



Article

Endpoint Device Risk-Scoring Algorithm Proposal for Zero Trust

Ui Hyun Park [†], Jeong-hyeop Hong [†], Auk Kim ^{*} and Kyung Ho Son ^{*}

Department of Convergence Security, Kangwon National University, Chuncheon-si 24341, Republic of Korea

^{*} Correspondence: kimauk@kangwon.ac.kr (A.K.); khson@kangwon.ac.kr (K.H.S.)[†] These authors contributed equally to this work.

Abstract: The rapid expansion of remote work following the COVID-19 pandemic has necessitated the development of more robust and secure endpoint device security solutions. Companies have begun to adopt the zero trust security concept as an alternative to traditional network boundary security measures, which requires that every device and user be considered untrustworthy until proven otherwise. Despite the potential benefits of implementing zero trust, the stringent security measures can inadvertently lead to low availability by denying access to legitimate users or limiting their ability to access necessary resources. To address this challenge, we propose a risk-scoring algorithm that balances confidentiality and availability by evaluating the user's impact on resources. Our contributions include (1) summarizing the limitations of existing risk scoring systems in companies that implement zero trust, (2) proposing a dynamic importance metric that measures the importance of resources accessible to users within zero trust systems, and (3) introducing a risk-scoring algorithm that employs the dynamic importance metric to enhance both security and availability in zero trust environments. By incorporating the dynamic importance metric, our proposed algorithm provides a more accurate representation of risk, leading to better security decisions and improved resource availability for legitimate users. This proposal aims to help organizations achieve a more balanced approach to endpoint device security, addressing the unique challenges posed by the increasing prevalence of remote work.

Keywords: zero trust; scoring system; risk score; security

Citation: Park, U.H.; Hong, J.-h.; Kim, A.; Son, K.H. Endpoint Device Risk-Scoring Algorithm Proposal for Zero Trust. *Electronics* **2023**, *12*, 1906. <https://doi.org/10.3390/electronics12081906>

Academic Editor: Aryya Gangopadhyay

Received: 8 February 2023

Revised: 11 April 2023

Accepted: 14 April 2023

Published: 18 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Remote work has expanded in many companies following the COVID-19 pandemic. Traditionally, the network boundary security concept has been used to maintain security for remote work. The network boundary security concept separates internal and external networks and restricts access at the network boundary. In this concept, remote access to internal networks is assumed to be access that can be trusted. However, as remote work increases, the limitations of boundary security are emerging owing to vulnerabilities in remote solutions and the large variety of access devices [1–3]. Before the expansion of remote work, access to company networks was possible on a limited number of devices that had previously been provided and inspected by the companies. However, as remote work expands, employees can use personal devices to access company data from outside company networks. The conventional boundary security concept has limitations in that it is difficult to control access from vulnerable or already infected devices when internal networks are accessed. As the limitations of boundary security became apparent, there was a need for new security methods to compensate for these limitations and allow for secure remote work. Consequently, the zero trust concept was proposed as a new security paradigm that improves upon boundary security. Zero trust is a security concept in which no activity is trusted, and verification must be performed constantly [4–6]. Contrary to the conventional boundary security approach, when zero trust is implemented, risk scoring

is performed to verify all devices each time a resource access request is made, and access authorization is determined according to the results of the scoring [5,7,8].

The advantage of zero trust is that it ensures the security and safety of company resources by not trusting and constantly performing risk scoring according to the same criteria when access requests arrive [9,10]. However, when risk scoring is performed according to the same criteria for all users, resources become more secure, but there is a disadvantage in that the resources may become less available. Availability refers to the degree to which a permitted user can access requested resources promptly. Decreased availability implies that it takes a considerable time or it is difficult to access resources when a user makes a request for access. For example, consider a shopping mall that has implemented zero trust where access requests are made by employees who are allowed to read customer information as well as interns who have low-level permissions. In this case, if risk scoring is performed according to the same high-level criteria that are used for employees with high-level permissions, interns with low-level permissions will only be allowed access after going through high-level security procedures. As such, it may be difficult to follow the security procedures, and accessing resources may become inconvenient, thus decreasing availability. Conversely, if risk scoring is performed according to the same low-level criteria, it can decrease confidentiality and even lead to security incidents if the standard for employee devices that can access customers' personal information is lowered and important resources are accessed. Consequently, a risk-scoring algorithm that inspects all users with the same criteria is inefficient in terms of system security and availability.

To increase the confidentiality and availability of systems that implement zero trust, in this study, we propose (1) a "dynamic importance metric" for evaluating importance according to the user's role and resource access permissions and (2) a risk-scoring algorithm based on the Common Configuration Score System (CCSS) base metric, which dynamically changes its security demands according to importance as evaluated by the dynamic importance metric. CCSS is a standard framework that assigns scores to the security vulnerabilities of equipment in IT systems and checks them.

The contributions of this study are as follows.

