Gestión de identidades y credenciales. Caso de estudio en aplicaciones no seguras.

Autor Kevin Carracedo Vázquez

Titulación Máster en Tecnologías y Aplicaciones en Ingeniería Informática

Tutor Manuel Torres Gil

Departamento **Departamento de Informática. Área de conocimiento:** Lenguajes y Sistemas Informáticos

Modalidad Trabajo Técnico

Palabras clave Keycloak, Vault, Docker, Terraform

1. Introducción

La idea de este desarrollo que se va a realizar como **Trabajo Fin de Máster** viene de conocer de manifiesto una situación cada vez más frecuente en la que muchas empresas de la industria que tienen distintas soluciones TI, les surge la **necesidad de proteger su información**, bien por su volumen, sensibilidad y acceso a la misma en aplicaciones web y servicios en línea.

El **manejo de la privacidad de los datos** con diferentes sistemas y protocolos de autentificación a través del uso de credenciales, gestión de usuarios, roles y permisos **cada día va ganado popularidad** al encontrarse soluciones efectivas ante esta situación. Sin embargo, de cara al usuario **la complejidad de manejar diversa variedad de credenciales** para su autentificación **en los distintos servicios y aplicaciones empiezan a no ser tan eficientes**, debido al volumen de estas, así como su caducidad, con tiempos de validez diferentes, llegando a dar lugar como consecuencia la pérdida de información.

El incorporar una mayor seguridad en las distintas soluciones software y dependiendo sobre el tipo de arquitectura que haya seguido en su desarrollo llega a una situación de un alto coste no asumible por las empresas, y por ello muchas veces deciden no invertir en proteger su información y apuestan solo por realizar el mantenimiento de su producto actual.

En este trabajo se propone **implementar una solución** que consiste en **añadir una capa extra de seguridad a que se encargue de gestionar la autenticación y la autorización de usuarios y recursos, gestión de credenciales, etc. en aplicaciones que no tienen implementada esta funcionalidad de forma nativa**, delegando todo ello a plataformas especializadas.

2. Objetivos

La implementación de esta idea se basa en una de las herramientas más novedosas en el ámbito del desarrollo mencionada: **Keycloak[1]** y **Vault[2]**.

Keycloak, es un servidor de código abierto desarrollado por Red Hat que **proporciona utilidades como es la de gestión de identidades, sistemas de autenticación, autorización, federación, SSO (Single Sign-On), etc.** siendo muy interesante su aplicación **en aplicaciones web y móviles**. Su finalidad en este proyecto será el controlar la autenticación y la autorización a ciertas rutas u operaciones, estableciendo para ello entorno con configuraciones específicas para cada una de ellas. **Keycloak tomaría apoyo de Vault que tendrá un papel importante en el refuerzo de la seguridad de la aplicación**, desde donde se obtendrían las credenciales de los usuarios para su autentificación.

Vault se trata de un sistema de código abierto desarrollado por HashiCorp **diseñado para proteger, almacenar y gestionar credenciales, claves de API, certificados, etc. con técnicas de cifrado avanzadas y medidas de seguridad, políticas de acceso o tokens de acceso con un tiempo limitado de validez**. Vault será utilizado para el almacenamiento y gestión de credenciales y refuerzo de su seguridad, siendo totalmente transparente para las aplicaciones.

Para conseguir que este sistema de gestión y autentificación de usuarios sea un servicio independiente o aislado, escalable para las aplicaciones en las que se integrará, se hará uso de la virtualización de contenedores. **Docker[3]** es una herramienta que **permite crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina** con Docker instalado, teniendo como resultado un sistema aislado que cuenta **únicamente con los recursos mínimos y las librerías que se van a necesitar**. Docker permitirá crear esta capa extra de seguridad a implementar con Keycloak, Vault y todo lo necesario, sin ocasionar ningún impacto sobre el código fuente original de las aplicaciones.

El despliegue automatizado se realizaría mediante el uso de **Terraform[4]**, una tecnología de la compañía de Hashicorp **utilizada para la automatización de infraestructuras** a través de código basado en programación declarativa, pudiendo así definir una configuración simple y legible, ya sea en una infraestructura en local o basada en la nube. Ésta se conoce bajo el concepto de **IaaS[5] (Infraestructure as a Service)**. Esta tecnología se aplicaría en el desarrollo de este proyecto para realizar el despliegue de toda la infraestructura con todas las configuraciones realizadas de manera automática.

El objetivo del presente **Trabajo Fin de Máster (TFM)** consiste en implementar un sistema de gestión y autentificación de usuarios y acceso a recursos sobre una aplicación conocida, aumentando de esta forma su seguridad en situaciones en las que por ejemplo puede no tener un sistema de login y/o acceso o manipulación de la información, mejorando con ello su seguridad actual sin ningún impacto sobre su código fuente, automatizando su configuración inicial y despliegue automatizado de toda la nueva infraestructura creada.

3. Fases de desarrollo

El desarrollo del TFM se divide en las siguientes fases:

I. **Estudio y aprendizaje de Keycloak** (40h).
II. **Estudio de los diferentes métodos de autenticación de usuarios** (50h).
III. **Estudio y aprendizaje de Vault** (30h).
IV. **Creación servicio gestión y autentificación de usuarios** (40h).
V. **Integración servicio gestión y autentificación de usuarios** (30h).
VI. **Automatización configuración del nuevo servicio** (60h).
VII. **Despliegue automatizado infraestructura** (20h).
VIII. **Documentación memoria TFM** (30h).

4. Materiales y métodos

- Git como sistema de control de versiones y Git Bash como cliente.
- Windows 11 como S.O anfitrión.
- Visual Studio Code como editor de textos y herramienta de desarrollo.
- Keycloak para la autenticación de los usuarios y la autorización a ciertas rutas u operaciones de la aplicación.
- Vault para el almacenamiento y gestión de credenciales y refuerzo de la seguridad de la información.
- Docker para la elaboración de los diferentes contenedores y Docker Desktop como cliente.
- Terraform como herramienta de automatización y despliegue.

5. Bibliografía básica

[1] Stian Thorgersen and Pedro Igor Silva, *Identity and Access Management for Modern Applications: Harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications*, 1st Edition. Birmingham, United Kingdom: Packt Publishing. 2021.

[2] Anubhav Hanjura, *Implementing HashiCorp Vault*. Birmingham, United Kingdom: Packt Publishing. 2018.

[3] Sean P. Kane and Karl Matthias, *Docker: Up & Running*, 2nd Edition. California, United States: O'Reilly Media. 2015.

[4] Amín Espinoza de los Monteros, *Terraform. Curso práctico de formación*, 1ª Edición. Chile: Alpha Editorial. 2021.

[5] Matthias Marschall, *Chef Infrastructure Automation Cookbook*, 2nd Edition. Birmingham, United Kingdom: Packt Publishing. 2015.

Firma del director (codirector):