Casablanca APPC Security/Vulnerability Report

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

Repository	Group	Impact Analysis	Action
appc	org.codehaus.jackson	There is no non vulnerable version of this component. False Positive Explaination: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. appc doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on appc. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization This is a dependency indirectly from jackson-jaxrs. We do not use Jackson-mapper-asl directly and do not use createBeanDeserializer() function which has the vulnerability. We were unable to find any reference to this Vulnerability from appc code.	No Action Required
appc	org.codehaus.jackson	There is no non vulnerable version of this component. False Positive Explaination: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. appc doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on appc. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization This is a dependency indirectly from jersey-json. We do not use Jackson-mapper-asl directly and do not use createBeanDeserializer() function which has the vulnerability. We were unable to find any reference to this Vulnerability from appc code.	No Action Required
appc	com.fasterxml.jackson.core	There is no non vulnerable version of this component. False Positive Explaination: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. appc doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on appc. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization appc codes does not use ObjectMapper.setDefaultTyping() https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dg/appc-dg-shared/appc-dg-dependency-model/src/main/java/org/onap/appc/dg/dependencymanager/helper/DependencyModelParser.java;h=11bff0237ce42af835d0afe752c6af0785182bb4;hb=15f4f0fa7c142be425f39647461c438d0e68103f	No Action Required
appc	com.fasterxml.jackson.core	There is no non vulnerable version of this component. False Positive Explaination: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. appc doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on appc. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization	No Action Required

appc	com.fasterxml.jackson.core	There is no non vulnerable version of this component. False Postive Explaination: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. appc doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on appc. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization appc codes does not use ObjectMapper.setDefaultTyping() https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-common/src/main/java/org/onap/appc/util/JsonUtil.java;h=7e6f5ef8d000bd2037cb7405f43dc1eb0cebda50;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-adapters/appc-dmaap-adapter/appc-message-adapter-api/src/main/java/org/onap/appc/adapter/message/event/EventMessage.java;h=d64d6d0cebd1728ef9a974eb802f704b9ab256c7;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-event-listener/appc-event-listener-bundle/src/main/java/org/onap/appc/listener/LCM/conv/Converter.java;h=6e303a5ff2cbb1269cca6a8dae8ccef4ca124d9b;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-event-listener/appc-event-listener-bundle/src/main/java/org/onap/appc/listener/util/Mapper.java;h=bd77041ecef1b4978cd09013070920d73eb612d1;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-config/appc-config-params/provider/src/main/java/org/onap/sdnc/config/params/parser/PropertyDefinitionNode.java;h=a9707b6a2d73776bcc52a31c76a03dd90f5b89e6;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-config/appc-config-params/provider/src/main/java/org/onap/sdnc/config/params/transformer/ArtificatTransformer.java;h=4ac4fbb9f9d11fe43397d843249c2b16f0ee3722;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-config/appc-config-params/provider/src/main/java/org/onap/sdnc/config/params/transformer/tosca/ArtifactProcessorImpl.java;h=f5dd7b6dbc00b6f6e33a03b37a89b2f87f403d6a;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-config/appc-data-services/provider/src/main/java/org/onap/appc/data/services/node/ConfigResourceNode.java;h=856648210d87d5e8a7b79b7a830dba38da89c55b;hb=15f4f0fa7c142be425f39647461c438d0e68103f https://gerrit.onap.org/r/gitweb?p=appc.git;a=tree;f=appc-config/appc-flow-controller/provider/src/main/java/org/onap/appc/flow/controller/node;h=68460c525de553dff2f626cccb1c4de48b9b6b5f;hb=15f4f0fa7c142be425f39647461c438d0e68103f	No Action Required
appc	com.att.nsa	org.onap.dmaap.messagerouter.dmaapclient has the 5 security vulnerabilities , out of these 4 security issues are related to the com.att.nsa:dmaapclient and another is related to the Jackson-core.jar, which we can't fix as all the versions are vulnerable. DMaaP client is not using the jackson-core.jar, in such a way that it will cause the vulnerability. a ticket #54030 with the LF by dmaap team. Please refer the following link for more details. Dmaap Security/Vulnerability - Beijing	~~APPC-1176~~ - dmaapClient-1.1.5 jar security report CLOSED created to track this issue
appc	org.apache.karaf.shell	It comes with org.onap.ccsdk.sli.core:dblib-provider:jar, org.opendaylight.controller:opendaylight-karaf-empty Apache karaf is vulnerable to `Improper Access Control`. Multiple functions in `SessionFactoryImpl`, `SecuredCommand` and `SecuredSessionFactoryImpl` class files process and execute the external commands without checking the scope of the input commands. This may allow an attacker to read from, or write to any file on the filesystem to which the Karaf process user has access.	*request CCSDK and OpenDayLight to fix
appc	org.apache.karaf.jaas	False Positive Explanation 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. We do not use `normalizePath()` function which has the vulnerability. We were unable to find any reference to this Vulnerability from appc code. and this is indirect from org.onap.ccsdk.sli.core:dblib-provider:jar:0.3.0-SNAPSHOT	No Action Required

