Resume of Vulnerabilities None 0 Low 14 Medium 0 High 8 Critical 0 Total 22
You are using the Insider open source version. If you like the product and want more features, visit http://insidersec.io and get to know our enterprise version. If you are a developer, then you can contribute to the improvement of the software while using an open source version DRA - Data Risk Analytics
File: main/webapp/WEB-INF/web.xml Dra: http://java.sun.com/dtd/web-app_2_3.dtd Type: url File: main/resources/Users.hbm.xml Dra: http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd Type: url File: main/webapp/header.jsp Dra: http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd Type: url File: main/webapp/header.jsp Dra: http://www.w3.org/1999/xhtml Type: url File: main/webapp/header.jsp Dra: http://www.w3.org/1999/xhtml Type: url File: main/webapp/footer.jsp Dra: http://www.cysecurity.org Type: url File: main/webapp/footer.jsp Dra: http://www.cysecurity.org Type: url
File: main/webapp/jquery.min.js Dra: http://jquery.com/ Type: url File: main/webapp/jquery.min.js Dra: http://jquery.org/license Type: url File: main/webapp/vulnerability/Injection/1.xsl Dra: http://www.w3.org/1999/XSL/Transform Type: url File: main/webapp/vulnerability/Injection/1.xsl Dra: http://www.w3.org/TR/xslt Type: url File: main/webapp/vulnerability/Injection/1.xsl Dra: http://www.w3.org/TR/xslt Type: url File: main/webapp/vulnerability/Injection/courses.xml Dra: https://www.udemy.com/hacking-securing-java-web-programming/ Type: url
File: main/webapp/vulnerability/Injection/courses.xml Dra: https://www.udemy.com/hacking-securing-php/ Type: url File: main/webapp/vulnerability/Injection/courses.xml Dra: https://www.udemy.com/certified-whitehat-hacker-level-1/ Type: url File: main/webapp/vulnerability/Injection/courses.xml Dra: https://www.udemy.com/certified-apt-defender/ Type: url File: main/webapp/vulnerability/Injection/xslt.jsp Dra: http://java.sun.com/jsp/jstl/core Type: url File: main/webapp/vulnerability/Injection/xslt.jsp Dra: http://java.sun.com/jsp/jstl/xml
Type: url File: main/webapp/vulnerability/unvalidated/OpenURL.jsp Dra: http://certified.cysecurity.org/ Type: url File: main/webapp/vulnerability/xss/flash/exss.jsp Dra: http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0 Type: url File: main/webapp/vulnerability/xss/flash/exss.jsp Dra: http://www.macromedia.com/go/getflashplayer Type: url Wulnerabilities CVSS: 7.4
Severity: Class: LoginValidator.java (52:108) VulnerabilityID :79b44bd4bfae6e9d945cf5dc28be17cb Method: rs=stmt.executeQuery("select * from users where username='"+user+"' and password='"+pass+"'"); Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/LoginValidator.java (52:108) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. CVSS: 7.4 Severity:
Class: Register.java (50:15) VulnerabilityID :055e3d0ea50188c339c04f4391a74677 Method: secret="nosecret"; Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (50:15) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. CVSS: 3.2 Severity:
Class: Register.java (71:27) VulnerabilityID :5a933ed5788baab52fdbeafa018fc1b1 Method: System.out.println("SQLException: " + ex.getMessage()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (71:27) Recomendation: CVSS: 3.2 Severity: Class: Pogister java (73:26)
Class: Register.java (72:26) VulnerabilityID :bfb47404d3613f24a9531cf64ef34144 Method: System.out.println("SQLState: " + ex.getSQLState()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (72:26) Recomendation: CVSS: 3.2 Severity: Class: Register.java (73:26)
VulnerabilityID :610a0511e3198a193c0c5244d54afda3 Method: System.out.println("VendorError: " + ex.getErrorCode()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (73:26) Recomendation: CVSS: 3.2 Severity: Class: Register.java (71:27)
<pre>VulnerabilityID :5a933ed5788baab52fdbeafa018fc1b1 Method: System.out.println("SQLException: " + ex.getMessage()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (71:27) Recomendation: CVSS: 3.2 Severity: Class: Register.java (72:26)</pre>
<pre>VulnerabilityID :bfb47404d3613f24a9531cf64ef34144 Method: System.out.println("SQLState: " + ex.getSQLState()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (72:26) Recomendation: CVSS: 3.2 Severity: Class: Register.java (73:26) WilnerabilityID :ClassIdasanasanasanasanasanasanasanasanasanasa</pre>
<pre>VulnerabilityID :610a0511e3198a193c0c5244d54afda3 Method: System.out.println("VendorError: " + ex.getErrorCode()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Register.java (73:26) Recomendation: CVSS: 3.2 Severity: Class: Install.java (174:23) WulnerabilityID :f7404b3d8c1a5fdeccas577ac4f42ff</pre>
<pre>VulnerabilityID :f7d04b3d8c1a5fdcecea5373ea4f42ff Method: System.out.println("SQLException: " + ex.getMessage()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (174:23) Recomendation: CVSS: 3.2 Severity: Class: Install.java (175:22) VulnerabilityID :40f40877b3a954d87b4631d5208dd8a5</pre>
