An Evolutionary Risk-based Access Control Framework for Enterprise File Systems

Shi-Cho Cha
Dept. of Information Management
National Taiwan University of Science and Technology
Taipei, R.O.C.
csc@cs.ntust.edu.tw

Yi-Hsuan Hsuan
Dept. of Information Management
National Taiwan University of Science and Technology
Taipei, R.O.C.
M11009137@mail.ntust.edu.tw

Kuo-Hui Yeh
Dept. of Information Management
National Dong Hwa University, Hualien, R.O.C.
Taipei, R.O.C.
khyen@gms.ndhu.edu.tw

Takeshi Ishihara
Kioxia Corporation
Kanagawa, Japan
takeshi3.ishihara@kioxia.com

Ohba Yoshihiro
Kioxia Corporation
Kanagawa, Japan
yoshihiro.ohba@kioxia.com

Wei-Nin Chen
Department of Information Management
National Taiwan University of Science and Technology
Taipei, R.O.C.
D10109301@mail.ntust.edu.tw

2022 IEEE 8th World Forum on Internet of Things (WF-IoT) | 978-1-6654-9153-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/WF-IOT54382.2022.10152041

***Abstract*—To enhance access control mechanisms, organizations need to monitor access requests issued from devices. Therefore, organizations can evaluate the trustworthiness or risks of the devices based on collected requests to adapt the access privileges. However, existing schemes usually do not address organizational authorization processes and may not be suitable for enterprise file systems. In light of this, this study proposes an Evolutionary Risk Adaptive Access Control (ERAAC) Framework for enterprise file systems. The proposed framework provides an extensible architecture for an organization to deploy different access control filters for different perspectives. An access control filter can filter out access requests based on access control policies. An organization can add a new access control filter without replacing its existing access control mechanism. In addition, the proposed framework enables organizations to define new risk labels for data entities, such as subjects and objects to be accessed, used in access control policies. The access control mechanism can adapt user privileges based on the risk labels. Even if organizations do not have enough data to generate risk labels, the organizations can set access control policies without risk labels. Therefore, the proposed framework enables organizations to progressively improve their access control mechanisms. To the best of our knowledge, the proposed framework is the first access control framework that can evolve with organizational maturity in risk management. This study also illustrates how the proposed framework satisfied the related tenets mentioned in NIST SP 800-207. Consequently, this study can hopefully contribute to helping an organization to implement zero trust architecture.**

***Index Terms*—access control, ZTA, security risk; risk-adaptive, access control**

I. Introduction

As the trends of work from home, people may work out of the scope of an organization. As users may not have appropriate security countermeasures at home, malicious people can easily hack into user devices. Then, malicious people can utilize the devices to attack resources in the organization as the device owners bring the devices to the organization and connect to the internal network of the organization. In this case, the effectiveness of traditional enterprise perimeter protection mechanisms (such as firewalls) is doubtful.

To prevent malicious people from utilizing devices of authorized users to move laterally in organizational network, the concept of zero trust architecture has been brought to the spotlight. Simply speaking, the zero trust architecture requests organizations to protect organizational resources even if user devices in the organizations are breached [1][2]. Organizations should not implicitly trust a requester and skip the authentication or access control processes. In addition, organizations should continuously monitor organizational requesters and obtain context of the requesters to evaluate the their risks. Therefore, organizations can utilize the risk information to determine user privileges. However, it is a challenge to consider risks in access control policies, especially for enterprise file systems.

First, traditional access control models usually do not address the process of privilege decisions. For enterprise file systems, files (and folders) usually have their owners. The file owners determine who can access the files. However, the file owners may not have enough knowledge to embed risk consideration in the authorization process. Second, risks come from different sources, such as the user's devices, the system where the resources are located, the network environment, etc. It is not an easy task to integrate the risk values from different perspectives. For example, if the risk of a user device is high and the confidential level of a document to be accessed is middle, we may have trouble in determining the overall risk for the user to access the document via the device. Third, organizations usually need to deploy security controls to collect information for risk evaluation. For instance, organizations may need to install agents on user devices to collect status of devices. However, organizations may not afford security controls for risk assessment.

