





El Alto DMaaP Security/Vulnerability Report

This table represents the known exploitable and non-exploitable vulnerabilities in third party packages used in the project.

Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
dmaap-datarouter	commons-codec	N/A	Ineffective		The Apache <code>commons-codec</code> package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>BCodec</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-datarouter	org.eclipse.jetty	CVE-2019-10241	Ineffective		In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the <code>DefaultServlet</code> or <code>ResourceHandler</code> that is configured for showing a Listing of directory contents. The <code>jetty</code> package is vulnerable to Cross-Site Scripting (XSS). The <code>sendDirectory()</code> function in <code>ResourceService.class</code> and <code>DefaultServlet.class</code> files and the <code>doDirectory()</code> function in the <code>ResourceHandler.class</code> file use the <code>getListHTML()</code> function in the <code>Resource.class</code> file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or <code>ResourceHandler</code> that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-datarouter	org.eclipse.jetty	CVE-2019-10247			In Eclipse Jetty version 7.x, 8.x, 9.2.27 and older, 9.3.26 and older, and 9.4.16 and older, the server running on any OS and Jetty version combination will reveal the configured fully qualified directory base resource location on the output of the 404 error for not finding a Context that matches the requested path. The default server behavior on <code>jetty-distribution</code> and <code>jetty-home</code> will include at the end of the Handler tree a <code>DefaultHandler</code> , which is responsible for reporting this 404 error, it presents the various configured contexts as HTML for users to click through to. This produced HTML includes output that contains the configured fully qualified directory base resource location for each context. The <code>jetty</code> package is vulnerable to sensitive Information Exposure. The <code>handle()</code> method of the <code>DefaultHandler.class</code> file discloses sensitive information via the <code>context</code> object. The method outputs the value of <code>context.toString()</code> within the error responses which will reveal the base resource path of each context.	Requesting exception. <div><input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components CLOSED</div>
dmaap-datarouter	org.eclipse.jetty	CVE-2019-10241	Ineffective		In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the <code>DefaultServlet</code> or <code>ResourceHandler</code> that is configured for showing a Listing of directory contents. The <code>jetty</code> package is vulnerable to Cross-Site Scripting (XSS). The <code>sendDirectory()</code> function in <code>ResourceService.class</code> and <code>DefaultServlet.class</code> files and the <code>doDirectory()</code> function in the <code>ResourceHandler.class</code> file use the <code>getListHTML()</code> function in the <code>Resource.class</code> file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or <code>ResourceHandler</code> that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed.
dmaap-datarouter	org.eclipse.jetty	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The <code>jetty-util</code> package running on Windows is vulnerable to sensitive Information Exposure. The <code>getListHTML()</code> method of the <code>Resource.class</code> file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception <div><input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components CLOSED</div>

Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
dmaap-dbcapi	com.fasterxml.jackson.core	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. <code>jackson-databind</code> is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed

dmaap-dbcapi	com.fasterxml.jackson.core	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-dbcapi	com.fasterxml.jackson.core	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-dbcapi	commons-codec	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32, Base64, and Base16 classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-dbcapi	org.eclipse.jetty	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-dbcapi	org.eclipse.jetty	CVE-2019-10247			In Eclipse Jetty version 7.x, 8.x, 9.2.27 and older, 9.3.26 and older, and 9.4.16 and older, the server running on any OS and Jetty version combination will reveal the configured fully qualified directory base resource location on the output of the 404 error for not finding a Context that matches the requested path. The default server behavior on jetty-distribution and jetty-home will include at the end of the Handler tree a DefaultHandler, which is responsible for reporting this 404 error, it presents the various configured contexts as HTML for users to click through to. This produced HTML includes output that contains the configured fully qualified directory base resource location for each context. The jetty package is vulnerable to sensitive Information Exposure. The handle() method of the DefaultHandler.class file discloses sensitive information via the context object. The method outputs the value of context.toString() within the error responses which will reveal the base resource path of each context.	Requesting exception  DMAAP-1324 - [DMAA P] CVE-2019-10247 vulnerability fix for DMaaP all components 
dmaap-dbcapi	org.eclipse.jetty	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-dbcapi	org.eclipse.jetty	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The jetty-util package running on Windows is vulnerable to sensitive Information Exposure. The getListHTML() method of the Resource.class file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception  DMAAP-1324 - [DMAA P] CVE-2019-10247 vulnerability fix for DMaaP all components 

