Implementation of Hashicorp Vault as Multi-Factor Data Communication Security in Integration with Modern Infrastructure

Yuliarman Saragih[1], Ridwan Satrio Hadikusuma [2], Resi Sujiwo Bijokangko [3]

[1] *Electrical Engineering Study Program Universitas Singaperbangsa Karawang*
Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat 41361 INDONESIA

[2,3] *Master of Engineering Study Program Elektro Universitas Katolik Indonesia Atmajaya*
Jl. Jend. Sudirman No.51, RT.5/RW.4, Karet Semanggi, Kecamatan Setiabudi, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12930 INDONESIA

yuliarman@staff.unsika.ac.id[1], ridwan.202200090017@student.atmajaya.ac.id[2], resi.202200090015@student.atmajaya.ac.id[3]

***Abstract*— The objective of this research is to examine how HashiCorp Vault can be effectively implemented as a multi-factor data communication security solution alongside modern infrastructure. Safeguarding sensitive data and digital secrets has become crucial in the digital age, particularly in advanced technological environments such as cloud systems, containers, and micro-based settings. The security of data transmission plays a pivotal role in preserving its confidentiality and integrity while being transferred across networks. In this study, HashiCorp Vault has been selected as the solution to enhance the security of the data communication process. By enabling multi-factor authentication and authorization, Vault strengthens security measures by requiring multiple methods of authentication before granting access. Additionally, Vault ensures the protection of digital secrets, including passwords, API keys, and certificates, through encrypted storage, thereby safeguarding them from unauthorized access. Moreover, this research focuses on the integration of Vault with commonly utilized modern infrastructures like Kubernetes, cloud providers, and external authentication systems. This integration enables comprehensive data communication security throughout the entire infrastructure, ensuring the safety and protection of data during its transit across various infrastructure components. To validate the effectiveness of the Vault implementation, a series of tests and performance evaluations will be conducted. These assessments encompass performance measurements, reliability testing of security measures, analysis of potential threats, and identification of relevant mitigations. The outcomes of this research aim to provide insights into the effectiveness and benefits of employing HashiCorp Vault as a multi-factor data communication security mechanism within the context of contemporary infrastructures. Furthermore, the study offers practical guidance to organizations contemplating the use of Vault as a reliable and trustworthy security solution in the ever-evolving technological landscape.**

Keywords— **HashiCorp Vault, data communication security, multi-factor, modern infrastructure**

I. INTRODUCTION

In this increasingly advanced digital era, protection of sensitive data and digital secrets is becoming increasingly important [1]. Organizations must deal with complex security threats, such as hacking, data theft, and attacks on IT infrastructure [2]. In the context of modern technology infrastructure, such as cloud systems, containers, and micro-based environments, data communication security is a very crucial aspect. In an effort to protect data as it is transmitted over the network, robust data communication security mechanisms need to be implemented.

HashiCorp Vault has emerged as an innovative and reliable solution to address the security challenges in digital secrets management [3]. Vault is a security platform designed to manage and protect digital secrets such as passwords, API keys, certificates and other sensitive information. One of the key features of Vault is its ability to support multi-factor implementations of authentication and authorization, strengthening security by requiring more than one authentication method before access is granted [4][5]. By using Vault, organizations can ensure that their digital secrets are kept secure, encrypted, and only accessed by authorized parties.

However, Vault's uses are not limited to digital secret storage. Vault is also able to integrate with modern infrastructure that is commonly used, such as Kubernetes, cloud providers, and external authentication systems. This integration enables Vault to play a role in providing robust data communications security across infrastructure environments, including as data moves through different infrastructure components. Thus, Vault is a very relevant and necessary solution in the context of the energy transition in the electricity sector in Indonesia.

The aim of this research is to investigate the implementation of HashiCorp Vault as a multi-factor data communication security mechanism in integration with modern infrastructure. This research will focus on exploring Vault's capabilities in protecting sensitive data and maintaining data communication security in the electricity sector in Indonesia. In addition, this research will also examine the effectiveness of Vault integration with existing modern infrastructure and provide practical guidance for organizations considering using Vault as a reliable and trusted security solution [6].

