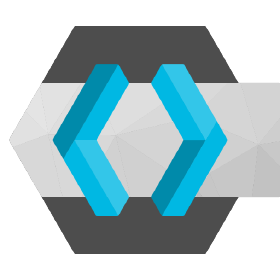
|  |
| --- |
| KeyCloak |
| Billy Hofland  DB03  2023/2024 |



Research Report

Research Report



Table of contents

[Introduction 3](#_Toc149312943)

[Problem description 3](#_Toc149312944)

[Main question: 3](#_Toc149312945)

[Sub-questions: 3](#_Toc149312946)

[Results 5](#_Toc149312947)

[Sub-question 1: 5](#_Toc149312948)

[Sub-question 2: 8](#_Toc149312949)

[Sub-question 3: 11](#_Toc149312950)

[Sub-question 4: 14](#_Toc149312951)

[Resolution 15](#_Toc149312952)

[Conclusion: 15](#_Toc149312953)

[Recommendation: 15](#_Toc149312954)

[References 16](#_Toc149312955)

[Version history 17](#_Toc149312956)

# Introduction

## Problem description

Users need to create and remember different credentials for each service associated with TeamVas, my online education platform. Because my application offers different services with multiple accounts this leads to a rather annoying experience and often results in forgotten passwords or the usage of overly simplistic, insecure passwords. Furthermore, without a centralized system, it’s difficult to track user activities and preferences across different services.

## Main question:

Which IAM best aligns with the specific needs, user management, and the demand for a centralized system in an online education platform?

## ­Sub-questions:

1. How does KeyCloak function inside of IAM?
2. Literature study: By looking at the documentation, information about strengths and weaknesses can be found of the most popular IAM systems.
3. How can Keycloak be integrated with the specific needs and features of an online education platform?
4. Community Research: By searching for various forums and user created posts, The most active community can be found for all of the popular frameworks.
5. Observation: By looking at what other people are using and how they are customizing their framework, and whether this could be a good fit for an online education platform.

1. What are strengths and weaknesses for the 3 most popular IAM Systems?
2. Problem analysis: Research what the most popular vulnerabilities and attacks are towards frameworks. And how different frameworks find solutions for these problems.
3. Document analysis: By researching online guides and forums, a general idea can be formed of strengths and weaknesses of the most popular frameworks.
4. Which OWASP risks are covered by KeyCloak and which do still exist after implementing KeyCloak?
5. Document analysis: By reviewing the official documentation of KeyCloak, I can determine what risks are covered by KeyCloak.
6. SWOT analysis: By conducting a SWOT analysis for KeyCloak, identify the strengths, weakness, opportunities, and threats.

# Results

## Sub-question 1:

**How does KeyCloak function inside of IAM?**

IAM stands for Identity and access management, it is used to control critical information about user access within the organization. Systems used for IAM include but are not limited to single sign-on systems and two-factor authentication. Signle Sign-On (SSO) allows users to access multiple applications and services with a single set of credentials. This reduces the number of attack surfaces as there is fewer passwords for potential breach. IAM systems can also be useful for the ability to securely store identity and profile data. (Gittlen, 2021)

KeyCloak is an open-source tool for IAM, it supports Single Sign-On as well as support for different protocols such as OAuth. The system can be distributed in 3 ways including Docker. The importance of KeyCloak is automating the process of authentication in addition to being logged in on several applications and resources of the network. This would be beneficial for an online education platform since it is based on multiple functionalities that require a user to be logged in before using them. (Żyliński, 2021)

The extra security that KeyCloak offers is MFA (multi-factor authentication) also referred to as two-factor authentication, it gives extra security to the sign-in process by requiring the user to provide an additional form of identification. This could something like a PIN or passcode, generated by an external device. MFA is very important in an application since it reduces the security login risks by a lot. (Kinzer, 2021)

**SWOT analysis for the three most popular frameworks:**

|  |  |  |
| --- | --- | --- |
| React | Positive | Negative |
| Intern | 1. Component-based architecture 2. Large active community | 1. State management is more difficult for larger projects. 2. Frequent updates may lead to version compatibility issues. |
| Extern | 1. Integration with tools like React Native 2. Improvements in developer experience | 1. React changes and updates frequently which results in more maintainability for the application. |

|  |  |  |
| --- | --- | --- |
| Angular | Positive | Negative |
| Intern | 1. Strong TypeScript support. 2. Structured architecture. | 1. Harder to learn. 2. Angular has a low performance in runtime. |
| Extern | 1. Support for large-scale applications. 2. Integration with tools like Firebase. | 1. Complexity of the framework may discourage developers. |

|  |  |  |
| --- | --- | --- |
| Vue.js | Positive | Negative |
| Intern | 1. Lightweight and flexible. 2. Easy to learn and use, for beginners. | 1. Smaller ecosystem and community compared to react and angular. 2. Less adoption in large-scale applications. |
| Extern | 1. Growing popularity and community. 2. Continued improvements in performance and tooling | 1. Depends on a single creator/maintainer |

