



# Portal Platform Security/Vulnerability Report (Dublin Release)









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

## High-level mitigation plan:

Regarding known issues like “DOS, Remote Code Execution (RCE), CORS attack, HTTP request smuggling”, the Portal’s code is not exposing these vulnerabilities directly due to many layers of encapsulation by APIs, so these are most likely false positives reported by NexusIQ scan, however to be on safe side the mitigation plan is to deploy Portal platform in a secure environment e.g. in private network inside the company firewall.

Repository	Group	Impact Analysis	Action
portal	com. fasterxml. jackson. core	<p>False positive.</p> <p>Analysis: This vulnerability is not exposed from the portal's code, because</p> <ol style="list-style-type: none"><li>1. The portal does not pass any untrusted data for deserialization, as there is XSS/XSRF validation enabled in the portal's backend code.</li><li>2. and the default typing (ObjectMapper.setDefaultTyping()) is not called as we use concrete java types.</li><li>3. and we use Spring Security 4.2.3 as recommended in the nexus-iq report.</li></ol> <p>Spring version 4.2.3 will take care of this.</p> <p>Comments from Nexus-IQ: Spring Security has provided their own fix for this vulnerability (CVE-2017-4995). If this component is being used as part of Spring Security, then you are not vulnerable if you are running Spring Security 4.2.3.RELEASE or greater for 4.x or Spring Security 5.0.0.M2 or greater for 5.x.</p>	Not vulnerable in ONAP
portal	moments  moment 2.1.0	<p>All available versions of moment.js are vulnerable. Upgrade is not an option.</p> <p>Analysis: Not vulnerable as all our date fields are reformatted and validated before being submitted. See below</p> <p>CVE 185 information: The moment package is vulnerable to Regular Expression Denial of Service (ReDoS). The monthsShortRegex(), monthsRegex(), weekdaysRegex(), weekdaysShortRegex(), and weekdaysMinRegex() functions in the moment.js, moment-with-locales.js, and regex.js files use a vulnerable regular expression while parsing the date input. A remote attacker can exploit this vulnerability by crafting a date input containing a very long sequence of repetitive characters which, when parsed, consumes available CPU resources and results in Denial Of Service. See JIRA ticket: <a href="#">PORTAL-531</a></p>	upgrade to moment 2.11.2+
portal, portal- sdk	elasticse arch : 2.2.0	<p>Description from CVElasticsearch Alerting and Monitoring in versions before 6.4.1 or 5.6.12 have an information disclosure issue when secrets are configured via the API. The Elasticsearch _cluster/settings API, when queried, could leak sensitive configuration information such as passwords, tokens, or usernames. This could allow an authenticated Elasticsearch user to improperly view these details.Explanation elasticsearch is vulnerable to Information Disclosure. The renderResponse() method in the RestClusterGetSettingsAction class fails to filter certain settings from the ClusterGetSettingsResponse object, and consequently exposes potentially sensitive information via the /_cluster/settings API endpoint. A remote authenticated attacker can exploit this vulnerability by sending a request to the affected cluster endpoint. This will result in the exposure of any sensitive information contained therein. See Jira ticket: <a href="#">PORTAL-532</a></p>	upgrade of Elasticsearch Alerting and Monitoring to versions after 6.4.1 or 5.6.12
portal, portal- sdk	angular	<p>Analysis: Cannot upgrade angular as this will require changes on all the Portal pages.</p> <p>From our analysis the vulnerability cannot be exploited because the portal application follows the below design recommendations provided by nexus-iq report.</p> <p>Recommendation by nexus-iq for this vulnerability (SONATYPE-2016-0064):</p> <p><b>It's best to design your application in such a way that users cannot change client-side templates.</b></p> <ul style="list-style-type: none"><li>• Do not mix client and server templates</li><li>• Do not use user input to generate templates dynamically</li><li>• Do not run user input through \$scope.\$eval (or any of the other expression parsing functions listed above)</li><li>• Consider using {@link ng.directive:ngCsp CSP} (but don't rely only on CSP)</li></ul>	Not vulnerable in ONAP
portal, portal- sdk	angular- sanitize 1.5.0,  Angularjs	<p>Explanation AngularJS is vulnerable to Cross-Site Scripting (XSS). The \$SanitizeProvider() function in the sanitize.js file doesn't account for user input within the xml:base attribute SVG anchors. A remote attacker can exploit this vulnerability by injecting malicious JavaScript into the xml:base attribute, which results in script execution when rendered by the browser.</p> <p>Detection</p> <p>The application is vulnerable by using this component only when enableSvg is enabled, and when using Firefox. By default, the svgEnabled is set to false in 1.5+ versions. See Jira ticket: <a href="#">PORTAL-533</a></p>	We will perform the upgrade along with angular.js. in further versions by default, the svgEnabled is set to false, so upgrade should be considered to 1.5+.

