Team 9: Heyt Patel, Wenting Zheng

We are using the openmrs.org, the open source provided in project description to test out and document a technical security reviews. This open-source system provides electronic health care functionality for resource-constrained environments.

## Black-Box Security Tests

| test case id | Instructions | Actual Result | Expected result |
| --- | --- | --- | --- |
| **SQLInjectionAtLogin**  (Injection) | **Preconditions:**     1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes .   **Steps:**   * Username : ' OR ''=' * Pasword: ' OR ''=' * Click on login button   Reference: <https://sqlzoo.net/hack/> | Result:  Login fails with an Error message : Invalid username/password. Please try again. | It should not let you login and through an error message of invalid entry of username or password. |
| **FindPasswordBySQLInjection**  (Injection) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes .   **Steps:**   * Username : admin * Pasword: ' OR EXISTS(SELECT \* FROM users WHERE username='admin' AND password LIKE '%A%') AND ''=' * Click on login button   Reference: <https://sqlzoo.net/hack/> | Result:  Login fails with an Error message : Invalid username/password. Please try again. | It should not let you login and through an error message of invalid entry of username or password. |
| **sessionTimeout**  (Broken Authentication and Session Management ) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes .   **Steps:**   * Username : admin * Password: Admin123 * You will driven home page or to this link: <https://demo.openmrs.org/openmrs/referenceapplication/home.page> * Close the browser tab * After an hour check the same link on the same browser | **Failing**!  Even after an hour you will find user “admin” logged in on same link. | Session should be timed out and person should not stay logged in for more than 10 min if account is inactive. |
| **strengthCheckOfPassword**  (Broken Authentication and Session Management ) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin12    **Steps:**   * Click on “admin” on top bar. * Select “my account” from the drop down menu * Next click on Password Change * Type:   **Try 1:**  Old Password: Admin123  New password: 87654321  Confirm password: 87654321   * Check the output message   **Try 1:**  Old Password: Admin123  New password: CoolCool  Confirm password: CoolCool   * Check the output message | After clicking SAVE :    For the Try1 message shown : “ Please choose password that contains both upper-case and lower-case letters.”  For the Try2 message shown : “ Please choose password that contains at least one number.”  This message shows that system does check against 3 rules out of 4 rules of complexity of password. | Given passwords are signs of weak passwords.  New password should get check against 3 out of the 4 complexity rules of password. |
| **onChangeXSS**  (Cross-site Scripting) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin12  **Steps:**   * Click on “Register a patient” option * Give a name : " onChange="alert(“To verify your name give full social security number on second input box”)” * Other all fields can be fill can per your choice. | The pop window should not displayed. | Right after jumping from name to middle name. Pop window containing “To verify your name give full social security number on second input box” should not be displayed. |
| **xssByURL**  (Cross-site Scripting) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin12  **Steps:**   * Click on “find a patient record” option * Give a name : “Patel” * Using the URL change patientId to <script> window.location.href ="https://www.example.com";</script> | Throws Null pointer exception with UI Framework Error! | This should throw an error stating that invalid input or out of range. |
| **Access Page after logout**(Broken access control) | **Preconditions**:   1. Click on the link below: [https://demo.openmrs.org/openmrs](https://demo.openmrs.org/openmrs/referenceapplication/home.page) 2. Login as admin with the following information:   Username: admin  Password: Admin123   1. Click “System Administration” 2. Click “Manage Accounts” 3. Copy the URL(<https://demo.openmrs.org/openmrs/adminui/systemadmin/accounts/manageAccounts.page>) 4. Logout 5. Go to the copied URL | The URL is redirected to login page. The “Manage Account” page should not show | The “Manage Account” page should not show |