For the very sake of that, this study proposes an Evolution-

ary Risk Adaptive Access Control (ERAAC) Framework for enterprise file systems. First, the framework allows an organization to deploy a set of access control filters for different access control considerations. If an organization does not have enough data to evaluate risk factors used in an access control filter, the organization can disable the filter until it collects necessary data. Second, the framework specifies different risk labeling systems to tag risks of subjects, objects, and other data entities that can be used in access control policies. In this case, this study defines an extensible model to specify access control policies based on the labels. Furhtermore, organizations can deploy new risk labeling systems for new risk considerations.

With the proposed framework, organizations can delegate different access control filters for different users to administrate access control policies from different perspectives. For example, an organization may deploy an access control filter for file owners to specify who can access their files and deploy another access control filter for administrators to present devices that have not updated its virus signatures from accessing a file. To the best of our knowledge, the proposed framework is the first access control framework that can evolve with organizational maturity in risk management. Therefore, organizations can improve their access control mechanisms progressively. This study also illustrates how the proposed framework satisfied the related tenets mentioned in US NIST SP 800-207. Consequently, this study can hopefully contribute to helping an organization to implement zero trust architecture.

The rest of this paper is organized as follows: Section II introduces preliminary information on access control models and zero trust architecture. Section III provides an overview of the proposed framework. Next, Section IV describe the key components of the proposed framework. Section V illustrates how the proposed framework satisfies the tenets of zero trust architecture. Conclusions are finally drawn in Section VI along with recommendations for future research.

II. Risk-based Access Control Models

Access control is critical in zero trust architecture. As described in US NIST SP 800-207 [1], every access request for organizational resources should be checked by a policy enforcement point (PEP). The PEP then enquiries the policy decision point (PDP) to decide whether a request can be allowed based on organizational access control policies. Currently, there are four major access control models for access control policy specification: the mandatory access control (MAC) model, the discretionary access control (DAC) model, the role-based access control (RBAC) model, and the attribute-based access control model (ABAC). This study focuses on how to use security risks to achieve adaptive access control, which is a key feature of zero trust architecture.

To consider security risks in access control, McGraw presented the concept of Risk-Adaptive Access Control (RAdAC) model in 2004 [3]. In the RAdAC model, organizations use a security risk determination function to calculate security levels based on characteristics of requesters, resources to be accessed, and other environmental or contextual factors. The organizations can then use the security levels in access control decision. However, McGraw does not mention how to generate security levels and how to use the security levels. After that, several researches have been proposed to address security risks in access control. In 2020, Atlam et al. have selected and summarized 44 papers about risk-based access control from 2007 to 2019 [4]. This study does not think of the RAdAC or risk-based access control as a new access control model. Instead, this study refers to the risk-based access control as a means of using security risks as factors in access control decision. A risk-based access control scheme usually needs to address the following three issues: (1) factors to estimate security risks; (2) schemes to estimate security risks; (3) methods to use security risks in access control.

Several factors can be used to assess security risks of access requests. In traditional MAC model, organizations can use sensitivity level of objects to be accessed to forbid unauthorized access. Early research usually utilizes static information of user profiles, IT components, and action consequences. With advances in information technology, organizations can further adopt dynamic information, such as user context and historical logs, for risk assessment. Current guidelines like the US NIST SP 800-207[1] further recommend organizations to consider external sources like cyber threat intelligence.

Among the above factors, organizations may regard factors like resource sensitivity and action severity as security risks directly. Organizations can further use the factors to estimate the potential loss of confidentiality, integrity, and availability for each access request [5]. Then, organizations can estimate a risk value of an access request based on weighted average of expected loss of related breaches. Weighted average would not be suitable for qualitative factors like sensitivity level. Therefore, organizations may use the fuzzy logic system for risk estimation [6]. Moreover, organizations can use machine learning techniques to assess risks based on historical data. For example, Molly et al. developed a SVM (Support Vector Machine)-based classifier for access decision and referred uncertainty of risks to adjust decision.

