.3.6.Final ☑ ~~DCAEGEN2-1388~~ - dcaegen2-services-bbs-event-processor security vulnerabilities CLOSED
onap-dcaegen2-services-bbs-event-processor	com.fasterxml.jackson.core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	CLOSED ☑ ~~DCAEGEN2-1388~~ - dcaegen2-services-bbs-event-processor security vulnerabilities CLOSED
onap-dcaegen2-services-bbs-event-processor	com.fasterxml.jackson.core: jackson-databind: 2.97			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	CLOSED ☑ ~~DCAEGEN2-1388~~ - dcaegen2-services-bbs-event-processor security vulnerabilities CLOSED
onap-dcaegen2-services-bbs-event-processor	com.fasterxml.jackson.core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	CLOSED ☑ ~~DCAEGEN2-1388~~ - dcaegen2-services-bbs-event-processor security vulnerabilities CLOSED
onap-dcaegen2-services-bbs-event-processor	com.fasterxml.jackson.datatype: jackson-datatype-jsr310: 2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	CLOSED ☑ ~~DCAEGEN2-1388~~ - dcaegen2-services-bbs-event-processor security vulnerabilities CLOSED

onap-dcaegen2-services-bbs-event-processor	com.fasterxml.jackson.core : jackson-databind : 2.9.8			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Analysis Note:BBS-ep does not use Jackson for its JSON serialization/deserialization logic (it uses Gson). Jackson databind artifact is only used at runtime by Swagger.	Request Exception
onap-dcaegen2-platform-cdapbroker	nothing				
onap-dcaegen2-platform-cli	nothing				
onap-dcaegen2-platform-deployment-handler	nothing				
dcaegen2-platform-servicechange-handler	org.json : json : 20131018			Found a License in the 'Use Restrictions' License Threat Group Likely false positive identified in the tool	False Positive
onap-dcaegen2-platform-plugins	nothing				
onap-dcaegen2-platform-policy-handler	nothing				
onap-dcaegen2-platform-servicechange-handler	nothing				
onap-dcaegen2-services-heartbeat	nothing				
dcaegen2/services/mapper	com.fasterxml.jackson.core : jackson-databind : 2.9.6			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x. Workaround: Do not use the default typing. Instead you will need to implement your own. > *It is also possible to customize global defaulting, using ObjectMapper.setDefaultTyping(…) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all aspects of type information. Builder itself is just a short-cut for building actual handlers.*	Upgrade to 2.9.7 for consistency ☑ ~~DCAEGEN2-1213~~ - dcaegen2/services/mapper security vulnerabilties CLOSED
dcaegen2/services/mapper	com.fasterxml.jackson.core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Request Exception

