Universidad de La Habana
Facultad de Matemática y Computación

Sistema de Autenticación Central

Autor:
Nadia González Fernández

Tutores:
Lic. Roberto Martí Cedeño

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Noviembre de 2022

github.com/nala7/central_authentication_system

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Índice de figuras

Ejemplos de código

Introducción

Hoy en día las personas usan muchos sistemas de software independientes con distintas identidades. La identidad del usuario es considerado un conjunto permanente o de larga vida de atributos asociados a la identidad de un usuario. Para obtener acceso a servicios muchos requieren que el usuario esté registrado y haya iniciado sesión. Muchas de las veces los usuarios están registrados en los distintos sitios con el mismo nombre de usuario y con la misma o similar contraseña, lo cual no es la mejor práctica para preservar la seguridad de las cuentas. Por otra parte, es fácil de olvidar las credenciales, lo cual obliga a los gestores del sistema a enviar un correo no encriptado con información confidencial.

Por ello el manejo de múltiples nombres de usuarios y contraseña es una tarea molesta del Internet actual.

Teniendo en cuenta todos esos problemas de seguridad, cada vez más administradores de sitios web deciden delegar esos servicios de administración y autenticación a terceros, entidades externas especializadas en ese tipo de actividad. Pueden alojar, almacenar, administrar y proteger los datos de los usuarios de un proveedor de servicios en particular. Ofrecen una interfaz de programación de aplicaciones (conocida también por la sigla API, en inglés, *Application Programming Interface*), que pueden proporcionar acceso a datos de usuario para aplicaciones externas. Estas soluciones permiten compartir los datos con más de un sistema web, permitiendo así el Inicio de Sesión Único (SSO) para aplicaciones de terceros.[1]

El Nodo Central de la Universidad de La Habana tiene entre sus responsabilidades dar las credenciales digitales a todos los usuarios de la Universidad. Los trabajadores y estudiantes de la institución registran sus datos personales en las bases de datos de recursos humanos y secretaria docente respectivamente. Esa información es utilizada más adelante para que los usuarios puedan autenticarse en los distintos servicios de la institución, respaldado por sus sistemas de origen.

En este trabajo se presenta una solución a los problemas de autenticación que se ajusta a las necesidades y posibilidades de la Universidad de La Habana.

Motivación

Actualmente vivimos en un mundo donde la tecnología tiene un papel protagónico. La Universidad de La Habana en los últimos años ha estado inmersa en el proceso de transformación digital que lleva a cabo nuestro país, como continuidad de la estrategia de informatización de la sociedad cubana, que pretende integrar las tecnologías digitales a todos los ámbitos de la sociedad.

En este sentido, nuestra sede universitaria ha avanzado en la digitalización de la información y los procesos y trabaja para facilitar el acceso a la red de todos sus estudiantes y trabajadores. A la par de estos avances, debe crecer también la seguridad de los sistemas para evitar brechas de información y asegurar la privacidad de los usuarios.

Garantizar un acceso seguro y sencillo a los recursos de la red es esencial para alcanzar este objetivo. Por consiguiente, se deben establecer mecanismos que cuenten con un alto nivel de seguridad y permitan identificar quién realmente está autorizado para acceder a los recursos del sistema.

La creación de una plataforma central de autenticación resolvería muchos de estos problemas y mejoraría la experiencia de los usuarios. Estos solo tendrían una cuenta para acceder a todos los servicios que ofrece la Universidad, por lo que no tendrían que lidiar con formularios de inicio de sesión cada vez que vayan a acceder a un servicio.

Antecedentes

Actualmente todos los usuarios de la Universidad de La Habana se almacenan en dos Protocolos Ligeros de Acceso a Directorios (en inglés: Lightweight Directory Access Protocol, también conocido por sus siglas como LDAP). En los dos directorios se guardan las cuentas de correo y los datos de sus usuarios. En uno ellos se registra la información de los estudiantes y en el otro la de los trabajadores. A partir de estos, todos los sitios web y aplicaciones de la universidad, individualmente, verifican la pertenencia del usuario a la institución.

