Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 5, No.2, Juli 2023, pp. 0230204-01 ~ 0230204-016
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v5i2.15795

Single Sign On Using Keycloak Integrated Public Key Infrastructure for User Authentication In Indonesia's Electronic Based Government System

Wawan Hermawan[1*], Hendrawan[2], Eueung Mulyana[3], Dini Fronitasari[4], Stella Bella Vita W[5], Farhan Ardiya Fernanda[6]

[1,2,3,5,6]School of Electrical Engineering and Informatics, Bandung Institute of Technology, Jl. II, Lebak Siliwangi, Coblong, Bandung City, West Java 40132, Indonesia

[4]National Research and Innovation Agency.

*33221037@std.stei.itb.ac.id

Abstract. *The government in carrying out its function as a public administration servant is regulated in law of the Republic of Indonesia number 25/2009 on public services. In this regulated about electronic government (e-government), many individuals use various web applications that require users to authenticate themselves to access each application. Many entities require various web- based applications for operational activities. This makes centralized access management for web-based applications very much needed. Currently, access management is often implemented using Single Sign On (SSO) with password authentication method. Security considerations arise against the use of passwords. This is because passwords have a vulnerability to brute forcing using a password list, and human nature often uses repeated or uncomplicated passwords. There is an alternative authentication method,* namely *Mutual TLS which utilizes Public Key Infrastructure (PKI). Users authenticate with X.509 digital certificates, so the authentication factor becomes something you have. This research aims to implement an integrated PKI SSO system and RBAC access automation. The approach of this project is research, design, implementation, and testing. The entire system is built with open-source software and implemented on a cloud infrastructure. The system has three subsystems, namely registration, login and RBAC access automation. All subsystems are tested according to the specified flow. The test results show that the registration subsystem has been successfully carried out as evidenced by the success of filling in personal data, approval flow, and downloading of certificates. The login subsystem was also successfully implemented, as evidenced by the existence of mTLS authentication with certificate validation. In testing the RBAC access automation subsystem, it is shown that the script created can perform access checks and access remediation if needed.*

Keywords: Digital government, E-government, SSO, PKI, RBAC

(Received 2023-06-22, Accepted 2023-07-31, Available Online by 2023-07-31)

1. Introduction

Nowadays Information Technology (IT) has been evolving significantly in many fields not only for which closely related to Information Technology or Information System, but it also penetrated into different fields including government's policies and businesses [1]. Electronic government (e-government) is the administration of government that uses information and communication technology to provide services to e-government users. Through the ministry of communication and information technology, the Indonesian government currently has a roadmap related to the Indonesia Digital Nation 2021-2024, which touches on four strategic sectors: digital infrastructure, digital government, digital economy, and digital society [2].

The first Digital Indonesia Roadmap is accelerating infrastructure to expand public access to the internet. Second, encourage technology adoption. Third, increasing digital talent and finally finalizing supporting regulations to prepare a digital society. In addition, the Ministry of Communications and Information Technology has prepared the construction of data centers that will be placed in four locations to realize One Indonesia Data. With the massive development of digital infrastructure and the provision of government data centers for digital government services, it is hoped that this can be utilized to develop Indonesia's digital economy. The digital roadmap is also expected to encourage further the success of digital transformation in Indonesia, which is more focused and can accelerate the digital transformation process in the government environment so that later it can change how the government provides public services for the community.

Figure 1. Indonesia Digital Nation Roadmap 2021-2024

Specifically for the digital government sector, the government has issued Presidential Regulation No. 95 of 2018 concerning the Electronic-Based Government System (SPBE). The presidential regulation regulates the governance of electronic-based government systems, management of electronic-based government systems, auditing of information and communication technology, administration of electronic-based government systems, acceleration of electronic-based government systems, and monitoring and evaluation of electronic-based government systems. The implementation of SPBE itself departs from the problems that have occurred so far, such as the occurrence of waste of ICT spending due to each ministry building its government applications. With such conditions, there is also the disintegration of government information systems, so the validity of government data is not fully believed.

Figure 2. Presidential Regulation No. 95 of 2018 concerning SPBE

Estonia is one country that has successfully implemented E-Government or an electronic-based government system. In the context of E-Government, Estonia is an example of extraordinary success and attracts international attention, especially in developing countries [3], [4]. The main factor for Estonia's success that influenced and contributed to the evolution of E-Government was its information systems research.