I. Summarize and describe the limitations of the risk-scoring system that is used in companies that currently implement zero trust.

II. Propose a dynamic importance metric that can measure the importance of resources that are accessible to users within systems that implement zero trust.

III. Propose a risk-scoring algorithm that uses the dynamic importance metric to efficiently increase security and availability for users who are accessing resources in systems that implement zero trust.

The remainder of this paper is organized as follows. Section 2 describes the necessity of risk scoring in systems that implement zero trust as well as trends in risk scoring. In addition, it describes CCSS, which is a framework that evaluates vulnerabilities in devices' system settings. Section 3 presents the proposed dynamic importance metric for evaluating importance according to users' roles and resource access permissions and the zero trust-based risk-scoring approach for user devices that employs the dynamic importance metric. Section 4 presents conclusions and future research directions.

2. Related Work

2.1. Trust Algorithm in Zero Trust

Zero trust is a security concept in which requests for access to resources are not trusted but continuously verified. Figure 1 shows the zero trust access. The core of zero trust comprise the (1) policy decision point (PDP) and (2) policy enforcement point (PEP). The PDP determines the policy set by the company regarding whether resource access is allowed, and the PEP allows or disallows all access according to the policy. In particular, users who request access to resources in a system that implements zero trust are allowed to access the resources after receiving verification via the PDP and PEP [5,7,9].

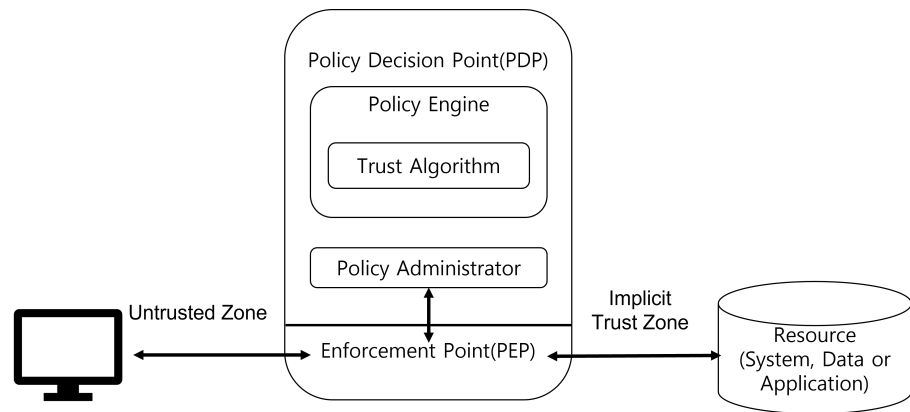


Figure 1. Zero Trust Access.

The PDP is similar to the brain of zero trust. Within the PDP, the trust algorithm is used to ultimately determine whether users’ resource access requests will be granted according to the policies set by the company. The trust algorithm is a planned process that is used to determine whether access to a resource is allowed. Figure 2 shows the trust algorithm considers the following sources: (1) access request, (2) subject database and history, (3) asset database, (4) resource policy requirements, and (5) threat intelligence and logs [5,11]. Figure 3 below illustrates the relationship between the trust algorithm and risk scoring in a zero trust.

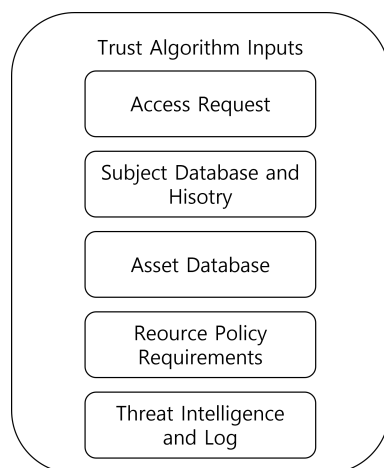


Figure 2. Trust Algorithm in ZeroTrust.

In the case of access requests, which are one of the trust algorithm’s sources, user information is verified in regard to requests to access resources. During verification, risk scoring is performed on the user device. Risk scoring refers to assigning scores regarding whether the user’s device has an adequate level of security. Each time a user requests access to a resource, the user is subjected to risk scoring [5,11,12]. In a company that implements zero trust, the scoring criteria for risk scoring are established by the trust algorithm’s policies, and access is only allowed if the user device’s risk score meets a certain standard. For example, if there is an e-commerce business, such as a mall that has implemented zero trust within the company, devices are inspected using the same risk scoring standard for all users to provide access to resources. If an employee named Alice accesses a company resource to perform remote work, she is subjected to a device inspection according to the risk-scoring algorithm. After inspection, access is allowed if it is judged that the risk score of Alice’s device is safe according to the policies established by the mall.

When creating a trust algorithm’s policies, it is necessary to consider the balance between security and availability [5]. For example, if a policy is created in which resources

can only be accessed when all users receive a good risk score, users who intend to access the resource may be inconvenienced because access is limited to a high-level security configuration. Such a policy increases resource security but reduces availability. In contrast, if a policy is created in which all users who request access to a resource are allowed despite receiving poor risk scores, availability is increased but security is reduced because vulnerable devices can access the resource [13].