Cada aplicación y servicio tiene un servicio de autenticación individual que dependen de los dos mencionados directorios que contienen a los usuarios.

Problemática

La Universidad brinda servicios como Wi-Fi, correo, proxy y EVEA (Entorno Virtual de Enseñanza Aprendizaje), imprescindibles para el funcionamiento de la institución. También brinda servicios más especializados como los sistemas de contabilidad, recursos humanos, inventarios, el registro de notas, entre otros. Todos estos

servicios utilizan la autenticación de forma individual, lo cual significa que un usuario tiene diversas cuentas a pesar de pertenecer todas a la misma institución.

El sistema implementado actualmente es poco eficiente y requiere de intervención humana constante para corregir y/o restablecer el apropiado funcionamiento de los servicios de autenticación. Esto genera tiempos elevados de respuesta y dificulta el mencionado proceso de transformación digital que está siendo llevado a cabo por nuestra Universidad.

Objetivo

Con el propósito de presentar una propuesta para solucionar la problemática expuesta anteriormente, se plantean los siguientes objetivos:

Objetivo General

- Diseñar e implementar un sistema de autenticación centralizada para todos los usuarios de La Universidad de La Habana.

Objetivos Específicos

- Generar los servicios de autenticación con compatibilidad con todas las tecnologías existentes y previstas en la institución.
- Implementar la gestión de control de acceso a todos los servicios ofrecidos por el Nodo Central de forma extensible a futuros servicios y fuentes de datos.

Estructura de la Tesis

Recuerda que tu estado del arte va sobre autenticación.
Puedes salirte un poco de autenticación central y hablar de otras formas de autenticar (radius, kerberos, active directory u otros protocolos que te encuentres)
Es mas, te invito a que lo hagas.

En este capítulo se brindan las definiciones de herramientas utilizadas para la autenticacióna. También se realiza un estudio sobre el estado del arte de las mismas. Además, se brindan razones para incluir su utilización como parte de la solución propuesta.

Capítulo 1

Estado del Arte

Si quieres en lugar de este pequeño parrafo introductorio haz un resumen de todo lo que hablas en el capítulo

1.1. Inicio de Sesión Único

Tienes una perfecta definición en el párrafo a continuación. Pero como que queda un poco seca no crees ? Quien lo usa, desde cuando se emplea. Que ventajas y desventajas posee. Ejemplos de softwares que lo usan. Son de código abierto ? Son de código cerrado ?
Por ahi tienes bastante tela por donde cortar. Lo mismo aplica para el resto de los protocolos de autenticación.

El Inicio de Sesión Único (en inglés *Single Sign-On* o también conocido por sus siglas SSO) ha sido ampliamente adoptado para la autenticación en línea debido a su utilidad y seguridad que ofrece. Este es un método de autenticación que permite a los usuarios iniciar sesión con un único conjunto de credenciales en varios sistemas de software independientes. El inicio de sesión único facilita que un usuario no tenga que iniciar sesión en cada aplicación que use. Con este servicio los usuarios pueden acceder a todas las aplicaciones necesarias sin tener que autenticarse con otras credenciales.[2]

1.2. OpenID Connect

Hay una noción o distinción importante a la hora de diferenciar protocolos y que puedes abordar en tu estado del arte. Cada uno de los protocolos diseñados tiene como objetivo una arquitectura de red predeterminada.
Si bien OpenId Connect se diseño para entornos de internet donde no se puede verificar la autenticidad de los dispositivos clientes, también sirve para entornos mas

controlados (como puede ser la universidad)
Sin embargo, kerberos, radius y ldap si son pensados para redes controladas. (privadas)

OpenID Connect es una capa de identidad simple implementada a partir del protocolo *OAuth 2.0*. Permite a los clientes verificar la identidad del usuario final en función de la autenticación realizada por un servidor de autorización, así como obtener información básica del perfil del Usuario final de manera interoperable y similar al protocolo REST. [3].