The security vulnerability of electronic systems is currently one of the things that must be a concern, including electronic-based government systems. We must understand that in the security of electronic systems, we are all attractive targets for cybercriminals. Some threats may include data theft, such as usernames, passwords, documents, and emails. Most cyberattacks are common and can happen to anyone, although personalized attacks can happen. One of the primary and common triggers of cyberattacks is human error.

According to [5], every online user leaves a digital imprint of his data and identity on the internet. Some of the activities carried out by online users are often unsafe, and many cyber criminals commit fraud and are always looking for new ways to get hold of someone's data. Identity plays a significant role in our daily activities because identity is a complex subject. Building an identity usually requires several supporting components such as a verification process, what you know (e.g., username, password, or PIN), and what you have (e.g., cellphone number or token generator) to allow individuals to complete the process whether the transaction is rejected or accepted.

With the current advancement in information system applications, access control is one of the crucial aspects within an organization to protect information [3]. One of the most popular methods of authentication is password-based access,which is vulnerable to attacks. A Secure Sockets Layer (SSL) based on Public Key Infrastructure (PKI), now popularly known as Mutual TLS Authentication (mTLS), provides a safer solution as an authentication method [4]. Recently, mTLS is gaining traction again caused by the Ease of Use increasing usage of security-critical activities online such as banking and healthcare, thus escalating the needs for mutual authentication [5].

Role Based Access Control (RBAC) is one of the access control models that is suitable for organizational structure [6]. It enables authorization within an application with the role asa reference of a user's permissions. PKI based RBAC can beachieved by embedding the role information within a user's certificate. Previous studies [4], [7], [8] have explored ways to use PKI for RBAC, mostly by adding an external database forRBAC and integrating it with LDAP. The use of RBAC within an organization is often related to the use of SSO, since it is now the common standard of managing access.

There are also similar platforms that use SSO as a single-sign-on app access feature. One of them is the OKTA platform. OKTA is an identity and access platform that provides authentication, authorization, and identity management services for cloud and internal applications. With OKTA, organizations can manage user identities, control access to applications and data, and provide a secure authentication experience for their users. Okta also uses RBAC as well as PKI and features X.509 certificates. However, OKTA also has some disadvantages compared to using keycloak, including cost, OKTA is a paid platform, costs can increase with the number of users and features used, so it needs to be considered in budget planning. OKTA's learning curve requires a fairly complicated and complex technical understanding and skills, OKTA is a cloud platform, so it relies on a reliable internet connection to be able to use it. Customization is limited. For this reason, researchers conducted validity and usability tests to compare the quality of the platform to be used with other similar platforms.

PKI is the basis of a security infrastructure whose services are implemented using the concept of public key cryptography [9]. PKI is an embedded technology that underlies the provision of security for Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) thereby enabling a user to securely access public networks such as the internet [10]. PKI provides digital certificates that can identify individuals or companies and provides a directory service that can store them and revoke them. Public-key cryptography is based on a key pair from a trusted authority. These key pairs (private key and public key) are stored separately and both keys are required for communication to take place [11].

Single sign-on (SSO) is a mechanism that uses a single authentication action to allow authorized users access to all associated software systems or applications without being asked to log back in on

each system during a specific session. In PKI-based SSO, the server and user authenticate each other by using their respective key pairs. Users can authenticate the server by challenging the server to decrypt the messages they send encrypted by the server's public key. In the same way, servers can authenticate users by challenging them to decrypt messages they send encrypted by the user's public key. Since only the true owner of the private key can decrypt it. User and server certification authorities can differ and if they differ there should be trust between certification authorities [12]. This approach utilizes public key cryptography for user authentication. This system relies on the role of the Certificate Authority (CA) for the issuance and management of digital certificates and users' digital identities [6].

RBAC is based on the given authority of the role the entity has assigned. Each user has a specific role that must be matched with the operating executive. RBAC can perform access control based on user roles in the organization. Access management based on the RBAC concept can be implemented by utilizing PKI. PKI binds access rights to entities provided by authorities through digitally signed data structures. This model can ensure protected system authorization and maximize user management flexibility [8].

This paper proposes an SSO integrated with PKI and RBAC automation system. The authentication method of SSO will be mTLS. This paper heavily focuses on the RBAC automation system that will help administrators to remediate access according to a desired configuration. All of the software components are fully open-sourced, thus making it cheaper and easier for installations and deployments.

2. Methods

2.1. Cloud Infrastructure

The overall cloud infrastructure has 5 GCE instances. Access to the GCP infrastructure using a VPN while in development, this is implemented to mitigate the threat of attacks via public IP which often occurs with automated scanners. The illustration of the cloud infrastructure is shown in Fig. 3.