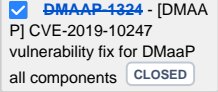
Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
------------	-------	--------------	------------------------	-----------------------	-----------------	--------

dmaap-messagerouter-dmaapclient	com.att.nsa	CVE-2017-7525			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. The application is vulnerable by using this component, when default typing is enabled.	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed
dmaap-messagerouter-dmaapclient	com.att.nsa	N/A	Ineffective		The hazelcast package is vulnerable to Remote Code Execution (RCE). Several functions in several files allow untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a file which contains a malicious serialized object. This would result in remote code execution if the application attempts to deserialize the file.	Hazelcast is not used in DMaaPCient in the way that it can be exploited. DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.
dmaap-messagerouter-dmaapclient	com.att.nsa	CVE-2014-0114	Ineffective		Apache Commons BeanUtils, as distributed in lib/commons-beanutils-1.8.0.jar in Apache Struts 1.x through 1.3.10 and in other products requiring commons-beanutils through 1.9.2, does not suppress the class property, which allows remote attackers to "manipulate" the ClassLoader and execute arbitrary code via the class parameter, as demonstrated by the passing of this parameter to the getClass method of the ActionForm object in Struts 1. Apache Commons BeanUtils is vulnerable to ClassLoader manipulation which can lead to Remote Code Execution (RCE). Access to the class property is not suppressed, exposing it by default. An attacker can construct malicious input using the class property in order to manipulate the ClassLoader potentially leading to arbitrary code execution. If you are the calling application, you are vulnerable by running this component without filtering the class property name. If this is a transitive dependency, you will want to contact the parent project to ensure they have added a mitigating control. commons-beanutils added a SuppressPropertiesBeanIntrospector which includes a specialized instance of itself as the SUPPRESS_CLASS constant in version 1.9.2 that specifically suppresses the class property. However, this is not enabled by default. We recommend filtering the class property name by using either: (1) The SUPPRESS_CLASS specialized instance of SuppressPropertiesBeanIntrospector, or (2) A custom instance of SuppressPropertiesBeanIntrospector that will suppress the class property. (3) implement a custom servlet filter as described in https://community.hpe.com/t5/Security-Research/Protect-your-Struts1-applications/ba-p/6463188#VCUfrhYvBaV .	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.
dmaap-messagerouter-dmaapclient	com.att.nsa	N/A	Ineffective		jackson-core is vulnerable to Denial of Service (DoS). The writeNumber() method in files UTF8JsonGenerator.class and WriterBasedJsonGenerator.class converts a big decimal value to its plaintext value without validating the size of the input exponent, when WRITE_BIGDECIMAL_AS_PLAIN setting is enabled. This leads to an overconsumption of Java heap memory and causes DoS. The application is vulnerable by using this component when WRITE_BIGDECIMAL_AS_PLAIN is explicitly enabled. By default, WRITE_BIGDECIMAL_AS_PLAIN is disabled. We recommend upgrading to a version of this component that is not vulnerable to this specific issue. If upgrading is not an option, we recommend disabling the WRITE_BIGDECIMAL_AS_PLAIN option.	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.
dmaap-messagerouter-dmaapclient	com.att.nsa	N/A	Ineffective		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. An attacker can exploit this vulnerability by crafting a POST request containing large amounts of data. When the data contains invalid JSON, an exception is thrown, which results in the consumption of available disk space when the error message is written to server.log along with the request data. Root Cause: dmaapClient-0.2.12-jar-with-dependencies.jar <= ReaderBasedJsonParser.class : [2.0.0-RC1, 2.8.6), dmaapClient-0.2.12-jar-with-dependencies.jar <= UTF8StreamJsonParser.class : [2.0.0-RC1, 2.8.6)	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.
dmaap-messagerouter-dmaapclient	com.att.nsa	N/A	Ineffective		The Apache httpcomponents component is vulnerable to Directory Traversal. The normalizePath() function in the UriBuilder class allows directory traversal characters such as ../. An attacker can exploit this vulnerability by sending a specially crafted request containing this sequence in the URL path, allowing the attacker to traverse beyond the allowed directory and retrieve the contents of arbitrary files from the server, leading to information disclosure. Root Cause: dmaapClient-0.2.12-jar-with-dependencies.jar <= UriBuilder.class : [4.2.1-RC1, 4.5.3)	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.
dmaap-messagerouter-dmaapclient	com.att.nsa	N/A	Ineffective		The jackson-databind package is vulnerable to XML eXternal Entity Reference (XXE) attacks. The parserFactory object, an instance of the DocumentBuilderFactory type in the DOMDeserialzier.class file does not prevent expansion of external entities by default. An attacker can exploit this vulnerability by submitting crafted XML which when parsed by the application leads to XXE attacks and further security issues. Root Cause: dmaapClient-0.2.12-jar-with-dependencies.jar <= DOMDeserialzier.class : (, 2.7.6)	DMaaPCient, as the name suggests, is a client side library provided as reference only example code provided for clients to get started and is not intended to be used as a production code as-is. No action needed.