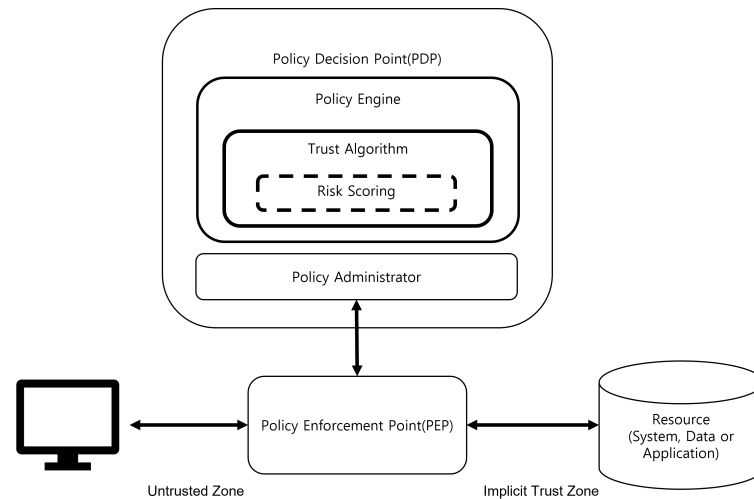


Figure 3. Risk Scoring in Zero Trust.

2.2. Device Risk Scoring Trends and Common Configuration Scoring System

The trust algorithm determines whether access is allowed by comparing the access request in the input with asset database information when examining the device's security level. At this time, it is necessary to perform risk scoring on the device to make an efficient determination. Systems that use zero trust should continuously protect systems and users through risk scoring and allow appropriate access. Most companies implement zero trust perform risk scoring, which can examine the security levels of device devices. To perform risk scoring, it is necessary to analyze security configurations and establish objective evaluation standards. To establish objective evaluation standards, one may use a risk-scoring standard framework, such as the CCSS, which evaluates vulnerabilities in device system settings. This section first describes risk-scoring trends in companies that have implemented zero trust. Next, it introduces CCSS, which is a typical risk scoring standard framework.

2.2.1. Corporate Trends

When Microsoft establishes zero trust within a company, the client's devices and IDs are registered and managed in Microsoft Intune [14,15]. Microsoft Intune is a Microsoft management tool that provides corporate and bring-your-own-device (BYOD) integrated endpoint device management. Here, Microsoft Secure Score performs risk scoring on devices that request access by linking with Intune's policies and Microsoft Defender, which is a product that manages device health checks. Microsoft Secure Score represents the device's security state as a number based on the system configuration, user actions, and other security-related measurements. Based on the results, resource access is determined via threat level classifications [16].

The Secure Score that is provided by Microsoft makes the inspection results visible and provides a variety of inspection items. It is used to effectively inspect the security levels of device devices in a system that implements zero trust and to suggest recommended security actions [17,18]. However, Microsoft's Secure Score does not perform scoring based on importance according to the user's role or resource access permissions. For example,

the default value of the perfect score (e.g., 9 points) can be set for “Ensure all users can complete multi-factor authentication for secure access”, which is one of the Secure Score inspection items; however, dynamic inspections are not performed for each user. Therefore, all devices in an organization are inspected according to the same standard.

Cisco is another company that implements zero trust and performs vulnerability scoring. Contrary to Microsoft, which scores user devices, Cisco creates scores for the vulnerabilities of its own products. Cisco has its own scoring system for examining systems to implement zero trust. However, like Microsoft, Cisco does not provide scoring based on importance according to the users’ roles and access permissions; instead, it performs scoring with standardized scores [19].

As such, the companies that are leading the way in introducing zero trust perform risk scoring on endpoints using their own solutions and determine whether a resource can be accessed based on the results. In addition, intensive research and solution development are being conducted on how to perform risk scoring on user devices.

Those companies provide Secure Score services that can examine their own services and products. However, these companies score all user devices according to the same standard. They are disadvantageous because they do not use a scoring algorithm that considers the roles and resources access permissions of users in the system.

2.2.2. Common Configuration Scoring System

To use device risk scoring in their security systems, companies can use their own algorithms or consider the use of a standard scoring framework for device security configurations, such as the CCSS. CCSS is a risk-scoring framework through which analysts can use three types of metrics (base, temporal, and environmental) to evaluate the vulnerabilities of IT system security configurations [20]. The base metrics measure vulnerabilities in devices’ basic configurations. Temporal metrics measure the expected rate of reduction in threats due to the attack cycle and supplementary measures. Environmental metrics measure values regarding the environment that is used to calculate the base metrics and temporal metrics. Here, the term “environment” refers to aspects such as the IT system’s facilities. When the CCSS framework is used, the base metrics must be used. The other metrics can be used optionally, but there are constraints on applying them to complex corporate-level system architectures. Therefore, this section only discusses the base metrics [20–22].