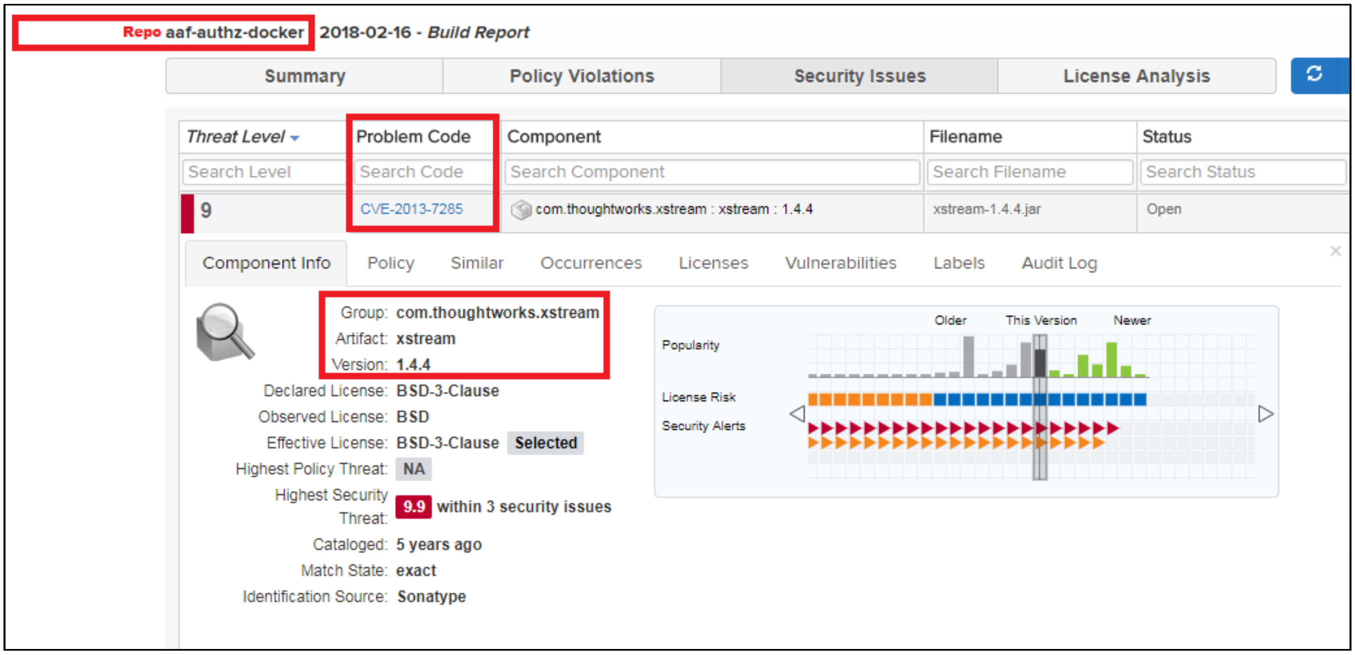
dcaegen2/services/mapper	com.fasterxml.jackson.core:jackson-databind:2.97			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2/services/mapper	com.fasterxml.jackson.core:jackson-databind:2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2/services/mapper	com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Request Exception
dcaegen2/services/mapper	dom4j : dom4j : 1.6.1			dom4j version prior to version 2.1.1 contains a CWE-91: XML Injection vulnerability in Class: Element. Methods: addElement, addAttribute that can result in an attacker tampering with XML documents through XML injection	No non-vulnerable version available. Request Exception
dcaegen2/services/mapper	org.apache.ant : ant : 1.9.4			Apache Ant is vulnerable to Path Traversal. The `extractFile()` method in the `Expand` class, which is responsible for extracting compressed archive files, allows archives to contain files with path traversal characters (like `../`) in the name. If the application extracts an archive that is provided via user input, or if an attacker has access to modify existing archives on the filesystem, this vulnerability can be exploited to reveal the structure and possibly contents of other directories on the filesystem.Detection The application is vulnerable by using this component when `stripAbsolutePathSpec` is false. You are not vulnerable if you are running the applicable .patch file available in 1.10.1 of rpm.	No non-vulnerable version available. Request Exception
dcaegen2/services/mapper	xerces : xercesImpl : 2.12.0			Apache Xerces-J is vulnerable to a Denial of Service (DoS) attack. The `setupCurrentEntity()` method in the `XMLEntityManager` class lacks a connection timeout mechanism. Detection The application is vulnerable by using this component if you are running any of the following versions of Java: *Java SE: 6u161, 7u151, 8u144, 9; Java SE Embedded: 8u144*	No non-vulnerable version available. Request Exception
dcaegen2/services/mapper	commons-fileupload : commons-fileupload : 1.3.3			Apache Commons FileUpload contains a resource leak which may lead to a Denial of Service (DoS) attack. The `FileItemIteratorImpl()` method in the `FileUploadBase` class does not close a file input stream on encountering an exception; this means that the application maintains an open reference to the file in question indefinitely, as the stream is continuously waiting to read more data. An attacker can exploit this vulnerability by sending a large number of requests specifically crafted to trigger this exception, causing the application to maintain a large number of open streams. Eventually, these streams will consume all available processing or memory resources and render the application unresponsive.	No non-vulnerable version available. Request Exception
dcaegen2/services/prh	com.fasterxml.jackson.core:jackson-databind:2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Request Exception
dcaegen2/services/prh	com.fasterxml.jackson.core:jackson-databind:2.97			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2/services/prh	com.fasterxml.jackson.core:jackson-databind:2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2/services/prh	com.fasterxml.jackson.datatype:jackson-datatype-jsr310:2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Request Exception