In risk-based access control, access decision mechanisms utilize estimated risk values to decide to accept or deny an access request. Organizations can treat risk values as attributes and establish a set of criteria for the attributes. An access request is allowed associated criteria are met. From this perspective, we may implement the risk-based access control mechanism with ABAC simply. To consider security risks in the RBAC model, organizations can assign risks on role assignment relationships and calculate the overall risk for a user to have a specific permission on resources [7]. In the research of Bijon et al. [8], organizations can continuously calculate risks of permissions and roles associated with the permissions. A user cannot activate a role if the risk after activating the role is higher than a threshold.

This study adopts traditional access control models in hybrid mode. In the proposed framework, each file has its owner. Therefore, a file owner can follow the DAC to set access control policies on his/her files. The access control policies

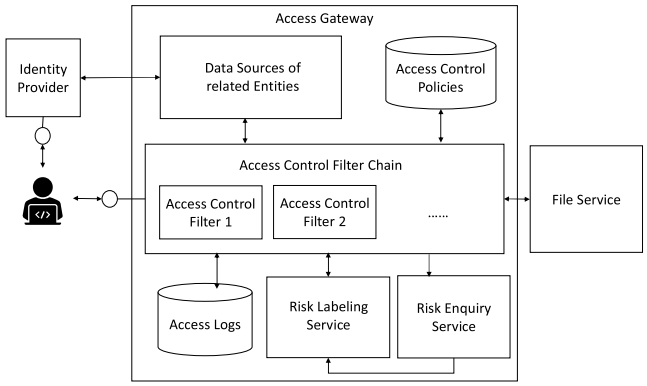

Fig. 1. Overview of the Proposed Framework

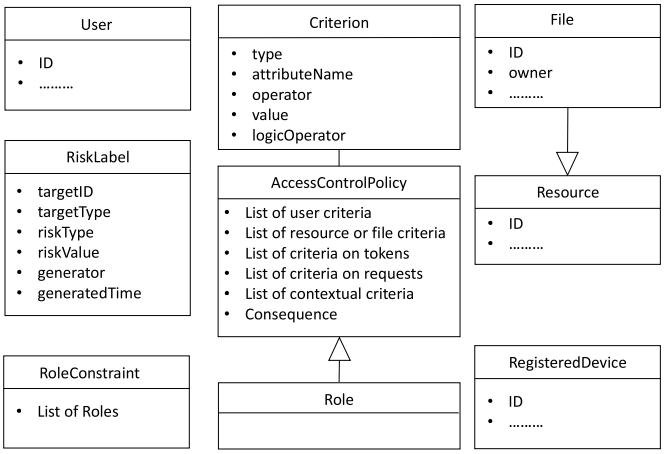

Fig. 2. Major Data Types of the Proposed Framework

follow the ABAC model. File owners and administrators can set rules for access control decision based on related attributes. Moreover, this study allows administrators to define roles and assign role membership based on attributes of data entities, such as subjects, objects, environmental context, etc., to support the RBAC model. Compared to existing risk-based access control research, this study uses a tagging system. The risk values are represented as tags to associated data objects. The tags can further be mapped to attributes of data objects for access control. This study illustrates the details in Section IV.

III. Overview of the Proposed Framework

Fig. 1 provides an overview of the proposed ERAAC framework. Before a user wishes to access a file, the user first authenticates himrself/herself with an *Identity Provider* to initiate a session. After verifying user identity, the identity provider issues an access token for the session. The token includes a unique identifier and is signed by the identity provider. Then, the user sends his/her request to the *Access Gateway* with the token.

The access gateway plays the role of the policy enforcement point (PEP) and the policy decision point (PDP) mentioned in the US NIST SP 800-207. The kernel of the access gateway is the *Access Control Filter Chain*. The ERAAC framework defines the standard interface of *Access Control Filters* and provides a base implementation based on the interface. Organizations can adjust the setting of access control filters for different purposes. Therefore, the administrator of an access gateway can specify a set of access control filters and arrange their orders to form a chain.

