Research report  
Keycloak

**­­­­­­­­­­­**

**Afbeelding met schermopname, Graphics, Lettertype, logo

Automatisch gegenereerde beschrijving**

Student name: Rowan van der Weel.

Student number: 4924576

Class: DB-04

Teacher: Tim Kurvers

Version: 1.0

Date: 21-09-2023

Table of contents

[**Introduction** 3](#_Toc150523526)

[Problem description 3](#_Toc150523527)

[Main question 3](#_Toc150523528)

[Sub questions 3](#_Toc150523529)

[**Results** 4](#_Toc150523530)

[Sub question 1 4](#_Toc150523531)

[Sub question 2 4](#_Toc150523532)

[Sub question 3 4](#_Toc150523533)

[Sub question 4 4](#_Toc150523534)

[**Resolution** 5](#_Toc150523535)

[Conclusion 5](#_Toc150523536)

[Recommendation 5](#_Toc150523537)

[**References** 6](#_Toc150523538)

[**Version History** 7](#_Toc150523539)

# **Introduction**

## Problem description

In today's digital world, keeping online information secure is crucial. Businesses and organizations need effective tools to manage who can access their apps and data, ensuring only the right people get in while keeping everything safe. This challenge calls for Identity and Access Management (IAM) solutions.

Keycloak, an open-source IAM tool, steps in to help solve these challenges. The problem we are tackling is how to manage user identities, control who can access what, and make sure everything stays secure in modern apps. Without a good IAM system, there is a risk of security issues, not meeting compliance standards, and a lack of control over who gets to use what.

The main purpose of this research report is to understand its features and how well it can manage identity and access management. My goal is to implement in my personal project: “GamifyWork” and to determine for which project(s) it could be suitable.

## Main question

How can Keycloak be integrated into 'GamifyWork' for efficient identity and access management?

## Sub questions

1. What is IAM and how does Keycloak play a part in it?
   1. **Literature study:** Searching for academic papers, articles, and blog posts that discuss about Keycloak and other IAM’s.
   2. **Expert interview:** Conduct interviews with an expert in the field of Identity and   
      Access Management (IAM) and Keycloak.
2. How does Keycloak address specific OWASP security risks?
   1. **Document analysis:** Reviewing the official documentation for Keycloak. Look for sections or documents that explicitly outline the security measures.
   2. **SWOT analysis:** Conduct a SWOT analysis for Keycloak. Identify the Strengths, Weaknesses, Opportunities, and Threats associated with the IAM tool.
3. How easily can Keycloak integrate with the specific features of GamifyWork.
   1. **Community research:** Engage with online communities, forums, and social media groups dedicated to each of these frameworks. Observe discussions, queries, and the level of activity within these communities. Take note of the number of members, frequency of posts, and responsiveness to inquiries.
   2. **Observation:** Observe and analyse real-world applications or projects built using Keycloak. Pay attention to how developers have customized it to meet specific requirements.
4. To what extent does Keycloak allow for customization of the user interface, and how can this be delivered to align with the branding of "GamifyWork"?
   1. **Prototyping:** Develop a demonstration prototype displaying the potential customization capabilities of Keycloak's user interface in alignment with the branding of "GamifyWork."
   2. **Usability testing:** Gather feedback on the prototype through usability testing sessions with stakeholders and potential users.

# **Results**

## Sub question 1

**What are identity access management systems?**Identity and access management (IAM) is a system that helps businesses manage digital identities. It allows IT managers to control user access to essential information. IAM includes tools like single sign-on and two-factor authentication for secure access. Additionally, it ensures that only necessary and relevant data is shared, promoting data security (Rosencrance & Gittlen, 2021).

IAM offers diverse key benefits.