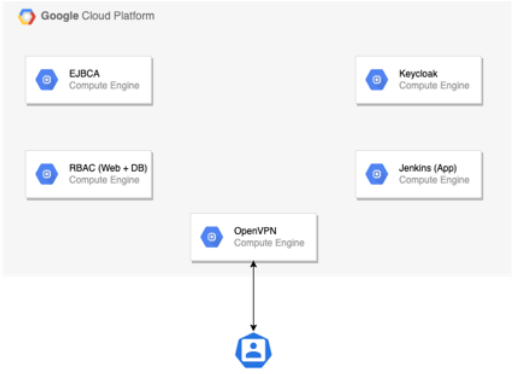

Figure 3. Cloud infrastructure of proposed system

The EJBCA instance is used for managing the PKI configuration and as a host to run the access automation script. Keycloak instance contains the software for the SSO login. The RBAC instance is used for configuring the RBAC and acts as a web application connected to SSO. Jenkins instance is also a web application connected to SSO. Meanwhile the OpenVPN instance is a VPN server to connect external traffics to the network.

2.2. Registration Flow

The registration flow of the proposed system is shown in Figure 4.

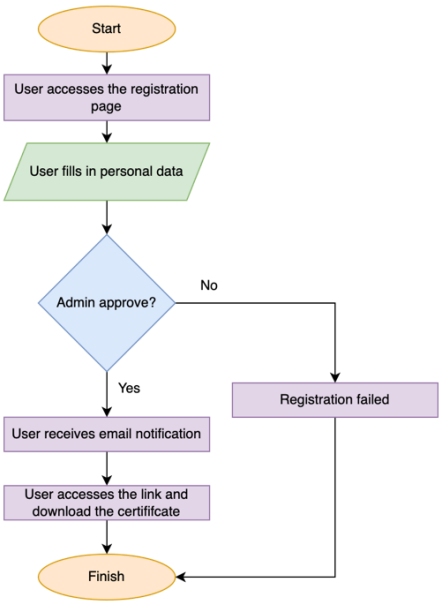

Figure 4. Registration flow

The registration process is held completely in the web. New users can access the registration page and fill in their data. The data will be read by an administrator and then the request can be approved or denied. If the request were denied, the registration process will fail. Meanwhile, if the request were approved, then the user will receive an email notification with a link to get download the certificate. The user can then access the link and provide the credential set when filling data to prove that the user is indeed the one who files the request.

2.3. Login Flow

The login flow of the proposed system is shown in Figure 5 below:

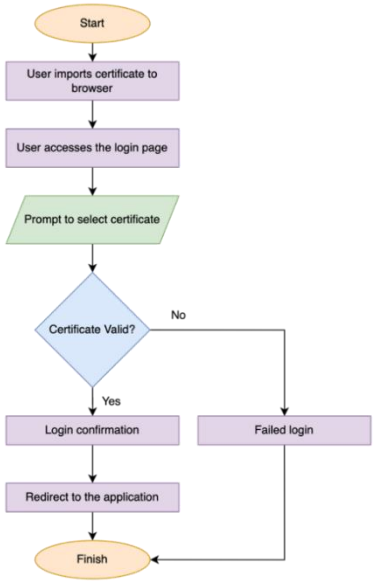

Figure 5. Login flow

The login process can be held after the user has completed the registration process. The downloaded certificate needs to be imported to the preferred browser. A user can then access an application or the SSO page to authenticate. The web prompts a selection of certificate to be used as the identity.

The SSO then checks whether the certificate is valid or not. If the certificate is invalid, then the login process is failed. Meanwhile if the certificate is valid, then the SSO shows a login confirmation and redirects to the application. The validation is done by checking the certificate issuer, user (email), expiry, and Certificate Revocation List.

2.4. RBAC Automation Flow

The RBAC automation flow of the proposed system is illustrated in Figure 6.