portal, portal-sdk	angular-ui-grid 3.0.7	<p><b>Explanation</b></p> <p>The ui-grid package is susceptible to CSV Macro Injection. The <code>exporter.js</code> file quotes strings in double quotes when exporting to CSV files. An attacker could potentially exploit this behavior by injecting a macro command into a cell in a spreadsheet, having a victim export that spreadsheet as a CSV and loading it into a local copy of Microsoft Excel, at which point the macro can execute arbitrary commands against the victim's computer.</p> <p><i>Advisory Deviation Notice:</i> The Sonatype security research team discovered that the vulnerability is present from version 3.0.0-rc.1 onward, and that the attack can take place as described in the associated issue, despite quoting strings on export.</p>	Will check if export function is being used. If not, we are not vulnerable.
portal	org. webjars. bower	<p><b>Explanation</b></p> <p>The AngularJS framework is vulnerable to Remote Code Execution (RCE) and Cross-Site Scripting (XSS). The <code>ensureSafeAssignContext()</code> function in <code>parse.js</code> processes malicious expressions that access the constructors. A remote attacker can exploit this vulnerability by crafting malicious expressions that, when processed, result in execution of arbitrary code.</p> <p><b>Recommendation</b></p> <p><i>Each version of Angular 1 up to, but not including 1.6, contained an expression sandbox, which reduced the surface area of the vulnerability but never removed it. In Angular 1.6 we removed this sandbox as developers kept relying upon it as a security feature even though it was always possible to access arbitrary JavaScript code if one could control the Angular templates or expressions of applications.</i></p> <p><i>Control of the Angular templates makes applications vulnerable even if there was a completely secure sandbox:</i></p> <p>See Jira ticket: <a href="#">PORTAL-533</a></p>	Should be the same comments as for angular.js. We will perform the upgrade along with angular.js.
portal	commons-beanutils	<p>All available versions of commons-beanutils are vulnerable. Upgrade is not an option.</p> <p><b>Analysis:</b> The portal code do not use classloader so it is not vulnerable in ONAP.</p> <p><b>CVE CWE: 20</b></p> <p><b>Description from CVE</b></p> <p>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 <code>getClass</code> method of the <code>ActionForm</code> object in Struts 1.</p>	Not vulnerable in ONAP
portal-sdk	org. apache. poi	<p><b>Analysis:</b> Not vulnerable as we do not use POI to read documents. We use only to generate XLS from our own data.</p> <p><b>CVE CWE:399:</b></p> <p>Apache POI in versions prior to release 3.17 are vulnerable to Denial of Service Attacks: 1) Infinite Loops while parsing crafted WMF, EMF, MSG and macros (POI bugs 61338 and 61294), and 2) Out of Memory Exceptions while parsing crafted DOC, PPT and XLS (POI bugs 52372 and 61295).</p> <p> <a href="#">PORTAL-446</a> - POI <span>CLOSED</span></p>	Not vulnerable in ONAP
portal, portal-sdk	org. springframework	<p>The impact of the springframework library is all over the project. So have to be very careful in upgrading the versions.</p> <p>At least trying to resolve the multiple version use in Dublin -</p> <p><input checked="" type="checkbox"/> <a href="#">PORTAL-423</a> - Align springframework version among all poms <span>CLOSED</span></p>	Request exception
portal, portal-sdk	io.netty : netty-handler : 4.0.56. Final	<p>Not clear what is the issue based on the Nexus IQ report information.</p> <p>See Jira ticket <a href="#">PORTAL-534</a></p>	Need to upgrade to version 4.1.10.final+
portal, portal-sdk	commons-fileupload	<p>If not false positive, can be handled with the new version upgrade which do not have vulnerability.</p> <p> <a href="#">PORTAL-443</a> - commons-fileupload <span>CLOSED</span></p> <p><b>Explanation</b></p> <p>Apache Commons FileUpload contains a resource leak which may lead to a Denial of Service (DoS) attack.</p>	Target fix in Dublin release