1. Enhanced Data Security: IAM strengthens cybersecurity by efficiently managing privileged access. It provides robust control over user access, reducing the risk of data breaches, identity theft, and unauthorized access to sensitive corporate information across various devices.
2. Facilitates Compliance: Integrating IAM into business operations aids in meeting regulatory requirements, including authentication methods, user access reviews, and resource location access.
3. Minimizes Human Errors: IAM tools automate access management, eliminating manual errors associated with privilege settings. This not only frees up the IT team from tedious tasks but also reduces the chances of human error, streamlining operations and cutting costs.
4. Ensures Data Confidentiality: IAM tools provide a secure way to grant access, maintaining confidentiality by restricting access to specific individuals or groups while safeguarding sensitive information.
5. Streamlines IT Workloads: IAM enables simultaneous updates of access privileges across the organization, reducing the number of IT tickets for password resets. This streamlines IT workflows and enhances efficiency. (Why Is Identity and Access Management Important?, sd)



**What is Keycloak?**

Keycloak, an open source "Identity and Access Management" tool under Apache License 2.0, serves as the upstream project for Red Hat SSO. Supporting multiple platforms based on chosen protocols, it was released in September 2014 and is actively developed by the Red Hat team, welcoming contributions. (Żyliński, 2021).

Additional feature highlights of Keycloak are:

* Single sign-on (SSO) support
* User federation (i.e., support for external identity providers)
* Support for popular protocols like OAuth2 and OpenID Connect
* Multifactor authentication (MFA)
* Fine-grained access control (Hmza, 2023).

### Expert interview

**Introduction**

In this interview, we have the privilege of conversing with Niko Köbler, a distinguished freelance Software- and IAM-Consultant with a profound expertise in Identity and Access Management (IAM) and a comprehensive understanding of Keycloak since its inception in 2015. Niko, who dedicates a significant 80% of his daily professional life to the intricacies of identity management, offers invaluable insights into the world of Keycloak. With a diverse portfolio, he not only provides public and private training sessions but also conducts meticulous architecture and configuration reviews. Furthermore, Niko actively engages in implementing custom extensions and integrations within existing landscapes, contributing significantly to the advancement of IAM practices. Beyond his consulting endeavours, he extends his knowledge-sharing commitment through a thriving YouTube channel, where he publishes informative videos focused on Keycloak, and as a speaker at major IT and software conferences. Join us as we delve into the wealth of Niko Köbler's experience and expertise, uncovering the nuances of Keycloak and the evolving landscape of identity management.

**What IAM solutions exist, and which one do you recommend, and why?**

“There are a lot of IAM solutions available, e.g. Wikipedia serves a long list of service providers and vendors. As my focus is on Keycloak IAM & SSO, I do recommend of course Keycloak. It was the first solution to be hosted and run in your local/private environment and keeping the souvereignty of the users data (credentials) in your company and not giving them to an outside solution hosted by a service provider.”

**What essential features make Keycloak stand out?**

“Keycloak is a full-featured and standalone solution for Identity Management and Single-Sign-On to be hosted and managed by yourself (see above). It follows the OpenID-Connect standard and comes with everything one might need for successfully authenticating users and clients. And it’s not only about authentication itself, there are features like registration, forgot password, multi-factor-authentication (MFA) with OTP, sign-in with social providers and brokering with custom IdPs, user federation with existing directories, like LDAP or AD and others. For managing the system, Keycloak comes with a comprehensive admin and management console where you can manage all aspects of the server, enabling/disabling features, creating and managing users and clients and managing policies, like e.g. the password-policies. Additionally there’s the account console, where users can manage their data, credentials and sessions themselves. If you need some custom functionality which Keycloak doesn’t provide out-of-the-box, there are a lot of extension interfaces, which can be used to implement the features you want or need by using Java code. Last but not least, the complete look-and-feel of the Keycloak UIs and forms can be customized with custom themes, so that it integrates also visually into your corporate identity.”

**What distinguishes Keycloak from other IAM solutions in the market (in your opinion)?**

“Keycloak is, like already mentioned above, a full-featured standalone solution, which can be hosted by the company itself. There’s no need to use a managed service provider and to give the most sensitive user data (credentials!) into foreign responsibility where you don’t know how this data will be processed and stored. With Keycloak you have the full control and power over your data!