CCSS Base Metrics and Base Score

The base metrics are used to evaluate vulnerabilities based on the fundamental properties of security configuration vulnerabilities, and the base score uses this evaluation to show the vulnerabilities’ degree of severity. Figure 4 shows the base metrics of CCSS. In the base metrics, a vulnerability’s degree of severity considers the exploitability of the security configuration and impact. The metrics that indicate exploitability are access vector (AV), authentication (Au), and access complexity (AC), and the metrics that indicate impact are confidentiality impact (C), integrity impact (I), and availability impact (A). The exploitation method (EM) indicates whether the vulnerability exploitation method is active or passive, and it is not used directly in scoring but is used to ameliorate configuration vulnerabilities [20].

The access vector is a metric value that depends on the attacker’s location, and it can be network, adjacent network, or local. Authentication is the number of times an attacker needs to be authenticated before maliciously using the configuration, and it can be none, single, or multiple. Access complexity indicates the degree to which the attacker must make prior preparations, and it can be high, medium, or low. For example, consider a case in which an attacker performs an attack after acquiring permissions through a malicious email file. The Access complexity can be found based on the permissions that must be acquired beforehand. For administrator permissions, the access complexity is set to high; for user permissions; it is set to medium; otherwise, it is set to low. The confidentiality impact, integrity impact, and availability impact can be complete, partial, or none according

to the degree of impact on the system's confidentiality, integrity, and availability (CIA Triad). Here, the results of measuring each metric are referred to as vectors. Algorithm 1 shows the equation for calculating the CCSS base score [20]. The base score indicates the security configuration's vulnerability using the following base equation and the metric values of the base metrics. The values of the metrics are inserted in place of constants that are proportional to the vulnerabilities to find the values of the impact, exploitability, and $f(\text{Impact})$ variables. Ultimately, a base score from 0 to 10 is produced [20,23,24].

Algorithm 1 Base score equation

$$\text{BaseScore} = \text{round_to_1_decimal}(((0.6 \times \text{Impact}) + (0.4 \times \text{Exploitability}) - 1.5) \times f(\text{Impact}))$$

$$\text{Impact} = 10.41 \times (1 - (1 - \text{ConfImpact}) \times (1 - \text{IntegImpact}) \times (1 - \text{AvailImpact}))$$

$$\text{Exploitability} = 20 \times \text{AccessVector} \times \text{Authentication} \times \text{AccessComplexity}$$

$$f(\text{Impact}) = 0 \text{ if } \text{Impact} = 0, 1.176 \text{ otherwise}$$

$$\text{AccessVector} = \text{case AccessVector of}$$

- requires local access*: 0.395
- adjacent network accessible*: 0.646
- network accessible*: 1.0

$$\text{Authentication} = \text{case Authentication of}$$

- requires multiple instances of authentication*: 0.45
- requires single instance of authentication*: 0.56
- requires no authentication*: 0.704

$$\text{AccessComplexity} = \text{case AccessComplexity of}$$

- high*: 0.35
- medium*: 0.61
- low*: 0.71

$$\text{ConfImpact} = \text{case ConfidentialityImpact of}$$

- none*: 0.0
- partial*: 0.275
- complete*: 0.660

$$\text{IntegImpact} = \text{case IntegrityImpact of}$$

- none*: 0.0
- partial*: 0.275
- complete*: 0.660

$$\text{AccessComplexity} = \text{case AccessComplexity of}$$

- none*: 0.0
- partial*: 0.275
- complete*: 0.660

Below is the process for finding the CCSS base score for "automatic login is set up" on a PC. (1) AV is local because it can be physically accessed. (2) AU is none because there is no need for additional authentication. (3) AC is low because there is no need to do extra work to find the password because automatic login is set up. In this case, the confidentiality impact, integrity impact, and availability impact values can be assessed as partial. At this point, the values of the vectors and CCSS base score can be expressed as follows: "AV:L/AC:L/Au:N/C:P/I:P/A:I", Base Score: 4.6.

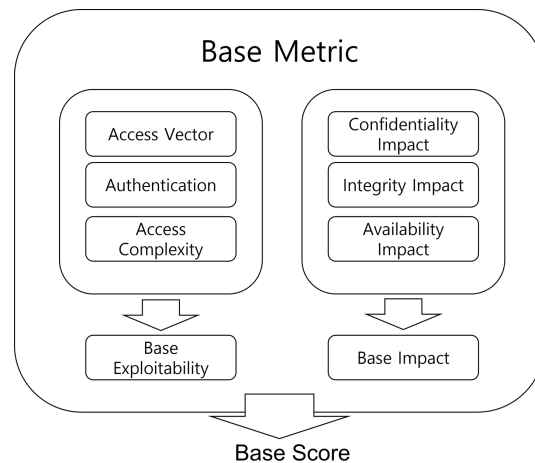


Figure 4. CCSS base metrics component.