For each access control filter, privileged users can set *access control policies* of the file service. Similar to the Policy element defined in XACML, an access control policy includes a set of attribute criteria based on the following information:

- *Attributes in access requests*. An access request usually includes files to be accessed, requested actions on the files, and other client side information embedded in the request.
- *Attributes in access tokens*. In addition to identifiers, identity providers may attach user and contextual information in access tokens. Therefore, an access control filter may utilize the information directly rather than request the data.
- *Attributes of data source entities maintained by the framework*. The ERAAC framework maintains data sources of users, devices, resources, and other data source entities for access control. An access control filter can extract related data based on access requests and tokens from the data sources for access control decision.
- *Risk labels of data source entities*. The *Risk Labeling Service* includes a set of *Risk Labelers*. A risk labeler evaluates security risks of users, files, user devices, and other entities. The evaluated risks are represented as risk labels, which include risk types, risk values, and evaluation time. A risk labeler then "attaches" the risk labels to data source entities.

Upon receiving an access request, an access control filter extracts associated access control policies. Then, the access control filter collects required data based on the criteria in the policies. Note that the access control filter may find that risk labelers have not generated required risk labels. Also, the access control filter may just obtain out-of-dated risk labels. In this case, an access control filter may ask the *Risk Enquiry Service* to request the associated risk labeler to generate related risk labels immediately. Therefore, the risk enquiry service can notify the access control filter to fetch the generated result. With the collected data, an access control filter can evaluate whether or not to accept a request based on the criteria in related access control policies. If an access control filter decides to accept the request, the filter forwards the request to the next access control filter in the access control filter chain. If the access control filer is the last filter in the chain, it sends the request to the file service. Finally, the access gateway logs the request process into the *Access Logs* storage. The logs can further be used by related risk labelers to produce risk labels.

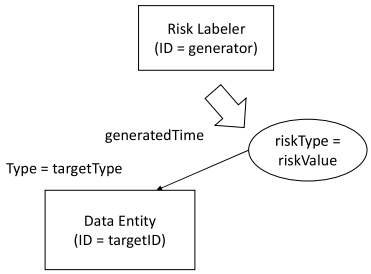

Fig. 3. Concept of a Risk Label

IV. Key Components

A. Data Models

The proposed framework provides data source entities for access control. This study assumes that each organization has maintained databases of users, devices allowed to access the system, and resources to be accessed. In our system, we view these entities as a data source for risk analysis. Therefore, the proposed framework can access databases, map the databases into a unified format, and enable access control filters to access the data through standard interfaces. As depicted in Fig. 2, current ERAAC framework focuses on the following four types of data source entities: (1) users; (2) registered devices; (3) resources; (4) files (and folders). Each data source entity has a unique id and other attributes. A file can be viewed as a subclass of resource. A file includes an owner attribute to identify who can modify access control policies of the file.

As depicted in Fig. 3, risks of data source entities are represented as risk labels. A risk label includes the source risk labeler that generates the label, the generation time, identity and type of the target data source entity, the risk value, and the type of the risk. Therefore, a risk labeler can generate different types of risk labels of a data source entity simultaneously.

An access control policy is composed of a set of criteria and the consequence to accept or deny the request. The type attribute in a criterion represents the type of the criterion. Currently, there are five types of criteria: (1)user criteria; (2) resource and file criteria; (3) token criteria; (4) request criteria; (5) contextual criteria. First, the criteria on users, resources, and files are used to specify suitable data source entities. For example, user criteria describe which users are applicable to the related access control policy. In addition, the proposed framework enables administrators to set criteria on attributes of access requests and the associated tokens. Note that because people can use the request criteria to specify applicable actions, such as reading, writing, deleting, and other operations, on resources, this study does not specify the action criteria in an access control policy. Finally, the contextual criteria enable administrators to define requirements on access time, device used, and other contextual factors. A criterion uses the attributes of attributeName, operator, and value to specify the range of attributeName. This study uses the ‘dot’ operator to denote the sub-attribute of an attribute. For example, if the attribute ‘name’ has a sub-attribute ‘lastname’, the lastname is represented as ‘name.lastname’. Moreover, while an access control policy may include more than one criterion, the logicOperator attribute in a criterion uses the values of ‘and’, ‘or’, and ‘not’ to describe how to combine related criteria.