Additionally, other solutions often come with multiple modules which you have to deploy and operate depending on what you need and what your requirements are. Not with Keycloak, you will get everything the Keycloak ecosystem offers in one compact server. And, if you need more, you have the possibility to extend Keycloak with several Service Provider Interfaces. Try that with a managed service…!”

**What are some best practices for implementing Keycloak in a new project, and any potential pitfalls?**

“This heavily depends on the requirements of the project and environment. There might be no big challenges in small projects, which are just using some standards and default settings. The bigger the projects become, and the more entities (users, clients, brokers, etc.) are involved, it might(!) become more complex.

Using OIDC as the authentication protocol in your clients (applications) is nowadays no more hassle, as nearly every programming language, library and ecosystem has some ready-to-use solutions, which you just have to integrate in your custom application. There’s no need to implement things on your own.

When it comes to authentication with the various available so called grant-types, one should be careful, what the used library uses, as there are some grant types which are deprecated and should and must not be used any more. It’s always a good advice to read and understand the OIDC spec and not just implement what some, maybe outdated, blog posts or tutorials on the internet will tell you. Just because it’s on the internet, it is not necessarily right. In my now 9 years of experience with Identity Management and OIDC, I think I’ve seen almost every failures and bull\*\*\*t companies can think of and unfortunately really do...”

**From your perspective, what are some emerging trends or developments in IAM, and how does Keycloak fit into them?**

“One of the most popular trend is of course the change from using password to using FIDO2 Passkeys. The use of Passkeys are based on the WebAuthn protocol, which is available in Keycloak already since 2019 or 2020 (? don’t know exactly), so Keycloak already knows how to use Passkeys, there are only a few customizations necessary to get it running in a convenient way (it’s already working ootb, but not that convenient as it can be).

There’s a lot of movement in the field of authentication, no matter if using OIDC in financial sector (FAPI), secure encryption standards (FIPS 140-2), Self-Sovereign Identity (SSI), Verifiable Credentials (VC), and so on… Due to the fact that Keycloak is an Open Source Software, there are SIGs (special interest groups) for nearly every topic, so that Keycloak will also be in future on of the first and leading solutions to provider new and up-to-date authentication possibilities.”

## Sub question 2

**OWASP**

The Open Web Application Security Project (OWASP) is a non-profit organization founded in 2001, with the goal of helping website owners and security experts protect web applications from cyber-attacks. OWASP has 32,000 volunteers around the world who perform security assessments and research (OWASP, sd).

I will research the [top 10](https://owasp.org/API-Security/editions/2023/en/0x11-t10/) security risks for Keycloak.

[**Broken Object Level Authorization**](https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/)

In Keycloak, handling Object Level Access Control (OLAC) involves setting up authorization policies and permissions based on object identifiers. Keycloak provides a flexible and customizable approach to implement object-level authorization checks. [Here's](https://www.youtube.com/watch?v=kBBf9k8RtrE) a full guide on how Keycloak manages Object Level Access Control.

[**Broken Authentication**](https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/)

Keycloak addresses Broken Authentication risks through diverse measurements. It supports secure authentication protocols like OAuth 2.0 and OpenID Connect. Multi-Factor Authentication adds an extra layer of security. Brute force protection is built-in, temporarily locking user accounts after multiple failed login attempts. Keycloak's session management controls session duration and idle timeouts to prevent unauthorized access. Administrators can enforce strong password policies, including length, complexity, and expiration. User self-service features enable password resets and profile updates. Identity federation integrates with external providers for enhanced security. (Server Administration Guide, 2022).