3. Proposed User Device Risk Scoring for Zero Trust Access

In this section, we propose (1) a dynamic importance metric that measures the dynamic importance effect that occurs during security events according to the users' roles and resource access permissions and (2) a risk-scoring algorithm that uses the dynamic importance metric. (1) The dynamic importance metric measures the importance of the resources that can be accessed by the user in the system. (2) The risk-scoring algorithm that uses the dynamic importance metric can perform scoring using different criteria for each user. This risk-scoring algorithm can perform risk scoring using stricter criteria for users who have important permissions and less strict criteria for users without important permissions. Ultimately, it can efficiently increase confidentiality and availability for users who access resources in the system. The dynamic importance metric comprises two elements: accessible resource and resource importance. These two importance-based elements are applied as weights to the values of the evaluations of user device configurations when users access resources.

Next, we propose a risk-scoring algorithm that adds the dynamic importance metric. The risk-scoring algorithm with the additional dynamic importance metric is based on the CCSS base score and adds role-based access control (RBAC) concepts, such as user company position permissions and departments. A risk-scoring approach that reflects users' resource access levels can aid in efficient corporate zero trust implementations. In zero trust, the users and systems that request access to resources must pass through strict user authentication and device risk-scoring. In short, resources cannot be accessed if an entity cannot be trusted. Thereby, performing risk scoring on all users who attempt to access resources is an important factor for companies that implement zero trust. However, these companies perform risk scoring with the same criteria rather than using different criteria for each user even though there are various users with various permissions and goals. In addition, relevant studies have not been conducted. Therefore, for companies that want to implement zero trust, in this study, we propose a user device risk-scoring algorithm that also reflects the importance of the resources that can be accessed.

3.1. Dynamic Importance Metric

This section describes the dynamic importance metric, which can measure the dynamic importance effect that occurs during security incidents according to the users' roles and resource access permissions. Figure 5 shows the overview of the dynamic importance metric. As shown in the figure, the dynamic importance metric comprises (1) accessible resource (AR), which measures the permissions that the user needs to access resources, and (2) resource importance (RI), which measures the importance of the resources. The new dynamic importance metric is used in the proposed risk-scoring algorithm in addition to the CCSS base metrics.

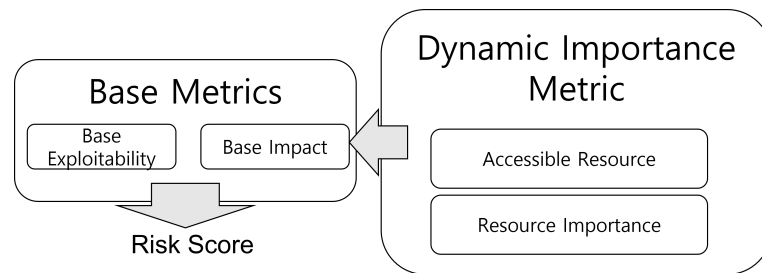


Figure 5. Overview of Dynamic Importance Metric with Base Metrics.

(1) Accessible resource shows the permissions needed to access resources. For example, there is a clear difference in the permissions needed by a company employee and a department manager to access company resources. Therefore, accessible resource refers to the extent of the company resource permissions that are granted by workers’ ranks, such as intern, manager, director, and president.

(2) Resource importance indicates the importance of company resources. For a resource that is subject to a user access request, resource importance indicates the effect that an attack on that resource would have on availability, confidentiality, and integrity. For example, suppose that an employee is a member of the human resources department. In this case, the employee can access human resources information. If the employee were a member of the security department, they could access the company security systems’ resources. Here, resource importance measures the effect of attacking each resource, and it evaluates the importance of the resources that can be accessed by the employees. Therefore, the dynamic importance metric, which comprises two elements, performs the role of assigning weight values to the confidentiality impact, integrity impact, and availability impact according to the extent of the permissions assigned to company resources according to the users’ ranks and departments.

Dynamic Importance Metric Vector Diagram

This section explains the dynamic importance value generation flow chart. In Figures 6 and 7, the dynamic importance metric vectors are added to the CCSS scores and used in the risk-scoring algorithm. Accessible resource and resource importance, which are elements of the dynamic importance metric, are evaluated according to the effect that user permissions have on CIA. For accessible resource, the metric’s value is set as none, partial, or complete by comprehensively considering whether the user can access confidential or important resources or has permission to add, modify, or delete resources.