appc	com.fasterxml.jackson.core	False Positive Explanation `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. appc doesn't use `UTF8StreamJsonParser` and `ReaderBasedJsonParser` classes https://github.com/FasterXML/jackson-core/pull/322 appc codes using JsonParser/JasonProcessingExection/type.TypeReference: https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-common/src/main/java/org/onap/appc/util/JsonUtil.java;h=7e6f5ef8d000bd2037cb7405f43dc1eb0cebda50;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=tree;f=appc-config/appc-flow-controller/provider/src/main/java/org/onap/appc/flow/controller/node;h=68460c525de553dff2f626cccb1c4de48b9b6b5f;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dg/appc-dg-shared/appc-dg-mdsal-store/appc-dg-mdsal-bundle/src/main/java/org/onap/appc/mdsal/impl/MDSALStoreImpl.java;h=fcd315bf6be4f8756c13b1663f8424d57c9d7e81;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dg/appc-dg-shared/appc-dg-netconf/src/main/java/org/onap/appc/dg/netconf/impl/NetconfDBPluginImpl.java;h=459ece9c1ead17a579895e344b15116e5bb1661a;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dg/appc-dg-shared/appc-dg-ssh/src/main/java/org/onap/appc/dg/ssh/impl/SshDBPluginImpl.java;h=c3dfc61d6930120a22eb2f566b33cdbb683e40a0;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dispatcher/appc-request-handler/appc-request-handler-core/src/main/java/org/onap/appc/messageadapter/impl/MessageAdapterImpl.java;h=ecc7f729c76fa85d034e4def5cbf690543c6bcbb;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-dispatcher/appc-request-handler/appc-request-handler-core/src/main/java/org/onap/appc/requesthandler/conv/Converter.java;h=5aac95a42bc230c5c7b7ea2fbbbf142bf0ea2df3;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-event-listener/appc-event-listener-bundle/src/main/java/org/onap/appc/listener/LCM/conv/Converter.java;h=6e303a5ff2cbb1269cca6a8dae8ccef4ca124d9b;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-event-listener/appc-event-listener-bundle/src/main/java/org/onap/appc/listener/LCM/impl/WorkerImpl.java;h=acf6d8bccc2dceeca918429e047c05bc441498b1;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-inbound/appc-design-services/provider/src/main/java/org/onap/appc/design/dbervices/DesignDBService.java;h=83ef0f914873e21bfd6648e6d593b7a00fb5b10e;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-inbound/appc-design-services/provider/src/main/java/org/onap/appc/design/validator/ValidatorService.java;h=7ba518d212cf9176294850c44b9fb0ac180c5248;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-oam/appc-oam-bundle/src/main/java/org/onap/appc/oam/messageadapter/Converter.java;h=152ffc9ccc20fd4aa464f24ab58ae8715fdb7d8f;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-oam/appc-oam-bundle/src/main/java/org/onap/appc/oam/messageadapter/MessageAdapter.java;h=91836cb406fd305588bc1a4d32e1a98964e4ddda;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-sdc-listener/appc-sdc-listener-bundle/src/main/java/org/onap/appc/sdc/artifacts/helper/DependencyModelGenerator.java;h=62212d74ca2aab916281cd763783c1666a9d07ec;hb=117c7e7210f00da7011275be4347aae8d500002a https://gerrit.onap.org/r/gitweb?p=appc.git;a=blob;f=appc-sequence-generator/appc-sequence-generator-bundle/src/main/java/org/onap/appc/seqgen/dgplugin/impl/SequenceGeneratorPluginImpl.java;h=f99ca4cfb0ef3cea75074e19a0da89c55de6d6c3;hb=117c7e7210f00da7011275be4347aae8d500002a	No action required