[Broken Object Property Level Authorization](https://owasp.org/API-Security/editions/2023/en/0xa3-broken-object-property-level-authorization/)

Keycloak effectively handles Broken Object Level Authorization (BOLA) through a robust access control framework. It employs roles, permissions, and resource-based authorization to define and manage access to objects. Keycloak's authorization policies, conditions, and mappers enable fine-grained control, allowing specific rules based on object identifiers. Scopes, client scopes, and dynamic authorization consent further contribute to precise access control. The platform supports custom authorization logic through its Service Provider Interface (SPI), providing flexibility for complex scenarios (Authorization Services Guide, sd).

[Unrestricted Resource Consumption](https://owasp.org/API-Security/editions/2023/en/0xa4-unrestricted-resource-consumption/)

Keycloak adeptly addresses Unrestricted Resource Consumption through a multifaceted approach. The platform incorporates rate limiting, allowing administrators to control the volume of incoming requests, preventing abuse and resource exhaustion. Token lifespan control, caching strategies, and optimized token validation processes are integral to resource optimization. Keycloak's scalability, load balancing, and performance monitoring capabilities ensure the system's resilience and prevent resource exhaustion in clustered environments (Authorization Services Guide, sd).

[Broken Function Level Authorization](https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/)

Keycloak effectively addresses Broken Function Level Authorization (BFLA) by employing Role-Based Access Control (RBAC), fine-grained policies. This ensures precise authorization checks for application functions. Keycloak's dynamic consent and token-based access control enhance flexibility. The platform's direct access to backend data eliminates the need for extensive data imports, streamlining authorization processes. Regular auditing and monitoring, along with adherence to best practices, further solidify Keycloak's robust handling of BFLA (Garvey, 2023).

[Unrestricted Access to Sensitive Business Flows](https://owasp.org/API-Security/editions/2023/en/0xa6-unrestricted-access-to-sensitive-business-flows/)

It allows precise control over user access at various levels, ensuring only authorized individuals can interact with sensitive business processes. Support for Multi-Factor Authentication (MFA) further enhance security. By following best practices and continuous monitoring, Keycloak establishes a robust defence against unauthorized access to critical business flows.

[Server Side Request Forgery](https://owasp.org/API-Security/editions/2023/en/0xa7-server-side-request-forgery/)

Keycloak guards against Server-Side Request Forgery (SSRF) by employing a URL whitelist, enabling administrators to specify permitted URLs. This restricts requests to predefined, trusted destinations, mitigating the risk of SSRF attacks. Additionally, Keycloak recommends secure configuration defaults, employs role-based access control, and promotes adherence to security best practices. By implementing these measures, Keycloak ensures a robust defence against SSRF vulnerabilities, reinforcing the platform's security posture. [Here](https://securityforeveryone.com/tools/keycloak-12-0-1-request-uri-blind-ssrf-unauthenticated-cve-2020-10770?) you can detect SSRF vulnerabilities.

[Improper Inventory Management](https://owasp.org/API-Security/editions/2023/en/0xa9-improper-inventory-management/)

Keycloak effectively mitigates Improper Inventory Management by implementing robust security measures. Utilizing Role-Based Access Control (RBAC), fine-grained authorization policies, and dynamic consent, Keycloak ensures precise control over user access to inventory resources. (RBAC for frontend and backend using Keycloak, 2022). Overall, Keycloak provides a resilient solution for managing identity and access while safeguarding against improper inventory management.

[Unsafe Consumption of APIs](https://owasp.org/API-Security/editions/2023/en/0xaa-unsafe-consumption-of-apis/)

The platform's controls over client registrations, token validation, and revocation further enhance the security of API consumption. Real-time monitoring and auditing capabilities enable the prompt detection of any unsafe API consumption practices, contributing to a robust and secure API environment. Most of the other measurements I already explained in the previous sections.