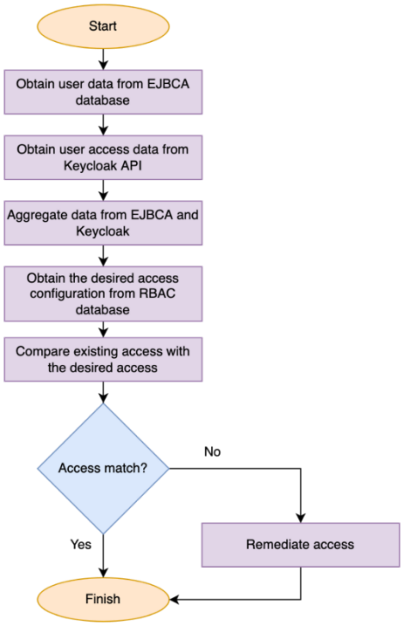

Figure 6. RBAC automation flow

The RBAC automation subsystem works by comparing the existing access and the desired access. Data regarding the current access is obtained by gathering all user information from EJBCA and Keycloak. Meanwhile data regarding the desired access is obtained from the RBAC database. If the automation subsystem finds any difference between the existing and desired access, then the subsystem will remediate the access accordingly.

3. Results and Discussion

3.1. Registration Subsystem

The PKI management system used is EJBCA [13] with the open-source version. EJBCA itself has a built in RA Web that can be used as the registration page. The user certificate that is generated will have the following properties:

- E : Email
- CN : Common Name
- OU : Organization Unit
- O : Organization

The CA structure of the proposed system contains a root CA and an intermediate CA. This is done to match the Keycloak mTLS requirement for CRL checking. There will be two roles that is implemented within EJBCA. One of them is a user role, which is only capable of requesting a user certificate. The other one is a super admin role, which can practically do anything as it is the default admin user. The privileges configured for the user role are as follows:

- /ca_functionality/create_certificate/
- /ra_functionality/view_end_entity/
- /ra_functionality/create_end_entity/
- /ra_functionality/delete_end_entity/
- /ca/[CA Name]/
- /endentityprofilesrules/[ProfileName]/view_end_entity/
- /endentityprofilesrules/[ProfileName]/create_end_entity/
- /endentityprofilesrules/[ProfileName]/delete_end_entity/

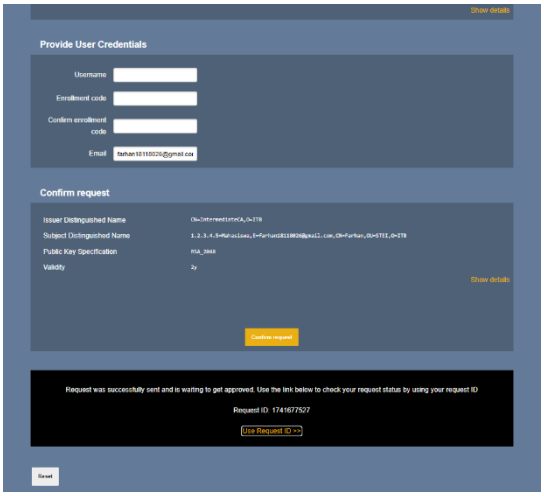

Figure 7. Registration request in EJBCA

The sample registration request in EJBCA is shown in Figure 7. The user can provide details about their identity and createa credential that acts as a passphrase of its private key and a password for downloading the certificate. After the request issent, it will have to be approved by the administrator.

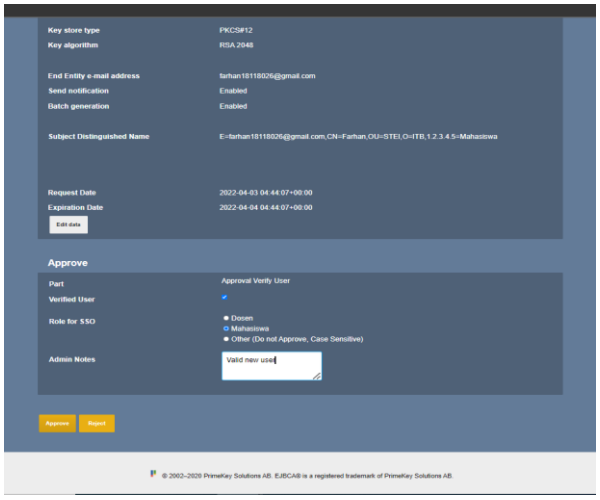

Figure 8. Registration approval in EJBCA

The registration approval page is shown in Figure 8. An administrator can review the details of the user's request. The page has a form for administrator to fill as a note. The form is highly customizable so that it can conform to any use cases.