## Sub-question 2:

To know how readable and comprehensive a community is we must know the certain attraction to a framework and how many people interact with it. One way to do this would be social media, how active is a popular framework on different social media sites and how many users interact with this framework? I have done some research and put these numbers in visual graphs. React is the framework with the most interaction.  
A graph of different colored bars

Description automatically generated

Users that are members within LinkedIn 1

A graph of different posts

Description automatically generated

Total likes in the past 3 post over different social media platforms

A graph of different colored columns

Description automatically generated with medium confidence

Total posts of the framework on all social media platforms

A graph of followers

Description automatically generated

Followers of a framework on different social media platforms

To further investigate the popularity and likings of the community a survey was conducted about which framework they used and posted it in a reddit community for frameworks. We posted this survey in 2 communities and this is the results we received:

A black rectangular object with a white line

Description automatically generated

Survey 1 about which framework people have worked with before in the past (<https://www.reddit.com/r/learnjavascript/comments/16ug7hz/with_which_javascript_framework_have_you_worked/>)

A black rectangular object with a white line

Description automatically generated

Survey 2 about which framework people have worked with before in the past (<https://www.reddit.com/r/webdev/comments/16ug9yd/with_which_javascript_framework_have_you_worked/?rdt=42659>)

## Sub-question 3:

JavaScript frameworks like Angular, Vue, and React are widely used for developing web applications. Because large enterprises as well as many individuals use these frameworks, they are a notable target for various types of cyberattacks. For the most popular attacks I will research whether the most popular frameworks have countermeasures for them.

**Attacks: Cross-Site Scripting (XSS) and injection attacks**

XSS attacks function by injecting malicious scripts into webpages viewed by other users. These scripts can then be used to steal information and carry out actions on behalf of the user without their consent. Each framework handles this vulnerability differently.

Vue:

“Vue mitigates XSS through a feature named “escaping”. Where any information entered can be taken as a text string.” (Matias, 2022)

Angular:

“Angular uses several security features to protect applications from attacks. For example, Angular uses Content Security Policy (CSP) to prevent cross-site scripting (XSS) attacks. CSP is a security policy that helps detect and prevent XSS attacks by allowing sources of content to be loaded into a webpage.” (Srivastava, 2023)

React:

“Typically, this attack would work in traditional HTML, JS based application. But React is clever! Instead of executing the malicious code, it would interpret the code as a string” (Vindula, 2023)

the way frameworks handle OWASP top 10 really differs, as developer you must understand what security procedures were build-in the framework you are using and apply them accordingly.

**Angular**

This JavaScript framework offers an embedded safeguard against XSS attacks. "Cross-site scripting (XSS) enables attackers to inject malicious code into web pages. Such code can then, for example, steal user and login data, or perform actions that impersonate the user. This is one of the most common attacks on the web" (Żyliński, 2021). Angular introduces functionalities that assist in safeguarding against two prevalent internet security issues: cross-site request forgery (CSRF or XSRF) and cross-site script inclusion (XSSI). Although these problems are primarily addressed server-side, Angular provides utilities that facilitate easier client-side management. Angular incorporates intrinsic protection against malicious code in web applications, utilizing features such as Content Security Policy (CSP). CSP helps thwart cross-site scripting (XSS) attacks by governing which scripts are permissible to run on a webpage. Within Angular, CSP permits developers to define which external resources may be loaded, minimizing the likelihood of unauthorized script execution, and bolstering the application's overall security (Angular, 2023). Angular supplies coherent and comprehensible documentation on the application of its security features.

**React**

React does not provide inherent safeguards against Cross-Site Scripting (XSS) attacks. The onus is on developers to manually manage input sanitization or utilize third-party libraries (Gittlen, 2021). Furthermore, React does not inherent support for Cross-Site Request Forgery (CSRF) protection; hence, developers are tasked with instating suitable measures independently. Although Content Security Policy (CSP) is not exclusive to React, it can be imposed at the server level to alleviate specific kinds of attacks. Nonetheless, a plethora of online resources, articles, and advice is accessible, providing guidance on focal points and methods to tackle security issues within React applications.