appc	org.apache.httpcomponents	Explanation 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. Detection The application is vulnerable by using this component. Recommendation We do not use `normalizePath()` function which has the vulnerability. We were unable to find any reference to this Vulnerability from appc code. This is indirect from org.onap.ccsdk.sli.adaptors:aai-service-provider:jar:0.3.0-SNAPSHOT	*request CCSDK (CCSDK may depend on OpenDayLight) to fix
appc	org.apache.httpcomponents	False Positive Explanation 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. Detection The application is vulnerable by using this component. We investigated from mvn tree output. Found out this jar only included in test purpose. Please see the tree dependency below: org.apache.maven.wagon:wagon-http:jar:2.10:test [INFO] | +- org.opendaylight.odlparent:karaf-util:jar:3.1.3:test [INFO] | | \- org.apache.maven.wagon:wagon-http:jar:2.10:test [INFO] | | +- org.apache.maven.wagon:wagon-http-shared:jar:2.10:test [INFO] | | | +- org.jsoup:jsoup:jar:1.7.2:test [INFO] | | | \- commons-lang:commons-lang:jar:2.6:test [INFO] | | +- org.apache.httpcomponents:httpclient:jar:4.3.5:test [INFO] | | | \- commons-codec:commons-codec:jar:1.11:test [INFO] | | +- org.apache.httpcomponents:httpcore:jar:4.3.2:test [INFO] | | \- org.apache.maven.wagon:wagon-provider-api:jar:2.10:test [INFO] | | \- org.codehaus.plexus:plexus-utils:jar:3.0.15:test	No action required
appc	org.glassfish.grizzly	False Positive Library not used by APPC code directly, but is contains in cdp-pal library. The dependency comes from cdp-pal; however, this should not be a security concern as CDP-PAL/woorea does not host any urls for incoming GET requests and from what we read about the vulnerability it should not apply as grizzly-http is only used for outgoing calls. It is not used to allow incoming get requests.	**request CDP-PAL to fix
appc	dom4j	False Positve 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. However QName.Qname class not used by APPC code. Appc code only uses QName.localName	No action required
appc	com.google.guava	this package comes with org.opendaylight.controller:sal-binding-api:jar:1.6.1, org.opendaylight.mdsal:mdsal-binding-api:jar:2.3.1 , org.onap.ccsdk.sli.core:dblib-provider:jar:0.2.3, org.opendaylight.controller:sal-binding-api:jar:1.6.1, org.opendaylight.odlparent:features-test:jar:2.0.5	Need to follow up will CCSDK and OpenDayLight community
appc	com.h2database	False Positive only occurred in appc-test-dependencies. This APPC: appc-test-dependencies package is only used during junit/mockito test.	No action required

appc	com.jcraft	False Postive library is contained in cdp-pal libraary. ONAP is not using Windows envrionment.	No action required
appc	javax.mail	this package comes with org.opendaylight.controller:config-persister-directory-xml-adapter:jar:0.8.3	*request OpenDayLight to fix
appc	com.sun.mail	False Positve this package comes with javax:javaee-api:jar:7.0 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. However `UniqueValue` class not used by APPC code.	No action required
appc	io.netty	False Positive this package comes with com.datastax.cassandra:cassandra-driver-core:jar:3.0.8 The `netty-handler` package is vulnerable to Improper Certificate Validation. The `OpenSsl` class and the `setSSLParameters()` method in the `ReferenceCountedOpenSslEngine` class do not validate hostnames of SSL certificates. An attacker can exploit this vulnerability by executing a Man-in-the-Middle (MitM) attack in order to intercept requests and provide a valid attacker-controlled certificate to the client. This allows the attacker to decrypt, read, and modify data in transit, thus effectively spoofing the vulnerable server. Detection The application is vulnerable by using the `OpenSSL` feature of this this component. However The `OpenSsl` class and the `setSSLParameters()` method in the `ReferenceCountedOpenSslEngine` class is not used by APPC code.	No action required
appc/cdt	com.fasterxml.jackson.core	False Positive 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. Detection The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. appc codes below does not use `createBeanDeserializer()` https://gerrit.onap.org/r/gitweb?p=appc/cdt.git;a=blob;f=CdtProxyService/src/main/java/org/onap/appc/cdt/service/MainApplication.java;h=c1605196866286a02a6a3fc6f71009e6f37ab8c5;hb=refs/heads/master	No action required
appc/deployment	com.fasterxml.jackson.core	False Positive 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. Detection The application is vulnerable by using this component, when default typing is enabled and passing in untrusted data to be deserialization. appc codes below does not use `createBeanDeserializer()` https://gerrit.onap.org/r/gitweb?p=appc/cdt.git;a=blob;f=CdtProxyService/src/main/java/org/onap/appc/cdt/service/MainApplication.java;h=c1605196866286a02a6a3fc6f71009e6f37ab8c5;hb=refs/heads/master	No action required

appc /deployment	com. github. fonimus	False Positive 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. Note: This vulnerability exists due to the incomplete fix for CVE-2017-7525, CVE-2017-15095, CVE-2017-17485, CVE-2018-5968, and CVE-2018-7489. Evidence of this can be found at https://pivotal.io/security/cve-2017-4995: appc codes below does not use `createBeanDeserializer()` https://gerrit.onap.org/r/gitweb?p=appc/cdt.git;a=blob;f=CdtProxyService/src/main/java/org/onap/appc/cdt/service/MainApplication.java;h=c1605196866286a02a6a3fc6f71009e6f37ab8c5;hb=refs/heads/master	No action required
appc /deployment	common s-beansutils	This pacakge comes from aaf-shiro-aafrealm-osgi-bundle.jar	AAF addresses: AAF R3 Casablanca Security /Vulnerability Threat
appc /deployment	org. apache. shiro	This package comes from aaf-shiro-aafrealm-osgi-bundle.jar	AAF addresses: AAF R3 (Casablanca) Security /Vulnerability Threat
appc /deployment	com. google. guava	This pacakge comes from org.opendaylight.mdsal:yang-binding:jar:0.10.1	*request OpenDayLight to fix