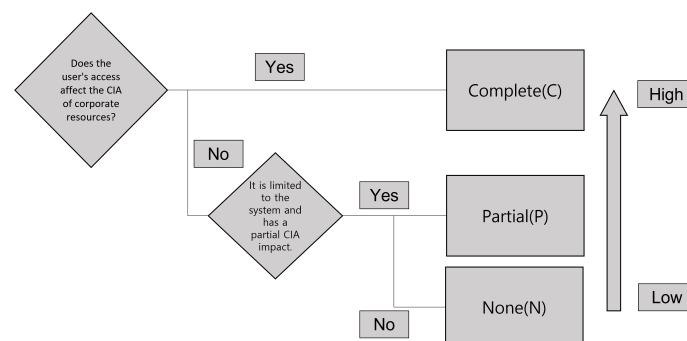


Figure 6. Accessible resource diagram.

One method for applying dynamic importance vectors to company employee device scoring is to generate vector values by differentiating users’ responsibilities and their position of office. For example, it is assumed that vulnerabilities in the security configurations of IT development team members will expose development source code or the structures of

enterprise databases and affect or harm most services. In such cases, the accessible resource and resource importance values can be considered complete.

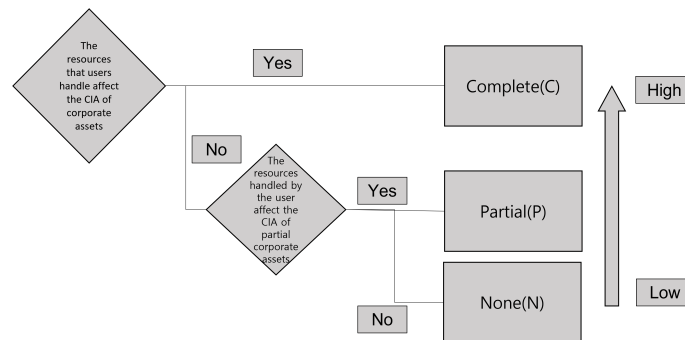


Figure 7. Resource importance diagram.

3.2. Device Risk-Scoring Algorithm for Zero Trust

In this section, we compare CCSS base scores with the risk-scoring algorithm and derive evaluation scores by applying an algorithm based on Algorithm 2 to several security items, and the result is in Table 1. In the proposed algorithm, constants that correspond to the dynamic importance metrics are applied as weight values to the base scores.

Algorithm 2 Risk scoring algorithm

$$RiskScore = round_to_1_decimal(((0.6 \times Impact) + (0.4 \times Exploitability) - 1.5) \times f(Impact))$$

$$Impact = (5.41 + DynamicImportance) \times (1 - (1 - ConfImpact) \times (1 - IntegImpact) \times (1 - AvailImpact))$$

$$DynamicImportance = AccessibleResource + ResourceImportance$$

AccessibleResource = case AccessibleResource of

- none: 0.0
- partial: 1.25
- complete: 2.5

ResourceImportance = case ResourceImportance of

- none: 0.0
 - partial: 1.25
 - complete: 2.5
-

The method for calculating scores using the risk-scoring algorithm is as follows. For example, suppose that two employees in the human resources department access human resources information resources. If risk scoring is performed for the aforementioned PC “automatic login is set up” risk-scoring item, the CCSS vector is “AV:L/AC:L/Au:N/C:P/I:P/A:I” and the base score is 4.6. By using the proposed dynamic-importance-metric-based risk-scoring algorithm, the following scores can be calculated for two company positions.

#1. In the case of an intern: The dynamic importance metric’s accessible resource (AR) is none because the user has few permissions for accessing resources. The resource importance (RI) can be considered complete because the resource is human resources information.

#2. In the case of a manager: The dynamic importance metric’s accessible resource (AR) is complete because the user has high-level permissions that can access resources and affect them by reading, writing, etc. The resource importance (RI) can be considered complete because the resource is human resources information.

Table 1. Example table of risk-scoring algorithm.

Item	CCSS Vector	CCSS Base Score	Risk Score (AR:C/RI:C)	Risk Score (AR:N/RI:N)
Auto login settings	AV:L/AU:N/AC:L /C:P/I:P/A:P/	4.6	4.6	2.5
USB AutoRun	AV:L/AU:N/AC:L /C:C/I:C/A:C/	7.2	7.2	3.8
Vulnerable password settings	AV:N/AU:N/AC:H /C:C/I:P/A:N/	6.1	6.1	4
Antivirus software installation	AV:N/AU:N/AC:L /C:C/I:C/A:C/	10	10	7.3

In the two examples, a value of “AR:N/RI:C/” is calculated for the intern, and a value of “AR:C/RI:C/” is calculated for the manager. Here, when the proposed risk scoring equation was used to calculate the dynamic importance metrics, the intern’s value was 3.3 points, and the manager’s value was 4.6 points. As a second example, the CCSS base score for “USB AutoRun” is calculated as follows. AV is local because the USB must be physically inserted. AU is none, and the AC is low because the USB automatically runs. In this case, there is an overall complete impact on the CIA of the user PC. Therefore, the CCSS vector is “AV:L/AU:N/AC:L/C:C/I:C/A:C”, and the base score is fairly high at 7.2 points. Here, the method for using the dynamic importance metric is explained using the two groups (1) AR:C/IR:C and (2) AR:N/RI:N as examples. AR evaluates the confidentiality, availability, and integrity of resources according to the permissions granted to the employee’s position in the company. It is divided into none, partial, and complete by considering access permissions for important documents. RI evaluates confidentiality, availability, and integrity according to the resources requested by the department, and it is divided into none, partial, and complete. An instance of a resource could be data that is public even to users with low-level permissions, or it could be confidential data that contains important internal company information. As such, various vectors from AR:C/RI:C to AR:N/RI:N could be used as the CCSS vector for “USB Autorun” according to the company resource.

