**Cyber 8420 Project Proposal**

**Open Source Project:** Keycloak

**Github Repository:** <https://github.com/DanielLucier/CYBER8420-SemesterProject>

**Authors:** Glenn Anderson, Dan Lucier, Chet Cyr, Kerolos Lotfy

**Team Name:** Cyber Wardens

Keycloak is a server to client application that adds authentication to applications by storing user's credentials on a company's own server. This allows for single sign on to both desktop and web applications including social websites using LDAP and active directory. Some of the key features that are offered include creating password policies and scalability. Keycloak is an open source application with its first open source commit in 2013. Keycloak is based on industry standard protocols and incorporates support for OpenID, OAuth 2.0, SAML, and OpenID Connect. To find out more about Keycloak’s primary features please visit their webpage at <http://www.keycloak.org/about.html>. Keycloak is also well documented. They host a documentation page on their website at <http://www.keycloak.org/documentation.html>. This documentation includes setup and how-to guides as well as API documentation. The Keycloak server and client applications are supported on Windows and Linux/Unix platforms. This project is written primarily in Java using JavaScript and CSS as the customer facing technology. Keycloak also offers a large community with over 190 different contributors with 8 main contributors and multiple returning users. A key reason for the strong community is that Keycloak is also backed by the RedHat community. Due to a large community, there are frequent builds and documentation updates including a release of version 3.3.0 on August 28th and version 3.2.1 on July 21st.

Keycloak uses the Apache License 2.0. For a full listing of details for the Apache License 2.0 please visit <https://www.apache.org/licenses/LICENSE-2.0>. The following process is required to contribute to the project. A full description of the process may be found at <https://github.com/keycloak/keycloak/blob/master/misc/HackingOnKeycloak.md>.

Here's a quick check list for a good pull request (PR):

* Discussed and agreed on Keycloak Dev mailing list
* One commit per PR
* One feature/change per PR
* No changes to code not directly related to your change (e.g. no formatting changes or refactoring to existing code, if you want to refactor/improve existing code that's a separate discussion to mailing list and JIRA issue)
* A JIRA associated with your PR (include the JIRA issue number in commit comment)
* All tests in testsuite pass
* Do a rebase on upstream master
* We only accept contributions to the master branch. The exception to this is if the fix is for the latest CR release and Final has not yet been released, in which case you can send the PR to both the corresponding branch and the master branch.

Once you're happy with your changes, go to GitHub and create a PR to the master branch.

As a contributor, one must first discuss any suggested changes with the developer mailing list and follow defined procedures to update the database schema and perform tests using testsuite. The Keycloak community does not enforce a specific coding style in their work, but they do recommend following the code style used by WildFly. A how-to guide for importing these formatting rules can be found at: <https://developer.jboss.org/wiki/ImportFormattingRules?_sscc=t>.

Due to the maturity of Keycloak, there have been many opened and closed issues. Keycloak offers multiple locations where issues can be documented. This includes the Github project and a secondary JIRA website for tracking the issues and feature requests. Currently, there are 42 open issues and over 4,000 closed issues on Github. Among those issues are security bugs that have been resolved (i.e.: XSS defense and integration with active directory).

With Keycloak both pulling information from a centralized location and interacting with the client's desktop/web applications, there are plenty of security requirements that must be considered. Since Keycloak supports multiple platforms, understanding the various platforms is a must to be able to securely deploy it across an entire organization. Keycloak provides documentation pertaining to proper security implementation steps for each supported platform. Also, with single sign on (SSO) an organization needs to be able to set and enforce security policies throughout various systems. According to Keycloak’s documentation, an organization is able to perform that task. Keycloak supports the most common social media services as well (i.e.: Github, Linkedin, StackOverflow, Google, Facebook, Twitter, and Microsoft). Some security areas that we will be looking for are policy implementation, authentication, and authorization processes.

When searching for a project, our motivation was to select one with obvious security requirements. A project that offered some form of user credential/password storage had the obvious security requirement of protecting a user’s information. Are passwords stored in plaintext or ciphertext? If passwords are saved in ciphertext, is an appropriate encryption algorithm used? These are just some requirements that we felt would need to be addressed to determine the security of a user credential/password storage project. We found a wealth of open source projects for this type of software. Most of the projects were either outdated or had very few main contributors/activity. Fortunately, we located Keycloak. Keycloak is a mature project with a decent amount of recent activity. As previously noted, the project is endorsed by Red Hat. In fact, Red Hat derives their own software from this project.