OpenID Connect es uno de los protocolos de tipo *Single Sign-On* más utilizados para delegar la autenticación. También tiene un formato simple, por lo que ha ganado popularidad y es soportado por grandes empresas como Google, IBM, Microsoft, Amazon y PayPal [4]. La nueva versión es compatible con *API* y puede ser usado por aplicaciones nativas y móviles. También define mecanismos opcionales más robustos para firmas y cifrados. [3]

Este protocolo está basado en *OAuth 2.0* por lo que tiene todas las ventajas de este protocolo. Sin embargo, lo extiende ya que ofrece facilidades para obtener más información de la identidad de los usuarios eficientemente. Permite un flujo de información adicional que genera un *id-token* que contiene datos del usuario. De esta forma las aplicaciones no solo tienen acceso a los permisos de los usuarios, sino también obtienen información sobre la identidad de los usuarios. [3][1]

1.3. SAML

The Security Assertion Markup Language (SAML) defines the syntax and processing semantics of assertions made about a subject by a system entity. In the course of making, or relying upon such assertions, SAML system entities may use other protocols to communicate either regarding an assertion itself, or the subject of an assertion. This specification defines both the structure of SAML assertions, and an associated set of protocols, in addition to the processing rules involved in managing a SAML system.
SAML assertions and protocol messages are encoded in XML [XML] and use XML namespaces [XMLNS]. They are typically embedded in other structures for transport, such as HTTP POST requests or XMLencoded SOAP messages. The SAML bindings specification [SAMLBind] provides frameworks for the embedding and transport of SAML protocol messages. The SAML profiles specification [SAMLProf] provides a baseline set of profiles for the use of SAML assertions and protocols to accomplish specific use cases or achieve interoperability when using SAML features. [5]
SAML, developed by the Security Services Technical Committee of the Organization for the Advancement of Structured Information Standards (OASIS), is an XML-based framework for communicating user authentication, entitlement, and attribute infor-

mation. As its name suggests, SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application. SAML is a flexible and extensible protocol designed to be used – and customized if necessary – by other standards. [6]

1.4. LDAP

El Protocolo Ligero de Acceso a Directorios (en inglés: *Lightweight Directory Access Protocol*, también conocido por sus siglas de LDAP) es un conjunto de protocolos de licencia abierta que son utilizados para acceder a la información que está almacenada de forma centralizada en una red. Este protocolo se utiliza a nivel de aplicación para acceder a los servicios de directorio remoto. [7]

LDAP está basado en estándares implementados sobre TCP/IP Y Open Id Connect ?. Permite a los clientes interactuar directamente con los servidores de los directorios: almacenar y consultar información, buscar datos filtrados, autenticar usuarios, entre otros.

Este protocolo es utilizado actualmente por muchos sistemas que apuestan por el software libre al utilizar distribuciones de Linux para ejercer las funciones propias de un directorio activo en el que se gestionarán las credenciales y permisos de los usuarios y estaciones de trabajo en redes LAN corporativas en conexiones cliente/servidor mientras que open id connect y el flujo de emisión de tokens esta pensado para redes abiertas (internet).

Un directorio remoto es un conjunto de objetos que están organizados de forma jerárquica, tales como: nombre, claves, direcciones, etc. Estos objetos estarán disponibles para una serie de clientes conectados mediante una red, normalmente interna o LAN, y proporcionarán las identidades y permisos para esos usuarios que la utilicen.

Sería bueno que a cada protocolo le incluyas una url con el rfc que lo define (cualquiera de ellos, el primero o el último) (puede haber mas de un rfc por protocolo)
Ahora si esta en tu objetivo de tesis entender un protocolo, su link lo pones en la bibliografía consultada, por el contrario si es adicional, ponlo como un pie de página.

LDAP está basado en el protocolo X.500 para compartir directorios, y contiene esta información de forma jerarquizada y mediante categorías para proporcionarnos una estructura intuitiva desde el punto de vista de la gestión por parte de los administradores.