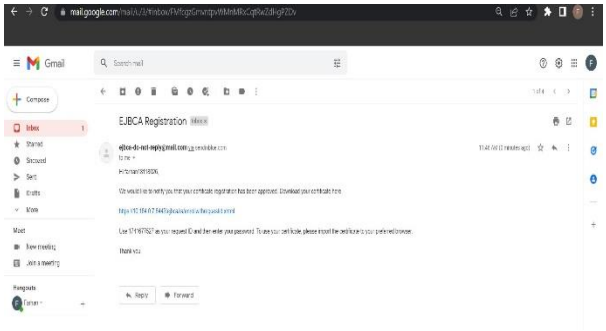

Figure 9. Email notification

If a request is approved by the administrator, the user will receive a notification email as shown in Figure 9. The email contains a link and a request ID to obtain the certificate.

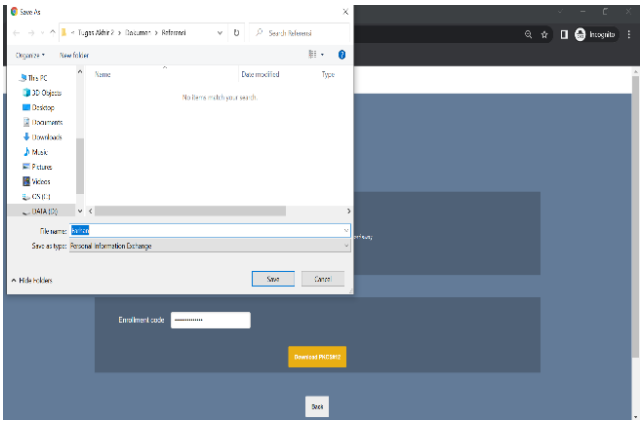

Figure 10. User certificate download

Downloading the certificate is shown in Figure 10. The user can access the link and enter the request ID to check whether the certificate is ready to be downloaded. To download the file, the user must enter the credential that was made during the making of registration request.

3.2. Login Subsystem

The proposed system uses Keycloak [14] as its SSO software, as it is an open-source software. To make it easy to install, keycloak is installed via docker. The container is defined by a dockerfile shown below.

```
1.  services:
2.    keycloak:
3.    image: quay.io/keycloak/keycloak:14.0.0
4.    command: -c standalone.xml
5.    environment:
6.      DB_VENDOR: h2
7.      KEYCLOAK_USER: admin
8.      KEYCLOAK_PASSWORD: admin
9.      X509_CA_BUNDLE:
    /etc/x509/https/rootCA.crt
10.   ports:
11.     - 8080:8080
12.     - 8443:8443
13.   volumes:
14.     -
    ./x509/localhost.crt:/etc/x509/https/tls.crt
15.     -
    ./x509/localhost.key:/etc/x509/https/tls.key
16.     -
    ./x509/rootCA.crt:/etc/x509/https/rootCA.crt
```

Before creating the container, Keycloak is registered a certificate for HTTPS and mTLS purposes. The issuer CA and Keycloak server certificates is imported to the container. The issuer CA certificate is used as a reference to validate mTLS authentication.

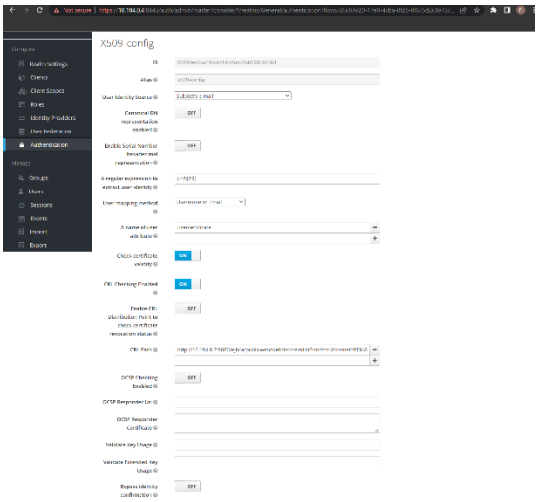

Figure 11. Keycloak mTLS configuration

The configuration of mTLS is presented in Figure 11. Keycloak maps a user by the email field provided in the certificate. Certificate validity checking is enabled to deny any expired certificates. The system checks for revocation status by accessing the issuer's CRL by URL.