| **Access page from the other tab after logout**  (Broken access control) | **Preconditions**:   1. Click on the link below: <https://demo.openmrs.org/openmrs> 2. Login as admin with the following information:   Username: admin  Password: Admin123   1. Click “System Administration” 2. Open “Manage Accounts” in another tab 3. Logout in the previous page 4. Refresh the “Manage Account” page | The page is redirected to the login page. The “Manage Account” page should not show | The “Manage Account” page should not show |
| **overlyInformativeErrorMessage**  (Misconfiguration) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm>   **Steps:**   * https://demo.openmrs.org/openmrs/manageAccounts.page | It throws an internal server error with stack trace. | It should throw an short error message stating the issue. |
| **notInformativeErrorMessage2**  (Misconfiguration) | Preconditions:   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin123   1. Add Provider to Julie Smith:   Click on System Administration  Click on the edit icon to edit Julie Smith  Click on Add Provide Account  Enter the following information:  Identifier: asdfg  Provider Role: Clerk  Click Save | Error message “Fail to save” is displayed and it doesn’t include more information about identifier or provider role. | Error message is shown without detailed information about the identifier or the provider role. |
| **verificationOfProtocols**  (Sensitive Data exposure) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm>   **Steps:**   * Explore all the possible pages to find if HTTPS converts to downgrade connection HTTP | There is no webpage in the software that uses HTTP. | There is no webpage in the software that uses HTTP. |
| **encryptedID**  (Sensitive Data exposure) | **Preconditions:**   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin12  **Steps:**   * Click on “Register a patient” option   Fill the information shown below:   * Name: Hiteshwari, H., Patel * Gender: Male * Birthdate: 23 year(s), 1 month(s) * Address: 900 Cottage Grove Rd, Bloomfield, Connecticut, 06002 * Phone Number: 9192345678 * Relatives: Hiteshwari H. Patel - Parent, mark zuckerberg * Go to home page and “find a patient record” * Search “Patel” * Look at the ID Patient ID 100JDJ * Check the patients ID in URL * And try to access patients using id 100JDJ | URL shows the patient’s ID which is encrypted well.    URL with actual ID will not display the Patient associated with ID. | URL shows the patient’s ID which is encrypted well.  URL with actual ID will not display the Patient associated with ID. |
| **Lockout after failed attempts**  (Insufficient attack protection) | Preconditions:   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin123   1. Add a new user:   Click on System Administration  Click on Manage Accounts  Click on Add New Account  Enter the following information:  Family Name: Sam  Given Name: Z  Gender: Male  Check Add User Account  Username: Sam  Privilege: Full  Password: Password1  Click on Save   1. Logout   Steps:   * Enter the following information for 10 times:   Username: Sam  Password: password1   * Enter the following information:   Username: Sam  Password: Password1 | Error message “Invalid username/password. Please try again” showed for the 10 failed attempts.  “Invalid username/password. Please try again” showed when the username and password are correct. | It should not let you login with the wrong password and display an error message.  The account should be locked out with 10 failed attempts. The error message should display saying the account is locked |
| **Logout after closing the window**  (Insufficient attack protection) | Preconditions:   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin123   1. Close the window 2. Click on the link below: <https://demo.openmrs.org/openmrs> | The admin page shows up. | The admin is automatically logout and the login page should display. |
| getPrivilegebyCookie  CSRF | Preconditions:   1. Click on the link below: <https://demo.openmrs.org/openmrs/login.htm> 2. This page will have username and Password input boxes . 3. Enter :   Username : admin  Password: Admin123  **Steps:**   * Click on “Register a patient” option   Fill the information shown below:   * Give a name : " onChange="alert(document.cookie)” * Other all fields can be fill can per your choice.   Check if jumping on to next steps gives cookie or not! | Not able to get Cookie! | If we can access cookie then we can use curl to get the privilege as admin and make changes to private information of patient. |
|  |  |  |  |