portal-sdk	xerces	<p>There is no non vulnerable version of this package.</p> <p> <a href="#">PORTAL-445</a> - xerces <span>CLOSED</span></p> <p>Explanation</p> <p>Apache Xerces2 is vulnerable to a Denial of Service (DoS) attack.</p>	Request exception
portal-sdk	bootstrap	<p>There is no non vulnerable version of this package.</p>	Request exception
portal, portal-sdk	org. bouncycastle	<p>If not false positive, can be handled with the new version upgrade which do not have vulnerability.</p> <p> <a href="#">PORTAL-444</a> - bouncy castle <span>CLOSED</span></p> <p>Explanation</p> <p>Bouncy Castle is vulnerable to Remote Code Execution (RCE).</p>	we will try to handle them in Dublin release based on the resource availability and priority
portal	org. codehaus.groovy	<p>If not false positive, can be handled with the new version upgrade which do not have vulnerability.</p> <p> <a href="#">PORTAL-447</a> - codehaus.groovy <span>CLOSED</span></p> <p>Explanation</p> <p>Groovy is vulnerable to insecure deserialization leading to Remote Code Execution (RCE).</p>	we will try to handle them in Dublin release based on the resource availability and priority
portal	org. eclipse.jetty jetty-util	<p>If not false positive, can be handled with the new version upgrade which do not have vulnerability.</p> <p> <a href="#">PORTAL-448</a> - jetty <span>CLOSED</span></p> <p>Explanation</p> <p>Eclipse Jetty Server is vulnerable to HTTP request smuggling.</p>	<p>we will try to handle them in Dublin release based on the resource availability and priority;</p> <p>Will upgrade to 9.2.14. v20151106: or will disable http1.1</p>
portal, portal-sdk	org. apache.lucene	<p>Not used, this will be removed.</p> <p> <a href="#">PORTAL-440</a> - Lucene libraries <span>CLOSED</span></p>	we will try to handle them in Dublin release
portal	org. apache.tomcat.embed  tomcat-embed-core : 8.5.28:	<p>There is no non vulnerable version of this component/package.</p> <p> <a href="#">PORTAL-449</a> - tomcat.embedded <span>CLOSED</span></p> <p>Explanation</p> <p>Apache Tomcat is vulnerable to a Cross-Origin attack due to the insecure default configuration of the CORS filter.</p>	The configuration for CorsFilter needs to change. We will change the urlPattern in web.xml for CorsFilter from * to *onap*
portal	org. apache.cxf	<p>False positive</p> <p>We do not use the below code, which is vulnerable.</p> <p>System.setProperty("java.protocol.handler.pkgs", "com.sun.net.ssl.internal.www.protocol");</p> <p> <a href="#">PORTAL-450</a> - cxf <span>CLOSED</span></p>	Not Vulnerable
portal	org. hibernate	<p>If not false positive, can be handled with the new version upgrade which do not have vulnerability.</p> <p> <a href="#">PORTAL-441</a> - Hibernate validator <span>CLOSED</span></p> <p>Explanation</p> <p>The Hibernate Validator (HV) package is vulnerable to a privilege escalation vulnerability.</p>	we will try to handle them in Dublin release based on the resource availability and priority
portal, portal-sdk	c3p0 : 0.9.5.2	<p>c3p0-0.9.5.2.jar The c3p0 component is vulnerable to XML eXternal Entity (XXE) attacks. See Jira ticket: <a href="#">PORTAL-535</a></p>	Will upgrade to 0.9.5.3. Dublin +