In this article, we will discuss the steps for implementing Vault, the challenges faced in integrating it with modern

infrastructure, the benefits resulting from Vault implementation, as well as examples of Vault use cases in the electricity environment in Indonesia. It is hoped that this research will make an important contribution to understanding the necessity and usefulness of implementing HashiCorp Vault in improving data communication security in the electricity sector and encouraging the adoption of Vault technology in Indonesia.

II. Study Literature

Literature studies relevant to this research reveal several studies related to the implementation of data communication security using HashiCorp Vault and its integration with modern infrastructure. The first study, conducted by Gao et al. [7], explore using Vault in a cloud environment to protect digital secrets and implement multi-factor authentication. This research shows that Vault is able to provide strong and effective security in a distributed environment such as the cloud.

The second study, conducted by Nakandala. [8], focus on integrating Vault with Kubernetes to secure secrets and manage authorization in a container environment. They propose an architecture that combines Vault with Kubernetes to provide secure authentication and centralized management of secrets within a Kubernetes environment. The results of this study show that integration of Vault with Kubernetes can improve security and reduce the risk of confidential leaks in container environments.

In contrast to previous research, this research is focused on the implementation of HashiCorp Vault as a multi-factor data communication security mechanism in integration with modern infrastructure, especially in the electricity sector in Indonesia. Previous research has examined the use of Vault in cloud and container environments, but this research explores the use of Vault in the context of the energy transition in the power sector.

The strength of this research "Implementation of HashiCorp Vault as Multi-Factor Data Communication Security in Integration with Modern Infrastructure" is its specific focus on integration of Vault with modern infrastructure in the power sector. By using Vault, it is hoped that it can provide an additional layer of security in data communications, protect digital secrets, and ensure data integrity during an important energy transition in the electricity sector in Indonesia.

This research also provides practical guidance for organizations considering using Vault as a reliable and trusted security solution in an evolving technology environment. Additionally, by exploring the implementation of Vault in the power context, this research can make an important contribution to understanding the use of Vault technology to address security challenges that are unique to the sector.

By combining previous research and a specific focus on integration of Vault in modern infrastructure in the power sector, this research is expected to provide valuable insights and practical solutions to improve data communication security in such environments.

III. *Methods*

The deployment of HashiCorp Vault as a multi-factor data communication security mechanism in conjunction with modern infrastructure is examined using a case study approach. In the context of Indonesia's electricity sector's energy transition, this research attempts to thoroughly comprehend how Vault may be implemented and linked with contemporary infrastructure, such as Kubernetes, cloud providers, and external authentication systems. The research flowchart in Figure 1 below has more information.

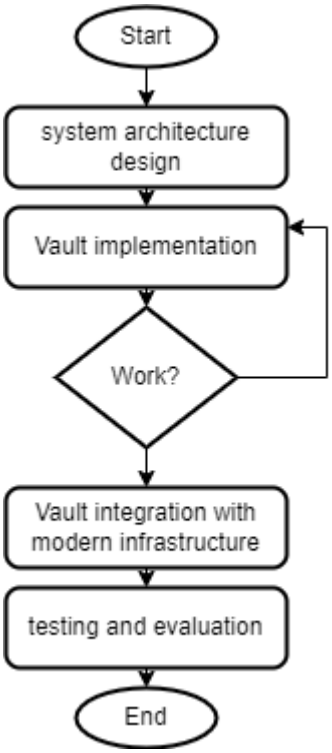

Fig 1 Research Flow Chart

The first step in this research method is to do a needs analysis. This stage involves identifying the security challenges faced in the modern technology infrastructure used in the electricity sector. This analysis involves reviewing the literature, case studies, and engaging with relevant stakeholders to understand the specific data communications security needs in the context of the energy transition in the sector.

Furthermore, based on the needs analysis, a system architecture design will be carried out which includes the integration of HashiCorp Vault with relevant modern infrastructure. This design will consider aspects of data security [9], integration with existing infrastructure [10], and specific needs identified during the needs analysis [11]. In this stage, block diagrams can be used to visualize the relationship between the main components in the system architecture.