#1 AR:C/IR:C: The user requests important confidential data within the company, and they have high-level resource permissions.

#2 AR:N/RI:N: The user requests data that is public to even users with low-level permissions, and the user has low-level resource permissions.

In case #1, a score of 7.2 points is found when the dynamic-importance-metric-based risk-scoring algorithm is applied because the vector is AR:C/IR:C. In case #2, a score of 3.8 points is found when the dynamic-importance-metric-based risk-scoring algorithm is applied because the vector is AR:N/RI:N. In the case of AR:C/IR:C, the CCSS base score is not affected by the dynamic importance metric, and a score of 7.2 points is produced. This is because in the case of AR:C/IR:C, important confidential data is accessed, and the user has high-level permissions; therefore, there is no need to relax the criteria for risk scoring. Conversely, in the case of AR:N/RI:N, a user with low-level permissions is requesting access to data that can easily be accessed by users with low-level permissions; therefore, relaxed criteria are applied, giving consideration to availability. The table below shows examples of applying the risk-scoring algorithm to typical security configurations for two groups, (1) AR:C/IR:C and (2) AR:N/RI:N.

4. Conclusions

As the number of companies implementing remote work is increasing, many employees are using personal devices to access company data from outside of company networks. Accordingly, there is a need for new security methods that can overcome the limitations of existing boundary security to allow for safe remote work. A new security paradigm

known as zero trust has emerged as a method to improve the existing boundary security approach, and many organizations are introducing zero trust solutions and services. In zero trust, inspections, such as risk scoring, which ensure security and identify vulnerabilities in devices that request access to an organization's resources, are important elements that must be performed each time a user requests a resource [25,26]. In this study, we proposed an algorithm that resolves the imbalance between security and availability that occurs when risk scoring is performed according to the same criteria for all users, as is currently performed at many companies.

We (1) analyzed trends in device risk scoring at companies that implement zero trust and have shown the limitations of risk scoring as it is currently performed, and (2) proposed a dynamic importance metric with weight values for scoring standards that vary according to resource importance as well as a risk-scoring algorithm that applies the dynamic importance metric. These tools make it possible to perform risk scoring, which changes dynamically according to the user's role and resource access permissions. In addition, we (3) described the use of case scenarios for the proposed algorithm in regard to several security configuration items.

If the proposed dynamic-importance-metric-based risk-scoring algorithm were to be introduced to systems that implement zero trust, users' resource access requests could be controlled efficiently by assigning weight values based on information regarding the users rather than solely using information regarding users' device security configurations when establishing policies on user resource access. This is expected to contribute to efficiently improving security and availability for users accessing resources in companies that implement zero trust. For future research directions, additional studies will be conducted on security items according to user impact, applying the dynamic-importance-metric-based risk scoring.

Author Contributions: Investigation, U.H.P. and J.-h.H.; writing—original draft, U.H.P. and J.-h.H.; supervision, K.H.S. and A.K.; writing—review and editing U.H.P., J.-h.H., A.K. and K.H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 1711193798, Zero Trust technology based access control and abnormal event analysis technology development for enterprise network protection in the untact era).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Waizenegger, L.; McKenna, B.; Cai, W.; Bendz, T. An affordance perspective of team collaboration and enforced working from home during COVID-19. *Eur. J. Inf. Syst.* **2020**, *29*, 429–442. [[CrossRef](#)]
2. Green, N.; Tappin, D.; Bentley, T. Working from home before, during and after the Covid-19 pandemic: Implications for workers and organisations. *N. Z. J. Employ. Relations* **2020**, *45*, 5–16. [[CrossRef](#)]
3. Mandal, S.; Khan, D.A.; Jain, S. Cloud-based zero trust access control policy: An approach to support work-from-home driven by COVID-19 pandemic. *New Gener. Comput.* **2021**, *39*, 599–622. [[CrossRef](#)] [[PubMed](#)]
4. Kindervag, J.; Balaouras, S. No more chewy centers: Introducing the zero trust model of information security. *Forrester Res.* **2010**, *3*.
5. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; Technical Report; National Institute of Standards and Technology: Washington, DC, USA, 2020.
6. Mehraj, S.; Banday, M.T. Establishing a zero trust strategy in cloud computing environment. In Proceedings of the 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 22–24 January 2020; pp. 1–6.
7. Department of Defense Chief Information Officer. Department of Defense Zero Trust Strategy. 2020. Available online: <https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf> (accessed on 2 December 2022).