portal	postgresql-9.1-901-1-jdbc4.jar	Description from CVEA weakness was found in postgresql-jdbc before version 42.2.5. It was possible to provide an SSL Factory and not check the host name if a host name verifier was not provided to the driver. This could lead to a condition where a man-in-the-middle attacker could masquerade as a trusted server by providing a certificate for the wrong host, as long as it was signed by a trusted CA.Explanation The postgresql package is vulnerable to Man-in-the-Middle (MitM) attacks. When using a non-default SSL Factory, the postgresql jdbc doesn't validate the hostname of SSL certificates. An attacker can potentially exploit this behavior to perform a MitM attack. See Jira ticket: <a href="#">PORTAL-536</a>	Remove this lib. May not be used anymore.
portal, portal-sdk	dom4j	Description from CVEdom4j version prior to version 2.1.1 contains a CWE-91: XML Injection vulnerability in Class: Element. Methods: addElement, addAttribute that can result in an attacker tampering with XML documents through XML injection. This attack appear to be exploitable via an attacker specifying attributes or elements in the XML document. This vulnerability appears to have been fixed in 2.1.1 or later.Explanation The dom4j package is vulnerable to XML Injection. The QName() function in the QName class file does not properly sanitize the QName input attribute value(s). A remote attacker can exploit this vulnerability by injecting an XML object that contains arbitrary code in the element and attribute names, hence leading to XML Injection. See Jira ticket: <a href="#">PORTAL-537</a>	Need to upgrade to version 2.1.1
portal, portal-sdk	wicket-util	Description from CVE The DiskFileItem class in Apache Wicket 6.x before 6.25.0 and 1.5.x before 1.5.7 allows remote attackers to cause a denial of service (infinite loop) and write to, move, and delete files with the permissions of DiskFileItem, and if running on a Java VM before 1.3.1, execute arbitrary code via a crafted serialized Java object.  See Jira ticket: <a href="#">PORTAL-538</a>	Need to upgrade to Apache Wicket 6.25.0
portal, portal-sdk	jquery 2.2.4	Explanation The jQuery package is vulnerable to Cross-Site Scripting (XSS). The parseHTML() function in the parseHTML.js, jquery.js files allow JavaScript to be executed immediately when it's embedded within the event attributes. An attacker can exploit this vulnerability by injecting malicious JavaScript containing events handlers which, when rendered, results in the execution of arbitrary script. See Jira ticket: <a href="#">PORTAL-539</a>	Need to upgrade to 3.2.0+
portal-sdk	jQuery 1.4.2	Explanation The jQuery package is vulnerable to Cross-Site Scripting (XSS). The parseHTML() function in the parseHTML.js, jquery.js files allow JavaScript to be executed immediately when it's embedded within the event attributes. An attacker can exploit this vulnerability by injecting malicious JavaScript containing events handlers which, when rendered, results in the execution of arbitrary script. See Jira ticket: <a href="#">PORTAL-540</a>	Need to upgrade to 2.0.0
portal, portal-sdk	org.webjars.bootstrap	Description from CVEIn Bootstrap before 4.1.2, XSS is possible in the data-container property of tooltip. Explanation The bootstrap package is vulnerable to Cross-Site Scripting (XSS). The show() function in the tooltip.js file allows HTML and scripts in the data-container tooltip attribute values in the DOM elements without proper sanitization. This can be misused to cause XSS. See Jira ticket: <a href="#">PORTAL-541</a>	Need to upgrade to 4.1.3
portal, portal-sdk	esapi	Description from CVEThe authenticated-encryption feature in the symmetric-encryption implementation in the OWASP Enterprise Security API (ESAPI) for Java 2.x before 2.1.0.1 does not properly resist tampering with serialized ciphertext, which makes it easier for remote attackers to bypass intended cryptographic protection mechanisms via an attack against the intended cipher mode in a non-default configuration, a different vulnerability than CVE-2013-5679.Explanation An attacker can manipulate the cipher transformation (e.g., changing the cipher mode from CBC to OFB or padding scheme) to adverse effect.	Will confirm if we are using ESAPI symmetric crypto . Most probably not being used in which case, we are not vulnerable.
portal, portal-sdk	org.apache.zookeeper:zookeeper:3.4.11		Doesnt seem to be vulnerable. Will confirm
portal, portal-sdk	org.owasp.antisamy:antisamy:1.5.3,  org.owasp.antisamy:antisamy:1.4.3	Description from CVEOWASP AntiSamy before 1.5.7 allows XSS via HTML5 entities, as demonstrated by use of &colon; to construct a javascript: URL.Explanation AntiSamy is vulnerable to Cross-Site Scripting (XSS). The package uses an HTML serializer that doesn't take HTML5 entities into consideration, such as &colon;, &lpar;, and &rpar;. An attacker can exploit this to inject JavaScript into the context of the page. See Jira ticket: <a href="#">PORTAL-542</a>	Need to upgrade to version 1.5.7+