**Vue.js**

In Vue, several vulnerability reports pertaining to Cross-Site Scripting (XSS) originate from instances where developers deliberately render unsensitized, user-supplied content as Vue templates. This practice is intrinsically hazardous and is not something Vue can mitigate. Moreover, initializing Vue on a page that includes server-rendered and user-contributed content can introduce analogous vulnerabilities. The recommended best practice advises against mounting Vue on nodes containing such content (Kinzer, 2021)."HTTP security vulnerabilities, such as cross-site request forgery (CSRF/XSRF) and cross-site script inclusion (XSSI), are primarily addressed on the backend, so they aren't a concern of Vue's. However, it's still a good idea to communicate with your backend team to learn how to best interact with their API, e.g., by submitting CSRF tokens with form submissions" (Kinzer, 2021).The official website provides warnings regarding potential risks in this area.

## Sub-question 4:

**What is OWASP?**

OWASP is an international organization that focuses on web application security. The OWASP top 10 are the most critical risks involved in web development. These are updated in real time and include but are not limited to Injection, broken authentication and sensitive data exposure. OWASP also provides recourses, tools, and best practices to assist in mitigating these risks. This includes guidelines for secure coding and official documentation further explanations.

**What OWASP risks does KeyCloak cover?**

**SWOT analysis**

# Resolution

## Conclusion:

As a result of this research report, the final conclusion is that React is the most suitable framework for an online education platform. This is because React has the largest community which is essential to the learning curve, since some developers have little to no experience with JavaScript frameworks (sub question 1-3). Pertaining to an online education platform, React offers the most flexibility which is needed for the different UI’s that an online education platform is going to need, and to make and pursue a flexible interface for users to chat/call in (sub question 4).

## Recommendation:

If you want to build a project that includes a flexible UI, as well as lots of help on outer community, choose React! The powerful JSX syntax that comes with React offers strong support for HTML-like features, if you have experience with HTML this will greatly help in understanding the framework.

## Reflection:

While writing this research report, I have chosen to apply differnt researching methods for different sub-questions, I did this because I have no past expierience with frameworks. What I found hard about writing this research report is determining what specific aspects of a framework work best for the project that I am going to make.

# References

Acunetix. (2023). *Cross-site Scripting (XSS)*. Retrieved from Acunetix: https://www.acunetix.com/websitesecurity/cross-site-scripting/

Bahrynovska, T. (2023, 1 31). *React security guide: 7u vulnerabilities and best practices to fix them*. Retrieved from Forbytes: https://forbytes.com/blog/react-security-best-practices/#:~:text=user%20authentication%20measures.-,React.,scripting%20and%20injection%2Dstyle%20attacks.

Hert. (2023, 10 5). *her*. Retrieved from hert: http://Google.com/

*How to Securfe Your Angular App*. (n.d.). Retrieved from https://www.telerik.com/blogs/how-to-secure-your-angular-app#:~:text=Angular%20uses%20several%20security%20features,be%20loaded%20into%20a%20webpage.

Laurent, L. (2023, 7 31). *AppMaster*. Retrieved from React's Component-Based Architecture: A Case Study: https://appmaster.io/blog/react-component-based-architecture

Mahajan, A. (2019, 12 05). *React vs. Angular vs. Vue: The Complete Comparison*. Retrieved from Dzone: https://dzone.com/articles/react-vs-angular-vs-vue-the-complete-comparison-to

Matias, M. (2022, 07 14). *Avoid XSS Attacks in Vue*. Retrieved from dev-academy: https://dev-academy.com/vue-xss/

Ritika. (2022, 12 1). *Invendus*. Retrieved from What is a Framework and Why use Frameworks in Software Development?: https://invedus.com/blog/what-is-a-framework-and-why-use-frameworks-in-software-development/#:~:text=The%20type%20of%20framework%20that,your%20specific%20needs%20and%20requirements.

Srivastava, V. (2023, 04 01 ). *How to Secure Your Angular App*. Retrieved from Telerik: https://www.telerik.com/blogs/how-to-secure-your-angular-app#:~:text=Angular%20uses%20several%20security%20features,be%20loaded%20into%20a%20webpage.

Thinkster. (2023, 10 26). *Realworld*. Retrieved from Github: https://github.com/gothinkster/realworld

Vindula, O. (2023, 06 6). *Mitigating XSS attackos in React applications*. Retrieved from medium: https://medium.com/@omalwijegunawardana/mitigating-xss-attacks-in-react-applications-194c46472379

You, E. (2023). *Security*. Retrieved from Vue.js: https://vuejs.org/guide/best-practices/security.html

# Version history

|  |  |
| --- | --- |
| **When?** | **What?** |
| 30/11/2023 | Initial setup of Main/Sub questions, problem description |