Documentation

The documentation I mostly used are:

* [Red Hat](https://access.redhat.com/documentation/en-us/red_hat_build_of_keycloak/22.0/html/server_administration_guide/index)
* [Keycloak docs](https://www.keycloak.org/docs/23.0.1/authorization_services/)

**SWOT-analysis**

|  |  |  |
| --- | --- | --- |
| **Keycloak security measurements** | **Positive** | **Negative** |
| **Intern** | * **Comprehensive Access Management** * **Powerful Security Implementation** | * **Challenging Deployment** * **Insufficient SPI Documentation** |
| **Extern** | * **Customization Capabilities** * **Integration with OWASP Resources** | * **Technical Support Limitations** * **Diverse Risk Analysis Approaches** |

**Strengths:**

Keycloak boasts comprehensive access management capabilities, offering organizations a robust solution for controlling user authentication and authorization. With a powerful security implementation, it ensures the safeguarding of sensitive data through features such as multi-factor authentication and role-based access control.

**Weaknesses:**

However, its implementation may pose challenges during deployment, requiring careful configuration to maximize security benefits. Additionally, users may encounter difficulties due to insufficient documentation of the Service Provider Interface (SPI), highlighting the need for comprehensive guidance to streamline integration and customization processes.

**Opportunities:**

Keycloak offers extensive customization capabilities, allowing organizations to tailor the identity and access management solution to meet specific business requirements. Its integration with OWASP resources reinforces its commitment to security by aligning with industry best practices for web application security.

**Threats:**

However, users may face limitations in technical support, requiring careful consideration of available resources and community assistance. Notably, Keycloak stands out with its diverse risk analysis approaches, providing organizations with flexible tools to assess and mitigate security risks effectively.

## Sub question 3

**Intrudocution**

Exploring how well Keycloak can mesh with the unique features of GamifyWork is our current study. We're looking into how Keycloak, which is a handy identity and access management tool, can smoothly fit with what makes GamifyWork stand out. The main aim is to figure out if they can work together easily, making GamifyWork more effective and secure in the process.

**Popularity**

Keycloak emerges as a popular choice within the community, reflecting a widespread adoption among users. However, the broad scope of Keycloak functionality is met with a shared sentiment of complexity during implementation. Many users acknowledge the challenges encountered when navigating its intricate features. Notably, the difficulty lies not in the popularity of Keycloak but in the comprehensive nature of its capabilities. On a different note, the discussion shifts to GamifyWork, where the focus is on simplicity. Opting for a straightforward approach, the plan involves initial steps in simple page customization. This phased strategy aligns with a pragmatic approach to gradually delve into integration, with the potential future incorporation of an existing database. This nuanced perspective highlights the balance between the popularity and intricacies of Keycloak, set against a deliberate and step-by-step implementation strategy for GamifyWork customization and integration efforts.

A screenshot of a computer

Description automatically generated

[Figure 1 Reddit poll](https://www.reddit.com/r/devops/comments/16tmdat/does_anyone_here_use_keycloak_as_their_main_idp/)

**Customization**

In a [Reddit](https://www.reddit.com/r/KeyCloak/comments/11nrju4/adding_a_custom_public_page_ftl_or_html_to/) discussion focused on adding custom pages to Keycloak, diverse perspectives emerged regarding the ease of the task. One user expressed difficulty, noting the absence of comprehensive tutorials for Keycloakify a potential solution. On the other hand, a contrasting opinion highlighted the simplicity achieved with Keycloakify. However, the limited availability of tutorials became evident, leaving room for improvement in guiding users through this process. Personal experimentation revealed that while achievable, customizations like changing backgrounds and adjusting box positions posed a notable challenge, requiring direct edits to Docker containers. This nuanced discussion underscores the mixed experiences users face in navigating customizations within the Keycloak framework, shedding light on areas that may benefit from enhanced documentation and user support.

**Observation**

## Sub question 4

# **Resolution**

## Conclusion

## Recommendation

# **References**

# **Version History**

|  |  |
| --- | --- |
| **When?** | **What?** |
| 10/11/2023 | First start, initialized it. |
| 30/11/2023 | Main- and sub questions with the specific methods. |
| 5/12/2023 | Subquestion 1 so far done, sub 2 working on. |
| 13/12/2023 | Subquestion 2 done |
| 20/12/2023 | Subquestion 3 haflway |