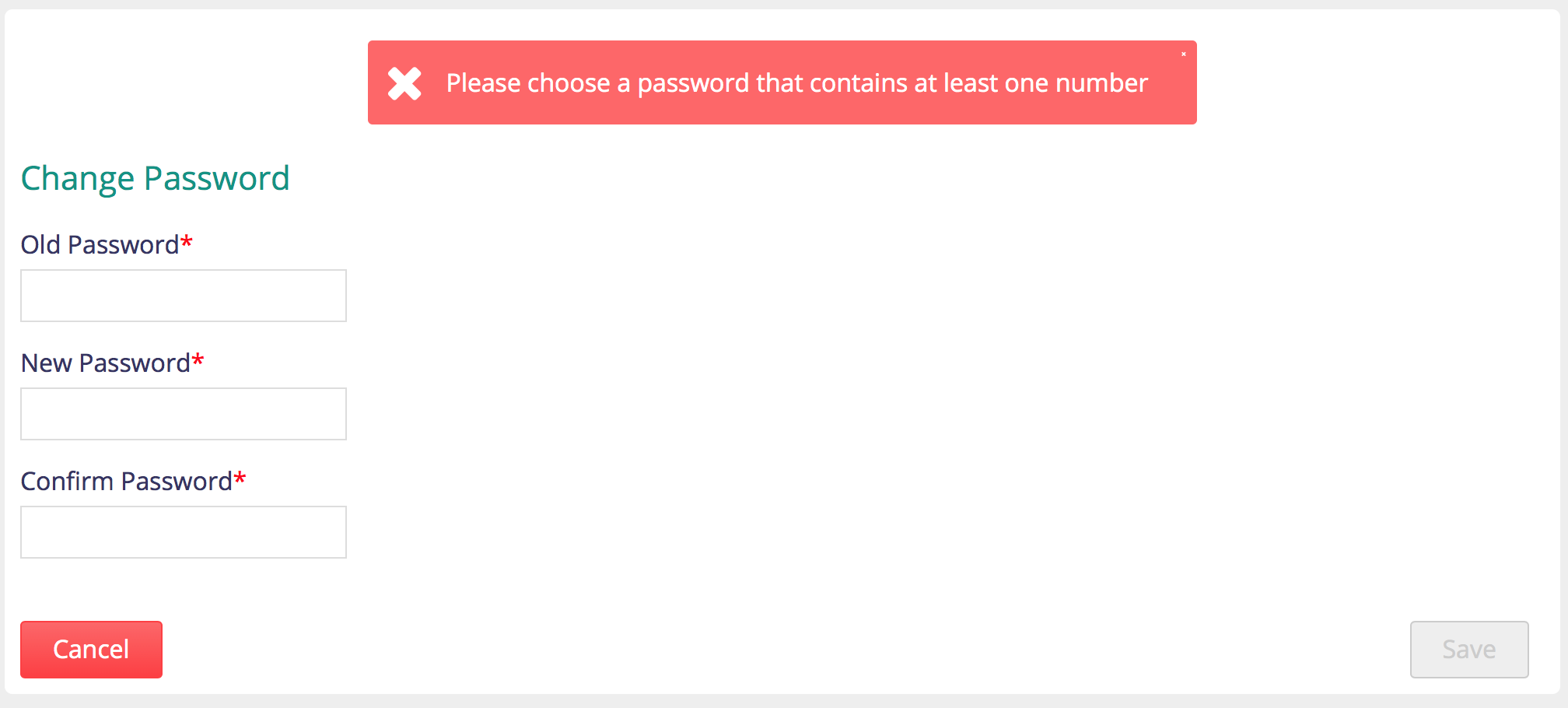