dcaegen2 /services/sdk	com. fasterxml. jackson. core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.	Request Exception
dcaegen2 /services/sdk	com. fasterxml. jackson. core: jackson-databind: 2.97			`jackson-databind` is vulnerable to Remote Code Execution (RCE). The `validateSubType()` function in the `SubTypeValidator` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2 /services/sdk	com. fasterxml. jackson. core: jackson-databind: 2.97			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. Workaround: Do not use the default typing. Instead you will need to implement your own.	Request Exception
dcaegen2 /services/sdk	com. fasterxml. jackson. datatype: jackson-datatype-jsr310: 2.9.7			The FasterXML `jackson-datatype-jsr310` package contains a Denial of Service (DoS) vulnerability. The `deserialize()` method in the `DurationDeserializer` class and the `_fromDecimal()` method in the `InstantDeserializer` class allow arbitrarily large `BigDecimal` initialization values. A remote attacker can exploit this vulnerability by crafting and submitting a request that causes the application to deserialize an inordinately large value, causing the application to hang and leading to a DoS situation. The application is vulnerable by using the `DurationDeserializer` or `InstantDeserializer` classes of this component to deserialize untrusted data.	Request Exception
onap-dcaegen2-services-son-handler	com. fasterxml. jackson. core : jackson-databind : 2.9.6			The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized. Note: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x. Workaround: Do not use the default typing. Instead you will need to implement your own. *It is also possible to customize global defaulting, using ObjectMapper. setDefaultTyping(…) – you just have to implement your own TypeResolverBuilder (which is not very difficult); and by doing so, can actually configure all aspects of type information. Builder itself is just a short-cut for building actual handlers.*	Remove this dependency if workaround exist; if not upgrade to 2.9.8 ☑ ~~DCAEGEN2-1275~~ - dcaegen2/services/son-handler security vulnerabilities CLOSED
onap-dcaegen2-services-son-handler	com. fasterxml. jackson. datatype : jackson-datatype-jsr310 : 2.9.6			Fasterxml Jackson version Before 2.9.8 contains a CWE-20: Improper Input Validation vulnerability in Jackson-Modules-Java8 that can result in Causes a denial-of-service (DoS). This attack appear to be exploitable via The victim deserializes malicious input, specifically very large values in the nanoseconds field of a time value. This vulnerability appears to have been fixed in 2.9.8.	Remove this dependency if workaround exist; if not upgrade to 2.9.8 ☑ ~~DCAEGEN2-1275~~ - dcaegen2/services/son-handler security vulnerabilities CLOSED
onap-dcaegen2-services-son-handler	org. codehaus. jackson : jackson-mapper-asl : 1.9.13			A deserialization flaw was discovered in the jackson-databind, versions before 2.6.7.1, 2.7.9.1 and 2.8.9, which could allow an unauthenticated user to perform code execution by sending the maliciously crafted input to the readValue method of the ObjectMapper.Explanation `jackson-databind` is vulnerable to Remote Code Execution (RCE). The `createBeanDeserializer()` function in the `BeanDeserializerFactory` class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object that will result in RCE if the application attempts to deserialize it.	No non-vulnerable version available. Request Exception
onap-dcaegen2-services-son-handler	org. postgresql : postgresql : 42.2.4			A weakness was found in postgresql-jdbc before version 42.2.5. It was possible to provide an SSL Factory and not check the host name if a host name verifier was not provided to the driver. This could lead to a condition where a man-in-the-middle attacker could masquerade as a trusted server by providing a certificate for the wrong host, as long as it was signed by a trusted CA. Explanation The `postgresql` package is vulnerable to Man-in-the-Middle (MitM) attacks. When using a non-default SSL Factory, the postgresql jdbc doesn't validate the hostname of SSL certificates. An attacker can potentially exploit this behavior to perform a MitM attack.	Switch to 42.2.5 CLOSED ☑ ~~DCAEGEN2-1275~~ - dcaegen2/services/son-handler security vulnerabilities CLOSED
onap-dcaegen2-services-son-handler	org. springframework : spring-web : 5.0.9. RELEASE			Spring Framework, version 5.1, versions 5.0.x prior to 5.0.10, versions 4.3.x prior to 4.3.20, and older unsupported versions on the 4.2.x branch provide support for range requests when serving static resources through the ResourceHttpRequestHandler, or starting in 5.0 when an annotated controller returns an org.springframework.core.io.Resource. A malicious user (or attacker) can add a range header with a high number of ranges, or with wide ranges that overlap, or both, for a denial of service attack.	Switch to 5.0.11.RELEASE CLOSED ☑ ~~DCAEGEN2-1275~~ - dcaegen2/services/son-handler security vulnerabilities CLOSED