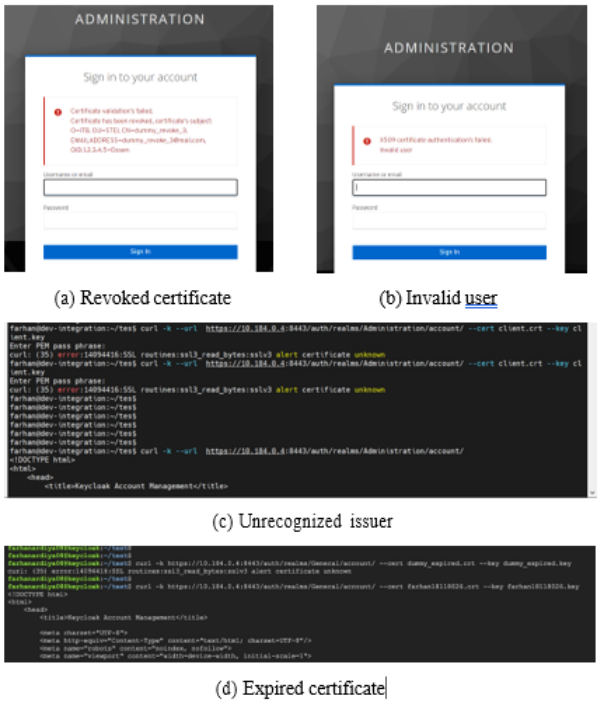

Figure 12. Invalid certificate errors

Figure 12 displays errors when trying to login using invalid certificates. The SSO can recognize whether a certificate is invalid because it has been revoked, thus giving a specific error. If the user is not in the SSO realm, but everything elsechecks out, the SSO gives an invalid user error. Meanwhile if the certificate is not issued by the trusted CA or it has expired, the errors will be similar (SSL errors).

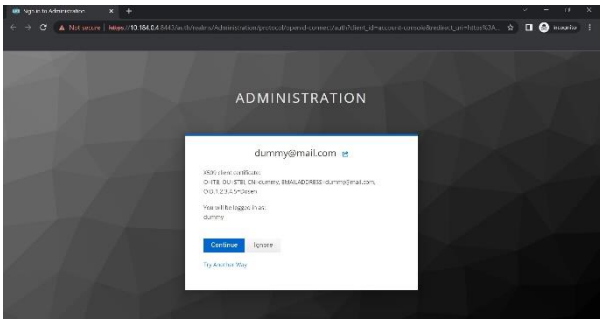

Figure 13. Valid certificate response

Figure 13 displays the response when logging in with a valid certificate. The page confirms that the user wants to continue as the presented username. It also shows the certificate information. When clicked continue, the user will be authenticated within the SSO realm.

3.3. RBAC Automation Subsystem

The RBAC automation uses a database as a reference of how the access should be provided. Then, a script can be run to check whether there is a remediation needed and take action accordingly. The script is capable of creating, deleting, and assigning groups for user access.

RBAC works by restricting access levels based on each user's role. Access levels, of course, govern what users can and cannot do. Admin is usually the highest access level. Admin users

have full control over the network or service at hand. But you can create other access levels with fewer privileges. The principle behind RBAC is that users should have access levels based on their roles.

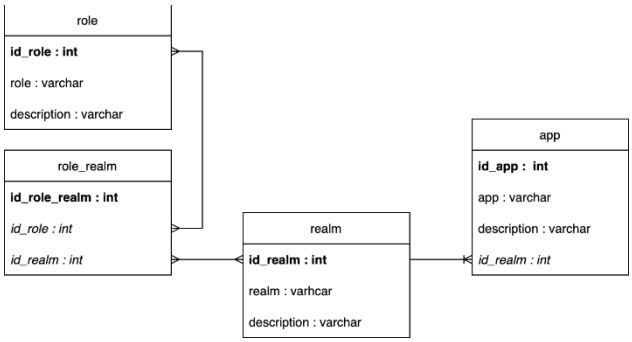

Figure 14. RBAC database entity relationship diagram

Figure 14 is an entity relationship diagram of the database used for RBAC management. There are four tables used, which are role, realm, app, and role_realm. The role table contains information about all of the roles in a system. The realm table contains information about all of the realms used in a system. The role_realm table is a junction table that stores the access mapping of each role. The app table is used to store information about apps and which realm it sits within the SSO.