Estos directorios se utilizan generalmente para contener información virtual de usuarios, para que otros usuarios accedan y dispongan de información acerca de los contactos que están aquí almacenados. Además es capaz de comunicarse de forma remota con otros directorios LDAP situados en servidores que pueden estar en el otro

lado del mundo para acceder a la información disponible. De esta forma se crea una base de datos de información descentralizada y completamente accesible.

El sistema de autenticación vigente en el Nodo Central verifica sus usuarios con dos sistemas implementados con LDAP. Este protocolo se adapta a las necesidades y condiciones actuales de la Universidad ya que es *Open Source* (OSS o código abierto). - It is a mature, flexible, and well supported standards-based mechanism for interacting with directory servers. It's often used for authentication and storing information about users, groups, and applications, but an LDAP directory server is a fairly general-purpose data store and can be used in a wide variety of applications.
LDAP is a tool in the User Management and Authentication category of a tech stack.

1.5. Keycloak

Keycloak es un software de código abierto que permite el *Single Sign-On* o Inicio de Sesión Único con *Identity Management* y *Access Management* para aplicaciones y servicios modernos. Esta herramienta facilita la protección de aplicaciones y servicios con poca o ninguna codificación. Un Proveedor de identidad (en inglés: *Identity Provider*, también conocido por sus siglas IdP), permite que una aplicación (a menudo llamada *Service Provider* o SP) delegue su autenticación. [8]

Este software está escrito en Java y es compatible de forma predeterminada con los protocolos de federación de identidad SAML v2 y OpenID Connect (OIDC) / OAuth2. Está bajo licencia de Apache y es compatible o mantenido? con Red Hat. [8]

1.5.1. Características

Los usuarios se autentican en Keycloak en lugar de hacerlo en las aplicaciones. Esto significa que no es necesario que cada aplicación tenga un formulario de inicio de sesión, autentique a los usuarios o almacene sus datos. Una vez entren en Keycloak, los usuarios no tendrán que iniciar sesión en las demás aplicaciones conectadas al software.

Lo mismo sucede cuando un usuario cierra sesión. Keycloak ofrece cierre de sesión único, lo cual significa que los usuarios solo tienen que desconectarse en una de las aplicaciones para salir de su cuenta en el resto.

Otra prestación de Keycloak son las federaciones de usuarios, que facilitan la compatibilidad con LDAP y otros servidores de directorios activos. También admite la implementación de servicios propios para usuarios guardados en otros tipos de almacenamientos como en bases de datos relacionales.

Keycloak ofrece como herramienta una consola de administración de cuentas, a través de la cual los usuarios pueden administrar sus propias cuentas. Pueden actua-

lizar su perfil, cambiar sus contraseñas y configurar la autenticación en dos pasos. También pueden administrar sus sesiones y visualizar el historial de su cuenta.

Otra característica es que es una herramienta extensible porque permite la eliminación, adición y modificación de las bases de datos de usuarios, los métodos de autenticación y los protocolos. Está basada en protocolos estándares y soportan OpenID Connect, OAuth 2.0 y SAML. [8]

Keycloak facilita añadir la autenticación y un servicio seguro a aplicaciones. Permite que los desarrolladores se centren en la funcionalidad empresarial al no tener que preocuparse por los aspectos de seguridad de la autenticación. También posibilita la unificación de los métodos de autenticación de distintas aplicaciones sin modificarlas.

Que empresa usa KeyCloack, Que empresa usa Gluee ? Desde que año estan en el mercado?
Gluu es otra de las tecnologías que tienen prestaciones y ventajas similares a Keycloak. Es un servicio *Open Source* que soporta *SMAL*, *OpenID Connect, SSO* y *OAuth 2.0*. Sin embargo Gluu es un sistema que requiere de 8 GB de RAM y 40 GB de espacio en disco, mientras que Keycloak solo necesita de 512 Mb de RAM y 1 GB de disco. Por ello Keycloak se ajusta más a los recursos que se tienen en la Universidad de La Habana.