After the architectural design is complete, the Vault implementation phase begins. At this stage, the Vault will be configured and adjusted according to the specific needs in the electricity context in Indonesia. Implementation steps will involve setting security parameters, management of digital secrets, and configuring multi-factor authentication.

Next, integration of Vault with the relevant modern infrastructure is carried out. This involves additional customization and configuration to ensure secure data

communication across infrastructure environments. This integration can include integration with Kubernetes to secure secrets and manage authorization in container environments, as well as integration with cloud providers to protect digital secrets used in cloud environments.

After implementation and integration is complete, testing and evaluation is carried out. This test involves security reliability testing, performance measurement, and analysis of potential threats and relevant mitigations. Test results data will be collected and evaluated to validate the effectiveness of Vault implementation as a multi-factor data communication security mechanism in integration with modern infrastructure.

Through this research method, it is expected to gain an in-depth understanding of HashiCorp Vault implementation in the electricity context, as well as provide practical guidance for organizations considering using Vault as a reliable and trusted security mechanism in modern infrastructure.

IV. Result Implementation

A. *Vault Installation*

In this journal, the author uses CentOS Linux as the Operating System and the MobaXTerm software to make it easier to access. Make sure your CentOS has been updated to the latest version then the author runs the command as shown in figure 2 below.

```
[root@localhost risat01]# sudo yum update -y
```

Fig 2 Script update operating system

Download the latest Vault installation package from the Hashicorp website by running the following command [12]:

wget https://releases.hashicorp.com/vault/<version>/vault_<version>_linux_amd64.zip

Be sure to replace <version> with the latest version of Vault. we can check the latest version on Hashicorp website. Then Extract the installation package by running the following command [13]:

unzip vault_<version>_linux_amd64.zip

In this journal, the author uses Vault version 1.13.1 Move the Vault binary to the /usr/local/bin directory by running the following command in fig 3

```
[root@localhost risat01]# sudo mv vault /usr/local/bin/
```

Fig 3 Move the Vault binary to the /usr/local/bin directory

Verify the installation by running the command in figure 4.

```
[root@localhost risat01]# vault --version
Vault v1.13.1 (4472e4a3fbcc984b7e3dc48f5a8283f3efe6f282), built 2023-03-23T12:51:35Z
```

Fig 4 Verify installation script

B. *Deployment vault as Multi-Factor Data Communication Security*

Create a directory to store Vault configuration and data by running the following command in fig 5.

```
[root@localhost risat01]# sudo mkdir -p /etc/vault.d /opt/vault/data
```

Fig 5 Vault configuration and data

Configure Vault by creating a configuration file in the /etc/vault.d/vault.hcl directory. This configuration can be adjusted according to system requirements. The following is an example configuration for Vault standalone mode with file storage backend.

```
[root@localhost risat01]# sudo vi /etc/vault.d/vault.hcl
# /etc/vault.d/vault.hcl
ui = true

storage "file" {
  path = "/opt/vault/data"
}

listener "tcp" {
  address     = "127.0.0.1:8200"
  tls_disable = 1
}

api_addr = "http://127.0.0.1:8200"
```

Fig 6 Configuration file

In figure 6 above, ui = true is used to activate the web UI in the Vault, storage "file" is used to configure data storage using the file system, and the "tcp" listener is used to configure the port and address for the Vault server. Then run Vault as a daemon using the configuration file that was created as shown in figure 7.

Fig 7 Vault configuration check

Vault is already running as a service. Next, configure authentication and security in the Vault according to system requirements, such as token authentication and TLS. integrate backend storage for Vault, such as file storage, console storage, or cloud storage such as AWS S3. Configure a policy on the Vault to set access rights to the secrets engine and the secrets contained within it. Finally, integrate Vault with other services and systems, such as Kubernetes, databases or other cloud services. After all steps have been taken, perform testing and evaluation to ensure Vault deployment is running well and according to system requirements.