The RBAC automation subsystem is a python script that will compare and remediate access. It is configured to be runperiodically so that the SSO is in sync with the desiredconfiguration. The subsystem consists of a main script, a toolscript, a configuration script, and a configuration file. The main script is the one that will be run, it executes the main functions defined in the tool script. Meanwhile, the configuration script reads the configuration file so that other files can call configuration variables easily.

```
1.  #__main__.py
2.  from tools import *
3.
4.  def main():
5.    ejbcaUsers = getUserEJBCA()
6.    keycloakUsers = getUserKeycloak()
7.
8.    currentAccess =
    getCurrentAccess(ejbcaUsers, keycloakUsers)
9.
10.   accessRole = getDBAdminMapping()
11.
12.   checkGroups()
13.
14.   add_access, delete_access =
    compareAccess(currentAccess, accessRole)
15.
16.   createUser(add_access)
17.   deleteUser(delete_access)
18.
19. if__name   == '__main__':
20.   main()
21.
```

The main script works by gathering data from EJBCA database and Keycloak API, then comparing it with the data in the RBAC database. Before comparing access, the script checks for groups within all of the realms and remediate access by adding or removing groups within realms. The compare Access function returns two outputs, which is list of access that needs to be added and a list of access that needs to be deleted. The actuator of access remediation is creating User and delete User function.

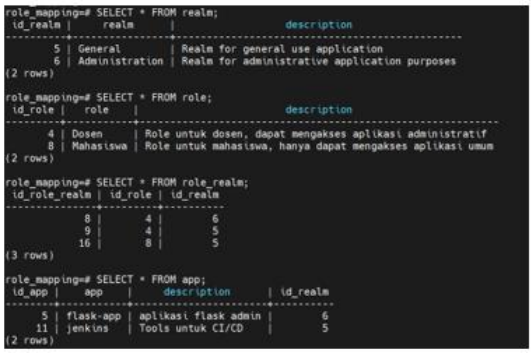

Figure 15. RBAC database configuration

Figure 15 illustrates the desired configuration for RBAC. The database will act as a reference for the automation script. Thescript will try to remediate access to reach the desired state configured.

Table 1. Access Before Remediation

Realm	Group	User
Administration	Alumni	-
	Dosen	**dummy_revoke_3**
		farhan18118026
		test_rbac_2
General	Dosen	**dummy_revoke_3**
		test_rbac_2
	Mahasiswa	**18118002**
		dummy_expired
		farisansafrialnafis
		test_rbac

Table 1 describes the state of access before the script is executed. As can be seen that there is a group that does not exist within the configuration. The script will be expected to remediate that group.

Table 2. Desired Access Configuration

User	Role	Desired Realm Access
18118002	Mahasiswa	**General**
dummy	Dosen	**General and Administration**
dummy_expired	Mahasiswa	**General**
dummy_revoke_3	Dosen	**General and Administration**
farhan18118026	Mahasiswa	**General**
farisansafrialnafis	Mahasiswa	**General**
test_rbac	Mahasiswa	**General**

test_rbac_2	Dosen	**General and Administration**
18118002	Mahasiswa	**General**
dummy	**Dosen**	**General and Administration**

Table 2 describes the access that is supposed to be implemented. There are two users that require access remediation which is farhan18118026 and dummy. The userfarhan18118026 is placed at the wrong realm and the user dummy has not been added to both realms.

Table 3. Access After Remediation

Realm	Group	User
Administration	Mahasiswa	-
	Dosen	dummy_revoke_3
		dummy
		test_rbac_2
General	Dosen	dummy_revoke_3
		test_rbac_2
		dummy
	Mahasiswa	18118002
		dummy_expired
		farisansafrialnafis
		test_rbac
		farhan18118026

Table 3 describes the state of access after the script is executed. The access configuration is the same as the desired access configuration described in Table II. There are three main remediations, which are the group change from Alumni to Mahasiswa, the user farhan18118026 is moved from Administration realm to General realm, and the user dummy is added to both realms.

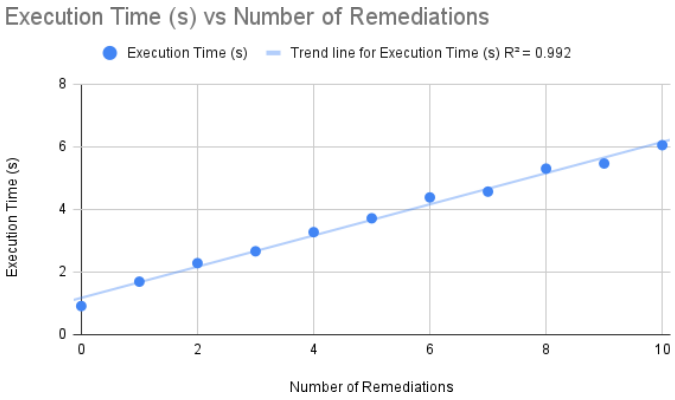

Figure 16. Automation execution time

Figure 16 illustrates the execution time needed against the number of remediations. The value of R square disapproaching 1, thus the linear trend line is a valid fit. The average increment of execution time is 0.5136 seconds. The increment of each data point is similar, since the script makesan API call to

Keycloak for each remediation action taken. With the performance of the script mentioned, it is wise for the script to be executed with a schedule that fits the behavior of access granting within an entity. The more frequent the access change, then the script needs to be performed more frequently.