dmaap-messagerouter-dmaapclient	com.att.nsa	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>Base64</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter-dmaapclient	com.fasterxml.jackson.core	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. <code>jackson-databind</code> is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter-dmaapclient	com.fasterxml.jackson.core	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. <code>jackson-databind</code> is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-dmaapclient	com.fasterxml.jackson.core	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation <code>jackson-databind</code> is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-dmaapclient	commons-codec	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>Base64</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.

Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. <code>jackson-databind</code> is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. <code>jackson-databind</code> is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed

dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload.Explanation 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The jetty-util package running on Windows is vulnerable to sensitive Information Exposure. The getListHTML() method of the Resource.class file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception 
dmaap-messagerouter-docker	com.att.ajsc	CVE-2019-10247			In Eclipse Jetty version 7.x, 8.x, 9.2.27 and older, 9.3.26 and older, and 9.4.16 and older, the server running on any OS and Jetty version combination will reveal the configured fully qualified directory base resource location on the output of the 404 error for not finding a Context that matches the requested path. The default server behavior on jetty-distribution and jetty-home will include at the end of the Handler tree a DefaultHandler, which is responsible for reporting this 404 error, it presents the various configured contexts as HTML for users to click through to. This produced HTML includes output that contains the configured fully qualified directory base resource location for each context. The jetty package is vulnerable to sensitive Information Exposure. The handle() method of the DefaultHandler.class file discloses sensitive information via the context object. The method outputs the value of context.toString() within the error responses which will reveal the base resource path of each context.	Requesting exception 
dmaap-messagerouter-docker	com.att.ajsc	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32, Base64, and BCodec classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter-docker	com.att.ajsc	N/A			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. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the src attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review
dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. jackson-databind is vulnerable to Information Exposure via Deserialization of Untrusted Data. 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 which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed

dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter-docker	com.att.ajsc	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The jetty-util package running on Windows is vulnerable to sensitive Information Exposure. The getListHTML() method of the Resource.class file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception 
dmaap-messagerouter-docker	com.att.ajsc	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32, Base64, and BCodec classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter-docker	com.att.ajsc	N/A			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. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the src attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review
dmaap-messagerouter-docker	com.fasterxml.jackson.core	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. jackson-databind is vulnerable to Information Exposure via Deserialization of Untrusted Data. 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 which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	com.fasterxml.jackson.core	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed

dmaap-messagerouter-docker	com.fasterxml.jackson.core	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload.Explanation 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-docker	org.apache.kafka	CVE-2018-17196			In Apache Kafka versions between 0.11.0.0 and 2.1.0, it is possible to manually craft a Produce request which bypasses transaction/idempotent ACL validation. Only authenticated clients with Write permission on the respective topics are able to exploit this vulnerability. Users should upgrade to 2.1.1 or later where this vulnerability has been fixed.	Requesting exception <input checked="" type="checkbox"/> DMAAP-1326 - [MR] CVE-2018-17196 fix vulnerability <input type="button" value="CLOSED"/>
dmaap-messagerouter-docker	commons-codec	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32, Base64, and BCodec classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter-docker	org.apache.zookeeper	CVE-2019-0201			An issue is present in Apache ZooKeeper 1.0.0 to 3.4.13 and 3.5.0-alpha to 3.5.4-beta. ZooKeeper's getACL() command doesn't check any permission when retrieves the ACLs of the requested node and returns all information contained in the ACL Id field as plaintext string. DigestAuthenticationProvider overloads the Id field with the hash value that is used for user authentication. As a consequence, if Digest Authentication is in use, the unsalted hash value will be disclosed by getACL() request for unauthenticated or unprivileged users. Apache ZooKeeper contains an Improper Access Control vulnerability. The processRequest() method in the FinalRequestProcessor class returns ACL ID field information as plaintext without checking the permissions of the requesting user. The ACL ID field may contain sensitive values, as is the case when using the Digest AuthenticationProvider as it injects the ACL ID field with sensitive, unsalted hash values that are used for user authentication. A remote attacker with the ability to call upon getACL() for an affected node can leverage this vulnerability to exfiltrate the aforementioned hashes which may be used to perform various other attacks. The application is vulnerable by using this component with the DigestAuthenticationProvider authentication method.	Requesting exception <input checked="" type="checkbox"/> DMAAP-1326 - [MR] CVE-2019-0201 fix vulnerability <input type="button" value="CLOSED"/>
dmaap-messagerouter-docker	org.eclipse.jetty	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter-docker	org.eclipse.jetty	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The jetty-util package running on Windows is vulnerable to sensitive Information Exposure. The getListHTML() method of the Resource.class file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception <input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components <input type="button" value="CLOSED"/>
dmaap-messagerouter-docker	org.springframework.security	N/A			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. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the src attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review

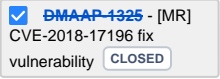
Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
------------	-------	--------------	------------------------	-----------------------	-----------------	--------

dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. jackson-databind is vulnerable to Information Exposure via Deserialization of Untrusted Data. 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 which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the DefaultServlet or ResourceHandler that is configured for showing a Listing of directory contents. The jetty package is vulnerable to Cross-Site Scripting (XSS). The sendDirectory() function in ResourceService.class and DefaultServlet.class files and the doDirectory() function in the ResourceHandler.class file use the getListHTML() function in the Resource.class file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The jetty-util package running on Windows is vulnerable to sensitive Information Exposure. The getListHTML() method of the Resource.class file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception <div><input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components CLOSED</div>
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2019-10247			In Eclipse Jetty version 7.x, 8.x, 9.2.27 and older, 9.3.26 and older, and 9.4.16 and older, the server running on any OS and Jetty version combination will reveal the configured fully qualified directory base resource location on the output of the 404 error for not finding a Context that matches the requested path. The default server behavior on jetty-distribution and jetty-home will include at the end of the Handler tree a DefaultHandler, which is responsible for reporting this 404 error, it presents the various configured contexts as HTML for users to click through to. This produced HTML includes output that contains the configured fully qualified directory base resource location for each context. The jetty package is vulnerable to sensitive Information Exposure. The handle() method of the DefaultHandler.class file discloses sensitive information via the context object. The method outputs the value of context.toString() within the error responses which will reveal the base resource path of each context.	Requesting exception <div><input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components CLOSED</div>
dmaap-messagerouter - messageservice	com.att.ajsc	N/A			The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32, Base64, and BCodec classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.

dmaap-messagerouter - messageservice	com.att.ajsc	N/A			The <code>spring-security-web</code> package is vulnerable to Cross-Site Request Forgery (CSRF). The <code>doFilter()</code> method in the <code>SwitchUserFilter</code> , which is reachable via a GET request, does not require any form of confirmation that the user sending the request intended to do so. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the <code>src</code> attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. <code>jackson-databind</code> is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. <code>jackson-databind</code> is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation <code>jackson-databind</code> is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the <code>DefaultServlet</code> or <code>ResourceHandler</code> that is configured for showing a Listing of directory contents. The <code>jetty</code> package is vulnerable to Cross-Site Scripting (XSS). The <code>sendDirectory()</code> function in <code>ResourceService.class</code> and <code>DefaultServlet.class</code> files and the <code>doDirectory()</code> function in the <code>ResourceHandler.class</code> file use the <code>getListHTML()</code> function in the <code>Resource.class</code> file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or <code>ResourceHandler</code> that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter - messageservice	com.att.ajsc	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The <code>jetty-util</code> package running on Windows is vulnerable to sensitive Information Exposure. The <code>getListHTML()</code> method of the <code>Resource.class</code> file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception 
dmaap-messagerouter - messageservice	com.att.ajsc	N/A			The Apache <code>commons-codec</code> package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>BCodec</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter - messageservice	com.att.ajsc	N/A			The <code>spring-security-web</code> package is vulnerable to Cross-Site Request Forgery (CSRF). The <code>doFilter()</code> method in the <code>SwitchUserFilter</code> , which is reachable via a GET request, does not require any form of confirmation that the user sending the request intended to do so. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the <code>src</code> attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review