Que otras soluciones de SSO existen en la industria ? (Google, Facebook, Incluso Telegram permiten integración con SSO)
Otro punto a seguir puede ser que soluciones de autenticación se poseen para redes corporativas (cerradas)
poner ejemplos, hablar de ellas y etc (keycloak y gluee sirven para eso también)
Y finalmente puedes hablar del estado de la autenticación de la Universidad de La Habana, puedo, si quireres por servicio, mencionarte el protocolo que se emplea para la autenticación
Recuerda, tu estado del arte va de autenticación. Hay mucha tela por donde cortar aca. No te limites a lo que hay en la uh. Todo lo que se autentique hoy en internet puede estar en tu tesis (protocolo)

1.5.2. Google Authenticator

Millions of web users today use their Google accounts to sign into millions of relying party websites. This is enabled through the Google authentication API, which allows third party application developers to embed Google sign-in into their application. However, regardless to the strength of the Google authentication mechanism, the majority of these applications have been identified to be infected with broken authentication, which made them vulnerable to cyber attacks. A major reason for this is mistakes that developers make while embedding Google sign-in into their ap-

plication. High complexity and lack of usability of the Google authentication API makes it difficult for programmers to use it correctly and lead them to make mistakes while using the API. [9]
We discussed how these usability issues would affect the security of the application that are developed using the Google authentication API and how the API should be improved to provide a better experience to application developers. [9]

1.6. Seguridad

Although security is a crucial aspect of any application, its implementation can be difficult. Worse, it is often neglected, poorly implemented and intrusive in the code. But lately, security servers have appeared which allow for outsourcing and delegating all the authentication and authorization aspects. Of these servers, one of the most promising is Keycloak, open-source, flexible, and agnostic of any technology, it is easily deployable/adaptable in its own infrastructure.

Capítulo 2

Propuesta

Capítulo 3

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

- El rendimiento del nuevo sistema es más eficiente ya que en cada iteración añade solo los nuevos usuarios en lugar de borrar y recrear cada noche todos sus usuarios. - El tiempo de activación de un usuario (el tiempo que transcurre entre que un usuario es dado de alta por recursos humanas y que el usuario tenga acceso a todos los servicios provistos por el Nodo Central como correo electrónico y acceso a internet) es más corto. - Sistema de autenticación más fiable

Recomendaciones

Recomendaciones

Bibliografía

[1] R. Kutera y W. Gryncewicz, «Single sign on as an effective way of managing user identity in distributed web systems. The ActGo-Gate project case study,» Informatyka Ekonomiczna, n.º 2 (40), 2016 (vid. págs. 1, 6).

[2] Microsoft. (2022). «What is single sign-on in Azure Active Directory?» Dirección: `https://learn.microsoft.com/es-es/azure/active-directory/manage-apps/what-is-single-sign-on` (vid. pág. 5).

[3] OpenID. (2022). «Welcome to OpenID Connect,» dirección: `https://openid.net/connect/` (vid. pág. 6).

[4] C. Mainka, V. Mladenov, J. Schwenk y T. Wich, «Sok: Single sign-on security—an evaluation of openid connect,» en 2017 IEEE European Symposium on Security and Pr IEEE, 2017, págs. 251-266 (vid. pág. 6).

[5] R. Philpott, N. Ragouzis, T. Wisniewski, E. G. Whitehead, H. H. Hinton, C. P. Cahill, J. Bradley, I. J. Hodges, I. J. Brennan, L. Alliance y col., «Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2. 0–Errata Composite,» 2015 (vid. pág. 6).

[6] T. Wisniewski, E. T. Nadalin, S. Cantor, I. J. Hodges y N. P. Mishra, «SAML V2. 0 executive overview,» OASIS, 2005 (vid. pág. 7).

[7] LDAP. (2022). «LDAP,» dirección: `https://ldap.com/` (vid. pág. 7).

[8] Keycloak Documentation, `https://www.keycloak.org/documentation` (vid. págs. 8, 9).

[9] C. Wijayarathna y N. A. Arachchilage, «An empirical usability analysis of the google authentication api,» en Proceedings of the evaluation and assessment on software engine 2019, págs. 268-274 (vid. pág. 10).