Based on the research that has been conducted by Chatterjee et al (2022) showed that the SSK implementation and configuration in the eCoach prototype system has effectively secured its microservice APIs from an attack in all the considered scenarios with 100% accuracy [15].

4. Conclusion

The PKI integrated SSO with RBAC automation system has been successfully implemented using EJBCA as PKI management, Keycloak as SSO software, and python as the script for RBAC automation. Main functionalities of the system include certificate registration, web based SSO login, and access granting automation. Registration subsystem is implemented using the RA Web functionality of EJBCA. Meanwhile the SSO login uses mTLS authentication method which validates user's digital certificate. The RBAC automation works by comparing the desired state stored in a database with the current state by querying information through EJBCA database and Keycloak API.

The benefits to using RBAC are the first, improved operational efficiency, with RBAC, companies can decrease the need for paperwork and password changes when they hire new employees or switch the roles of existing employees. Second, Enhanced compliance, companies generally prefer to implement RBAC systems to meet the regulatory and statutory requirements for confidentiality and privacy, as executives and IT departments can more effectively manage how the data is accessed and used. Third, increased visibility, RBAC gives network administrators and managers more visibility and oversight into the business. Fourth, reduced costs and the fifth, decreased risk of breaches and data leakage.

References

[1] A. Susanto and R. B. Bahaweres, "Preliminary research on e-government development overview: An assessment on e-Government capabilities in Indonesia," in *2013 International Conference of Information and Communication Technology (ICoICT)*, Mar. 2013, pp. 444–447, doi: 10.1109/ICoICT.2013.6574617.

[2] M. of C. and I. Technology, *The Strategic Plan of the Ministry of Communications*. 2020.

[3] T. Kalvet, "Innovation: a factor explaining e-government success in Estonia," *Electron. Gov. an Int. J.*, vol. 9, no. 2, p. 142, 2012, doi: 10.1504/EG.2012.046266.

[4] A. Kalja, J. Pold, T. Robal, and U. Vallner, "Modernization of the e-government in Estonia," 2011.

[5] M.-J. Sule, M. Zennaro, and G. Thomas, "Cybersecurity through the lens of Digital Identity and Data Protection: Issues and Trends," *Technol. Soc.*, vol. 67, p. 101734, Nov. 2021, doi: 10.1016/j.techsoc.2021.101734.

[6] T. Bazaz and A. Khalique, "A Review on Single Sign-on Enabling Technologies and Protocols," vol. 151, no. 11, 2016.

[7] U. M, I. S, and A.-N. A, "A dynamic access control model using authorising workflow and task-role-based access control," in *IEEE Access*, 2019, pp. 166676–166689.

[8] Yuping Deng, Xiaowei Guo, and Xiamu Niu, "A New Design Scheme of Role-Based Access Control Based on PKI," in *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, 2006, vol. 3, pp. 669–672, doi: 10.1109/ICICIC.2006.387.

[9] W. Xia et al., "Illuminate the Shadow: A Comprehensive Study of TLS Client Certificate Ecosystem in the Wild," in *28th International Conference on Telecommunications (ICT)*, 2021, pp. 1–5.

[10] A. Suganthy and V. P. Venkatesan, "An Introspective Study on Dynamic Role-Centric RBAC

Models," in *IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019, pp. 1–6.
[11] Zhengxian Gao, Zhongxue Li, and Yaqing Tu, "Design and completion of digital certificate with authorization based on PKI," in *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, 2004, pp. 462–466, doi: 10.1109/IRI.2004.1431504.
[12] L. Hongxin, G. Keqing, and W. Yugang, "The Application of RBAC Model in E-Commerce System," in *International Conference on Electrical and Control Engineering*, 2010, pp. 3059–3062.
[13] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*, 2nd ed. Boston: Addison-Wesley, 2003.
[14] Keycloak, "Single-Sign On," *https://www.keycloak.org/*, 2022. https://www.keycloak.org/.
[15] A. Chatterjee and A. Prinz, "Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study," *Sensors*, vol. 22, no. 5, p. 1703, Feb. 2022, doi: 10.3390/s22051703.