dmaap-messagerouter - messageservice	com. fasterxml. jackson. core	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. jackson-databind is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com. fasterxml. jackson. core	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. jackson-databind is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	com. fasterxml. jackson. core	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation jackson-databind is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter - messageservice	org. apache. kafka	CVE-2018-17196			In Apache Kafka versions between 0.11.0.0 and 2.1.0, it is possible to manually craft a Produce request which bypasses transaction/idempotent ACL validation. Only authenticated clients with Write permission on the respective topics are able to exploit this vulnerability. Users should upgrade to 2.1.1 or later where this vulnerability has been fixed.	Requesting exception <input checked="" type="checkbox"/> DMAAP-1326 - [MR] CVE-2018-17196 fix vulnerability <input type="button" value="CLOSED"/>
dmaap-messagerouter - messageservice	commons-codec	N/A			The Apache <code>commons-codec</code> package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>BCodec</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.
dmaap-messagerouter - messageservice	org. apache. zookeeper	CVE-2019-0201			An issue is present in Apache ZooKeeper 1.0.0 to 3.4.13 and 3.5.0-alpha to 3.5.4-beta. ZooKeeper's <code>getACL()</code> command doesn't check any permission when retrieves the ACLs of the requested node and returns all information contained in the ACL Id field as plaintext string. <code>DigestAuthenticationProvider</code> overloads the Id field with the hash value that is used for user authentication. As a consequence, if Digest Authentication is in use, the unsalted hash value will be disclosed by <code>getACL()</code> request for unauthenticated or unprivileged users. Apache ZooKeeper contains an Improper Access Control vulnerability. The <code>processRequest()</code> method in the <code>FinalRequestProcessor</code> class returns ACL ID field information as plaintext without checking the permissions of the requesting user. The ACL ID field may contain sensitive values, as is the case when using the <code>DigestAuthenticationProvider</code> as it injects the ACL ID field with sensitive, unsalted hash values that are used for user authentication. A remote attacker with the ability to call upon <code>getACL()</code> for an affected node can leverage this vulnerability to exfiltrate the aforementioned hashes which may be used to perform various other attacks. The application is vulnerable by using this component with the <code>DigestAuthenticationProvider</code> authentication method.	Requesting exception <input checked="" type="checkbox"/> DMAAP-1326 - [MR] CVE-2019-0201 fix vulnerability <input type="button" value="CLOSED"/>
dmaap-messagerouter - messageservice	org. eclipse. jetty	CVE-2019-10241			In Eclipse Jetty version 9.2.26 and older, 9.3.25 and older, and 9.4.15 and older, the server is vulnerable to XSS conditions if a remote client USES a specially formatted URL against the <code>DefaultServlet</code> or <code>ResourceHandler</code> that is configured for showing a Listing of directory contents. The <code>jetty</code> package is vulnerable to Cross-Site Scripting (XSS). The <code>sendDirectory()</code> function in <code>ResourceService.class</code> and <code>DefaultServlet.class</code> files and the <code>doDirectory()</code> function in the <code>ResourceHandler.class</code> file use the <code>getListHTML()</code> function in the <code>Resource.class</code> file to fetch resource list as an HTML directory listing. This allows any JavaScript present in the list items to get fetched and rendered without proper sanitization of user-supplied input, leading to XSS.	In DMaaP there is no servlet or ResourceHandler that is configured to show a directory listing. This vulnerability will not be exploited with DMaaP. No action needed
dmaap-messagerouter - messageservice	org. eclipse. jetty	CVE-2019-10246			In Eclipse Jetty version 9.2.27, 9.3.26, and 9.4.16, the server running on Windows is vulnerable to exposure of the fully qualified Base Resource directory name on Windows to a remote client when it is configured for showing a Listing of directory contents. This information reveal is restricted to only the content in the configured base resource directories. The <code>jetty-util</code> package running on Windows is vulnerable to sensitive Information Exposure. The <code>getListHTML()</code> method of the <code>Resource.class</code> file reveals the resource base path as it does not properly generate HTML content and includes the base path in the result.	Request exception <input checked="" type="checkbox"/> DMAAP-1324 - [DMAAP] CVE-2019-10247 vulnerability fix for DMaaP all components <input type="button" value="CLOSED"/>