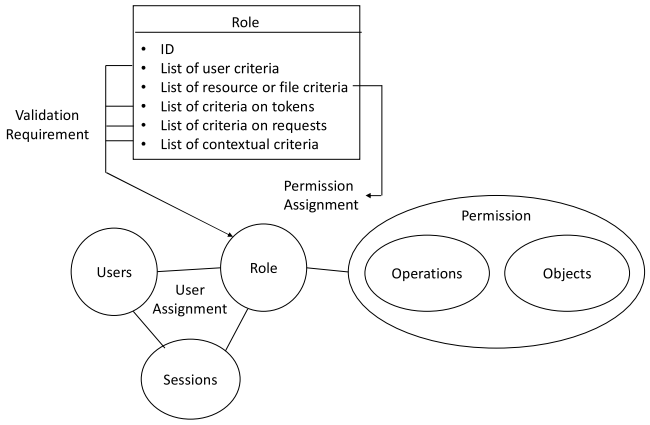

Fig. 4. Concepts of Mapping the Role Class in Fig. 2 to RBAC

This study enables administrators to specify risk labels in criteria to achieve risk-based access control: Administrators can use the prefix of ‘risk.’ in attributeName to represent that the criteria are related to risk requirement. Then, administrators can indicate the risktype behind the ‘risk.’. For example, if the value ‘risk.sensitivityLevel’ is shown in the attributeName attribute in a criterion for resources, the criterion focuses on the risk labels with targetType = ‘resources’ and riskType=‘sensitivityLevel’. Moreover, administrators can append the postfix of ‘.generatedTime’ to put restriction on the risk label generation time.

Finally, to simplify the complexity of access control policy administration and to enforce separation of duties, this study enables us to define roles and place constraints on roles. Fig. 4 illustrates how to use the role class depicted in Fig. 2 to implement RBAC. The role class or data source entities can be viewed as a sub-class of the AccessControlPolicy class. Administrators specify permission assignment with the resource criteria. Other criteria are used to set validation requirements on roles. This study does not deal with user assignment in the role class. Instead, the proposed framework uses the attribute role in user class to store the roles assigned to a user. For the static separation of duties (SSoD) scenario, this study assumes that the user data management mechanism checks that a user cannot have conflicted roles. For the dynamic separation of duties (DSoD) scenario, if a user have conflicted roles, the user can only active one of the conflict roles when the user authenticates with the identity provider. Then, the identity provider only stores the activated role in the access token.

B. Access Control Filters

The proposed framework enables organizations to deploy a set of access control filters to form a filter chain. The proposed framework defines the standard interfaces and pro-