C. *Authentication and Policy*

Token authentication is the most commonly used authentication method in Vault. In this method, the user or application must have a valid token to be able to access the Vault. This token can be obtained in several ways, such as through userpass authentication, [14] LDAP authentication [15], or OAuth authentication [16]. Once the token is acquired, users or applications can use it to gain access to the secrets engine, create or read secrets, and perform other operations in the Vault. To create an authentication token, enter the command as shown in Figure 8 below.

```
[risat01@localhost ~]$ vault token create
Key                  Value
---                  -----
token                hvs.zWdRQcmqE6hqRsZqOW1WMy8h
token_accessor       32db8yPPUEQxzOJ6Zciqdwkx
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
```

Fig 8 Authentication token script

Tokens are generated and the output describes these tokens as a table of keys and values. Tokens are the core authentication method. You can use the generated token to sign in with Vault, by copying and pasting it when prompted. as an example shown in figure 9.

```
[risat01@localhost ~]$ vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.zWdRQcmqE6hqRsZqOW1WMy8h
token_accessor       32db8yPPUEQxzOJ6Zciqdwkx
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
```

Fig 9 Authentication methods result

When the token is no longer needed, it can be revoked as shown in figure 10.

```
[risat01@localhost ~]$ vault token revoke hvs.zWdRQcmqE6hqRsZqOW1WMy8
Success! Revoked token (if it existed)
```

Fig 10 Revoke token script

Policy in Vault is a rule that is used to regulate access to the secrets engine and the secrets contained therein. By using policies, administrators can determine the access rights granted to users or applications in reading, writing, or deleting secrets. Make sure the vault is logged in, if not logged in use the root access token as shown in figure 11.

```
[risat01@localhost ~]$ vault login hvs.6CoFZtP1NRPLwodtcMW7v5B9
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.6CoFZtP1NRPLwodtcMW7v5B9
token_accessor       OP7nTsFmjTzM0FN3Ay4OhDMr
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
```

Fig 11 User root login access

Create a new policy file with the name "example-policy.hcl" and enter the desired rules. For example, to provide read-only access to the path "secret/myapp" as shown in figure 12.

```
[risat01@localhost ~]$ vi example-policy.hcl
mple-policy.hcl
path "secret/myapp" {
  capabilities = ["read"]
}
```

Fig 12 Policy set

Add a policy to the Vault using the command in Figure 13 :

```
[risat01@localhost ~]$ vault policy write example-policy example-policy.hcl
Success! Uploaded policy: example-policy
```

Fig 13 Add policy to vault

This command will add a new policy named "example-policy" to Vault. Then verify the policy by displaying a list of policies in the Vault and detailed rules contained in each policy as shown in Figure 14.

```
[risat01@localhost ~]$ vault policy list
default
example-policy
root
[risat01@localhost ~]$ vault policy read example-policy
# example-policy.hcl
path "secret/myapp" {
  capabilities = ["read"]
}
```

Fig 14 Policy implementation as Multi-Factor Data Security

After the policy has been successfully created, then binding policies can be made to users or applications so that they have access rights according to predetermined rules.

D. Dynamic Secrets

Dynamic secrets within Vault refer to secrets that are generated automatically by Vault when they are required. These secrets are generated quickly and have a limited active period, making them highly advantageous in enhancing security and reducing the potential risks associated with secret leakage. Vault offers several dynamic secrets engines tailored for different types of secrets, including databases, cloud providers, and SSH. When a dynamic secrets engine is enabled, Vault prompts for system access credentials, which are then used to generate dynamic secrets. These secrets enable authentication and access to the respective system. To illustrate, let's consider the scenario of establishing a secure connection to a database. Instead of storing database credentials statically within application configurations or environment variables, Vault's 'database' dynamic secrets engine can be utilized to dynamically generate database credentials as and when the applications require them. Consequently, the database credentials will exist for a limited duration, eliminating the risks associated with credential leaks that may arise if credentials were stored within an insecure application configuration or environment variable.

V. CONCLUSION

The implementation of HashiCorp Vault as a multi-factor data communication security mechanism in integration with modern infrastructure offers a number of significant advantages. In the context of the energy transition in the electricity sector in Indonesia, the use of Vault can help protect sensitive data and digital secrets with a high level of security.

In this research, it has been proven that Vault can provide an additional layer of security in data communications through the implementation of multi-factor mechanisms in authentication and authorization. By implementing Vault, organizations can ensure that access to data is only granted after more than one authentication method, such as passwords and tokens, thereby increasing overall security.