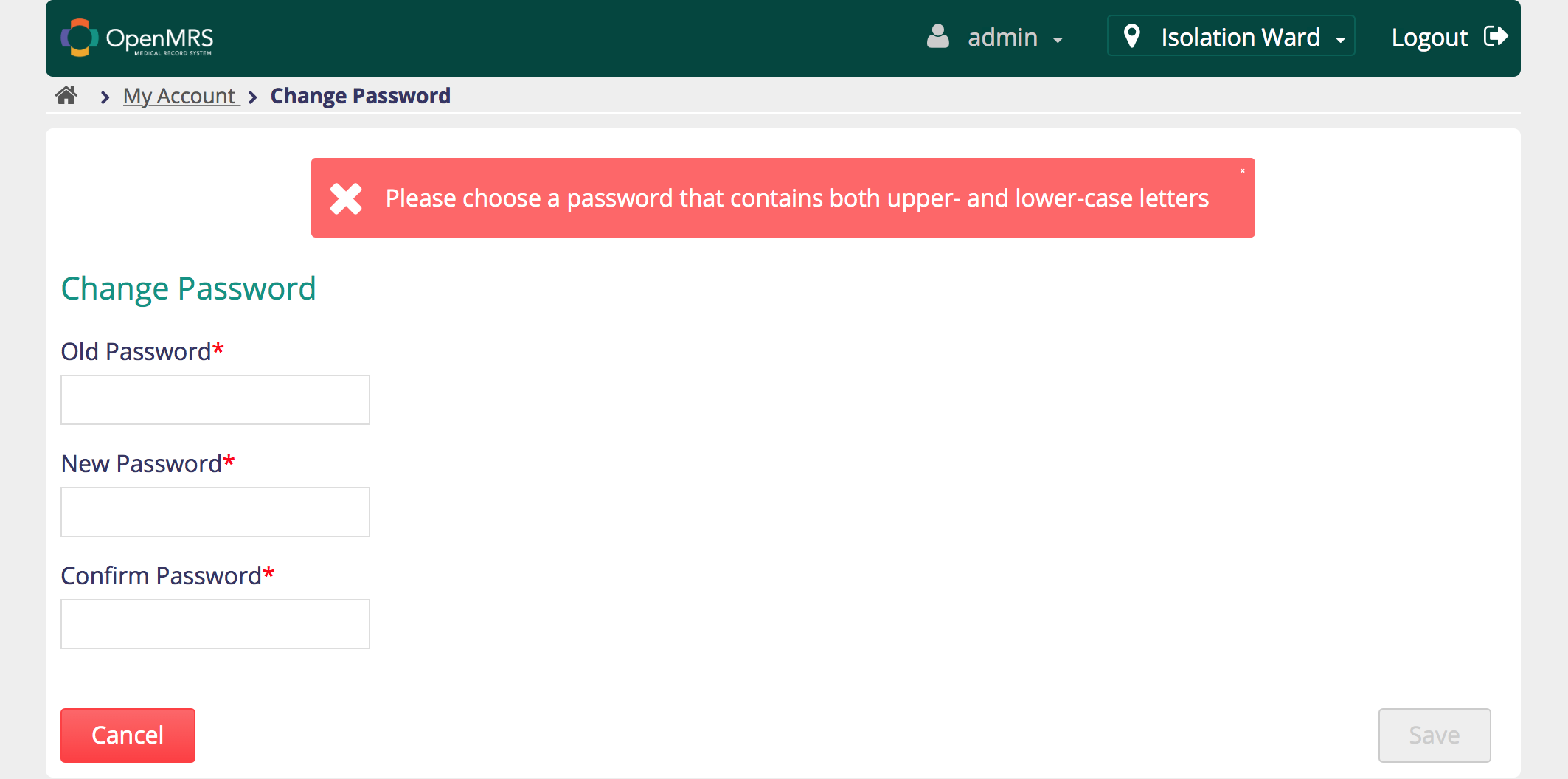
**Injection:**