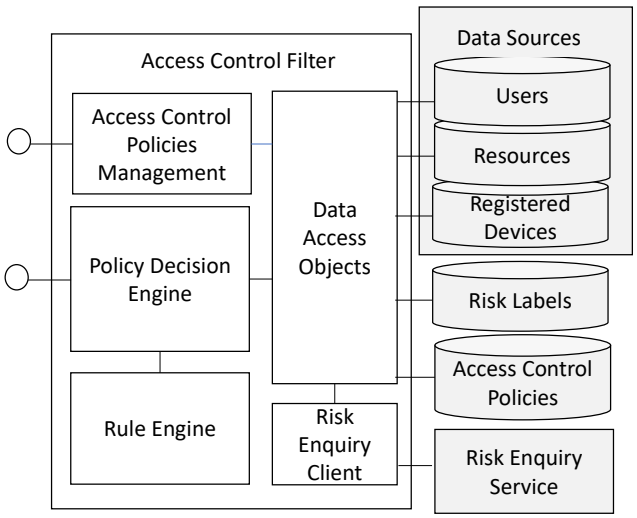

Fig. 5. Architecture of an Access Control Filter

vides fundamental implementation of an access control filter. Organizations can use the fundamental access control filter to deploy different access control filters with different settings for different purposes. Moreover, organizations can follow the defined interfaces and extend the fundamental access control filter based on their needs. Fig. 5 illustrates the architecture of a fundamental access control filter. This study uses blocks in gray scale to represent the external components.

First, each access control filter has its access control policies. An access control filter provides for administrators to manage access control policies. Note that this study only defines the API for policy management currently. Organizations can implement their own GUIs. The fundamental access control filter adopts the Data Access Objects (DAO) pattern. External data are mapped into data access objects. Therefore, access control filters can deal with the heterogeneous data in a unified way.

The *Policy Decision Engine* is the kernel of an access control filter. Upon receiving an access request, the policy decision engine extracts associated access control based on the access request. For each matched access control policy, the policy decision engine extracts data (or facts) to evaluate the access control policy. Recall that the proposed framework uses risk labelers to represent risks of data source entities with risk labels. If the policy decision engine finds that it does not obtain the up-to-dated risk labels, it requests the *Risk Enquiry Client* to trigger the risk label generation process. The risk enquiry client then collect the generated risk labels and provides the labels for further evaluation. The policy decision engine then forwards the access control policy and associated data to a *Rule Engine* to obtain access control decision of the policy. If there is more than one matched access control policy, the policy decision engine collects the decision of each access control policy and combines the decisions for final decision. Although the XACML specification defines several rule combining algorithms, this study adopts the "deny override" algorithm. That is, the policy decision engine declines an access request if any matched access control policy declines the request. In other words, an access request is allowed if every match access control policies allow the request.

V. Framework Evaluation

The US NIST publishes the NIST SP 800-207 [1]. The document summarizes the following tenets of zero trust architecture:

1) Identify all resources that could be accessed.
2) Secure all communications, even in organizational internal networks.
3) Re-authenticate users and evaluate their trustworthiness every-time the users request to access organizational resources.
4) Include behavioral and environmental attributes to determine user privileges dynamically.
5) Continuous monitoring the security posture of organizational resources and mitigating observed vulnerabilities.
6) Strengthen the authentication and access control mechanisms to enforce access control policies.
7) Collect information to improve security of existing architecture.

This study describes how the proposed framework helps an organization to satisfy the tenets of zero trust. The first tenet requires organizations to identify all resources. The proposed framework can prevent unauthorized access to resources. However, organizations still need asset management systems or other tools to discover their resources.

The second tenet requests organizations to transfer data on secure channels. The proposed framework can support the tenet easily by enforcing secure communication between the components of the proposed framework.

The proposed framework requests users to authenticate themselves with the identity providers to obtain access tokens and attach the tokens in their requests. Each token is only valid in specific time intervals. If a token becomes invalid, a user needs to re-authenticate himself/herself with the identity provider to obtain a new token. Moreover, as users should send every request to the access gateway, the access control filters in the access gateway can evaluate trustworthiness or risks of the requests and decide whether or not to accept the requests. Therefore, the proposed framework fulfills the third tenet.

The fourth tenet requests organizations to adopt dynamic access control policies that consider the state of users and resources.The proposed framework includes contextual criteria in access control policies. Besides, the proposed framework enables risk labelers to interact with external systems to update the state of users and resources with risk labels. For example, an organization can deploy a risk labeler to receive warnings of the Endpoint Detection and Response (EDR) system and tag users using devices with unpatched vulnerabilities. Therefore, the proposed framework satisfy the fourth tenet.