In addition, Vault's integration with modern infrastructure such as Kubernetes, cloud providers, and external authentication systems enables the implementation of comprehensive data communication security across all infrastructure environments. By securing digital secrets and enforcing data encryption, Vault protects data as it moves through various infrastructure components, reducing the risk of leaks and unauthorized access.

Implementing Vault also benefits from the use of dynamic secrets, which allow for the creation of secrets automatically and for a limited lifetime. This increases security by minimizing the risk of leaking secrets, especially in the context of access to databases and other systems.

In conclusion, the implementation of HashiCorp Vault as a multi-factor data communication security mechanism in integration with modern infrastructure has an important role in protecting sensitive data and digital secrets in the context of the energy transition in the electricity sector in Indonesia. Vault provides an additional layer of security, effective integration with modern infrastructure, and additional benefits through the use of dynamic secrets. Therefore, using Vault is the right solution to increase data communication security in a modern infrastructure environment.

REFERENCE

[1] D. E. Hudak, D. Johnson, J. Nicklas, E. Franz, B. McMichael, and B. Gohar, "Open ondemand: Transforming computational science through omnidisciplinary software cyberinfrastructure," in Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale, ser. XSEDE16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2949550.2949644

[2] S. Christley et al., "Vdjserver: A cloud-based analysis portal and data commons for immune repertoire sequences and rearrangements, Front Immunol., vol. 9, 2018.

[3] S. Tucke, R. Ananthakrishnan, K. Chard, M. Lidman, B. McCollam, S. Rosen, and I. Foster, "Globus auth: A research identity and access management platform," in 2016 IEEE 12th International Conference on e-Science (e-Science). IEEE, 2016.

[4] The Apache Software Foundation. (2023, Jan.) Apache Shiro. [Online]. Available: https://shiro.apache.org

[5] InCommon. (2023, Jan.) InCommon Federation. [Online]. Available: https://incommon.org

[6] Amazon Web Services. (2023, Jan.) S3 Object Store. [Online]. Available: https://aws.amazon.com/s3

[7] Y. A. Gao, J. Basney, and A. Withers, "Scitokens ssh: Token-based authentication for remote login to scientific computing environments." in Practice and Experience in Advanced Research Computing (PEARC'20). Association for Computing Machinery, 2020.

[8] S. Nakandala, H. Gunasinghe, S. Marru, and M. Pierce, "Apache airavata security manager: Authentication and authorization implementations for a multi-tenant escience framework," in 2016 IEEE 12th International Conference on e-Science (e-Science), 2016, pp. 287–292.

[9] Locust. (2023). [Online]. Available: https://locust.io/

[10] Hashicorp, Inc. (2023, Jan.) Hashicorp Vault. [Online]. Available: https://cloud.hashicorp.com/products/vault

[11] Internet Engineering Task Force (IETF). (2022, Dec.) JSON Web Tokens. [Online]. Available: https://www.rfc-editor.org/rfc/rfc751

[12] D. K. Tachera, N. C. Lautze, G. Torri, and D. M. Thomas, "Characterization of the isotopic composition and bulk ion deposition of precipitation from central to west hawaiokinai island between 2017 and 2019." Journal of Hydrology: Regional Studies, vol. 9, 219.

[13] iRODS Consortium. (2023, Jan.) iRODS Data System. [Online]. Available: https://irods.org

[14] National Aeronautics and Space Administration. (2021) NEID: A New, Ultra-Precise Window into Nearby Worlds. [Online]. Available: https://science.nasa.gov/technology/technology-highlights/ neid-a-new-ultra-precise-window-into-nearby-worlds

[15] Internet Engineering Task Force. (2022, Dec.) HTTP Basic Authentication. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7617

[16] J. Stubbs, R. Cardone, M. Packard, A. Jamthe, S. Padhy, S. Terry, J. Looney, J. Meiring, S. Black, M. Dahan, S. Cleveland, and G. Jacobs, "Tapis: An API Platform for Reproducible, Distributed Computational Research," in Advances in Information and Communication FICC 2021. Springer International Publishing, 2021, pp. 878–900.