onap-dcaegen2-services-son-handler	org.springframework.data : spring-data-jpa : 2.0.9.RELEASE			The Spring `spring-data-jpa` package is vulnerable to Information Disclosure. The `postProcess()` method in the `JpaRepositoryConfigExtension` class, the `build()` method in the `JpaQueryCreator$PredicateBuilder` class, the `create()` method in the `JpaQueryLookupStrategy()` class, the `next()` method in the `ParameterMetadataProvider` class, the `prepare()` method in the `ParameterMetadataProvider$ParameterMetadata` class, the `createCreator()` method in the `PartTreeJpaQuery$QueryPreparer` class, the `getQueryLookupStrategy()` method in the `JpaRepositoryFactory` class, and the `createRepositoryFactory()` method in the `JpaRepositoryBean` class allow control characters in `LIKE` expressions.	CLOSED - 04/29 Switch to 2.0.14.RELEASE ☑ ~~DCAEGEN2-1456~~ - on ap-dcaegen2-services-son-handler - 2019-04-20 CLOSED
onap-dcaegen2-services-son-handler	dom4j : dom4j : 1.6.1			Description from CVEdom4j version prior to version 2.1.1 contains a CWE-91: XML Injection vulnerability in Class: Element. Methods: addElement, addAttribute that can result in an attacker tampering with XML documents through XML injection. This attack appear to be exploitable via an attacker specifying attributes or elements in the XML document. This vulnerability appears to have been fixed in 2.1.1 or later.Explanation The `dom4j` package is vulnerable to XML Injection. The `QName()` function in the `QName` class file does not properly sanitize the `QName` input attribute value(s). A remote attacker can exploit this vulnerability by injecting an XML object that contains arbitrary code in the element and attribute names, hence leading to XML Injection.	No non-vulnerable version available. Request Exception
onap-dcaegen2-services-son-handler	org.springframework.data : spring-data-commons-core : 1.0.0.RELEASE			Spring Data Commons, versions 1.13 to 1.13.10, 2.0 to 2.0.5, and older unsupported versions, contain a property path parser vulnerability caused by unlimited resource allocation. An unauthenticated remote malicious user (or attacker) can issue requests against Spring Data REST endpoints or endpoints using property path parsing which can cause a denial of service (CPU and memory consumption).	No non-vulnerable version available. Request Exception

Sample of CLM Report

Repository	Group	Artifact	Version	Problem Code	Impact Analysis	Action
aaf-authz-docker	com.thoughtworks.xstream	xstream	1.4.4	CVE-2013-7285	Q-How does this vulnerability impact your project	JIRA Link In the JIRA ticket, you will explain findings and action plan Add the Label "Security"
					False Positive Add the explanation why you believe it is a false positive	Not applicable



Repo aaf-authz-docker 2018-02-16 - *Build Report*

Threat Level	Problem Code	Component	Filename	Status
Search Level	Search Code	Search Component	Search Filename	Search Status
9	CVE-2013-7285	com.thoughtworks.xstream : xstream : 1.4.4	xstream-1.4.4.jar	Open

Group: com.thoughtworks.xstream
Artifact: xstream
Version: 1.4.4
Declared License: BSD-3-Clause
Observed License: BSD
Effective License: BSD-3-Clause Selected
Highest Policy Threat: NA
Highest Security Threat: 9.9 within 3 security issues
Cataloged: 5 years ago
Match State: exact
Identification Source: Sonatype