INF226 Obligatory assignment

Hvar Eggereide and Syver Storm-Furru

2016-10-02

Contents

1 Introduction

For our obligatory assignment we were tasked with analysing OpenMRS, a medical patient journal system. The analysis is performed both using static and dynamic code analysis tools (HPE Fortify and FindBugs for static analysis and OWASP ZAP for dynamic), as well as a thorough run-through of the installation process and software usage.

2 Software

2.1 OpenMRS

OpenMRS is, according to the website, the "world's leading open source enterprise electronic medical record system platform".[1] It is used in hospitals and medical facilities all over the world, for example in Nigeria, South Africa, India and the United States, and is supported by many different governments, NGOs, and both for- and non-profit organisations. The software has a stated goal of being usable with no programming knowledge, and to be a common platform for which medical informatics efforts in developing can be built.

Technical Specifications OpenMRS is a client-server platform, with a web front end. It is programmed in Java 7, using Tomcat 6 or 7 as the server framework, and MySQL 5.6 as the database backend. It also exposes a programming API to users, and is modular and extendable.

Setup The setup process of OpenMRS is quite involved and time consuming when attempting to do so on a personal computer, requiring both Tomcat, Java and MySQL to be setup. The official documentation is useful, but different parts of it uses different versions of e.g. Tomcat, so it can be confusing. It also provides install instructions for Windows and Linux distributions with Aptitude, but not for OSX or other Linux distributions.

As mentioned, OpenMRS doesn't run on the newest version of MySQL (at the time of writing MySQL 5.7), and the install instructions do not mention this. The process of figuring this out, and of removing and reinstalling a previous version of MySQL, proved to be a lengthy detour on an already long road. The instructions are also not very specific when noting which files you need to run OpenMRS in Tomcat, whether it is the source code, which was difficult to build and only return a test suite on a normal compile, a readily packaged complete install (which did not work properly), or a .war file that should be uploaded to the Tomcat server.

We first attempted setting up OpenMRS on OSX 10.11, but ran into problems when trying to install the correct version of MySQL, and therefore retried in an empty virtual machine running Linux (tested with both Lubuntu and Kali Linux). Following the install instructions were a lot easier when running Debian based distributions containing Aptitude, but we still had to find and install a previous version of MySQL.

Once everything was installed OpenMRS had some extra setup that was required, done through a web interface. This was mostly easy once the correct version of MySQL was in place. You were also given a first username and password that was, respectively, 'admin' and 'Admin123'.

Usage

[1] OpenMRS. *About OpenMRS*. `http://openmrs.org/about`. 2016.

2.2 HPE Fortify

Fortify is a code security tool suite, developed by Hewlett-Packard Enterprise (HPE). It aims to "make application security a natural part of the new SDLC, enabling time to market by building security in".[2] It contains such tools as WebInspect, a dynamic code analysis tool, and the Fortify Static Code Analysis tool.

Fortify is a proprietary solution, but is available with an academic license for free.

Audit Workbench Audit Workbench is the tool used to organise the the output of HPE's static code analysis software contained in the Fortify package. It is a GUI application built on top of Eclipse, specially designed for organising and presenting output for the HPE tools.

The installation process for Audit Workbench was straight-forward, but running the program required changing variables for the Eclipse backend, without information about how this is done readily available. The software also was a large RAM consumer, needing 5 gigabytes of RAM to analyze a relatively large project (OpenMRS) The software also was a large RAM consumer, needing 5 gigabytes of RAM to analyze a relatively large project (OpenMRS.)

Audit Workbench Rapport The first scan yielded a few messages of high concern. The OpenMRS core includes unit-test which gives quite a few erroneous warnings concerning security risks and more general bad coding practices.

There was an error concerning how the cookie was coded which may be interesting to look more into.

2.3 OWASP ZAP

ZAP is a dynamic analysis tool developed by OWASP, the Open Web Security Project. The software acts as a proxy between the host computer and a web application, performing different types of automatic scans, as well as having tools for manual searches for security vulnerabilities.[3]

We ran ZAP version 2.5.0 for our tests.

Scan of Front End The initial Quick Start scan of our test setup of the OpenMRS service only had the login front page to crawl. This yielded very few results considering this is just one page. It found some information about jQuery, ZAP also warned about the implementation of the cookie.

[2]HPE. *Application Security.* `http://www8.hp.com/lamerica_nsc_carib/en/software-solutions/application-security`. 2016.

[3]OWASP. *OWASP ZAP 2.4 Getting Started Guide.* 2016.

2.4 FindBugs

FindBugs is a tool for scanning java code looking for potential errors in the implementation. It is distributed under Lesser GNU Public License. The project originated from the University of Maryland.[4]

We ran FindBugs 3.0.1 for our tests.

scanning code Running the scan seems easy and most of the work is obviously analysing the results. The reports might be a bit hard to navigate and there was some trouble generating a html report that potentially is more readable. Again the unit-test generated false positives.

[4]University of Maryland. *FindBugs - Find Bugs in Java Programs*. 2015.