dmaap-messagerouter-messageservice	org. springframework. security	N/A			The <code>spring-security-web</code> package is vulnerable to Cross-Site Request Forgery (CSRF). The <code>doFilter()</code> method in the <code>SwitchUserFilter</code> , which is reachable via a GET request, does not require any form of confirmation that the user sending the request intended to do so. An attacker can exploit this vulnerability by crafting a malicious application containing links to the vulnerable endpoint, HTML tags that use the vulnerable endpoint in the <code>src</code> attribute, or malicious JavaScript designed to send the request to the vulnerable endpoint. When a victim visits the malicious page, their browser will be made to send requests to the vulnerable endpoint, taking action as the victim without the victim's knowledge or consent. The application is vulnerable by using this component if the Switch User Processing Filter is configured.	Under review
------------------------------------	--------------------------------	-----	--	--	--	--------------

Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
dmaap-messagerouter-msgtr	com. fasterxml. jackson. core	CVE-2018-11307	Ineffective		An issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.5. Use of Jackson default typing along with a gadget class from iBatis allows exfiltration of content. Fixed in 2.7.9.4, 2.8.11.2, and 2.9.6. jackson-databind is vulnerable to Information Exposure via Deserialization of Untrusted Data. The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> class allows untrusted Java objects to be deserialized. A remote attacker can exploit this by uploading a malicious serialized object which will result in the exfiltration of sensitive information if the application attempts to deserialize it. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP does not use iBatis and hence this vulnerability cannot be exploited. No action needed
dmaap-messagerouter-msgtr	com. fasterxml. jackson. core	CVE-2018-12022	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Jodd-db jar (for database access for the Jodd framework) in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. jackson-databind is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Jodd-db jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-msgtr	com. fasterxml. jackson. core	CVE-2018-12023	Ineffective		An issue was discovered in FasterXML jackson-databind prior to 2.7.9.4, 2.8.11.2, and 2.9.6. When Default Typing is enabled (either globally or for a specific property), the service has the Oracle JDBC jar in the classpath, and an attacker can provide an LDAP service to access, it is possible to make the service execute a malicious payload. Explanation jackson-databind is vulnerable to Remote Code Execution (RCE). The <code>validateSubType()</code> function in the <code>SubTypeValidator</code> 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. The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialized.	DMaaP components do not use Oracle JDBC jars. This vulnerability cannot be exploited. No action needed
dmaap-messagerouter-msgtr	org. apache. kafka	CVE-2018-17196			In Apache Kafka versions between 0.11.0.0 and 2.1.0, it is possible to manually craft a Produce request which bypasses transaction/idempotent ACL validation. Only authenticated clients with Write permission on the respective topics are able to exploit this vulnerability. Users should upgrade to 2.1.1 or later where this vulnerability has been fixed.	Requesting exception 
dmaap-messagerouter-msgtr	commons-codec	N/A			The Apache <code>commons-codec</code> package contains an Improper Input Validation vulnerability. The <code>decode()</code> method in the <code>Base32</code> , <code>Base64</code> , and <code>BCodec</code> classes fails to reject malformed Base32 and Base64 encoded strings and consequently decodes them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.	The Base64 functionality identified in this vulnerability cannot be exploited as the DMaaP components in rare case are using Base64. decode only to decode the Authorization header, which if modified by a malicious user is only going to result in Authorization errors. This vulnerability will not directly impact DMaaP. No action needed.