As the proposed framework focuses on access control schemes, the proposed framework does not include tools to

monitor and measure the integrity and security posture of organizational resources. As described in the previous paragraph, the proposed framework can collaborate with external systems to collect security states of organizational resources for access control. Therefore, although the proposed framework does not achieve the fifth tenet directly, organizations can integrate the proposed framework with other security monitoring tools to realize the fifth tenet of zero trust.

In the proposed framework, every access request should be processed via an access gateway. The access gateway verifies each access request to ensure that the requester of the request is authenticated by a trustworthy identity provider. The access gateway then sends the request to the access control filter chain to decide whether or not to forward the request to its target. As organizations can integrate their services with the access gateway to filter out unauthorized requests, the proposed framework can help organizations to achieve the sixth tenet.

Finally, the proposed framework logs details of access request processing. In addition to using the logs to evaluate risks, the logs can further be used to improve the effectiveness of risk evaluation algorithms. Moreover, organizations can deploy new access control filters as their capability become more mature. Therefore, the proposed framework satisfies the seventh tenet.

VI. Conclusion and future work

To address the issues that an organization may not have enough data to consider security risks in access control, this study has proposed the Evolutionary Risk Adaptive Access Control (ERAAC) Framework for enterprise file systems. The proposed framework maintains data source entities about users, resources, files, and other basic factors for access control. Organizations can define access control policies with the basic data source entities. In addition, the proposed framework defines a scheme to annotate a data source entity with risk labels. Therefore, organizations can deploy risk labeling systems to evaluate risks of data source entities and add risk labels on the data source entities. In this case, the proposed framework provides a means to use risk labels in access control policies. Therefore, even if organizations do not have enough resources to deploy risk labeling systems, the organizations can just implement their access control mechanisms with basic data source entities. In addition, an organization can only use the risk labels generated by available risk labeling systems to set access control policies.

Moreover, the proposed framework enables an organization to deploy different access control filters for different perspectives. The proposed framework defines the standard interfaces and provides fundamental implementation of an access control filter. Organizations can use the fundamental access control filter to deploy different access control filters with different settings for different purposes. Moreover, organizations can follow the defined interfaces and extend the fundamental access control filter based on their needs. For enterprise file systems, files (and folders) usually have their owners. The file owners then determine who can access the files. However, the file owners may not have enough knowledge to embed risk consideration in the authorization process. Therefore, with the proposed framework, organizations can deploy two different access control filters for different purposes. Further, an organization can add a new access control filter without an existing access control mechanisms. Therefore, the proposed framework enables organizations to improve their access control mechanisms progressively.

This study also illustrates how an organization uses the proposed framework to achieve the tenets mentioned in US NIST SP 800-207. Therefore, this study can hopefully contribute to helping an organization to implement zero trust architecture.

References

[1] S. W. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," 2020, NIST Special Publication 800-207. [Online]. Available: https://www.nist.gov/publications/zero-trust-architecture

[2] J. R. Biden, "Executive order on improving the nation's cybersecurity," May 2021. [Online]. Available: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

[3] R. W. McGraw, "Securing content in the department of defense's global information grid," in *Secure Knowledge Management Workshop*, 2004.

[4] H. F. Atlam, M. A. Azad, M. O. Alassafi, A. A. Alshdadi, and A. Alenezi, "Risk-based access control model: A systematic literature review," *Future Internet*, vol. 12, p. 103, 2020.

[5] N. N. Diep, S. Lee, Y.-K. Lee, and H. Lee, "Contextual risk-based access control," in *Security and Management*, 2007.

[6] H. F. Atlam and G. B. Wills, "An efficient security risk estimation technique for risk-based access control model for iot," *Internet of Things*, vol. 6, p. 100052, 2019.

[7] L. Chen and J. Crampton, "Risk-aware role-based access control," in *Security and Trust Management*, C. Meadows and C. Fernandez-Gago, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 140–156.

[8] K. Z. Bijon, R. Krishnan, and R. Sandhu, "A framework for risk-aware role based access control," in *2013 IEEE Conference on Communications and Network Security (CNS)*, 2013, pp. 462–469.