<pre>WulnerabilityID :40f4087/03a9540870408704087040870808085 Method: System.out.println("SQLState: " + ex.getSQLState()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (175:22) Recomendation: CVSS: 3.2 Severity: Class: Install.java (176:22) VulnerabilityID :f90f90edafedbe72789660a1f27687a5</pre>
Method: System.out.println("VendorError: " + ex.getErrorCode()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (176:22) Recomendation: CVSS: 3.2 Severity: Class: Install.java (180:24) VulnerabilityID :f4dd873184f146b9169d8ebcbd0372db
<pre>Method: System.out.print("JDBC Driver Missing: Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (180:24) Recomendation: CVSS: 3.2 Severity: Class: Install.java (174:23) VulnerabilityID: f7d04b3d8c1a5fdcecea5373ea4f42ff</pre>
<pre>Method: System.out.println("SQLException: " + ex.getMessage()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (174:23) Recomendation: CVSS: 3.2 Severity: Class: Install.java (175:22) VulnerabilityID : 40f40877b3a954d87b4631d5208dd8a5</pre>
<pre>Method: System.out.println("SQLState: " + ex.getSQLState()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (175:22) Recomendation: CVSS: 3.2 Severity: Class: Install.java (176:22) VulnerabilityID :f90f90edafedbe72789660a1f27687a5</pre>
Method: System.out.println("VendorError: " + ex.getErrorCode()); Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (176:22) Recomendation: CVSS: 3.2 Severity: Class: Install.java (180:24) VulnerabilityID: f4dd873184f146b9169d8ebcbd0372db Method: System.out.println("VendorError: " + ex.getErrorCode()); Description: The App logs information. Sensitive information should not be logged. (176:22) Recomendation:
<pre>Method: System.out.print("JDBC Driver Missing: Description: The App logs information. Sensitive information should not be logged. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/Install.java (180:24) Recomendation: CVSS: 7.4 Severity: Class: XPathQuery.java (50:67) VulnerabilityID :7e30dcc63817dff1ce95a704e1ea484d Method: String xPression="/users/user[username='"+user+"' and password='"+pass+"']/name";</pre>
Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/java/org/cysecurity/cspf/jvl/controller/XPathQuery.java (50:67) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. CVSS: 7.4 Severity: Class: HashMe.java (16:32) VulnerabilityID :c3ca4c48c76640b05e0cf60a2faa7bdd Method: MessageDigest md = MessageDigest.getInstance("MD5");
Description: MD5 is a hash algorithm considered weak and can return the same result for two different contents, which can cause collisions and in extreme cases it can cause a security breach. https://en.wikipedia.org/wiki/Collision_resistance ClassMessage: main/java/org/cysecurity/cspf/jvl/model/HashMe.java (16:32) Recomendation: It is recommended to use some CHF (Cryptographic Hash Function), which is mathematically strong and not reversible. SHA512 would be the most recommended hash for storing the password and it is also important to adopt some type of Salt, so that the Hash is more secure. CVSS: 7.4 Severity: Class: HashMe.java (16:32)
VulnerabilityID :c3ca4c48c76640b05e0cf60a2faa7bdd Method: MessageDigest md = MessageDigest.getInstance("MD5"); Description: MD5 is a hash algorithm considered weak and can return the same result for two different contents, which can cause collisions and in extreme cases it can cause a security breach. https://en.wikipedia.org/wiki/Collision_resistance ClassMessage: main/java/org/cysecurity/cspf/jvl/model/HashMe.java (16:32) Recomendation: It is recommended to use some CHF (Cryptographic Hash Function), which is mathematically strong and not reversible. SHA512 would be the most recommended hash for storing the password and it is also important to adopt some type of Salt, so that the Hash is more secure. CVSS: 7.4
Severity: Class: adminlogin.jsp (19:108) VulnerabilityID :ec718171f0f05800a97ce8192864b7e4 Method: rs=stmt.executeQuery("select * from users where username='"+user+"' and password='"+pass+"' and privilege='admin'"); Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/webapp/admin/adminlogin.jsp (19:108) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. CVSS: 7.4 CVSS: 7.4 CVSS: 7.4
Severity: Class: login.jsp (6:9) VulnerabilityID :eca9ae43a178686716e06dfcffbf9ec8 Method: String password=""; Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/webapp/login.jsp (6:9) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. CVSS: 7.4 Severity:
Severity: Class: changepassword.jsp (40:52) VulnerabilityID :ffd4d7a28df7fe6f688ed76dde9292a8 Method: stmt.executeUpdate("Update users set password='"+pass+"' where id="+id); Description: File contains sensitive information written directly, such as usernames, passwords, keys, etc. ClassMessage: main/webapp/vulnerability/csrf/changepassword.jsp (40:52) Recomendation: Credentials must not be stored in the Git code or repository. There are 'Secrets Management' solutions that can be used to store secrets or use Pipeline resources. Copyright 2020 insidersec.io
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Score Security 26/100