* The attack supposed to let us read the details from the database as an external user.
* The first command intended to let us login as one of the valid user from the database.
* The second test **FindPasswordBySQLInjection** supposed to help attacker login if minimal information about password is leaked.

However, it looks like the website is checking for input at client side which does not let script injected.

**Broken Authentication and Session Management :**

* The test “**sessionTimeout**” is failing. To mitigate this, there should be timer for the session which should counting the time if session is inactive for more than certain amount of time.
* The test **“strengthCheckOfPassword” fulfills three rules of complexity out of four.**



**Cross-site Scripting**

* To mitigate XXS, the system needs to validate and sanitize all malicious input before sending them into backend. In website, there are piece of codes on HTML pages and class “[PersonNameValidator](https://github.com/openmrs/openmrs-core/blob/4a0feb8da351088f25fdc4e6d324a1f277aa3410/api/src/main/java/org/openmrs/validator/PersonNameValidator.java)” that indicates the source of XXS prevention in the software.

**Broken Access Control**

* The “Access Page after logout” test is intended to access the page as admin after the admin logs out. The URL is redirected to the login page. The system successfully checks the authentication before letting the user access to the page.
* The “Access page from the other tab after logout pass” is intended to access the page as admin in the other tab after the admin logs out in the previous tab. The system redirects the page to the login page.

**Insufficient attack protection**

* The “Lockout after failed attempts” is intended to try to login with wrong password for several times. The account is locked out but error message still shows “Invalid username/password. Please try again.”. The error message should show something meaningful. For example, your account is locked out due to too many failed attempts. Please contact the admin. This way, the user knows his/her account is targeted and will let the admin know.
* The “Logout after closing the window” is failing. The account still stay logged in after closing the window. To mitigate it, the account will be automatically logged out after closing the window. You can also give the user a choice by popping up a window saying “Your account will be logged out after you close the window. Do you want to continue?”. In this case, the user will have a chance to choose if s/he wants to stay or not.

**Misconfiguration:**

* When you try to redirect the links, it throws an internal server error with stack trace. Stack trace contains information about classes used in software. This can be mitigated by handling internal server error by replace with custom error message.

**Sensitive Data exposure**

* All the the webpages in the system system uses HTTPS protocol as shown below. Also, as shown below it uses the advanced standard security technology like SSL or TSL to ensure that all the data passed between the browser and the web server is encrypted and remains private. Application of HTTPS sessions provides a secure communication protocol.

## 

* Also, Patients’ id’s are encrypted well when it presented in URL. We are not able to use actual patients ID in URL to open patient’s profile.

## 

## 

## 

## **Dependencies**

Third-party software libraries: <https://github.com/openmrs/openmrs-core/blob/master/pom.xml>

Core dependencies:

javax.servlet-api(3.0.1), jsp-api(2.0), jstl(1.1.2), request(1.0.1), response(1.0.1), standard(1.1.2), page(1.0.1), velocity(1.6.2), velocity-tools(2.0), commons-collections(3.2.2), commons-io(2.5), commons-lang3(3.6), spring-core(4.1.4.RELEASE), spring-beans(4.1.4.RELEASE), spring-context(4.1.4.RELEASE), spring-aop(4.1.4.RELEASE), spring-orm(4.1.4.RELEASE), spring-tx(4.1.4.RELEASE), spring-jdbc(4.1.4.RELEASE), spring-web(4.1.4.RELEASE), spring-webmvc(4.1.4.RELEASE), spring-oxm(4.1.4.RELEASE), spring-context-support(4.1.4.RELEASE), javassist(3.22.0-GA), hibernate-core(4.3.9.Final), hibernate-c3p0(4.3.9.Final), hibernate-ehcache(4.3.9.Final), hibernate-validator(4.3.9.Final), hibernate-search-orm(4.3.9.Final), lucene-queryparser(4.10.4), lucene-queries(4.10.4), liquibase-core(2.0.5), modify-column(2.0.2), identity-insert(1.2.1), type-converter(1.0.1), log4j(1.2.15), slf4j-api(1.6.0), jcl-over-slf4j(1.6.0), slf4j-log4j12(1.6.0), hapi-base(2.0), hapi-structures-v23(2.0), hapi-structures-v24(2.0), hapi-structures-v25(2.0), hapi-structures-v26(2.0), dom4j(1.6.1), xercesImpl(2.8.0), xstream(1.4.3), mail(1.4.1), commons-beanutils(1.9.3), commons-fileupload(1.3.3), mysql-connector-java(5.1.45), mysql-connector-mxj(5.0.11), postgresql(9.0-801.jdbc4), jackson-core-asl(1.9.13), jackson-mapper-asl(1.9.13), jackson-core(2.9.0), jackson-annotations(2.9.0), jackson-databind(2.9.0), reflectutils(0.9.14), spring-test(4.1.4.RELEASE), junit(4.12), mockito-core(1.10.19), powermock-module-junit4(1.6.6), powermock-api-mockito(1.6.6), hamcrest-core(1.3), hamcrest-library(1.3), h2(1.4.196), dbunit(2.5.4), validation-api(1.0.0.GA), sonar-jacoco-listeners(3.1), ehcache(2.10.0)