8. Patil, A.P.; Karkal, G.; Wadhwa, J.; Sawood, M.; Reddy, K.D. Design and implementation of a consensus algorithm to build zero trust model. In Proceedings of the 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 10–13 December 2020; pp. 1–5.
9. Uehara, M. Zero Trust Security in the Mist Architecture. In Proceedings of the Complex, Intelligent and Software Intensive Systems: Proceedings of the 15th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2021), Asan, Republic of Korea, 1–3 July 2021; Springer: Cham, Switzerland, 2021; pp. 185–194.
10. Cybersecurity Framework. Available online: <https://www.nist.gov/cyberframework/updates/nist-cybersecurity-framework-journey-csf-20> (accessed on 6 December 2022).
11. Kerman, A. Zero Trust Cybersecurity: ‘Never Trust, Always Verify’. *NIST Blog*, 2020. Available online: <https://www.nist.gov/blogs/taking-measure/zero-trust-cybersecurity-never-trust-always-verify> (accessed on 6 December 2022).
12. Dimitrakos, T.; Dilshener, T.; Kravtsov, A.; La Marra, A.; Martinelli, F.; Rizos, A.; Rosetti, A.; Saracino, A. Trust aware continuous authorization for zero trust in consumer internet of things. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 1801–1812.
13. What Is a Zero Trust Architecture—Paloaltonetworks.com. Available online: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture> (accessed on 6 December 2022).
14. Staff, I.T. Implementing a Zero Trust Security Model at Microsoft—Inside Track Blog—Microsoft.com. Available online: <https://www.microsoft.com/insidetrack/blog/implementing-a-zero-trust-security-model-at-microsoft/> (accessed on 6 December 2022).
15. Conway, A. New Data from Microsoft Shows How the Pandemic Is Accelerating the Digital Transformation of Cyber-Security—Microsoft Security Blog—microsoft.com. Available online: <https://www.microsoft.com/en-us/security/blog/2020/08/19/microsoft-shows-pandemic-accelerating-transformation-cyber-security/> (accessed on 6 December 2022).
16. Brenduns. Configure Microsoft Defender for Endpoint in Microsoft Intune—Learn.microsoft.com. Available online: <https://learn.microsoft.com/en-us/mem/intune/protect/advanced-threat-protection-configure> (accessed on 6 December 2022).
17. Siosulli. Microsoft Secure Score—Learn.microsoft.com. Available online: <https://learn.microsoft.com/en-us/microsoft-365/security/defender/microsoft-secure-score?view=o365-worldwide> (accessed on 6 December 2022).
18. Katzer, M.; Katzer, M. Microsoft Secure Score. In *Securing Office 365: Masterminding MDM and Compliance in the Cloud*; Apress: California, MA, USA, 2018; pp. 97–156.
19. Samaniego, M.; Deters, R. Zero-Trust Hierarchical Management in IoT. In Proceedings of the 2018 IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, USA, 2–7 July 2018; pp. 88–95. [CrossRef]
20. Scarfone, K.; Mell, P. *The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities*; NIST Interagency Report; NIST: Gaithersburg, MD, USA, 2010.
21. Kasprzyk, R.; Stachurski, A. A concept of standard-based vulnerability management automation for IT systems. *Comput. Sci. Math. Model.* **2016**, *3*, 33–38. [CrossRef]
22. Torkura, K.A.; Sukmana, M.I.; Meinig, M.; Kayem, A.V.; Cheng, F.; Graupner, H.; Meinel, C. Securing cloud storage brokerage systems through threat models. In Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 16–18 May 2018; pp. 759–768.
23. Wicaksana, A.; Wira, J.C. Security Analysis of Private Blockchain Implementation for Digital Diploma. *Int. J. Innov. Comput. Inf. Control* **2022**, *18*, 1601–1615.
24. Yu, X.; Shu, Z.; Li, Q.; Huang, J. BC-BLPM: A multi-level security access control model based on blockchain technology. *China Commun.* **2021**, *18*, 110–135. [CrossRef]
25. Albuali, A.; Mengistu, T.; Che, D. ZTIMM: A zero-trust-based identity management model for volunteer cloud computing. In Proceedings of the Cloud Computing—CLOUD 2020: 13th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, 18–20 September 2020; Springer: Cham, Switzerland, 2020; pp. 287–294.
26. Ge, Y.; Zhu, Q. Trust Threshold Policy for Explainable and Adaptive Zero-Trust Defense in Enterprise Networks. In Proceedings of the 2022 IEEE Conference on Communications and Network Security (CNS), Austin, TX, USA, 3–5 October 2022; pp. 359–364.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.