Velocity-tools(2.0)’s [vulnerabilities](https://issues.apache.org/jira/browse/VELTOOLS-163):

* The software failed to suppress the class property. [Link](https://nvd.nist.gov/vuln/detail/CVE-2014-0114)
* The software failed to redirect access to the method in a proper way. [Link](https://nvd.nist.gov/vuln/detail/CVE-2014-0113)

Postgresql(9.0-801.jdbc4) has [vulnerabilities](https://www.cvedetails.com/vulnerability-list/vendor_id-336/product_id-575/Postgresql-Postgresql.html):

* The hacker can modify some system files. [Link](https://www.cvedetails.com/cve/CVE-2018-1115/)
* The access complexity is low, so it’s easier to exploit. [Link](https://www.cvedetails.com/cve/CVE-2018-1052/)

Database management system software: mysql (5.1.28)

Mysql has [vulnerabilities](https://www.cvedetails.com/vulnerability-list/vendor_id-185/product_id-316/Mysql-Mysql.html):

* The access complexity is medium, so it takes some extra effort to exploit. [Link](https://www.cvedetails.com/cve/CVE-2014-6559/)
* The hacker can modify some system files. [Link](https://www.cvedetails.com/cve/CVE-2015-2575/)

## 

## 

## 

## **Privacy**

Look for a [privacy policy](https://openmrs.org/privacy/) for the software system.

Code of Fair Information Practices

* *There must be no personal data record-keeping systems whose very existence is secret*
  + *“...we have a very permissive (non) privacy policy that mirrors our public mission”. “Please assume that everything you contribute intentionally to OpenMRS is public”.* There will not be a secret system to collect personal data because all the information is intended to be public. The privacy policy highlights it so that the users will be extra careful about what to post in OpenMRS.
* *There must be a way for a person to find out what information about the person is in a record and how it is used.*
  + “*There will be cases where you are invited to share private information for various purposes, and denote that information as such. In those cases, we commit to only disclosing that information without your permission upon receipt of a valid subpoena and following notice from OpenMRS to you*”. The user will know when and what information will be collected and how it will be used when s/he is invited to share.
* *There must be a way for a person to prevent information about the person that was obtained for one purpose from being used or made available for other purposes without the person's consent.*
  + “*In those cases, we commit to only disclosing that information without your permission upon receipt of a valid subpoena and following notice from OpenMRS to you*”. The information will only be collected with the person’s consent unless there is a valid subpoena. The user can prevent the information from being collected when s/he is invited to share.
* *There must be a way for a person to correct or amend a record of identifiable information about the person.*
  + The privacy policy doesn’t explicitly address this principle, but it doesn’t really need to. “*Please assume that everything you contribute intentionally to OpenMRS is public*”. In this case, the identifiable information, or the sensitive personal information, should not be posted in OpenMRS to make it public. Therefore, it wouldn’t be a problem to correct the identifiable information.
* *Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take precautions to prevent misuses of the data*
  + The privacy policy doesn’t explicitly address this principle, but again, the user shouldn’t post any identifiable information in OpenMRS. However, the privacy policy could have explained how to assure the data is reliable and how to prevent the data from misuse.